Blockchains

Empowering Technologies and Industrial Applications

Edited by
Anwer Al-Dulaimi
Octavia A. Dobre
Chih-Lin I

IEEE PRESS

IEEE SERIES ON DIGITAL & MOBILE COMMUNICATION
John B. Anderson, *Series Editor*

WILEY

Blockchains

IEEE Press
445 Hoes Lane
Piscataway, NJ 08854

Blockchains

Empowering Technologies and Industrial Applications

Edited by
Anwer Al-Dulaimi
EXFO Inc., Canada

Octavia A. Dobre
Memorial University, Canada

Chih-Lin I
China Mobile Research Institute, China

IEEE PRESS
WILEY

Published by John Wiley & Sons, Inc., Hoboken, New Jersey.
Published simultaneously in Canada.

Library of Congress Cataloging-in-Publication Data

Names: Al-Dulaimi, Anwer, 1974- editor. | Dobre, Octavia A., editor. | I, Chih-Lin, editor.
Title: Blockchains : empowering technologies and industrial applications / Anwer Al-Dulaimi, Octavia A. Dobre, Chih-Lin I.
Description: Hoboken, New Jersey : Wiley-IEEE Press, [2024] | Includes index.
Identifiers: LCCN 2023025340 (print) | LCCN 2023025341 (ebook) | ISBN 9781119781011 (cloth) | ISBN 9781119781028 (adobe pdf) | ISBN 9781119781035 (epub)
Subjects: LCSH: Blockchains (Databases)
Classification: LCC QA76.9.B56 B5745 2024 (print) | LCC QA76.9.B56 (ebook) | DDC 005.74–dc23/eng/20230623
LC record available at https://lccn.loc.gov/2023025340
LC ebook record available at https://lccn.loc.gov/2023025341

Cover Design: Wiley
Cover Image: © whiteMocca/Shutterstock

Set in 9.5/12.5pt STIXTwoText by Straive, Chennai, India

TO EXFO Team

This book is dedicated to my colleagues at EXFO, who made every day at work a joy. Your camaraderie and humor helped to make even the toughest challenges feel manageable.

To my team, your tireless work ethic and dedication to excellence inspired me every day, and I am forever grateful for the knowledge and experience I gain while working alongside each of you.

To the leaders of EXFO, who created a culture of innovation, inclusion, and growth. Your unwavering commitment to your employees and customers alike is truly admirable, and I am privileged to be part of such a forward-thinking organization.

Anwer Al-Dulaimi

Contents

About the Editors

Anwer Al-Dulaimi is currently a Senior Manager of Emerging Technologies and a Distinguished Member of Technical Staff at EXFO, Montreal, Canada. He received the PhD degree in electrical and computer engineering from Brunel University, London, UK, in 2012 after obtaining MSc and BSc honors degrees in communication engineering. He was a Postdoctoral Fellow in the Department of Electrical and Computer Engineering, University of Toronto, sponsored by Blackberry's advanced research team. In his current role, Anwer is responsible for identifying future trends in mobile technology evolutions and the adaptation phases that EXFO needs to take for compliance. He is the chair of the newly established IEEE 5G/6G Innovation Testbed Project, which is working to develop a virtual testing platform for E2E network industry testing. He is the chair of the IEEE 1932.1 "Standard for Licensed/Unlicensed Spectrum Interoperability in Wireless Mobile Network." He is also representing EXFO in many industrial forums such as One6G and other collaborative projects. He has published many papers, edited books, and developed patents focusing on new generations of mobile networking technologies. He is the editor of the IEEE Future Networks Series on 5G and 6G published by *IEEE Vehicular Technology Magazine*, the editor of the Vehicular Networking Series in *IEEE Communication Standards Magazine*, and a guest editor of many other IEEE series issues. His research interests include 5G/6G networks, cloud computing, IoT, and cybersecurity. He is a Fellow of the Institution of Engineering and Technology (FIET) and has registered as a Chartered Engineer (CEng) by the British Engineering Council since 2010. He is a member of the NSERC discovery grants committee, a voting member of Mobile

Communication Networks Standards Committee (MobiNet-SC), a senior member of IEEE, and an IEEE ComSoc Distinguished Lecturer.

Octavia A. Dobre is a Professor and Research Chair at Memorial University, Canada. Previously, she was with New Jersey Institute of Technology and Stevens Institute of Technology, USA, after receiving her PhD degree from the Polytechnic Institute of Bucharest, Romania, in 2000. Dr. Dobre was a Visiting Professor at the Massachusetts Institute of Technology, USA, and Université de Bretagne Occidentale, France. Her research interests encompass wireless communication and networking technologies, as well as optical and underwater communications. She has co-authored more than 400 refereed papers in these areas. Dr. Dobre serves as the Director of Journals and Editor-in-Chief (EiC) of the *IEEE Open Journal of the Communications Society*. She was the EiC of the *IEEE Communications Letters* and a senior editor, editor, and guest Editor for various prestigious journals and magazines. She also served as General Chair, Technical Program Co-Chair, Tutorial Co-Chair, and Technical Co-Chair of Symposia at numerous conferences. Dr. Dobre was a Fulbright Scholar, Royal Society Scholar, and Distinguished Lecturer of the IEEE Communications Society. She obtained Best Paper Awards at various conferences, including IEEE ICC, IEEE Globecom, IEEE WCNC, and IEEE PIMRC. Dr. Dobre is an elected member of the European Academy of Sciences and Arts, a Fellow of the Engineering Institute of Canada, a Fellow of the Canadian Academy of Engineering, and a Fellow of the IEEE.

Chih-Lin I is CMCC Chief Scientist of Wireless Technologies, spearheading major initiatives covering 5G/6G, C-RAN/O-RAN, and Green technologies. She received PhD in EE from Stanford University. She has won the 2005 IEEE ComSoc Stephen Rice Prize, the 2018 IEEE ComSoc Fred W. Ellersick Prize, the 7th IEEE Asia-Pacific Outstanding Paper Award, and the 2015 IEEE Industrial Innovation Award for Leadership and Innovation in Next-Generation Cellular Wireless Networks. She is the Chair of O-RAN Technical Steering Committee; an O-RAN Executive Committee Member, the Chair of FuTURE 5G/6G SIG;

the Chair of WAIA (Wireless AI Alliance) Executive Committee; an Executive Board Member of GreenTouch; a Network Operator Council Founding Member of ETSI NFV; a Steering Board Member and Vice Chair of WWRF; a Steering Committee member and the Publication Chair of IEEE 5G and Future Networks Initiatives; the Founding Chair of IEEE WCNC Steering Committee; the Director of IEEE ComSoc Meetings and Conferences Board; a Senior Editor of *IEEE Transactions of Green Communication and Networking*; an Area Editor of *ACM/IEEE Transactions of Networking*; Executive Co-chair of IEEE Globecom 2020, IEEE WCNC 2007, IEEE WOCC 2004 and 2000; a member of IEEE ComSoc SDB, SPC, and CSCN-SC; and a Scientific Advisory Board Member of Singapore NRF. She has published over 200 papers in scientific journals, book chapters, and conferences and holds over 100 patents. She has co-authored the book *Green and Software-Defined Wireless Networks – From Theory to Practice* and has also co-edited two books: Ultra-dense Networks – Principles and Applications and *5G Networks – Fundamental Requirements, Enabling Technologies, and Operations Management*. She is a Fellow of IEEE and a Fellow of WWRF. Her current research interests center around ICDT Deep Convergence: "From Green and Soft to Open and Smart."

About the Contributors

Anwer Al-Dulaimi is currently a Senior Manager of Emerging Technologies and a Distinguished Member of Technical Staff at EXFO, Montreal, Canada. He received the PhD degree in electrical and computer engineering from Brunel University, London, UK, in 2012 after obtaining MSc and BSc honors degrees in communication engineering. He was a Postdoctoral Fellow in the Department of Electrical and Computer Engineering, University of Toronto, sponsored by Blackberry's advanced research team. In his current role, Anwer is responsible for identifying future trends in mobile technology evolutions and the adaptation phases that EXFO needs to take for compliance. He is the chair of the newly established IEEE 5G/6G Innovation Testbed Project, which is working to develop a virtual testing platform for E2E network industry testing. He is the chair of the IEEE 1932.1 "Standard for Licensed/Unlicensed Spectrum Interoperability in Wireless Mobile Network." He is also representing EXFO in many industrial forums such as One6G and other collaborative projects. He has published many papers, edited books, and developed patents focusing on new generations of mobile networking technologies. He is the editor of the IEEE Future Networks Series on 5G and 6G published by *IEEE Vehicular Technology Magazine*, the editor of the Vehicular Networking Series in *IEEE Communication Standards Magazine*, and a guest editor of many other IEEE series issues. His research interests include 5G/6G networks, cloud computing, IoT, and cybersecurity. He is a Fellow of the Institution of Engineering and Technology (FIET) and has registered as a Chartered Engineer (CEng) by the British Engineering Council since 2010. He is a member of the NSERC discovery grants committee, a voting member of Mobile Communication Networks Standards Committee (MobiNet-SC), a senior member of IEEE and an IEEE ComSoc Distinguished Lecturer.

Antonio Albanese received his B.Sc. in Electronic and Telecommunication Engineering from Politecnico di Bari, Italy, in 2016 and his M.Sc. degree in Telecommunications Engineering from Politecnico di Milano, Italy, in 2018. Currently, he is the Chief Technology Officer at Flyhound Co., New York, USA while he is pursuing his Ph.D. in Telematic Engineering at Universidad Carlos III de Madrid, Spain. His research field covers millimeter waves, reconfigurable intelligent surfaces, aerial networks, and applied mathematical optimization, with a particular interest in wireless localization and prototyping.

Ismael Arribas is Co-founder, Kunfud®, Chief Compliance Officer at KRON WORLD S.L. Global and Collective Entrepreneur from Kingdom of Spain, and driving an independent compliance firm named Kunfud® since 2006. Ismael is committed with Standards at ISO TC 307, CEN-CENELEC JTC19, Liaison Officer between CEN-CENELEC and ETSI ISG- PDL, ITU-T FGDLT, and Q-22 at Study Group 16. He is Founding member of INATBA (https://inatba.org/) and Co-chair at Standards Committee in INATBA. He is also Principal Advisor for standardization at LaCChain (https://www.lacchain.net/#/home?lang=en) and IDB Lab. He is Co-Founder of various Startups like Blue Future Organization, Lumiversity, and CLAUDIA.

Hans Walter Behrens received his B.S. in Computer Science from the University of California, Irvine, and his M.S. and Ph.D. in Computer Science from Arizona State University. He serves as a visiting student researcher at Los Alamos National Laboratory, and has spent several years as Chief Technical Officer in several startups as well. His primary research interests include the security and privacy implications of distributed and decentralized systems, and in the creation of robust and resilient techniques that maintain

functionality in the presence of active adversaries. His work has been published in conferences such as CCS, VLDB, ICBC, and ICC, and has been recognized by several awards, including the Dean's Fellowship and Herbold Foundation Graduate Engineering Scholarship. To learn more, visit behrens.dev/.

Dragan Boscovic is a research professor in the School of Computing, Informatics, & Decision Systems Engineering (CIDSE) at Arizona State University and Research Director of AZ Blockchain Applied Research Center, and Distinguished Visiting Scholar, mediaX, at Stanford University. Additionally, he is CEO and Founder of technology VizLore Group, headquartered in Arizona, and focuses on delivering innovative solutions related to IoT, Data Analytics, Blockchain distributed computing, and digital asset management. He holds a Ph.D. in EE and CS, Numerical Electromagnetic Modeling from the University of Bath, United Kingdom (1991). He has 25 years of high-tech experience acquired in an international setup (UK, France, China, USA) and is uniquely positioned to help data-driven technical advances within today's global data-intensive technology arena. He is a lateral thinker with broad exposure to a wide range of scientific methods and business practices. He has a proven track record in conceiving strategies and managing the development, investment, and innovation efforts related to advanced data analysis services, ML/AI applications, and mobile and IoT solutions and platforms.

Kasim Selçuk Candan is a Professor of Computer Science and Engineering at the Arizona State University (ASU) and the Director of ASU's Center for Assured and Scalable Data Engineering (CASCADE). His primary research interest is in the area of management and analysis of (non-traditional, heterogeneous, and imprecise) data. He has published over 200 journal and peer-reviewed conference articles, one book, several book chapters, and has several patents. Prof. Candan served as an associate editor of the *Very Large Databases* (VLDB) journal, *IEEE Transactions on Cloud Computing*, and *IEEE Transactions on Multimedia*. He is currently serving as Associate Editor for the *ACM Transactions on Database Systems*

and *IEEE Transactions on Knowledge and Data Engineering*. He is a founding Managing Editor for the *ACM Proceedings of the Management of Data* (PACMMOD). He served as a member of the Executive Committee of ACM SIGMOD and is an ACM Distinguished Scientist. You can find more information about his research and an up-to-date resume at http://aria.asu.edu/candan.

Xiaofeng Chen, PhD, Zhejiang University, is senior technical standards expert and senior software testing and development expert. The main work direction includes engineering efficiency improvement, automated testing, continuous integration testing, NoSQL database research and testing, blockchain technology research and testing, application system and related platform demand analysis and testing, multi-platform and cross-platform project test method research, and test efficiency optimization. The author has actively participated in the development of blockchain standards of IEEE, ITU-T SG16 Q22, ISO/TC 307, and other international standardization organizations, the research work of national standards, industry and group standards, and the standard direction covers the basic technology, architecture, performance, security, and real economy applications of the blockchain. Xiaofeng led and participated in a team for development of more than 100 blockchain international, national, industry, and group standards. The author is currently working at Hangzhou Qulian Technology Co., Ltd., and is responsible for the company's quality system, safety system, and standardization system construction, improving product quality, improving and optimizing engineering efficiency, and enhancing the company's domestic and foreign influence.

Alexander Chuburkov is a national standardization expert (Russia, GOST R). Alexander leads the work in the WG "Smart contracts" of the Technical Committee on Standardization "Hardware and Software for Distributed Ledger Technologies" (TC159, Mirror Committee for ISO/TC307). Alexander led the work of the regulatory framework development at the ITU-T Focus Group on the Application of Distributed Ledger Technology (FG DLT).

Xavier Costa-Pérez is Head of Beyond 5G Networks R&D at NEC Laboratories Europe, Scientific Director at the i2Cat R&D Center and Research Professor at ICREA. His team contributes to products roadmap evolution as well as to European Commission R&D collaborative projects and received several awards for successful technology transfers. In addition, the team contributes to related standardization bodies: 3GPP, ETSI NFV, ETSI MEC and IETF. Xavier has been a 5GPPP Technology Board member, served on the Program Committee of several conferences (including IEEE Greencom, WCNC, and INFOCOM), published at top research venues and holds several patents. He also serves as Editor of IEEE Transactions on Mobile Computing and Transactions on Communications journals. He received both his M.Sc. and Ph.D. degrees in Telecommunications from the Polytechnic University of Catalonia (UPC) in Barcelona and was the recipient of a national award for his Ph.D. thesis.

Tasos Dagiuklas received the Engineering Degree from the University of Patras-Greece in 1989, the MSc from the University of Manchester, UK, in 1991, and the PhD degree from the University of Essex-UK in 1995, all in Electrical Engineering. He is a leading researcher and expert in the fields of Internet and multimedia technologies for smart cities, ambient assisted living, healthcare, and smart agriculture. He has been a principal investigator, a co-investigator, a project and technical manager, a coordinator, and a focal person of over 20 internationally R&D and capacity training projects with total funding of approximately č5.0m from different international organizations. He is currently the Leader of the SuITE Research Group, London South Bank University, where he also acts as the Head of the Division in Computer Science. His research interests include smart internet technologies, media optimization across heterogeneous networks, QoE, virtual reality, augmented reality, and cloud infrastructures and services.

Ming Ding (Senior Member, IEEE) received the BS and MS degrees (with first-class Hons.) in electronics engineering from Shanghai Jiao Tong University (SJTU), Shanghai, China, and the Doctor of Philosophy (PhD) degree in signal and information processing from SJTU, in 2004, 2007, and 2011, respectively. From April 2007 to September 2014, he worked at Sharp Laboratories of China in Shanghai, China, as a Researcher/Senior Researcher/Principal Researcher. Currently, he is a senior research scientist at Data61, CSIRO, in Sydney, NSW, Australia. His research interests include information technology, data privacy and security, machine learning and AI, etc. He has authored over 140 papers in IEEE journals and conferences, all in recognized venues, and around 20 3GPP standardization contributions, as well as a Springer book *Multi-point Cooperative Communication Systems: Theory and Applications*. Also, he holds 21 US patents and co-invented another 100+ patents on 4G/5G technologies in CN, JP, KR, EU, etc. Currently, he is an editor of *IEEE Transactions on Wireless Communications* and *IEEE Wireless Communications Letters*. Besides, he has served as Guest Editor/Co-Chair/Co-Tutor/TPC member for many IEEE top-tier journals/conferences and received several awards for his research work and professional services.

Hui Ding is senior standard expert of Ant Group and responsible for technology collaboration in Digital Technology BG of Ant Group. She is a member of IEEE-SA RevCom, Secretary of IEEE CTS DFESC, member of Blockchain Committee of China Computer Federation, and member of Blockchain Committee of China Institute of Communications. She led/participated in a number of blockchain and IoT-related standard projects in IEEE (P2418.3, P2144, P2418.10, P2418.10, etc.) and also contributed to ITU-T FG-DLT and FG-DPM while she was the co-founder of Chaincomp Technologies. She worked with ITU-T SG13, ONF, OIF, IETF and participated in more than 20 SDO projects including SDN northbound API, Intent-based API, information modeling, open-source tooling development, and ONF/OIF global interoperability demo as researcher and standard engineer in China Academy of Information and Communications Technology, China. She obtained a Ph.D. degree from Beijing University of Posts and Telecommunications.

Octavia A. Dobre is a Professor and Research Chair at Memorial University, Canada. Previously, she was with New Jersey Institute of Technology and Stevens Institute of Technology, USA, after receiving her PhD degree from the Polytechnic Institute of Bucharest, Romania, in 2000. Dr. Dobre was a Visiting Professor with the Massachusetts Institute of Technology, USA, and Université de Bretagne Occidentale, France. Her research interests encompass wireless communication and networking technologies, as well as optical and underwater communications. She has (co-)authored +400 refereed papers in these areas.

Dr. Dobre serves as the Director of Journals and Editor-in-Chief (EiC) of the *IEEE Open Journal of the Communications Society*. She was the EiC of the IEEE Communications Letters, Senior Editor, Editor, and Guest Editor for various prestigious journals and magazines. She also served as General Chair, Technical Program Co-Chair, Tutorial Co-Chair, and Technical Co-Chair of symposia at numerous conferences.

Dr. Dobre was a Fulbright Scholar, Royal Society Scholar, and Distinguished Lecturer of the IEEE Communications Society. She obtained Best Paper Awards at various conferences, including IEEE ICC, IEEE Globecom, IEEE WCNC, and IEEE PIMRC. Dr. Dobre is an elected member of the European Academy of Sciences and Arts, a Fellow of the Engineering Institute of Canada, a Fellow of the Canadian Academy of Engineering, and a Fellow of the IEEE.

Jörn Erbguth is a consultant on blockchain and data protection (GDPR). With majors in computer science and law, he takes a multidisciplinary approach to new technology. He is an enabler of privacy by design where legal and technological aspects need to be tightly integrated. Jörn previously worked as a software developer, a product manager, and was head of ICT and CTO for legal information systems in Germany and Switzerland. Jörn publishes about technology and law and lectures at the Geneva School of Diplomacy, the University of Lucerne, and the University of St. Gallen. He is active in Blockchain standardization at the International Telecommunication Union (ITU) and was a member of DIN SPEC 4997 Privacy by Blockchain Design. He works with Geneva Macro Labs as Head of Technology Insights. Jörn also serves as a board member for the German

EDV-Gerichtstag (German Association for Computing in the Judiciary) and the Swiss entscheidsuche.ch association. The latter engages in making Swiss court cases easily and freely available to the public. He is also member of the expert panel of the European Blockchain Observatory and Forum.

Saptarshi Ghosh received the BSc (Hons.) degree in computer science from the University of Calcutta, Kolkata, India, in 2010, the ME degree in software engineering from Jadavpur University, Kolkata, in 2016, and the MSc degree in smart networks from the University of the West of Scotland, Glasgow, UK, in 2017. He is currently working toward the PhD degree in computer science and informatics with London South Bank University, London, UK.

He is a module Leader of several core CS modules with London South Bank University. He is JNCIA (DevOps) certified, and was a Software Developer and Network Engineer. He has contributed to several research and software engineering projects funded by Erasmus+, Innovate UK, and Defence Science and Technology Laboratory. His research interests include SD-WAN, network programmability and automation, cognitive-routing, and deep reinforcement learning.

Ghosh is a recipient of GATE and Erasmus-Mundus Scholarship. His PhD research is under the EU-Horizon 2020 project, supported by Marie-Curie Fund with the research area focused in machine learning's application to self-organized SDN for 5G and beyond.

Dr. Thomas Hardjono is the CTO of Connection Science and Technical Director of the MIT Trust-Data Consortium at MIT in Cambridge, MA. USA. He is an early pioneer in the field of digital identities and trusted hardware, and instrumental in the development and broad adoption of the MIT Kerberos authentication protocol. His activities include leading standard development efforts, notably at the IETF (Internet Engineering Task Force), IEEE, Trusted Computing Group, Confidential Computing Alliance, and others. He has published over 70 technical conference and journal papers, and authored several books. He is currently involved in several startups around the MIT community. His current area of interest is Web3 Digital Assets, with focus on the interoperability of asset networks and survivability of these networks against cybersecurity attacks.

Chih-Lin I is CMCC Chief Scientist of Wireless Technologies, spearheading major initiatives covering 5G/6G, C-RAN/O-RAN, and Green technologies. She received PhD (EE) from Stanford University. She has won 2005 IEEE ComSoc Stephen Rice Prize, 2018 IEEE ComSoc Fred W. Ellersick Prize, the 7th IEEE Asia-Pacific Outstanding Paper Award, and 2015 IEEE Industrial Innovation Award for Leadership and Innovation in Next-Generation Cellular Wireless Networks.

She is the Chair of O-RAN Technical Steering Committee and an O-RAN Executive Committee Member, the Chair of FuTURE 5G/6G SIG, the Chair of WAIA (Wireless AI Alliance) Executive Committee, an Executive Board Member of GreenTouch, a Network Operator Council Founding Member of ETSI NFV, a Steering Board Member and Vice Chair of WWRF, a Steering Committee member and the Publication Chair of IEEE 5G and Future Networks Initiatives, the Founding Chair of IEEE WCNC Steering Committee, the Director of IEEE ComSoc Meetings and Conferences Board, a Senior Editor of *IEEE Trans. Green Comm. & Networking*, an Area Editor of *ACM/IEEE Trans. Networking*; Executive Co-chair of IEEE Globecom 2020, IEEE WCNC 2007, IEEE WOCC 2004 and 2000; a member of IEEE ComSoc SDB, SPC, and CSCN-SC; and a Scientific Advisory Board Member of Singapore NRF.

She has published over 200 papers in scientific journals, book chapters, and conferences and holds over 100 patents. She is co-author of the book *Green and Software-defined Wireless Networks – From Theory to Practice* and has also co-edited two books: *Ultra-dense Networks – Principles and Applications* and *5G Networks – Fundamental Requirements, Enabling Technologies, and Operations Management*. She is a Fellow of IEEE and a Fellow of WWRF. Her current research interests center around ICDT Deep Convergence: "From Green & Soft to Open & Smart."

Muddesar Iqbal received the PhD degree from Kingston University in 2010 with a dissertation titled "Design, development, and implementation of a high-performance wireless mesh network for application in emergency and disaster recovery." He has been a principal investigator, a co-investigator, a project manager, a coordinator, and a focal person of over 20 internationally teamed research and development, capacity building, and training projects. He is an established researcher and expert in the fields of mobile cloud

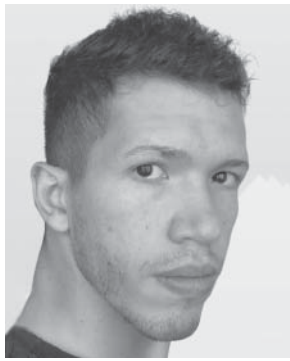
computing and open-based networking for applications in education, disaster management, and healthcare; community networks; and smart cities. He is currently a Senior Lecturer in mobile computing with the Division of Computer Science and Informatics, School of Engineering, London South Bank University. His research interests include 5G networking technologies, multimedia cloud computing, mobile edge computing, fog computing, Internet of Things, software-defined networking, network function virtualization, quality of experience, and cloud infrastructures and services. He was a recipient of the EPSRC Doctoral Training Award in 2007.

Petar Jevtić is an Assistant Professor at Arizona State University, School of Mathematical and Statistical Sciences, USA. He held an Assistant Professor position at McMaster University, Department of Mathematics and Statistics, Canada, where he also completed his Postdoctoral Fellowship. In 2013 he gained his PhD at the Department of Economic, Social, Mathematical and Statistical Sciences at University of Turin, Italy. During his PhD, he was visiting scholar at the Department of Statistics at The Wharton School of the University of Pennsylvania. He received an MS degree at the Faculty of Economics, University of Belgrade, Serbia. He received Dipl. Ing. in Computer Science and Engineering at School of Electrical Engineering, University of Belgrade, Serbia. His research focus is on mathematical modeling of risk with an emphasis on actuarial science and mathematical finance. He has published in premier actuarial, statistics, and theoretical mathematics journals.

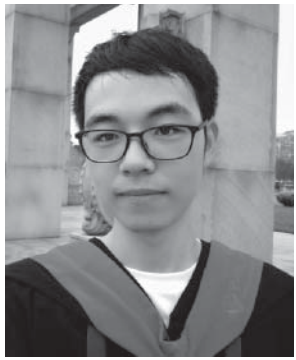
Xiangjuan Jia holds a Master's degree from Lanzhou University. Xiangjuan is a standardization expert and software testing engineer. The author has actively participated in the development of blockchain standards of IEEE, ITU-T SG16 Q22, ISO/TC 307, and other international standardization organizations, and served as the secretary of the IEEE P3203 working group. Xiangjuan participated in the writing of national standards, industry standards, corporate standards, group standards, and participated in the writing of whitepapers and development reports related to blockchain industry applications. The standard direction covers the basic technology of blockchain, energy, government affairs, finance and other fields. Xiangjuan obtained the standardization manager (senior) certificate. Xiangjuan

is currently working at Hangzhou Qulian Technology Co., Ltd., engaged in the research and testing of blockchain technology, and Standard preparation, comparative study of domestic and foreign standards, establishment of standard system.

Paulo Valente Klaine received his B. Eng. degree in electrical and electronic engineering from the Federal University of Technology - Parana (UTFPR), Brazil in 2014, the MSc. degree from the University of Surrey, Guildford, U.K., in mobile communications systems in 2015, and the PhD degree in electrical and Electronics engineering from the University of Glasgow, U.K., in 2019. He has 3 filed patents and authored/co-authored over 20 publications. He received the IEEE ICE-ICT'21 Best Paper award. His research interests include wireless blockchain, machine learning in wireless networks, and massive MIMO. He is currently working as an experienced researcher at Ericsson K. K. Japan.



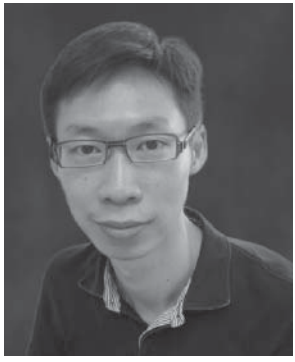
Axel La Salle was born in July 1992. He is currently pursuing a PhD degree in applied mathematics from the School of Mathematical and Statistical Sciences at Arizona State University in Tempe, Arizona. Studying under supervision of Dr. Lanchier, he has conducted research on stochastic modeling, percolation theory, and graph theory with applied focus on Distributed ledger Technology (DLT) platforms and areas within like cyber risk modeling, performance evaluation of DLT platforms.



Ziliang Lai is a Ph.D. student of Computer Science and Engineering (CSE) at the Chinese University of Hong Kong (CUHK). His research mainly focuses on database transaction processing, blockchain and video analytic systems.

Nicolas Lanchier obtained his PhD in mathematics at the University of Rouen, France, in 2005, and is now a Professor at Arizona State University, School of Mathematical and Statistical Sciences. His research is in the field of probability theory, with a focus on interacting particle systems (spatial stochastic processes). He is the author/co-author of more than 50 papers published in some of the main probability journals, the author of the textbook *Stochastic Modeling* (Springer), and the creator of a YouTube channel in probability theory. He is also the recipient of several grants from the National Science Foundation and the National Security Agency.

Shancang Li (Senior Member, IEEE) received the BSc and MSc degrees in mechanics engineering and the PhD degree in computer science from Xi'an Jiaotong University, Xi' an, China, in 2001, 2004, and 2008, respectively. He is currently a Senior Lecturer in School of Computer Science and Informatics, Cardiff University, UK. His current research interests include digital forensics for emerging technologies, network security, cyber attacks, wireless sensor networks, the Internet of Things, and the lightweight cryptography in resource-constrained devices.



Eric Lo is an Associate Professor of Computer Science and Engineering (CSE) at the Chinese University of Hong Kong (CUHK). His research mainly focuses on video analytics, upercomputing, database, distributed systems, data science, and blockchain. Supported by the Croucher Scholarship, Eric received his Ph.D. from ETH Zurich's Computer Science Department. He was a member of ETH's Systems group. Before joining CUHK, he got stints at Google and Hong Kong Polytechnic University.

Dinh C. Nguyen (Graduate Student Member, IEEE) is currently pursuing the PhD degree at the School of Engineering, Deakin University, VIC, Australia. His research interests focus on blockchain, deep reinforcement learning, mobile edge/cloud computing, network security, and privacy. He is currently working on blockchain and reinforcement learning for Internet of Things and 5G networks. He has been a recipient of the prestigious Data61 PhD scholarship, CSIRO, Australia.

Kyeong Hee Oh is CEO of TCA services, Outside director of Korea Internet & Security Agency. She is an information security professional with more than 20 years' experience. She had participated in the establishment of the evaluation criteria for information security product, of the national PKI, of ISMS certification scheme in Korea and developed the first draft of Privacy Information Management Systems in Korea. She has been involved with the International standardization from 2010, she is now Co-rapporteur of ITU-T SG 17 Q14 Security aspects of distributed ledger technologies, liaison officer to TC 307 and FG-DLT, Korean delegation of ISO TC 307 Blockchain and distributed ledger technologies, and Head of Korean delegation of ISO/IEC JTC 1/SC 27 WG 1 Information Security Management.

Oluwakayode Onireti (Member, IEEE) received the BE degree (Hons.) in electrical engineering from the University of Ilorin, Ilorin, Nigeria, in 2005, and the MSc degree (Hons.) in mobile and satellite communications and the PhD degree in electronics engineering from the University of Surrey, Guildford, U.K., in 2009 and 2012, respectively. He is currently a Lecturer with the University of Glasgow, Glasgow, U.K. He has been actively involved in projects, such as ROCKET, EARTH, Greencom, QSON, DARE, and Energy proportional EnodeB for LTE-Advanced and Beyond. His main research interests include self-organizing cellular networks, millimeter-wave communications, energy efficiency, wireless blockchain networks, multiple-input–multiple-output, and cooperative communications.

Pubudu N. Pathirana (Senior Member, IEEE) was born in 1970 in Matara, Sri Lanka, and was educated at Royal College Colombo. He received the B.E. degree (first class honors) in electrical engineering and the B.Sc. degree in mathematics in 1996, and the PhD degree in electrical engineering in 2000 from the University of Western Australia, all sponsored by the government of Australia on EMSS and IPRS scholarships, respectively. He was a Postdoctoral Research Fellow at Oxford University, Oxford, a Research Fellow at the School of Electrical Engineering and Telecommunications, University of New South Wales, Sydney, Australia, and a Consultant to the Defence Science and Technology Organization (DSTO), Australia, in 2002. He was a visiting professor at Yale University in 2009. Currently, he is a full Professor, Head of Discipline Mechatronics Electrical and Electronic Engineering, and the Director of Networked Sensing and Control group at the School of Engineering, Deakin University, Geelong, Australia and his current research interests include Bio-Medical assistive device design, human motion capture, mobile/wireless networks, rehabilitation robotics, and radar array signal processing.

Sasa Pesic is a teaching assistant at the Department of Mathematics and Informatics, Faculty of Science, University of Novi Sad, Serbia. Sasa is a Visiting Researcher at the School of Computing, Informatics, and Decision Systems Engineering (CIDSE) at Arizona State University (Tempe, AZ, USA) at the Blockchain Research Laboratory and a Research Consultant at the Electrical Engineering and Computer Science (EECS) department at Khalifa University (Abu Dhabi, UAE). He also is a seasoned blockchain engineer with 5 years' experience in industry settings. In his research work, he deals with highly distributed Internet of Things and edge computing systems, analyzing their robustness, security, and operating capacity and stability. His research interests include distributed ledger technologies, their security and interdisciplinary application in the domains of energy, finance, security of IoT systems, and peer-to-peer insurance. He is the author/coauthor of 15 scientific publications. He is actively working on two Horizon2020 research projects: Interconnect and Dedicat-6G, and in the past two years he has worked on PhasmaFOOD, Vicinity, AgileIoT, BlockIS, and SymbIote.

Dr. Sina Rafati Niya received his Ph.D. from the University of Zürich (UZH) on the topic of "Efficient Designs for Practical Blockchain-IoT Integration" in 2021. Since 2016 he has been conducting continuous research on blockchain-based Decentralized Applications (dApps) and on protocols in the Identity Management, Internet-of-Things (IoT), Know Your Customer (KYC), Peer-to-peer trading, Supply Chain Tracking, and Decentralized Finance (DeFi) domains. Sina has been pursuing his research in the blockchain analytics area since 2022 as a senior research associate at the Blockchain and Distributed Ledger Technologies (BDLT) group at UZH. Sina has published multiple scientific articles in the blockchain-based service management area in recent years.

Saqib Rasool holds an MS degree in Computer Science from the National University of Science and Technology (NUST), Islamabad, Pakistan. He is currently pursuing PhD studies and is also serving as senior lecturer at Department of Computer Science, University of Gujrat (UoG), Gujrat, Pakistan. His research interests are Blockchain, Internet/Web/Cloud of Things, Reflection and Meta-programming, Declarative DSLs, DevOps and scalable cloud/fog services.

Vincenzo Sciancalepore received his M.Sc. degree in Telecommunications Engineering and Telematics Engineering in 2011 and 2012, respectively, whereas 2015, he received a double Ph.D. degree. Currently, he is a senior 5G researcher at NEC Laboratories Europe in Heidelberg, focusing his activity on network virtualization and network slicing challenges. He is currently involved in the IEEE Emerging Technologies Committee leading the initiatives on SDN and NFV. He is the Chair of the Emerging Technologies Initiative (ETI) on Reconfigurable Intelligent Surfaces (RIS). He was also the recipient of the national award for the best Ph.D. thesis in the area of communication technologies (Wireless and Networking) issued by GTTI in 2015.

Aruna Seneviratne (Senior Member, IEEE) is currently a Foundation Professor of telecommunications with the University of New South Wales, Australia, where he holds the Mahanakorn Chair of telecommunications. He has also worked at a number of other Universities in Australia, UK, and France, and industrial organizations, including Muirhead, Standard Telecommunication Labs, Avaya Labs, and Telecom Australia (Telstra). In addition, he has held visiting appointments at INRIA, France. His current research interests are in physical analytics: technologies that enable applications to interact intelligently and securely with their environment in real time. Most recently, his team has been working on using these technologies in behavioral biometrics, optimizing the performance of wearables, and the IoT system verification. He has been awarded a number of fellowships, including one at British Telecom and one at Telecom Australia Research Labs.

Prof. Dr. Burkhard Stiller received the Diplom-Informatiker (MSc) degree in computer science and the Dr. rer.-nat. (PhD) degree from the University of Karlsruhe, Germany. He chairs as a full professor at the Communication Systems Group CSG, Department of Informatics IfI, University of Zürich UZH since 2004. He held previous research positions with the Computer Laboratory, University of Cambridge, UK; the Computer Engineering and Networks Laboratory, ETH Zürich, Switzerland; and the University of Federal Armed Forces, Munich, Germany.

He coordinated various Swiss and European industrial and research projects, such as AAMAIS, DAMMO, SmoothIT, SmartenIT, SESERV, and Econ@Tel, besides participating in others, such as CONCORDIA, M3I, Akogrimo, EMANICS, EC-GIN, FLAMINGO, and ACROSS. His main research interests are published in well over 300 research papers and include charging and accounting of Internet services, systems with a fully decentralized control (blockchains, clouds), network and service management, economic management, telecommunication economics, and IoT.

Lisa J. Y. Tan's work as the founder and lead economist at Economics Design has made her a pioneer in the design and engineering of digital ecosystems. With a track record of over 30 token economies and 50 token analyses, Lisa's work is characterized by a research-focused approach and a deep understanding of the potential of blockchain technology. As a highly sought-after speaker at conferences and forums worldwide, Lisa's expertise in token economics and DeFi has established her as a respected authority in the field of digital ecosystems.

Hao Xu received the B.Eng. degree in aerospace engineering from the University of Sheffield, Sheffield, U.K., in 2017, the M.Sc. degree in avionics from Cranfield University, Cranfield, U.K., in 2018, and the Ph.D. degree in electrical engineering from the University of Glasgow, Glasgow, U.K., in 2022. His research interests cover wireless communication, wireless blockchain consensus, blockchain-enabled radio access network, and the next generation of decentralized physical infrastructure.

Lanfranco Zanzi received his B.Sc. and M.Sc. in Telecommunication Engineering from Politecnico of Milan (Italy) in 2014 and 2017, respectively. He is currently enrolled as Ph.D. candidate at the Technical University of Kaiserlautern, and works as research scientist at NEC Laboratories Europe. His research interests include network virtualization, machine learning, blockchain, and their applicability to 5G mobile networks in the context of network slicing.

Foreword

Blockchain technology is revolutionizing the way we do business. It is rapidly transforming various industries, such as finance, healthcare, cybersecurity, and networking. Blockchains are based on decentralized, distributed ledgers that are transparent, secure, and immutable. It allows parties to transact with each other without the need for intermediaries. As a result, it is expected to have a significant impact on various industries and drive innovation in multiple sectors.

The underlying technology has come a long way, with underpinning principles introduced as early as 1979. The breakthrough, however, came in 2008 with the release of the Bitcoin whitepaper by Satoshi Nakamoto. The technology has undergone significant improvements over the years, leading to the development of various blockchain platforms and applications.

The first blockchain, Bitcoin, was designed as a peer-to-peer electronic cash system that could operate without intermediaries such as banks. However, the limitations of Bitcoin, such as slow transaction speeds and limited scalability, led to the development of other blockchain platforms that addressed these issues.

Ethereum, launched in 2015, was a significant milestone in the evolution of blockchain technology. It introduced the concept of smart contracts, which are self-executing contracts that automatically enforce the rules and regulations of the contract. Smart contracts have since become a fundamental feature of most blockchain platforms, enabling the creation of decentralized applications (DApps) that can perform various functions, such as asset management, supply chain tracking, and voting systems.

Another significant development in the evolution of blockchain technology is the emergence of private and consortium blockchains. Unlike public blockchains like Bitcoin and Ethereum, private blockchains are permissioned networks that allow only authorized participants to access and transact on the network. Consortium blockchains, on the other hand, are shared among a group of organizations

that collaborate to achieve a common goal, such as supply chain management. Permissioned blockchains offer increased privacy, scalability, and flexibility, making them suitable for various enterprise applications.

Blockchain technology has the potential to transform other technology fields through enabling secure communication channels, authentication mechanisms, and data protection through encryption. In terms of cybersecurity, blockchain technology can be used to prevent data breaches and fraud by providing tamper-proof records of transactions and other activities. It can also provide secure identity management systems that can reduce the risk of identity theft and other types of fraud.

From a communication perspective, blockchain technology can be used to provide secure messaging platforms that offer end-to-end encryption and prevent unauthorized access to user data. It can also provide a decentralized system for file storage and sharing, enabling secure and efficient collaboration between individuals and organizations.

The future of blockchain technology is promising, with new developments and innovations expected to emerge, leading to increased adoption and integration into various industries. This book aims to provide readers with a comprehensive understanding of blockchain technology, its underlying principles, and its industrial applications. It is also suitable for professionals in various industries who want to understand how blockchain can be applied in their respective fields. The book is written in a clear and concise manner, and the technical depth is just at the right level to ensure that readers can easily understand the concepts presented. It greatly complements other books in the field, such as "Blockchains in 6G: A Standardized Approach to Permissioned Distributed Ledgers."

The book covers blockchain architecture, starting with distributed storage and how data is broken down into small, encrypted fragments and distributed across multiple nodes or computers on the network. Then, it studies permissioned blockchains in which access to participate and interaction with the network is restricted to authorized parties. The book then moves into industries that use blockchain to improve security and data exchange between elements. For example, a blockchain-based V2V communication system can enable vehicles to share information such as location, speed, and road conditions, allowing them to avoid accidents and traffic congestion. In a subsequent chapter, the book shows how blockchain technology can enhance privacy in IoT applications by enabling users to control their data and decide who has access to it. In this case, IoT devices can authenticate and communicate with each other securely without the need for centralized intermediaries or third-party providers. Moreover, the book shows how blockchain and AI can be combined to create more intelligent and efficient data processing systems. By using blockchain technology to securely store and share data, AI algorithms can have access to more accurate and trustworthy data,

enabling them to make better decisions and predictions. The book also studies advanced use cases of O-RAN, healthcare, and blockchain standardization by different regulatory bodies.

In conclusion, the potential for blockchain technology in many fields of technology is immense, and this book lays down the ground for some of the most innovative sectors. As blockchain adoption continues to grow, it will be important for academic and industrial researchers to have a solid understanding of this technology and how it can be applied to their respective fields. Whether you are a student, entrepreneur, or industry professional, this book will equip you with the knowledge and tools necessary to stay ahead of the curve in this rapidly evolving field.

Finally, the editors are prominent researchers from world-class industries and academic institutions. Their selection of the topics is engaging and thought-provoking, while also ensuring that the content is informative and well-researched. They understand the importance of balancing current trends with timeless themes that will have lasting relevance as well as opportunities for industrial adoption.

Prof Mischa Dohler
VP Emerging Technologies
Ericsson Inc, Silicon Valley, US

Advisory Board FCC (TAC) & Ofcom (Spectrum)
Visiting Professor, King's College London, UK
Fellow IEEE, RAEng, Royal Society of Arts

Preface

The digital transformation has changed the way we store and process data across multiple industries once and forever. In today's world, most of the data is stored in the cloud, and users from across the planet can access those repositories assuming they have the right authentication. However, it is important to have technology that could provide a record for data alteration to enable us to understand what the changes could be to stored data and who could have done them. In the first instance, it is assumed that such a technology could guarantee data integrity and provide alternative storage for unaltered data copies. The emergence of blockchain was an enabler for more advanced and complex approaches to store and secure digital data. Since blockchain started in 1991 as an open ledger, use cases have evolved from preventing information change with prior agreement from all data users to blocks with unique hash identifiers linked to chain of records. The latter evolution enabled verification and traceability of multistep transactions at higher processing rates with reduced compliance costs. All these features helped to support employing blockchains in platforms that manage contracts and audit the origins of a product. Therefore, blockchain has become the key technology for many industries and use cases, such as securely storing of casted votes in voting platforms, visualizing the status of properties and deeds through full access to their history for real estate agencies, and most remarkably, banking services and crypto currencies.

By definition, blockchain framework is a software solution that compresses different modules and components in a decentralized fashion that operates on the basis of smart contracts. Given its immutable public ledger, blockchain-recorded transactions cannot be overwritten and they are secured by encryption mechanisms. Therefore, any transaction will be added as a new entry to the ledger upon source authentication. This also makes blockchain a potential key technology for governmental systems that help individuals securely access official services. Clearly, blockchain technology has the perspective of becoming an

integrated element of many systems including health care, vehicular networking, cellular communications. This demonstrates the importance of analyzing the technology from the vertical's point of view and how the technology could be further advanced to meet the users needs. Moreover, the unprecedented levels of connectivity and programmability between humans, machines, and services further leverage the benefits of blockchain. This helps administer an automated fraud system that pairs infrastructure and defines shareholder rights. Obviously, the financial sector would become the most promising user of this technology, considering payment transfers and regulatory oversights.

Today, an entire crypto industry has evolved using the blockchain framework. The wide adoption of blockchain to facilitate secure transactions of billions of dollars proves the need to investigate this technology closer from a service perspective, as well as how it could influence other industries. The opportunity for further adoption increases with the cloudification movement and the attachment of billions of users to the cloud data platforms as their backend storage. This means that blockchain could evolve to become the basic element of many new platforms that we use in our day-to-day engagements. Therefore, this book intends to leverage new understandings of the blockchain technology by identifying the embedded modules for development and use cases for new deployments. To achieve this mission, we provide insights into the blockchain concepts but start moving quickly to address the most relevant modules for development and use cases. The provided solutions are supported by explanations for the conceptual framework for operation, workflow within the bigger ecosystem for the industry, and verification results. The given ideas are meant to create new understandings of the scope of blockchain technology and how it will evolve over the coming years to support new services.

The book is addressed to both academic and industrial communities. Book readers can go through the basics of the blockchain technology and recognize how it can be used on platforms, thus allowing them full understanding of the entire ecosystem. For the academic domain, this book will help postgraduate and faculty researchers identify the key elements of blockchain that require further analysis and evolution. For example, distributed storage, security and trust, and permissioned blockchain are all topics on which scholars can dive deep into new enablers that could improve the blockchain operations and performance. For industry, the book provides many examples of blockchain deployment in systems that serve the Internet-of-Things (IoT), cybersecurity, artificial intelligence, healthcare, sixth generation (6G) communications, etc. This would be a valuable source for system designers, field engineers, architects, and chief technology officers who are willing to create solutions for various verticals. Finally, the book is concluded with a chapter that surveys the previous and ongoing standard activities around the blockchain. This chapter would help readers, especially from standardization

bodies, to have a comprehensive understanding of the standard status of this technology and how they intend to move forward in the coming years.

Since blockchain technology is still a relatively new and rapidly evolving field, there are enormous opportunities for further evolution in its architecture and integrated use cases. For example, improving interoperability will allow different blockchain networks to communicate with each other for a more comprehensive share of information and connectivity. Similarly, scalability solutions can significantly increase the number of transactions they can process per second. Privacy-focused blockchains and techniques such as zero-knowledge proofs and homomorphic encryption could also provide greater confidentiality. The energy consumption required for some blockchain systems is a concern given the growing cost of energy and environmental concerns. While technology is evolving, we should also expect new governance models to regulate these systems. Finally, blockchain could be applied to many new fields that could change the world we live in. In this book, we provide the conceptual thoughts for all those evolutions to help accelerate innovations.

Finally, we hope this book will become an informative reference for researchers and a guidance work manual for developers to create innovative solutions around blockchain. We would also like to thank all our colleagues who contributed to this project. This book is simply an effort to advance technology and create better visions for future platforms that improve human life.

Anwer Al-Dulaimi, Octavia A. Dobre, and Chih-Lin I

1

Introduction

Anwer Al-Dulaimi[1], Octavia A. Dobre[2], and Chih-Lin I[3]

[1] *5G Center of Excellence, EXFO, Montreal, Canada*
[2] *Faculty of Engineering and Applied Science, Memorial University of Newfoundland, St. John's, Canada*
[3] *China Mobile Research Institute, China*

1.1 Exploring Blockchain Technology

The blockchain is composed of a distributed database where digital pieces of information are made up in form of blocks that are stored in chains of public datasets. These blocks store information about: (i) transactions like the date, time, and value number of purchases; (ii) records of participations in transactions; and (iii) a unique code called a "hash" that distinguishes a block from another one [1]. A single block on the blockchain can store up to 1 MB of data allowing it to store a few thousand transactions, depending on the size of these operations. The state-of-the-art for blockchain consists of multiple blocks strung together. Therefore, any new data stored by any block will be added to the blockchains. On the other hand, to add a new block, it is necessary for a transaction to have occurred, verified by a network of computers, stored in a block, and that block to have been assigned a unique hash. Once a block is added to the blockchain, it becomes publicly available for anyone to view and that is what make data safe and not altered. For example, Bitcoin's blockchain allows everyone have access to transaction data, along with information about when ("Time"), where ("Height"), and by who ("Relayed By") the block was added to the blockchain [2].

To start, let's clarify the difference between public blockchain and private or federated blockchains. Public blockchains, like Bitcoin and Ethereum, are open to anyone who wants to participate in the network. They are decentralized and allow anyone to create and validate transactions without the need for permission or trust from a centralized authority, as shown in Figure 1.1. The security of the network is ensured through consensus mechanisms that incentivize participants to act in the best interest of the network. On the other hand, private or federated

Blockchains: Empowering Technologies and Industrial Applications, First Edition.
Edited by Anwer Al-Dulaimi, Octavia A. Dobre, and Chih-Lin I.

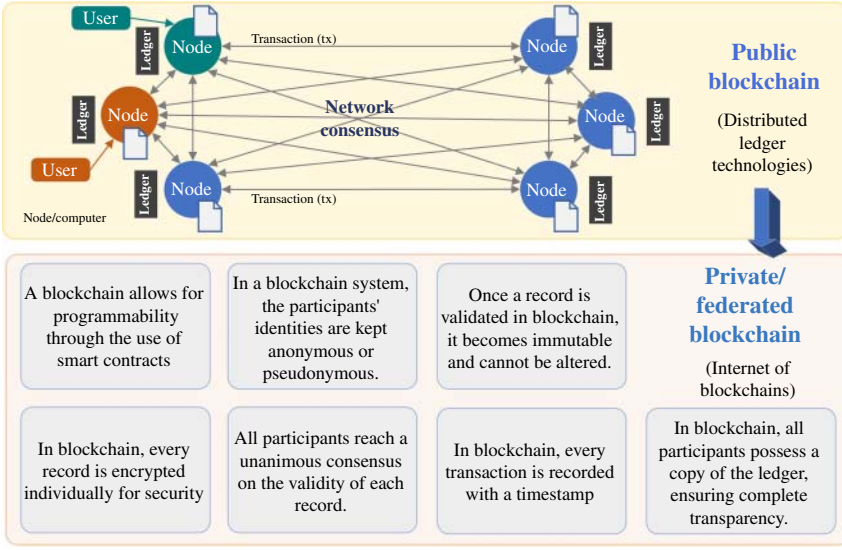


Figure 1.1 Blockchains & Distributed Ledger Technologies.

blockchains are controlled by a group of trusted entities who have been granted access to the network. These blockchains are often used in enterprise settings, where there is a need for greater control over the network and its participants. The nodes on the network are typically operated by known entities, such as businesses or organizations, and transactions are validated by a consensus mechanism agreed upon by those entities. The key difference between public and private or federated blockchains is the level of openness and accessibility of the network. Public blockchains are designed to be open to anyone, while private or federated blockchains are more restrictive and require permission to access. Additionally, public blockchains are often associated with cryptocurrencies and have a greater emphasis on security, while private or federated blockchains prioritize efficiency and control [3].

Each node or computer in the blockchain network maintains a local copy of the blockchain, which means that there are thousands or even millions of copies of the same blockchain. This distribution of blockchain copies makes the information more difficult to manipulate. Also, changing the contents of an existing block will change its hash code and that will require hackers almost to change every single block add afterwards. Those hash codes are created by a math function that transforms digital information into a string of numbers and letters. A hash code will change immediately if information is edited in any way making almost impossible for hackers to alter data. Similarly, blockchains employ "Proof of Work" system for any new participant computers to "prove" that they have done "work"

by solving a complex computational math problem. A computer becomes eligible to add a block only if it was able to solve one of these problems. However, adding blocks to the blockchain (Aka: mining) is very complicated, and the odds of solving a block are low. This is due to the design of the Bitcoin network's consensus mechanism, which is based on proof-of-work (PoW) [4]. In the Bitcoin network, miners compete to solve a complex mathematical problem called a hash function. The hash function is designed to be difficult to solve, requiring a significant amount of computational power and time. The first miner to solve the hash function is rewarded with a set amount of newly minted bitcoins and any transaction fees included in the block. The difficulty of the hash function is adjusted periodically to ensure that new blocks are added to the blockchain at a predictable rate. As more miners join the network and compete to solve the hash function, the difficulty is increased to maintain a consistent block time. The odds of solving a block on the Bitcoin network depend on several factors, including the current difficulty level, the amount of computational power dedicated to mining, and the randomness of the hash function. As of March 2023, the odds of a single mining node on the Bitcoin network solving a block and earning the block reward are estimated to be around 1 in 15 trillion attempts. However, it is important to note that the difficulty of mining new blocks is not the same across all blockchain networks. Some networks, such as Ethereum, use a different consensus mechanism called proof-of-stake (PoS) [5], which does not require the same level of computational power as PoW [6]. In a PoS system, block validators are chosen based on their stake in the network rather than their computational power, which can make the process of adding blocks to the blockchain less computationally intensive.

To understand the blockchain technology, it can be helpful to show the milestones to build the underlying components and functions of a sample blockchain. A typical process may include the following steps:

- **Define the target design for the blockchain**: Decide on the structure of the blockchain, including the block size, block interval time, consensus mechanism, and data storage format. This involved choosing the right programming language and development platform.
- **Write the code**: Use the chosen programming language to write the code for the blockchain. This includes creating the data structures, writing the smart contracts, and implementing the consensus algorithm.
- **Test the code**: This step will ensure that written code works as intended. This includes checking for bugs, verifying that the consensus mechanism is working correctly, and testing the smart contracts.
- **Deploy the blockchain**: Once the code has been tested and verified, it can be deployed on a network. This may involve any public blockchain network like Ethereum or creating a specialized private network.

- **Monitor the blockchain**: Once the blockchain is deployed, monitor it to ensure that it continues to function correctly. However, updates or modifications might be necessary on regular basis to address any issues that arise during operation.

Building a blockchain can be a complex undertaking that requires careful consideration of the use case and adherence to best practices in blockchain development. However, blockchain software development may consider prioritizing security at every stage through regular security audits and vulnerability testing. To test for vulnerabilities, it is necessary to perform various tests on the blockchain network and smart contracts. The tests may include penetration testing, fuzz testing, and code review [7]. Each of those tests will help to improve blockchain resiliency. For example, penetration testing involves simulating an attack on the system to identify potential weaknesses in the network, while fuzz testing involves providing unexpected inputs to the system to test how it responds to different inputs. Finally, code review involves analyzing the blockchain code to identify any vulnerabilities or weaknesses. Typically, vulnerability testing will be conducted regularly to identify and fix any potential security issues or defects that emerge during normal operations.

In summary, Blockchain technology has both short-term and long-term impacts on various industries. For short term, the implementation of blockchain has enabled companies to reduce costs and increase efficiency. By eliminating the need for intermediaries in transactions, blockchain has reduced transaction fees, processing times, and the risk of errors or fraud. It has also allowed for increased transparency, accountability, and traceability in supply chains, which can improve product quality and safety. For long term, blockchain has the potential to transform industries by enabling new business models and disrupting traditional ones. For example, blockchain-based decentralized marketplaces can allow for peer-to-peer transactions without the need for middlemen, while smart contracts can automate contractual agreements and eliminate the need for lawyers or other intermediaries. Overall, the impact of blockchain on industry will continue to evolve as the technology advances and more use cases are discovered.

1.2 Developing and Testing Blockchains: Software Development Approach

There are several software methods used to develop blockchain technology, some of the most well-known have been explained in Section 1.1 such as PoW and PoS, while delegated proof-of-stake is abbreviated as DPoS. As mentioned, PoW

is the original method used in the development of Bitcoin and involves miners solving complex mathematical problems to validate transactions and create new blocks on the blockchain. PoS, on the other hand, uses a consensus algorithm where validators must hold a certain amount of the cryptocurrency in order to validate transactions and create new blocks. DPoS is a variant of PoS that allows users to vote for delegates who can validate transactions and create new blocks on their behalf [8]. Other notable software methods used in blockchain development include Byzantine fault tolerance (BFT) and directed acyclic graph (DAG). BFT is a consensus algorithm that ensures the integrity of the blockchain even if some nodes on the network fail or are malicious [9]. DAG, on the other hand, uses a different structure than traditional blockchains and allows for more scalability and faster transactions [10]. Ultimately, the choice of software method used to develop a blockchain will depend on factors such as the goals of the project, the level of security required, and the size and complexity of the network [11]. These factors are carefully considered when choosing the most appropriate software method to ensure the success of blockchain project.

While conducting blockchain development, it is important to use specified test procedures in verifying the functionality of individual code modules. The test procedure typically involves writing test cases that simulate various inputs and outputs for each module, and then running these tests to ensure that the module behaves as expected. This can help identify any errors or bugs in the code early on, before the module is integrated into the larger blockchain system. Test procedures are particularly important in blockchain development, as errors in code can have significant consequences, such as security vulnerabilities or transaction errors. Figure 1.2 shows the key testing process that developers can use to verify the reliability and stability of code applications. The testing procedures are mapped to each module block to ensure that each function is developed as intended to operate in real-world deployments.

The computational resources needed for blockchain testing, in post-development validations, will depend on several factors, such as the complexity of the blockchain application, the type of testing being performed, and the size of the test network. In general, testing a blockchain application can be resource-intensive, as it involves running multiple nodes and simulating various scenarios to ensure the application's performance and security. To characterize the computational resources needed for blockchain normal operations, there are several factors that could be taken into consideration such as the specific blockchain network, the number of nodes participating in the network, the size of the blockchain, and the complexity of the consensus algorithm used. Typically, blockchain nodes require a computer with enough processing power, memory, and storage to

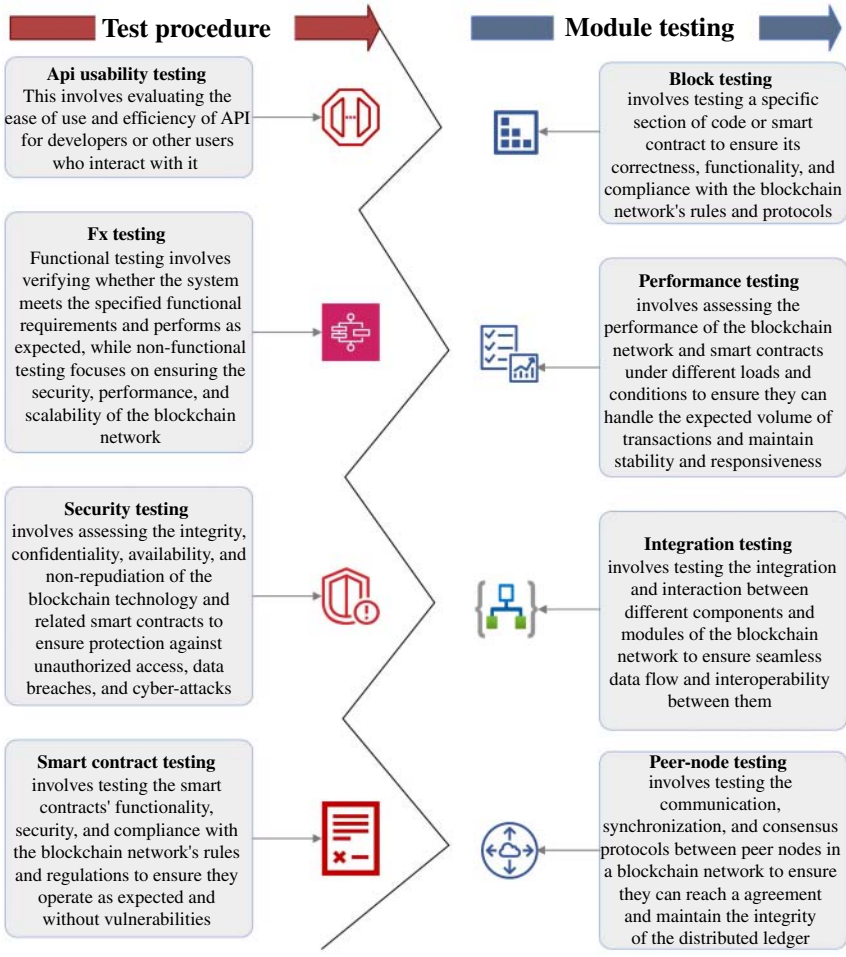


Figure 1.2 Testing Procedures and Relevant Blockchain Modules.

participate in the network and validate transactions. Some blockchain networks may also require specialized hardware, such as GPUs or ASICs, for mining new blocks. For example, Bitcoin's blockchain network requires nodes to have at least 2 GB of RAM, a multi-core processor, and around 300 GB of free disk space to store the entire blockchain. Ethereum, on the other hand, recommends a minimum of 4 GB of RAM, a 64-bit processor, and 200 GB of free disk space [12]. In addition to the hardware requirements, blockchain nodes also require a stable internet connection and sufficient bandwidth to send and receive transactions from other nodes in the network.

1.3 Blockchains and Cloud Integration

Cloud systems refer to the use of remote servers hosted on the internet to store, manage, and process data and applications. Instead of relying on local hardware and software resources, cloud systems provide on-demand access to computing power, storage, and other resources via the internet [13]. Cloud systems can be classified into three main categories: Software as a Service (SaaS), Platform as a Service (PaaS), and Infrastructure as a Service (IaaS). SaaS provides access to software applications, while PaaS provides a platform for developing and deploying applications, and IaaS provides virtualized computing resources such as servers, storage, and networking. Cloud services continue to dominate innovations through features such as scalability that allows cloud platforms to scale up or down depending on the needs of the user, accessibility that provide connectivity anywhere anytime to enabling remote work and collaboration, security measures to protect client's data, and automated backups and disaster recovery to ensure that data is not lost in the event of a disaster, etc. The commercial cloud service providers continue to evolve their platforms with new tools for software production and automated testing for telecom, finance, management, and many other applications. Integrating blockchain in cloud platforms can provide enhanced security and transparency in data transactions and storage, as well as improve efficiency and cost-effectiveness by reducing the need for intermediaries and increasing automation. For example, blockchain provides a secure and tamper-proof way of storing and sharing data, which can be useful for sensitive data such as financial transactions, medical records, or personal identity information. On the other hand, cloud computing provides a scalable and flexible way of storing and processing large amounts of data, but it can also be vulnerable to cyberattacks. Therefore, it is necessary to think out of the box and try to integrate blockchains as part of cloud-based systems to reduce the risk of data breaches and cyberattacks.

One approach is to look at creating trustable platforms equipped with blockchains at the cloud service provider sites. In one approach, future trustable computing platforms can employ Tactics, Techniques, and Procedures (TTPs) components as integral part of their design. The TTP provides the mechanism to perform a variety of actions to encounter cyberattacks against the platform such as tactics that can exploit attacker weaknesses, techniques that can neutralize the attack, procedures to counter an attack that may be able to disrupt the procedures themselves. The countering TTPs develop a deep understanding of the attacker capabilities and they are very flexible in adapting to changing circumstances [14]. Embedding blockchain technology into TTP can play a significant role in enhancing the security and effectiveness of hosting platforms. Blockchain technology

can secure TTPs so that only authorized personnel have access to management dashboards. The decentralized nature of the blockchain can also help to prevent unauthorized modifications or tampering of TTPs, ensuring that the information remains accurate and up to date. In addition, blockchain technology can enable the tracking of TTPs across multiple elements and interfaces allowing for real-time updates and coordination between different modules. This can improve the overall effectiveness of cloud operations, enabling faster response times and more effective decision-making. From a design perspective, blockchain-based identity solutions can securely and anonymously manage access to TTPs and other sensitive information without fear of being tracked or compromised. Clearly, blockchain enables new encounter mechanism that protects TTP itself and allows higher resiliency against advanced cyberattacks that targets defensive systems of victim platforms.

Another approach is to combine blockchain with cloud computing to improve cloud features such as scalability. Blockchain technology is designed in nature to be highly scalable and able to handle a large volume of transactions without sacrificing performance or security. Cloud computing also provides scalability by allowing cloud service providers to easily add or remove computing resources as needed. By combining these two technologies, cloud service providers can build highly scalable and efficient systems that can handle a large volume of transactions and data processing. Similarly, integrating blockchain can improve cost-effectiveness through more elastic pricing model where users pay only for the computing resources they need. Since blockchain technology reduces the need for intermediaries, cloud service providers would be able to reduce financial transaction costs by eliminating mediator fees and reducing processing times. In some other examples, blockchain could be the enabler for many other use cases such as:

- **Supply chain management**: Blockchain technology can be used to track the movement of goods along the supply chain, while cloud computing can provide a platform for sharing this information securely between different parties in the supply chain.
- **Decentralized storage**: Cloud computing can be used to provide a decentralized storage platform, where data is stored across multiple nodes in a blockchain network. This can provide enhanced security and reliability, as well as greater privacy for users.
- **Identity management**: Blockchain technology can be used to create a decentralized identity management system, while cloud computing can provide the infrastructure for hosting and managing this system.
- **Smart contracts**: Smart contracts are self-executing contracts with the terms of the agreement directly written into code. Combining blockchain and cloud computing can allow for more efficient execution and automation of these contracts.

- **Decentralized finance (DeFi)**: Blockchain technology can be used to create DeFi applications that operate on a peer-to-peer network, while cloud computing can provide the infrastructure for hosting and managing these applications [15].
- **Internet of Things (IoT)**: Blockchain and cloud computing can be used to create a secure and decentralized platform for managing IoT devices and data, allowing for greater privacy, security, and control for users [16].

Clearly, there are enormous advantages for combining blockchain and cloud computing. This would be a long journey for industry and academic to develop and test new ideas before defining what could be the cloud of the future.

1.4 Blockchain and Mobile Networking

The emergence of the fifth generation (5G) as a cloud-based network has been a game-changer in the evolution of mobile networks. Deploying 5G network functions, applications, and services as cloud-native containers creates complex architectures that can be scaled and updated easily without downtime. This approach builds upon software development features that enable cloud-based applications to become more scalable, resilient, and portable. In the context of 5G, cloud-native development empowers developers to create applications that can leverage the high bandwidth and low latency capabilities of 5G networks, including those that require real-time data processing such as augmented reality, virtual reality, and autonomous vehicles.

5G technology employs Massive-MIMO at the radio access network (RAN) side, enabling the use of multiple antennas at both the transmitter and receiver. This increases the capacity and efficiency of wireless networks. The edge cloud is another computing resource data center that is deployed closer to the edge of the network, the RAN, rather than in centralized data centers. The edge cloud fosters access to user plane traffic, leading to reduced latency and improved quality of experience (QoE) for real-time applications, such as video streaming and online gaming. 5G enables multiple vertical services operating in the form of different virtual networks on top of a physical network infrastructure. This technique is called network slicing, involving separating user services based on the user-requested Slice Service Type (SST), which is tagged to incoming user traffic. In such virtual and fully cloud-native domains, orchestrators are the software components responsible for coordinating and managing network resources and services to meet the requirements of different applications and users. The orchestrator automates processes such as rolling out services, scalability, and service recovery. These operations are performed through an orchestrator

application programming interface (API) that interfaces workflows and processes in a distributed computing environment.

Integrating blockchain technology into 5G and beyond systems is a complex process that requires careful planning and execution. The motivation for this integration is the fact that 5G and beyond are hosted by commercial clouds, which makes it mandatory to secure data exchange, identity management, and embedding smart contracts in data transactions. Therefore, choosing the right blockchain platform is also highly dependent on the targeted use cases for blockchain integration. There are several options to choose from, including public blockchains like Ethereum and private blockchains like Hyperledger Fabric. In such network-blockchain architecture, it may be necessary to determine the nodes, consensus mechanism, and data storage methods required for the blockchain system. In more advanced step, blockchain technology can be integrated with 5G system using two approaches:

- **Improving 5G functional design**: This includes developing APIs for data exchange between the blockchain and 5G systems and ensuring that the security protocols are in place to protect the integrity of the data. This could also see new 5G network functions (NFs) embedded with blockchain to manage the data exchange between NF modules or to secure data transactions between different core NFs or mobile edge clouds (MECs) and Core NFs.
- **Improving 5G operational design**: Integrating blockchain technology with a network orchestrator API is also another interesting approach to manage and coordinate workflows and processes in a distributed computing environment. Blockchain technology provides a decentralized, immutable, and tamper-proof ledger that can be used to record transactions and data in a secure and transparent way. When combined with an orchestrator API, it can provide a powerful tool for managing and coordinating complex workflows and processes. However, it is important to note that implementing such a system can be complex and requires a deep understanding of both blockchain technology and orchestrator APIs.

As technology moves toward sixth-generation (6G) networks, Artificial Intelligence (AI) is expected to be integral part of the telecom and hosting cloud systems leading to more sophisticated, smart, and optimized complex networks. Similarly, combining AI and blockchain technology in 6G networks holds immense potential to revolutionize the way we communicate and exchange data. With AI-powered algorithms, 6G networks can leverage predictive analytics to optimize network performance, anticipate network congestion, and enhance overall user experience. At the same time, blockchain technology can be used to secure and authenticate data exchanges, ensuring data integrity, privacy, and confidentiality. Both AI and blockchain can lead to more efficient and secure 6G networks, enabling a

wide range of innovative applications, such as autonomous vehicles, smart cities, and financial services. However, implementing AI and blockchain in 6G networks also presents significant challenges, such as management of real-time operations, interfacing with AI, and compliance with regulatory frameworks. Therefore, it is essential to develop a robust and standardized structure that can address these challenges and unlock the full potential of AI and blockchain on top of commercial cloud platforms.

1.5 Open Architecture and Blockchains

Open architecture refers to a system or platform that allows for interoperability and integration with other systems and platforms. In the context of software and technology, open architecture is typically characterized by the use of open standards, open protocols, and open interfaces, which enable different components to communicate and work together seamlessly. Open architecture is gaining more attention from mobile operators, particularly in terms of managing access to OTT (over-the-top) services. From a network infrastructure perspective, open architecture can enable greater interoperability with other networks and devices, as well as faster deployment of new services and applications. This style of architecture can also help mobile operators reduce vendor lock-in, as they can choose from a wider range of vendors and solutions that are compatible with open standards.

In terms of services, open architecture can enable mobile operators to offer more innovative and custom services to their customers. Open APIs allow mobile operators to collaborate with third-party developers to create new and unique services that leverage the capabilities of the network. For example, an open architecture approach can enable mobile operators to offer location-based services, such as personalized advertising or emergency services. Open architecture can also lead to greater transparency and collaboration between mobile operators and their customers. By using open standards and interfaces, mobile operators can provide customers with greater visibility into their network performance and usage, as well as more control over their services and data. This can help build trust and loyalty among customers, increase customer satisfaction and retention. Clearly, open architecture can be a key driver of innovation and differentiation for mobile operators since it enables them to leverage the power of collaboration and community-driven development. Blockchains, on the other hand, are decentralized digital ledgers that record transactions and other data in a secure and tamper-proof manner. Blockchains are typically used in the context of cryptocurrencies, such as Bitcoin, but they can also be used for a variety of other applications, such as supply chain management, identity verification, and digital

voting. When it comes to the intersection of open architecture and blockchains, there are a few key points to consider: First, many blockchain platforms are built with open architecture in mind, allowing developers to build on top of them using open standards and protocols. This can lead to greater interoperability and ease of integration with other systems and platforms. Second, some blockchain platforms, such as Ethereum, enable the creation of decentralized applications (DApps) that can run on top of the blockchain [17]. These DApps can be rebuilt using open architecture principles, allowing for greater flexibility and innovation. This approach transforms the concept of mobile network architecture from providing connectivity to traffic to facilitating accessibility to that service traffic.

In the context of mobile networks, the use of DApps in an open architecture can also enhance security to mobile networks by reducing the risk of data breaches and hacking attempts. This is because DApps are designed to operate on a distributed network, which makes it harder for a single point of failure to compromise the entire system. DApps can also streamline processes and reduce inefficiencies in mobile networks. For example, DApps can be used to automate billing processes or facilitate peer-to-peer payments, reducing the need for intermediaries and associated fees. This can provide users with a more seamless and user-friendly experience than traditional mobile applications. DApps has the flexibility to operate on a variety of devices and platforms providing a consistent experience across all connected devices. Clearly, this simplifies the interactions between mobile network infrastructure and 3rd-party providers when delivering real-time services to connected users. Therefore, it is likely to see a significant increase of DApps in mobile networks over the coming years. Open architecture creates more open, transparent, and inclusive system environment that empower users and promote collaboration. This is where industry is heading and this demonstrates more promising future for blockchain technologies in the field of mobile communications.

1.6 Open API and Monetization of Mobile Network Infrastructure

Open API is being promoted as a way to monetize mobile network infrastructure by providing third-party developers with open and standardized access to create and offer value-added services on top of the network. This can lead to new revenue streams for mobile network operators (MNOs) and increase the network's value for both customers and developers. To achieve this, MNOs need to offer APIs that expose network resources and functionality to third-party developers, who

can then create and offer services that utilize these APIs. MNOs can monetize these APIs by charging developers for access, usage, or revenue sharing models.

From blockchain perspective, there could be two different integration scenarios.

1.6.1 Using Blockchain Technology to Tokenize API Access

This involves the creation of a decentralized system for managing API access using digital tokens. Each token can have a unique identifier that is stored on the blockchain and can be used to track its ownership and usage [18]. Users can store their tokens in a digital wallet that is linked to their blockchain account. When a user wants to access an API, they transfer the appropriate amount of tokens from their wallet to the API provider's wallet. The API provider verifies the token transfer and grants access to the API for a specified period of time. The user can then use the API to access the relevant services or data. When the access period expires or the user no longer needs API access, they can redeem their tokens for the original payment or a proportionate refund. As all token transfers and API access events are recorded on the blockchain, operators can easily track the token ownership and usage, as well as detection of any fraudulent activity. This model provides a secure and transparent system for managing API access, while also allowing for easy tracking and auditing of token usage. It can also enable new business models, such as pay-per-use APIs, and provide a more flexible and decentralized alternative to traditional API access models.

1.6.2 Monetize Mobile Network Infrastructure

Another approach is using blockchain technology to monetize mobile network infrastructure considering resources and assets. In such model, mobile network infrastructure can be tokenized, and the tokens can be traded on a blockchain-based marketplace. These tokens can represent network resources such as bandwidth, processing power, and storage. By tokenizing network resources, MNOs can monetize underutilized network resources and provide an additional revenue stream. Obviously, MNOs can use blockchain technology to facilitate decentralized network sharing. By creating a blockchain-based platform, MNOs can allow other operators to access their network infrastructure on a pay-per-use basis. Smart contracts play an essential role here by automating billing and settlement processes. They can be used to automate the charging of network resources and the settlement of payments between different parties involved in the network. This can help to reduce transaction costs and increase the efficiency of the billing and settlement processes. Similarly, MNOs can use

blockchain technology to allow users to monetize their data by sharing it with third-party applications and services. The blockchain can provide transparency and security, ensuring that user data is only used for authorized purposes and that users are compensated fairly for their data. This model for infrastructure monetization offers solutions for deploying Private Network as a Service (PNAS) on top of 5G networks. It also helps to monetize resources using by 3rd-party applications, specifically IoT service providers.

1.7 Resiliency of Current Blockchain Models

Enabling any of current and future use cases running on a technology platform requires system resiliency and immunity against sudden failures. Overall, blockchain technology is designed to be resilient and decentralized, which means that it can continue to operate even if some nodes on the network fail or are compromised. However, the level of resiliency can vary depending on the specific blockchain model being used and there are some factors that could be used to evaluate blockchain resiliency. One of the biggest risks to blockchain resiliency is the 51% attack, where an individual or group gains control of over 50% of the computing power of a blockchain network. This can allow them to manipulate transactions, double-spend coins, and potentially compromise the integrity of the entire blockchain. Other risks include software bugs, which can lead to vulnerabilities and exploits, and human error, where mistakes in coding or configuration can cause significant damage. Additionally, the increasing centralization of blockchain networks due to the concentration of mining power in a few large mining pools can also pose a risk to resiliency. While blockchain technology is designed to be resilient, it is important to be aware of these potential risks and take steps to mitigate them to ensure the continued security and reliability of blockchain-based systems [19, 20].

The network size seems to have implications on resiliency. This is because a larger network can absorb more nodes going offline without compromising the overall network security. Bitcoin, for example, has a large network of nodes that are distributed around the world, which makes it highly resilient. By implementing the right security measures, blockchain platform can demonstrate resiliency. For example, multi-factor authentication and encryption can help prevent unauthorized access to nodes on the network, which can help maintain its resiliency. However, as with any technology, it is important to continuously monitor and improve security measures to ensure that the network remains resilient against potential threats. The quality assurance (QA) plays an important role in ensuring that blockchain-based systems are reliable, secure, and functional [21]. This validation cycle of monitoring, testing, and reporting should take place

on regular if not on real-time basis to identify any defects or vulnerabilities in the code. This includes functional testing, performance testing, security testing, and integration testing. For example, they may test smart contracts to ensure they behave as expected and do not have any vulnerabilities that could be exploited by attackers. This would be normally followed by code review to identify any potential issues in the codebase. In a typical industrialization process, automated testing is scheduled to ensure that blockchain-based systems meet regulatory requirements and standards. Later all testing results and limitations are documented for send-users. Since blockchain technology is anticipated to become a key driving element of many other systems, platforms, and networks, QA is likely to become an ongoing automated procedure with various testing profiles that are deployed per use case. It is also expected that AI will take the lead of this QA process to improve defects identification and probably manage automated fixes that will reshape the blockchains of the future.

1.8 Next Evolution in Blockchain Functions

Blockchain technology has rapidly evolved since the creation of the first blockchain, Bitcoin, in 2009. Since then, blockchain has found applications across various industries, from finance and healthcare to logistics and supply chain management. However, as the technology continues to mature, the question arises - what could be the next evolution in blockchain functions?

One potential area for blockchain's evolution is in the area of interoperability. Currently, most blockchains operate in silos, and there is no easy way to transfer value or data between them. However, interoperability solutions are being developed, such as cross-chain bridges and interoperability protocols, which could enable different blockchains to communicate with each other. This could facilitate seamless transactions between different blockchains and lead to the creation of an interconnected blockchain ecosystem. Another area for potential evolution is the integration of blockchain with other emerging technologies, such as AI, IoT, and edge computing [22]. Blockchain can provide secure and transparent data sharing, which could be leveraged by these technologies to enhance their capabilities. For example, IoT devices can be used to collect data, which can then be securely stored on a blockchain. AI algorithms can then analyze this data and provide insights that can be used to improve business processes. There is also a good opportunity for future evolution in blockchain functions is in the realm of governance and probably to become part of larger platform such as digital twins. Currently, most blockchain networks are governed by decentralized communities of users, which can make decision-making slow and difficult. However, new governance models are being developed that will

allow for more efficient decision-making while still maintaining the decentralized nature of blockchain networks.

Finally, we can expect to see significant advancements in the use cases for blockchain technology in the future. While blockchain networks are already being used for a wide range of applications, from supply chain management to digital identity verification [23], there are still many areas where blockchain technology could be applied. As new use cases are developed, we can expect to see even more sophisticated functions being developed on blockchain networks. However, if these challenges happen, blockchain technology has the potential to revolutionize many industries and change the way we interact with each other online.

1.9 Book Objectives and Organization

The main objective of this book is to provide a thorough understanding of the most recent developments in blockchains from both theoretical and industrial perspectives. The contributions in this book include all blockchain research initiatives that identify and discuss technical challenges as well as potential applications that continue expanding at an astonishing rate. From supply chain management to digital identity verification, blockchain technology has the potential to revolutionize many industries and change the way we interact with each other online. As such, there is a growing need for a new book on blockchains that explores the latest developments in this field and provide readers with a comprehensive view to blockchain use cases. By providing readers with a solid foundation in blockchain technology, the book can help to dispel misconceptions and promote a better understanding of its potential applications. Another important objective of this book is to provide readers with an understanding of the different types of blockchains that currently exist, including a review of public and private blockchains, as well as permissioned and permissionless blockchains. The book also reviews current and emerging use cases for blockchain technology, as well as an exploration of its potential to transform industries such as finance, healthcare, and mobile networking. By identifying the potential applications of blockchain technology, the book can help to inspire entrepreneurs and innovators to explore new ways of leveraging this technology.

Like any technology, blockchain faces technical challenges and this book shows methods to overcome drawbacks in design that could lead to wider adoption of blockchain technology, as well as an exploration of the social and ethical implications of its use. The book provides readers with analysis of the most recent research and development in this field, as well as a review of the latest trends and emerging innovations. By keeping readers up to date on the latest developments

in blockchain technology, this book can help to inspire new ideas and encourage further innovation. Finally, the book provides a discussion of best practices for implementing blockchain technology in real industrial domains that are associated with human life. This include a review of the technical, legal, and regulatory considerations involved in implementing blockchain technology to promote the responsible use of this technology and improve the chances of success for organizations that are considering its adoption. Our goal was always to discuss the technology enablers and provide real-life examples to improve readers' understanding and foster innovation at all levels.

To achieve the above objectives, this book has 11 chapters organized as following:

Chapter 1: Introduction

This chapter aims to educate readers on the functions of blockchain and explore various use cases, as well as futuristic improvements. Through this discussion, readers will gain a better understanding of the potential of blockchain technology.

Chapter 2: Enabling Technologies and Distributed Storage

This chapter examines the variety of storage systems that have been developed to address the shortcomings of centralized storage systems.

Chapter 3: Consensus and Distributed-Transaction Systems

This chapter provides a summary of the key developments that have enabled fault-tolerant consensus to be used in real-world applications. Additionally, it discusses recent developments, particularly within the context of distributed ledger systems and blockchain.

Chapter 4: Security, Privacy, and Trust of Distributed Ledgers Technology

This chapter describes the evolution of distributed databases into blockchain technology and provides guidance on integrating these databases with both new and legacy systems.

Chapter 5: Permissioned Blockchains

This chapter discusses the architecture and optimization of permissioned blockchains, with a focus on improving performance in a business setting.

Chapter 6: Attestation Infrastructures for Automotive Cybersecurity and Vehicular Applications of Blockchains

This chapter discusses automotive cybersecurity and the role of blockchain and distributed ledger technology in this field.

Chapter 7: Blockchains and Internet of Things

This chapter characterizes how the blockchain solution is well-suited to meet the decentralized requirements envisioned in the 5G context.

Chapter 8: Blockchains for Cybersecurity and AI Systems

This chapter examines the cybersecurity vulnerabilities of both public and private blockchain networks.

Chapter 9: 6G Resource Management and Sharing: Blockchain and O-RAN
This chapter explores the potential of blockchain for resource management and sharing in 6G through multiple application scenarios.
Chapter 10: Blockchains for Smart Healthcare Systems
This chapter explores the applications of blockchain in smart healthcare and the InterPlanetary File System (IPFS) for facilitating healthcare service delivery.
Chapter 11: Blockchain Standards
This chapter discusses the ongoing blockchain standards and relevant regulatory bodies responsible for their creation.

We would like to take a moment to express our sincere appreciation to the authors of each chapter in this book. Thank you for your hard work, dedication, and contribution to the field of blockchain. We hope that our readers will find these chapters informative, insightful, and enjoyable to read. Happy reading and exploring the fascinating world of blockchain!

References

1 Lee, S. and Seo, S.-H. (2022). Design of a two layered blockchain-based reputation system in vehicular networks. *IEEE Transactions on Vehicular Technology* 71 (2): 1209–1223.
2 Zhang, Y., Gai, K., Xiao, J. et al. (2022). Blockchain-empowered efficient data sharing in internet of things settings. *IEEE Journal on Selected Areas in Communications* 40 (12): 3422–3436.
3 Pourmajidi, W., Zhang, L., Steinbacher, J. et al. (2023). Immutable log storage as a service on private and public blockchains. *IEEE Transactions on Services Computing* 16 (1): 356–369.
4 Wang, T., Huang, D., and Zhang, S. (2022). Consensus algorithm analysis in blockchain: pow and raft. In: *Wireless Blockchain: Principles, Technologies and Applications*, 27–72. IEEE.
5 Yang, J., Paudel, A., and Gooi, H.B. (2021). Compensation for power loss by a proof-of-stake consortium blockchain microgrid. *IEEE Transactions on Industrial Informatics* 17 (5): 3253–3262.
6 Liu, Y., Wang, K., Lin, Y., and Xu, W. (2019). LightChain: a lightweight blockchain system for industrial internet of things. *IEEE Transactions on Industrial Informatics* 15 (6): 3571–3581.
7 Li, B., Pan, Z., and Hu, T. (2022). ReDefender: detecting reentrancy vulnerabilities in smart contracts automatically. *IEEE Transactions on Reliability* 71 (2): 984–999.

8 Xu, G., Liu, Y., and Khan, P.W. (2020). Improvement of the DPoS consensus mechanism in blockchain based on vague sets. *IEEE Transactions on Industrial Informatics* 16 (6): 4252–4259.

9 Jalalzai, M.M., Feng, C., Busch, C. et al. (2022). The hermes BFT for blockchains. *IEEE Transactions on Dependable and Secure Computing* 19 (6): 3971–3986.

10 Li, L., Huang, D., and Zhang, C. (2023). An efficient DAG blockchain architecture for IoT. *IEEE Internet of Things Journal* 10 (2): 1286–1296.

11 Li, G., Fan, Z.-P., and Wu, X.-Y. (2023). The choice strategy of authentication technology for luxury e-commerce platforms in the blockchain era. *IEEE Transactions on Engineering Management* 70 (3): 1239–1252.

12 "Running A Full Node", BitcoinCore Website. https://bitcoin.org/en/full-node#disable-listening (accessed 17 March 2023).

13 Tekin, N., Acar, A., Ahmet Aris, A. et al. (2022). Energy consumption of on-device machine learning models for IoT intrusion detection. *Internet of Things* 21: 1–13, article ID: 100670.

14 Timmins, J., Knight, S., and Lachine, B. Offensive cyber security trainer for platform management systems. In: *2021 IEEE International Systems Conference (SysCon)*, Vancouver, BC, Canada, vol. 2021, 1–8.

15 Li, Z., Xiao, B., Guo, S., and Yang, Y. (2023). Securing deployed smart contracts and DeFi with distributed TEE cluster. *IEEE Transactions on Parallel and Distributed Systems* 34 (3): 828–842.

16 Ghosh, U., Chakraborty, C., Garg, L., and Srivastava, G. (2022). *Intelligent Internet of Things for Healthcare and Industry*. Springer International Publishing.

17 Yue, K. et al. (2021). A survey of decentralizing applications via blockchain: the 5G and beyond perspective. *IEEE Communications Surveys & Tutorials* 23 (4): 2191–2217.

18 Wang, K.-Y., Lin, G., Kuo, K. et al. (2020). An empirical study of an open ecosystem model for inclusive financial services. In: *2020 IEEE International Conference on Services Computing (SCC)*, Beijing, China, 412–417.

19 Ping, J., Yan, Z., and Chen, S. (2023). A privacy-preserving blockchain-based method to optimize energy trading. *IEEE Transactions on Smart Grid* 14 (2): 1148–1157.

20 Aponte-Novoa, F.A., Orozco, A.L.S., Villanueva-Polanco, R., and Wightman, P. (2021). The 51% attack on blockchains: a mining behavior study. *IEEE Access* 9: 140549–140564.

21 Wu, H., Düdder, B., Wang, L. et al. (2022). Blockchain-based reliable and privacy-aware crowdsourcing with truth and fairness assurance. *IEEE Internet of Things Journal* 9 (5): 3586–3598.

22 Lo, S.K. et al. (2023). Toward trustworthy AI: blockchain-based architecture design for accountability and fairness of federated learning systems. *IEEE Internet of Things Journal* 10 (4): 3276–3284.

23 Sun, Z.-H., Chen, Z., Cao, S., and Ming, X. (2022). Potential requirements and opportunities of blockchain-based industrial IoT in supply chain: a survey. *IEEE Transactions on Computational Social Systems* 9 (5): 1469–1483.

2

Enabling Technologies and Distributed Storage

Sina Rafati Niya[1] *and Burkhard Stiller*[2]

[1] *Department of Informatics IfI, Blockchain and Distributed Ledger Technologies (BDLT), University of Zürich UZH, Zürich, Switzerland*
[2] *Department of Informatics IfI, Communication Systems Group CSG, University of Zürich UZH, Zürich, Switzerland*

2.1 Introduction

Communication systems, digital devices, computers, cellphones, and other technologies depend on data storage systems; thus, throughout the evolution of computer science and Information Technology (IT) data storage has been an imperative requirement of software and applications. The interconnected human life is laying base increasingly for communications and data collection on Internet-connected hardware such as Internet-of-Things (IoT) devices including smartphones. Moreover, the vision of "Internet of Everything" in the fifth generation (5G) of wireless communications and the emerging Web 3 paradigm, where distribution of IT systems and service providing is sought for in a broad range of data-driven applications [1], outline the necessity of data transmission technologies. Hence, the new human life style is dependent on the efficiency and reliability of distributed data storage systems.

Data objects being produced by digital devices are mostly stored within files, which initially were stored in locally hosted databases, i.e. centrally. However, omnipresent data flows between users, applications, and Web sites over the Internet have caused serious concerns with respect to security, transparency, control over the stored data, and availability of such centralized approaches. As a solution of those challenges, computer science and electronic engineering have addressed different data storage mechanisms, resulting in continuous improvements over time with respect to their design, architecture, efficiency, and capacity.

Blockchains: Empowering Technologies and Industrial Applications, First Edition.
Edited by Anwer Al-Dulaimi, Octavia A. Dobre, and Chih-Lin I.

Therefore, this chapter studies the range of storage systems that have revolved around tackling centralized storage systems deficits. The two main categories overviewed here include Cloud Storage systems and distributed storage systems (DSS). While being different in various dimensions, these two paradigms employ distributed file systems (DFS) in their foundation. This approach is backed by distinguishing between distributed systems and consensus-oriented Blockchains (BC), distributed ledgers (DL), and distributed hash tables (DHT). Finally, by outlining widely adopted DSS implementations, a comparison of those DSSs with a set of determined metrics is performed.

2.2 Data Storage

A fundamental expectation from data storage systems is preserving a data object without making any changes on it or without tempering any part of the data. In the centralized data storage world, it is the responsibility of that hardware, operating system, and the application software to maintain the data safe and unchanged. For example, a Portable Document Format (PDF) file including the ASCII characters or an image or the data records in databases shall be kept safe without any random changes. Thus, data storage systems need to ensure users that no changes have occurred on a particular data object by being tamper-proof.

There have been several approaches to guarantee tamper-proofness of storage systems. These approaches include (i) hardware-based mechanisms, e.g. Hardware Security Modules (HSM) which assures the device hardware is not manipulated, (ii) software-based functions, e.g. Hash functions which assure the integrity of data (cf. Section 2.3), and (iii) combinatory approaches, e.g. Physically Unclonable Functions (PUF) that ensure a specific hardware and software is used with no unexpected changes [2].

Additionally, in some cases, data storage systems are expected to offer change traceability by time stamping the updates and changes made on data objects. Traceability shares common semantic in many ways as in the "Git"-based version controlling for files. Programmers mostly use git to monitor the changes on a program code, and if needed, revert to a state in the past.

To offer a reliable ecosystem, centralized approaches, even if efficient, have proven to be the bottleneck and single-point-of-failure. As shown in Figure 2.1, in a centralized storage all nodes are connected to (and depend on) a central entity for storing their data. Hence, a compromised central node could cause essential losses for data owners. Moreover, centralized storage systems hinder efficacy of processing large files, e.g. for Big Data analysis (cf. Section 2.2.1).

To confront the safety and efficiency risks of centralized storage, decentralized storage enables the connection of nodes to more storage hosts. These storage hosts

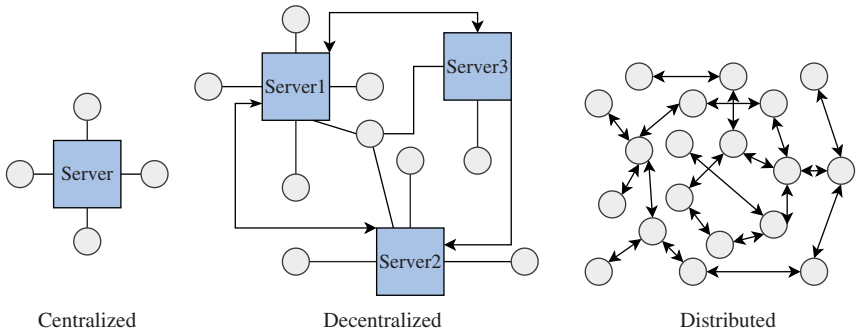


Figure 2.1 Storage network types.

are usually interconnected. Distributed storage networks, however, remove the central storage entities and delegate the storage to all the nodes in the network which are connected to other peers.

Different operating systems on computers and digital devices use their internal clock for time stamping. In decentralized and distributed networks, however, a common notion of time and time stamping has to be integrated explicitly. Otherwise, keeping track of changes would result in non-synchronized decisions.

2.2.1 Distributed File Systems

Over time, centralized data storage systems have evolved by progressing the file system management combined with Operating Systems (OS), such as with the Unix Network File System (NFS) [3]. While OSes enhanced the efficiency of their processing units and also by distributing data storage and processing tasks between multiple computing entities to reduce data transmission delays especially in decentralized systems [4, 5], distributed approaches optimized on operational dimensions. Thus, as shown in Figure 2.2 a range of different types of data storage systems and their scope exists.

A driving factor in decentralizing data storage systems was the growth in data volume, collecting data in/from different applications, leading to "Big Data" and data mining approaches. Big Data analysis demands fast processing on massive date volume, well exceeding TeraByte (TB) sizes and slowly reaching the scale of PetaByte (PB). To store and analyze such huge data amounts, distributed file system frameworks lay the only feasible path to follow.

Many different DFS, such as the Google File System (GFS) [3], NFS, and the Hadoop Distributed File System (HDFS) [3], have been developed during the last three decades. These DFSes have resulted not only in efficiency and performance of storage and processing, but also in transparency (e.g. location,

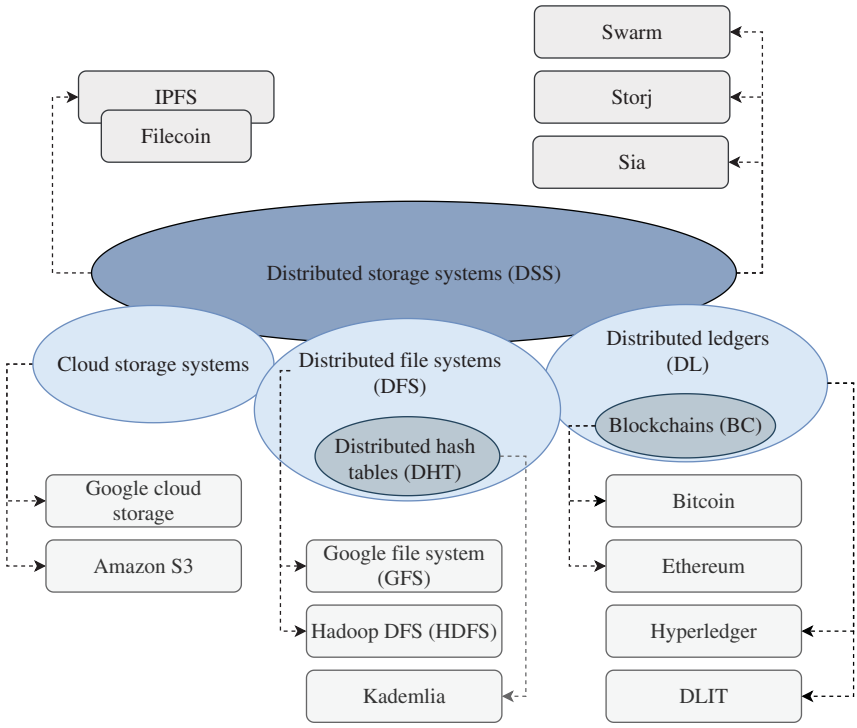


Figure 2.2 Scope of distributed storage systems – The big picture.

naming, access, and replication), redundancy, user mobility, ease of access, and high availability [3].

The distribution of data via DFS is extended in different directions, too. The first dimension addresses computational units, which are placed in the same location, e.g. within the same rack(s) in a building. The second dimension refers to files being hosted remotely, i.e. on Cloud storage systems. While HDFS, GFS, and NFS focus mostly on the first dimensions, cloud storage systems, such as Amazon S3 and Google Cloud Storage, have followed the second path. The third dimension is referred to as data storage, which is made possible without any single entity in charge of the data storage. This is the case with DSS, such as Interplanetary File System (IPFS) [6] and Sia [7, 8].

One specific group of DSSs are Blockchain (BC) or DL-based approaches. These approaches can be divided into two categories: the first category deploys BCs or DLs as their underlying data storage infrastructure. The second category relies on BCs and DLs, but does not use BC or DL consensus mechanism for data validation.

2.2.2 Cloud Storage Systems

Cloud data storage persists data via a cooperating storage service provider, which may use multiple devices in remote locations, i.e. distributed storage with centralized management. Cloud storage providers, such as Amazon Web Services (AWS) S3 [9] and Google Cloud Storage [10, 11], offer a wide range of data storage services: Google Cloud Storage materializes the configuration of user data storage settings and life cycle management features. Thus, an automated transition to lower-cost storage classes happens, whenever the state of data usage meets those criteria users specified, such as reaching a certain age data has been stored or users have stored a newer version of their data. Moreover, users can store data with automatic redundancy options to optimize response time or to create a robust disaster recovery plan.

Cloud storage services offer data object versioning by storing old copies of data, even when they are erased or overwritten. Users of such systems can define minimum retention periods in which data objects must be stored before they may be deleted. Data owners can place a hold on data to prevent deletion.

To protect user data, cloud storage services encrypt data with keys created by users and stored with the storage providers' key management services that users manage. Access control lists (ACL) configured by data owners prevent users from a uniform access to the shared data. Data owners can automate payment requests via cloud storage services that require data accessors to be charged for network, operation, and data retrieval.

User data can be corrupted upon uploading to or downloading from the Cloud storage, e.g. due to noisy network links, memory errors along the path, or software bugs. Cloud storage providers encourage users to employ hash functions before/while transmitting data to the Cloud and after downloading the data to detect corrupted files. For instance, Google Cloud storage recommends its users to employ a CRC32c hash function [12].

Cloud storage systems reduce infrastructure costs and maintenance effort of data storage for enterprises, but they are still not addressing all concerns experienced with centralized approaches. Users may not know the exact location where their data is stored or whether the storage service provider is abusing their data. In case of no Service-Level Agreements (SLA) or a legal contract being concluded between data storage providers and users, users will be left without legal rights to protect their data from being abused. National and international regulations, such as the European General Data Protection Regulation (GDPR) [13], have been defined and enable certain enforcement of user data storage rights and precautions to be met by Cloud service providers.

2.3 Blockchains

Several concepts and algorithms lay the foundation of BC. By considering their core structure and functionality, BCs define overlay networks of P2P nodes and that are comparable to DSSs, since data in these networks is stored in a distributed manner. As shown in Figure 2.3, a BC is organized as a chain of blocks storing transactions (TX) in backward-linked blocks, created and maintained via a network of distributed entities, in which actively block persisting nodes are called "miners." This data stored in blocks contains TXs, sent by BC users to the network, and specific meta data of that block. BCs, in contrast to other DSS types, store data directly inside the chain after verifying their validity by miners.

Any two consecutive blocks on a chain are linked through pointers based on their content's hash values. Hash functions can be used to map data of arbitrary size to fixed-size values. A hash function $h : \{0,1\}^* \rightarrow \{0,1\}^k$ maps an input string $a \in \{0,1\}^*$ of arbitrary length to an output string $b \in \{0,1\}^k$ of fixed length k such that $\forall a \in \{0,1\}^* \exists b \in \{0,1\}^k \wedge k \in \mathbb{N}$. Thus, the key characteristic of a hash function $h : \mathbb{A} \rightarrow \mathbb{B}$ is its resilience to collisions in the co-domain. This means it is very difficult to find two *distinct* input values with the same hash output such that for $a, b \in \{0,1\}^* \wedge a \neq b$ the hash outputs are equal, i.e. $h(a) = h(b)$.

BCs depend on the collision-resistance of hash functions. Thus, for BCs hash functions like SHA2 and SHA3 are applied to a block (cf. below) and the output will be specific to that block. In turn, any malicious or erroneous change in this block's content can be detected. Therefore, each block stores two hashes, i.e. the current hash and the hash of the previous block, its parent block. Thus, BCs are tamper-proof, since the chain of blocks, i.e. the blockchain, serves as the persisted distributed data storage.

2.3.1 Building Elements of Blockchains

Since BCs operate over a P2P overlay network to facilitate communications between miners and clients, the data shared and distributed throughout the network of miners is part of the ledger of TXs persisted within blocks. While clients hold wallets for initiating TXs only, miners may need to store a full copy of the DL. Thus, by running the BC's consensus mechanism across all BC nodes,

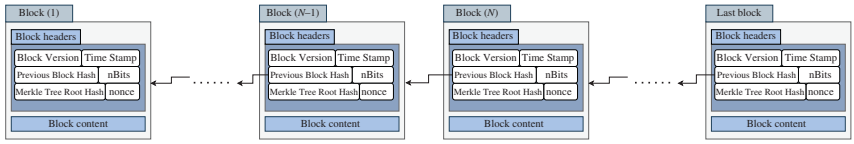


Figure 2.3 Chain of blocks in a Blockchain.

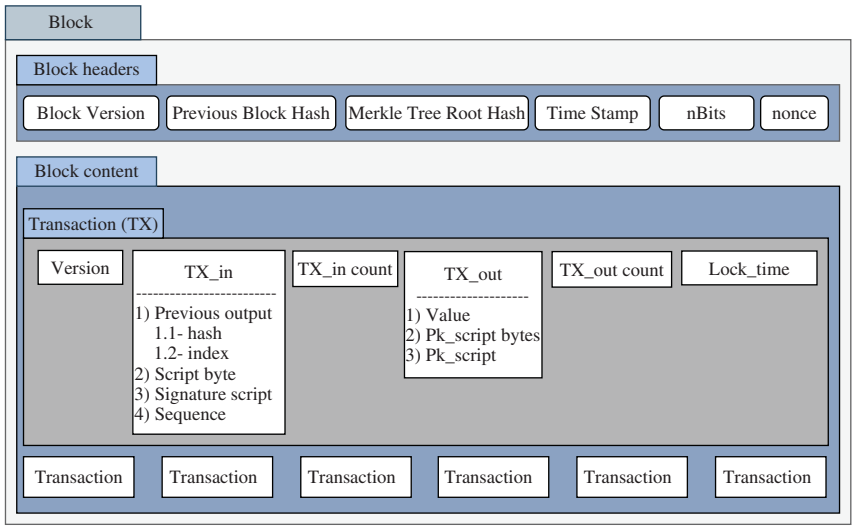


Figure 2.4 An example of a block content in the Bitcoin Blockchain [14].

decentralized rules for the mining processes are adhered to so that the consensus mechanism forces miners to verify TXs, confirm those, and persist the data in the chain, which is considered to be the major difference to other non BC-based DSSs.

In general, as shown in Figure 2.4, each block consists of a header and a content part. A Bitcoin block header includes the following data [14]:

- "version" is a number that indicates which set of block validation rules to be followed. Four versions are available as of today, which refer to a specific forked version of Bitcoin.
- "parent block's hash." As a miner mining a recent block has already received the previous block in the chain, it knows its parent's block hash and has to add that hash in the newly mined block to persist the chain.
- "hash value" of the Merkle tree root. A Merkle tree root is constructed using all TX IDs of TXs in this block. It is the SHA256(SHA256()) of those TXs paired in a binary tree. The ordered list of TXs construct the leaves of this tree. The hash of concatenations of these TXs are paired and concatenated and hashed again until only one root value is created (cf. Figure 2.5). A Merkle root is used to verify the integrity of data in many different DSSs.
- "timestamp" is the time at which this block was mined.
- "nBits" is an encoded representation of the target difficulty threshold of this block. Which means, this block's hash needs to be less than or equal to the value determined via nBits.

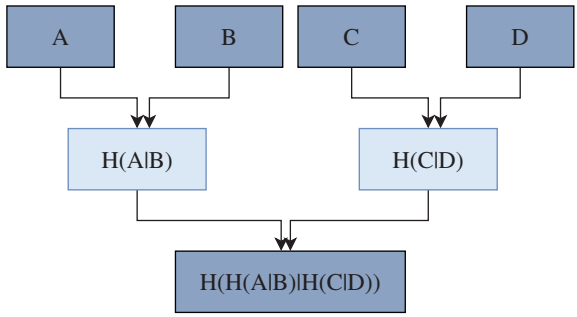


Figure 2.5 A simplified example of the Merkle Tree Construction.

- "nonce" is a randomly selected number used at most once. Miners use the nonce by which the hash of the concatenated block's content and the nonce solves the partial hash collision ("Crypto Puzzle") defined to match the difficulty level specified. The process of finding a suitable nonce and hashing data demands a massive amount of computational power, which leads to the consumption of a high volume of energy [15].

The block's content contains all newly mined TXs. Different BCs employ various types of TXs to enable P2P communications within their ecosystem. Generally, BCs can be labeled as "transactional DSSs," since they employ TXs for changing the state of the BC, i.e. their length. This means that every single change can be accepted only if recorded via a dedicated TX. In this regard, for instance, Bitcoin has implemented 25 TX types [16]. From a high-level perspective, a block in Bitcoin contains several fields in each TX to interpret the data stored within that TX. For instance, the "Raw" TX format used in Bitcoin (cf. Figure 2.4) includes the version, input, output, counters, and a lock time field.

BCs provide data integrity and authenticity by requiring miners to sign TXs using predetermined Public Key Cryptography (PKC) [17]. As generally for asymmetric cryptography, for PKC a mathematically associated pair of public and secret/private keys (PK and SK, respectively) is used to avoid the distribution of encryption material between different parties in a communication. Thus, the SK has to be kept secret and in possession of only one entity. The public key is known publicly, i.e. by the other side of the communication. When a TX is encrypted by a user's PK, it can only be decrypted by its corresponding SK. PKC algorithms, such as RSA and ElGamal [17], have proven to be hard to be broken, i.e. knowing the PK will not lead to the disclosure of the corresponding SK [18].

BCs employ PKC-based schemes known as the Digital Signature Algorithm (DSA) or the Digital Signature Standard [17]. Elliptic curve variants of DSA, i.e. EC-DSA [17], are used within BCs to enable the address generation by applying hash functions additionally (cf. Figure 2.6) and to preserve data integrity, too.

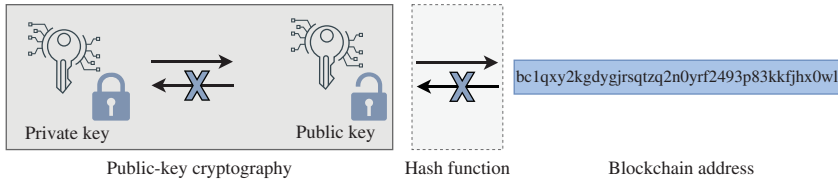


Figure 2.6 Blockchain address generation based on Public Key Cryptography.

2.3.2 Mining in Blockchains

Mining is referred to a task performed by miners, which depends on the consensus mechanism in use. For PoW, which was introduced by Bitcoin, miners have to validate "open TXs" to mine a new block. The validation process is mandatory and it prevents double-spending of coins or malicious behaviors of BC users. As mentioned above, Bitcoin miners are in an ongoing competition to solve these Crypto Puzzles, i.e. finding the nonce, by which the block content hash meets the condition of mining a new block according to the current difficulty level. That is precisely defined by the number of leading zero bits the block hash output starts with. This range or "difficulty" is determined by the PoW consensus algorithm of Bitcoin for that state (height) of the BC. In Bitcoin, all these tasks need to be conducted by miners in less than 10 min within a highly competitive and time-sensitive situation. As a result, a miner who finds the nonce and validates the block faster than others will add its block to the chain and, consequently, earn two rewards, first the reward for mining that new block and second a reward per validated TX.

The Bitcoin network has been criticized for its high power consumption and low scalability. Although many different consensus mechanisms have been introduced [19], such as Proof-of-Stake (PoS), Byzantine Fault Tolerance (BFT), Proof of Authority (PoA), Proof-of-Space-Time (PoST), and hybrid mechanisms, such as in Libra [20] and DLIT [21], all of these consensus mechanisms are intended to overcome PoW's deficits, especially its scalability and privacy concerns. Moreover, enterprise solutions of DL, such as Hyperledger [22], introduced a paradigm shift toward "private" BCs with limited access or contribution rights, but higher scalability by removing or reducing the mining dependency.

2.3.3 Blockchain-Based Data Storage

Besides these operations of BCs as discussed, the storage of data is provided via DL. However, in contrast to distributed data storage alone, the 1991 introduction of time stamping of electronic documents [23] introduced the mark-up functionality of a time stamp, which provided for a unique "label" of a file, such

that no-one else could successfully counter-argue on its validity on generating that label. This determines basic functionality BCs inherited, since the sequence of writes is essential to maintain a backward-linked list in a decentralized manner for which the time stamp serves as a respective proof.

In addition, the manipulation of files as such in a distributed and decentralized file system served as a key concern, until 2002 the Byzantine storage concept had been defined [24]. Such services are essential for distributed systems, which offer availability, data integrity, and redundancy. While redundancy require the maintenance of a software functionality used across all nodes participating, the minimization of such a number of nodes required to achieve a fault-tolerant storage service is operationally essential. Note that a Byzantine error is defined as a condition of a distributed system, where (i) any number of nodes may fail and (ii) non-global information on whether a node failed or not exists. Therefore, DSS based on BCs require this service to be embedded as otherwise the storage is of no practical use.

2.3.4 Blockchain Types

Newer approaches either improve the consensus mechanism's efficiency compared to PoW-based BCs or trade-off selected attributes of BCs, such as being publicly accessible or being permissionless. Thus, these solutions limit the availability of BCs to a specific group in private settings and restrict respective data storage options, too.

Since BCs are backward-linked blocks of data, regardless of their type (public or private), they all share and operate as a DL. Recent studies have distinguished these different types of BCs and DLs, which do fit overall into four categories (cf. Figure 2.7 [25, 26]). Accordingly, public permissionless DLs are specifically determining BCs and all other types "remain" a DL. From a data storage point of view, all DL types store data identically; however, for permissioned versions only selected contributors are granted access from one authority in the DL, usually the DL ecosystem owner.

Although the Bitcoin BC was the first one representing the role model of a "public permissionless BC," many other and different DLs have been introduced in the past decade, including for instance Ethereum, Hyperledger, IOTA, Libra, EOS, Litecoin, Monero, NEO, Polkadot, Ripple, Steem, Stelar, Tether, XTZ, and Zchash, a DLIT [19, 27].

The Ethereum BC has distinguished itself as the first BC to promote distributed autonomous computation, thus, enabling data storage in a distributed setting, too [28]. Ethereum was started by a proprietary PoW-based consensus, but with higher scalability than Bitcoin. Ethereum introduced Smart Contracts (SC) as distributed applications developed like programs, i.e. written by programming

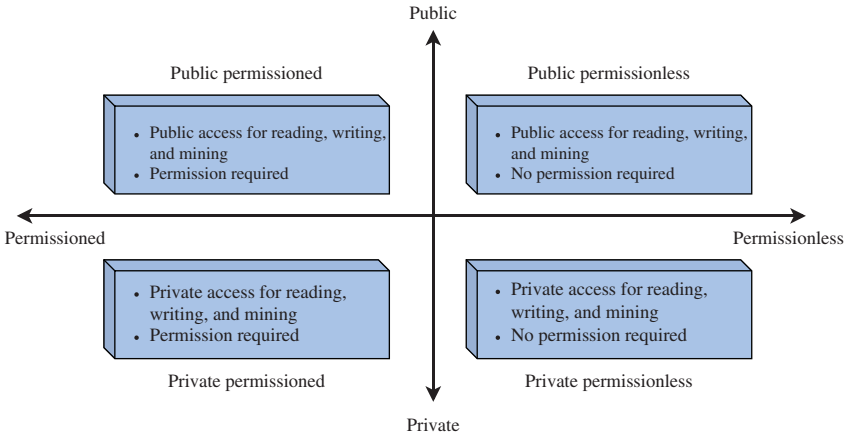


Figure 2.7 Distributed ledger types.

languages, but run decentrally and accessing decentrally stored data. The invention of SCs has been a linchpin for decentralized applications growth in different use cases and distributed computation, which led to the emergence of many different use cases [29]. Ethereum has evolved even further to reach higher scalability in its version 2.0 by employing PoS as its consensus mechanism and sharding techniques [30].

As an instance of DLs, DLIT [21] is a recently introduced research-oriented DL for IoT data. Data persistence is achieved in DLIT by two consensus layers of BFT and PoS, which are maintained by two entities of committees and validators. DLIT reduces or in many cases even removes the need for inter-shard communications. Based on a secure block validation mechanism, DLIT implements a slashing mechanism to fine malicious nodes and divides the TX assignment, validation, verification, and storage responsibility between nodes. As the first DL with such a technique for the storage of IoT data, the employed TX aggregation mechanism reaches a moderate DL growth as a sharded DL.

2.4 Distributed Storage Systems

DSS offer a set of functions and services for the distributed storage of data. These services include storing, retrieving, and maintaining data and, optionally, auditing and accounting of operations for nodes involved. In general, DSSs encompass six different layers, namely (i) Networking, (ii) Consensus, (iii) Data, (iv) Execution, and (v) Application [26, 31], which is now and here augmented by an operationally necessary (f) Management layer.

2.4.1 DSS Layers

In case of an operational DSS three entity types are distinguished for a DSS implementation, i.e. (i) clients or users, (ii) storage hosts, and (iii) owners or managers. Storage nodes need to store and return data upon request. These nodes exist as separate devices with an access to hard discs, who receive rewards by providing data storage service to others. Storage nodes are differentiated and selected to store data based on distinct measures, such as responsiveness (e.g. ping time, up-time, or latency), throughput (e.g. data read or write rate), data upload and download bandwidth, available disk space, geographic location, or reputation, i.e. the history of responding accurately to audits.

From the **Network Layer's** perspective, storage nodes within a DSS construct P2P communications like in BCs. These nodes establish partial or full redundancy of data and metadata storage. DSS Clients and storage nodes shall communicate without any risk of eavesdroppers. Hence, all storage nodes (i.e. peers) on the distributed storage network communicate via standardized networking protocols, which ensure peer reachability, authentication, and privacy of storage nodes and DSS users.

Distributed storage providers need to look up peer network addresses by a unique identifier (ID) such that, given a peer's ID, any other peer can connect to it, similarly to the Internet or Domain Name System (DNS)-based networks, using an overlay network, enabled by, e.g. Chord, Pastry, or Kademlia, which is operating on top of the P2P communication protocol. Data storage security is the crucial characteristic; thus, those communication protocols employed in DSSs ensure that all communications are private by default. Moreover, DSSs need to audit operations within the network and facilitate in case of accounting and payments needs.

The **Consensus Layer** is designed and enforced throughout the DSS network with partially a similar purpose as with the consensus in BCs (cf. Section 2.3). Several implementations of consensus mechanisms, such as the Proof-of-Space (PoSp) in Filecoin (cf. Section 2.4.3), exist to ensure the integrity and reliability of DSSs. The key difference between DSSs and BCs is the employment of light-weight consensus mechanisms and the elimination of TX (data) validation processes in DSSs. Depending on the business model, DSS users may need to pay the storage fee either to DSS owners or storage hosts.

While the data storage specification and TX validation are defined in the Consensus layer, participants of this process are not directly accountable for the data retrieval. In a BC like Bitcoin or Ethereum, miners mine a block and distribute it over the BC network. When a node (client or miner) looks for a block or TX, the block's initial miner is not needed, and any miner in the BC can provide the data. For retrieving a TX or block, its ID or number is enough since every TX

and Block has a unique ID. In practice, on one hand, data retrieval by BC clients is usually conducted via BC explorer applications such as Etherscan [32]. On the other hand, active miners of a BC already have a full copy of the chain locally, and if a (few) blocks were missing in one miner's local database, that miner can send a request to its neighbor miners and ask for them.

The data retrieval processes in consensus-free DSSs like IPFS are different from BCs. In such DSSs, the data may be stored in one or more storage nodes and replicas. When a user sends a data request query, the request is sent to the network and forwarded to the data storage node. Thus, the storage nodes are accountable in retrieving the user data, even if they are not participating in a consensus process.

The **Data Layer** covers all data storage, access, security, and maintenance of data within a DSS. DSSs typically employ DHT to access data. DHTs are distributed key-value tables that point peers to that location, where the data is stored in the distributed network. DHTs are designed to ensure fault tolerance and scalability to a DSS by coordinating each peer with only $O(logn)$ nodes in a network with "n" peers. The key space partitioning in a DHT indicates that peer, which is storing the data and its corresponding key. The key usually is calculated as the hash of the file name. To access the data stored, a client searches files with their name, which is translated in DSSs to a search request by the file's name hash. Every peer in a DSS receives the request and forwards it toward the responsible storage peer(s). Several implementations of DHTs had been performed, such as Kademlia, Chord, Cassandra, TomP2P, and Pastry. Thus, DHTs employ overlay networks for P2P communications, message routing, node discovery, and delivering user queries [33, 34].

Data storage by DSSs needs to be highly available and "durable." Considering the "churn" (i.e. the average rate of storage nodes that go offline or become active again) and ensuring a redundant approach, DSSs need to offer (very) high up-time values and have to guarantee no data loss. The need for durability stems due to the fact that an unknown fraction of storage nodes may go offline temporarily or permanently. Thus, DSSs reach durability by managing the redundancy of the network with approaches such as "erasure coding" to promote high durability and replication with low network expansion rates. Erasure coding is based on encoding a data object into n unique data units, where only by having $k \leq n$ of them at hand, the data object can be rebuilt if needed.

The **Execution Layer** of DSS nodes can be elaborated from two perspectives. Firstly, executing as a DSS node is possible via a broad range of commodity and virtual machines (VM), which may be Cloud-hosted. As long as DSS storage nodes can provide reliable storage space and communicate with other nodes, they can operate as part of the network. However, for those storage hosts that tend to collect rewards, the storage space and income shall be proportional. Secondly, the execution of low-level machine code by DSS nodes has to be possible, for instance,

Ethereum-based nodes need to run an EVM (Ethereum Virtual Machine) to mine blocks and run Smart Contracts [26].

The **Application Layer** covers all applications of DSSs in different areas for using DSSs and interacting with them, such as the applications for writing into and reading from them. For instance, BC client apps and wallets enable easy P2P fund transfers between BC users while visualizing the user account status regarding the summary of TXs and current balance. DSS applications are now covering an ever-growing list of use cases [35]. Thus, different tools have been developed for data retrieval from the corresponding BCs, such as BC explorers. However, accessibility scope and utilization of such tools are determined by the BC types. For example, if a BC is private, no one out of the BC network can use any of such tools to read the data from a particular private BC. Input/Output (I/O) efficiency of DSSs, rely on the application layer as a vital gate for user interactions. Section 2.4.4 overviews a selected set of use cases and applications of DSSs.

Above all, the **Management Layer** exists in most cases for non-BC-based DSSs' management. That is for private or enterprise DSSs such as Storj (cf. Section 2.4.3). All settings for storage host controlling, auditing, storage fee rate setting, configuring host reputation measurement techniques, and distributed management of DSSs, e.g. via Smart Contracts, are part of this management layer.

2.4.2 Distributed Storage Challenges

Next to the determination and discussion of DSS basics and their related reliable ecosystem, DSSs still face challenges.

2.4.2.1 Security

Any decentralized storage must ensure data storage's privacy and security by mitigating an additional layer of complexity and risk associated with data storage on inherently non-trusted nodes. Due to the fact that decentralized storage platforms cannot take similar approaches that centralized approaches can, such as establishing firewalls, decentralized storage must be designed to support end-to-end encryption and enhanced security and privacy by offering an overall layer covering the data storage layers architecture in full.

Data storage platforms need to comply with data storage regulations, such as the European (GDPR) [36] in Europe or the US legislation for the Health Insurance Portability and Accountability Act (HIPAA) [37]. Moreover, data owners may not be interested to see their data traversing and being stored across the globe. In general, data storage systems' customers should be able to evaluate that a data storage is secure and resistant to attacks. In this regard, open-source software provides initial transparency and selected means to verify implementations.

Furthermore, different attacks challenge the integrity of DSSs. While every DSS may show specific security vulnerabilities, certain attacks can, others cannot be easily mitigated.

Common attacks on DSSs include (*a*) hijacking or Spartacus attacks, that happen where a node ID is compromised and its messages are sent to another malicious node. (*b*) Sybil attacks, that happen where the reputation and consensus mechanism of a system is subverted by creating large number of identities in an attempt to disrupt network operation by hijacking or dropping messages. (*c*) Eclipse attacks, that happen where malicious nodes separate and isolate a node or set of nodes in the network graph by ensuring that all of their outbound connections reach only to malicious nodes. (*d*) Honest Geppeto attacks, that happen where the attacker runs many storage nodes on the network, gaining reputation and data over time. After a certain threshold, by performing a hostage attack it owns the data stored in that node or disconnects its storage node(s) from the network. (*e*) Hostage bytes attacks, that happen where malicious nodes refuse to transfer pieces of data fully or partially to clients, to extort additional payments.

2.4.2.2 Reliability

DSSs have to guarantee that user data will not be lost. Thus, durability and resilience in case of failures must be considered. Compared to Cloud storage systems, guaranteeing a high data availability, this level of availability determines a notable challenge for DSSs.

2.4.2.3 Economic Incentives

While the market for Cloud storage providers increased during the last decade, DSSs' situation is not as straightforward as for centralized and cloud-based models. To be successful, DSSs need to consider four groups of users: (i) End users, who would like to see – alongside the distribution – added value, such as higher capacity, durability, security, and performance. (ii) Storage node operators, who need to be economically compensated for their contribution to the maintenance of data storage within the DSS network. These nodes are the key players for a reliable network and its expansion. (iii) Demand providers, including businesses and developers, who need to be convinced that the distributed storage will bring sustainability to their business, and (iv) Network operators, who invest in the DSS development, sustainability of functions, and network maintenance.

Scalability is concerned with the following four categories specifically.

- **Scalability of the underlying BCs or DLs**: Despite selected cryptocurrencies' success, notable performance drawbacks have been experienced with the scalability and energy efficiency of the PoW-based DLs and DSSs [38]. For instance, Bitcoin's energy consumption is due to all miners participating in

the competition to solve the cryptographic puzzle [39]. Such a computational demand causes lower scalability and throughput of PoW-based BCs or DSSs, measured in terms of TPS achieved [40]. Thus, if many TXs are submitted by users simultaneously, which is the situation of large networks with many users, the volume of TXs will exceed the storage capacity of a single block, and some users will have to wait until their TXs are appended to a block and the chain overall, which can lead to extensive delays.

- **Migration from Cloud storage systems**: the implementation of migration functionality for Cloud storage users is difficult for DSS providers, such as Storj. To migrate from Amazon S3 storage, i.e. moving user data to another DSS, the Application Programing Interfaces (API) must be integrated into the DSS application to connect the cloud storage provider to that DSS. Thus, major software engineering tasks are required.
- **Bandwidth**: Uneven distribution of Internet access for DSS users and hosting nodes in different locations impacts the user experience. Since the "download speed" provided by Internet Service Providers (ISP) is a key parameter for ISPs, the upload speed is usually not considered by users, when applying for an Internet subscription. However, to upload data the bandwidth provided needs to offer a sufficiently fast speed, which is even more relevant for distributed data storage network nodes. These nodes need to communicate within the data storage network, not only for data transmissions to users, but also for data maintenance or failure repairs inside the distributed storage network. Hence, a DSS node located in a location with lower access bandwidth can cause network delays and decrease user satisfaction.
- **Data object size**: Data storage providers show different definitions of "large" or "small." While, e.g. Storj considers 4 MB or larger as "large," data size may reach PetaByte, DSSs require to treat large files differently than small ones, especially in optimizing storage, search, and streaming.

2.4.2.4 Coordination

A basic notion and a fundamental requirement of distributed systems is its coordination between nodes. It is required to synchronize the state of these nodes throughout the network. While being an essential element of distributed systems' healthiness, coordination comes with networking costs and delays. In case of DSSs, on one hand, many of the recent BC-based approaches maintain a shared ledger throughout the distributed network by relying on and enforcing all storage nodes' participation. On the other hand, certain approaches such as Storj prefer coordination avoidance-based approaches, trading off BC-based DSSs' advantages for higher performance.

2.4.2.5 Monetization

As DSS providers need to benefit from the storage service provisioning financially, they need to compete with centralized and Cloud Storage systems. Successful DSSs need to offer as reliable and cost-efficient storage solutions as Cloud Storage systems do. Thus, DSS providers need to incentivize reliable and long-lasting commitments of storage hosts by offering paid compensation for their services. The financial profitability and the need for active engagement of storage hosts require storage pricing models, which calculate storage costs depending on a set of metrics, such as storage amount, uptime, networking, operation, and accounting. Considering the storage pricing model in Cloud Storage systems of 0.10 $ per 1 GB month^{-1} [41, 42], monetization of DSS services can be considered a "delicate" issue.

2.4.3 DSS Implementations

While a wider range of implementation of DSSs exists, major players of DSS as of 2021 encounter the following five (cf. Table 2.1):

I. Storj proposes a distributed data storage framework that scales to exabytes of data storage globally. The Storj network stores, encrypts, shards, and distributes data to nodes for storage. This system is designed to prevent breaches by being modular, i.e. consisting of independent components with task-specific jobs. Storj aims at achieving high security, performance, reliability, and cost-efficiency [13]. Metadata servers, object storage servers, Satellites, and clients are the major

Table 2.1 Comparison of selected DSSs.

	Storj	Sia	Swarm	IPFS	Filecoin
Security	TLS, S/Kademlia, PK Hashing	Threefish Hash	Keccak256 Hashing	Encrypted Data Transmissions	Encrypted Data Transmissions
Consensus	Byzantine Altruistic Rational (BAR)	Storage Proofs: Merkle Tree Root	Binary Merkle Tree (BMT) Chunk	—	Storage Proofs
Payment	STORJ Token	Sia Coin	Ether	—	FIL
Execution	Storj Software	File Contracts	EVM and SCs	IPFS Client App	Filecoin Client App

actors in the Storj network. Storj defines a set of nodes as "Satellites" which are responsible for discovering the storage server nodes and keeping track of metadata and accounting and payments. Storj clients and storage server nodes are connected to each other and to the satellite nodes via P2P connections [13]. Object storage servers store the data objects in the system. These nodes need to install and configure the Storj software locally. Storage servers in Storj have to configure disk space and per-Satellite bandwidth allowance. In the node discovery phase, storage nodes will advertise how much bandwidth and hard disc space is available for a new inquiry, and their designated STORJ token wallet address. Metadata servers keep track of data objects by storing their metadata and their location. Storj clients are meant to provide a cohesive view and easy access for users by communicating with data and metadata storage servers.

Storj's strategy in enabling durability and replication is using the Reed–Solomon erasure code. Their approach is considering four different variables of $k \leq m \leq n \leq o$; k and n are the same variables as explained in Section 2.4, "m" is the minimum safe, and "o" represents the optimal value for replicas. The mechanism is as follows. If a satellite node notices a lack of available data object duplicates has dropped below m, it calls a repair request at once, to maintain k or more pieces of a data object. Storj determines o as the number of required copies of data objects, which establishes the desired durability [13].

Storj's consensus mechanism is a Byzantine Altruistic Rational (BAR) model. Every node in Storj is self-managed, while the majority of storage nodes are expected to be rational and a minority to be Byzantine. Storj does not rely on altruistic nodes, i.e. good actors that participate in a proposed protocol even if the rational choice is not profitable. Storj avoids using BFT or Tangle-like (Consensus in IOTA) to avoid coordination requirements of this consensus mechanism and enable history pruning [13]. Storj relies on the gRPC protocol for P2P communications and implements the Transport Layer Security (TLS) protocol for privacy provision and authentication in communications.

Storj employs S/Kademlia and avoids ID hijacking attacks. In this method, IDs are public key hashes of nodes, and messages are signed by the sender's private key. Thus, an attacker cannot reproduce the private key of other nodes and cannot replace others. S/Kademlia proof of work identity generation reduces the Storj vulnerability to Sybil attacks. Further, the Storj reputation system demands an initial vetting stage for its storage nodes to be trusted with substantial data or membership in Kademlia routing tables.

Storj confronts the Eclips attacks by relying on public key hashes and signatures and makes sure the new nodes are connected to at least one reputable node. To prevent the Honest Geppeto attack, Sotrj expects its Satellites to distribute data pieces among as distinct storage nodes as possible [13].

II. Sia is an open-source BC-based DSS. Sia ensures data recovery for its users by redundantly dividing each data file into 30 segments. These segments are encrypted – using the Threefish [43] hash function– and stored in a distributed fashion. Using the Reed–Solomon erasure coding, Sia facilitates the recovery of files even if 20 out of the 30 segments are lost [7, 8].

Sia enforces its users to set up contracts, i.e. "file contracts," between themselves and the data storage hosts. These contracts indicate the service specifications such as uptime commitment and pricing. File contracts are automatically applied using Smart Contracts, initiated by users and stored on the Sia BC, without the need for any intermediaries. Sia employs these Smart Contracts to set up Service-Level Agreements (SLA) [7]. Sia nodes, including users and storage hosts, pay and get paid by Siacoin. The payment is conducted off-chain analog to P2P payment channels. An important design element in Sia is that storage hosts are discouraged from leaving the network, since they have to pay a collateral. Storage hosts in Sia prove their commitment in data storage of files according to file contracts and a submission of "storage proofs," which consist of Merkle tree roots. Using Merkle trees, Sia assures the persistence of the data by a host. Each host has to submit the storage proof to the Sia BC within a specific time span to be paid for.

III. Swarm is the DSS of the Ethereum BC. Swarm introduces a 4-layer design starting with a P2P layer as its transport layer. The next layers include an overlay network layer, data access layer, and the application layer [28, 44]. In Swarm, nodes are running a client application and shape the "underlay" network of Swarm. Each node obtains an address and can listen to the network, dial-in other peers directly, and make live connections. Each node in Swarm is identified by its 256-bit overlay network address, which is determined based on that node's Ethereum address. Swarm enables authentication and integrity verification since Ethereum addresses are generated by hashing the corresponding node's PK. Swarm topology in overlay network is shaped with the Kademlia DHT and routing mechanism [45]. Kademlia guarantees the presence of a path between any two nodes with $O(log(n))$ hops. Swarm establishes quasi-stable peers over Kademlia relying on TCP live channels.

Swarm disassembles large files into storage units of maximum of 4 KByte called "chunks." Chunks in Swarm are content addressed, which means the data storage address is calculated according to the content, more precisely speaking, by its content hash. Uniformity, collision resistance, and irreversibility are the key characteristics of Swarm addressing, which are achieved via its binary Merkle tree chunk (BMT chunk). Chunks are encrypted in Swarm using Keccak256 hashing [44].

Swarm enables redundancy within the network of storage nodes by duplicating the chunks in nearest neighbors of storage nodes. A Chunk in Swarm has redundant retrievability $r + 1$, which is if r storage nodes leave the network, the data

chunk is retrievable by at least one node. Swarm guarantees eventual consistency and confronts storage nodes' churn by duplicating the chunks to maintain the $r + 1$ retrievability in the network. Swarm allows its users to retrieve data via concurrent retrieve requests to avoid potential problems in forwarding processes in sending data from a storage node to users.

IV. Interplanetary File System (IPFS) is a P2P public DSS for storing and accessing data objects, files, websites, and applications. IPFS utilizes content addressing, i.e. assigning addresses and fetching files according to data object content – hash of the content–, instead of its location in the distributed storage network [6].

IPFS employs the Interplanetary Linked Data (IPLD) to translate between hash-linked data structures allowing for its clients to explore data regardless of the underlying protocol. These links between content are embedded within that content address through a Directed Acyclic Graph (DAG), i.e. Merkle DAGs [6]. If a web site is updated, only updated files receive new content addresses. Thus transferring large datasets are made efficient since only the updated parts need to be transferred.

IPFS uses a DHT as a database of keys to values. A DHT is scattered across all the storage nodes (i.e. peers) of IPFS. To find content, clients need to request these peers. IPFS employs Bitswap to reply to these requests.

In IPFS, data transfer is encrypted, but the essential metadata is published publicly. This metadata includes unique node identifiers and addresses of the data blocks. For further privacy provisioning, users can use an IPFS "public gateway" for data transfers.

V. Filecoin is a public DSS. It enables a dynamic distributed data storage marketplace for data storage providers (storage miners) and users. The pricing and availability of the storage is designed to be decentralized. Payments in Filecoin are through the FIL cryptocurrency, the native currency of Filecoin BC. The BC in the Filecoin system is used for storing the TXs between users and storage miners, and the storage proofs are provided by the storage miners [46, 47].

Filecoin is built on IPFS, and every node in Filecoin can communicate with other IPFS nodes. However, not every IPFS node is a Filecoin node, and they cannot be paid by Filecoin users or vice versa. Filecoin nodes do not join and contribute to the IPFS DHT. In IPFS, a piece of data exists as long as there is a node that stores it. Filecoin attempts to incentivize data storage providers to join the IPFS system and provide storage services to others. Filecoin enables a data retrieval market in which storage providers are get paid for caching data for faster retrieval. This type of miners in Filecoin is called "retrieval miners." In this case,

to respond a data retrieval request, user requests do not need to necessarily be sent into the IPFS network since the paid retrieval miners can return user data back to them faster.

In comparison to IPFS, where data storage is only done by volunteers, and there is no control on the centralized pinning process, i.e. storing files, in Filecoin paid members, i.e. storage miners are providing data storage and those miners who do not respect the storage deal will be penalized. Thus, Filecoin offers higher reliability as a DSS to its users. In essence, both IPFS and Filecoin are using the same file format, i.e. IPLD, networking protocol, i.e. libp2p, and data transfer mechanism, i.e. Graphsync and Bitswap. However, Filecoin is designed for higher efficiency for larger files' storage and longer and guaranteed service provision. Filecoin dedicates three types of miners for storage, repair, and retrieval [46, 48].

2.4.4 DSS Use Cases

Throughout the last decade, BCs as consensus-based DSSs have different use cases successful. The initial and main utilization domain of BCs is the financial area, especially the Fintech and Decentralized Finance (DeFi) paradigm. These use cases include P2P payments with cryptocurrencies, stock exchanges, and P2P trading [49], all of which benefited from BCs not only from the removal of intermediaries, but also from an enhanced trust and transparency. However, BC-based applications in the Fintech area suffer from the financial volatility of certain cryptocurrencies and different scalability issues, i.e. the number of TXs a BC can validate and add to the chain within a time unit.

A second domain of measurable potential for BCs includes IoT-integrated applications. BC and IoT integration is employed in Supply Chain Tracking (SCT) [25], health care, environmental monitoring [50], smart cities, smart agriculture, industrial use cases [51], and IoT Data marketplaces [52]. Likewise, many challenges still exist related to the required energy efficiency, the high cost of storing data in BCs, and the lack of user privacy, which impacts those applications' performance and usability [51].

These challenges experienced in using BCs have impacted DSS designers to cautiously integrate a subset of BC characteristics only. Such new approaches employed can be categorized into three main groups: (i) DSSs with data integrity and reliability by dedicating replicas and reputation mechanisms with no mining or TX validation in place (cf. Section 2.4). (ii) Computationally expensive consensus mechanisms are not employed by DSSs by relying on DLs, which are not PoW-based. (iii) dApps employ an integration of DSSs and BCs or DLs.

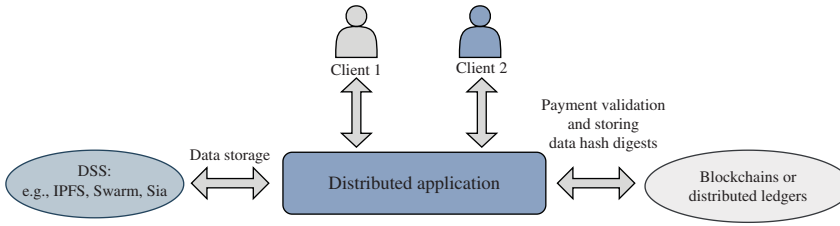


Figure 2.8 Co-existence of DSSs and DLs for dApp data storage.

Therefore, DSS-based dApp clients benefit from BCs solely with respect to the validation of P2P payments and storage of a hash digest of data. As Figure 2.8 indicates, user or application data, e.g. collected by an SCT dApp, is stored on a DSS. In such a case, (i) the dependency on computationally expensive processes (e.g. as mining) is reduced, (ii) data storage costs are reduced, (iii) scalability of the entire system is increased, and (iv) tamper-proofness and time stamping is provided.

Decentralization of data-oriented applications concurrent to the evolution of the Web3 has caused numerous use cases to employ DSSs. For instance, firstly, IPFS [6] is used as the underlying infrastructure of file-sharing applications, such as Arbore, which enables a secure transmission of files, pictures, and documents. Secondly, Enzypt.io allows its users to sell their data by only sharing a link with data buyers using IPFS for data storage. Furthermore, DSSs are being used for collaboration on files (e.g. written documents), version controlling (i.e. distributed version of git), message exchanges, connecting event attendants, video hosting platforms (e.g. P2P video streaming, live streaming), co-hosting large data sets, or parallel Big Data analysis [6].

2.4.4.1 SCT dApps

Although SCT defines a special use case area, SCT dApps use DSSs frequently since now the physical and a digital world are mapped into each other via such dApps. The interaction of producers and users is influenced by the trust and transparency of the Supply Chain monitoring tools and applications, especially the sensors used to measure SCT-essential parameters, such as temperature, CO2 emission, weight, or quality. A Supply Chain monitored by an SCT dApp throughout the production, transportation, and storage processes offers important information to its customers. This data collected is typically used not only for end-users' (i.e. consumers) satisfaction, but also for inter-organizational product controlling and quality measurements.

Data collected by SCT dApps serves, if stored in tamper-proof DSSs, as key evidence of actions did endure throughout their chain of processing steps. The timestamped data includes actions tagged digitally, assigned to a particular

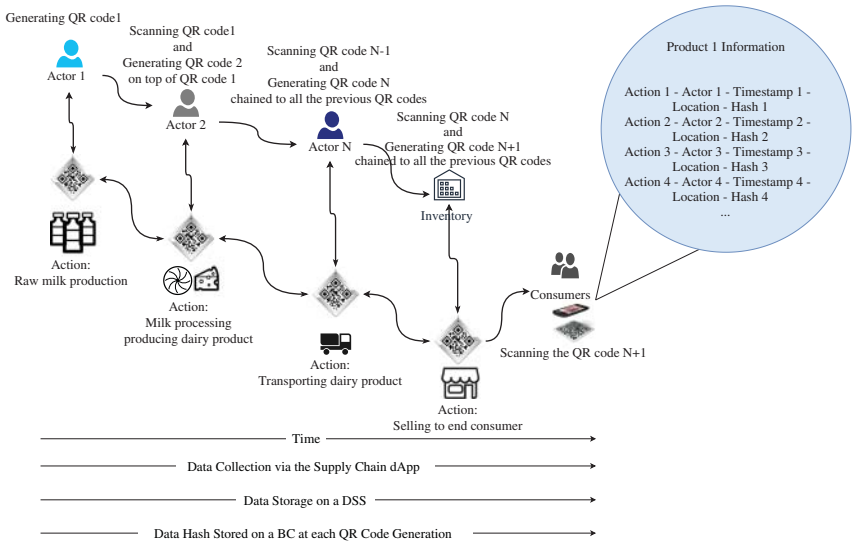


Figure 2.9 Dairy supply chain tracking with a DSS and BC integrated dApp.

product, and the precise location of conducting those actions. Depending on the specific case they can add value to the final product [25, 53].

2.4.4.2 SCT dApp Food Chain Example

A sample SCT dApp design with the complementary use of BCs and DSSs in the Diary product SCT use cases is illustrated by Figure 2.9 [25, 53]. Throughout the Supply Chain, each actor adds the action(s), time, and location to the produced product –preferably in an automated fashion. For ease of usability, a QR code is generated and attached physically to the product at every step. The product may be transported to the next steps for further processes or storage. At each step, an actor reads the information stored in DSS and BC by scanning the QR code attached to the product. The DSS stores the data collected throughout the Supply Chain monitoring, where its hash digest is stored on the BC. Accordingly, high data storage costs are avoided if the hash of data objects is smaller than the information collected. The consumer only scans one QR code by the SCT dApp, which retrieves all the product information and presents them on the user interface.

2.4.5 Performance Evaluation of DSSs

DSS performance can be measured from various angles. The processes undertaken within a DSS include many steps such as transmitting data which cause

(and affected by) network delays, consensus mechanism used by miners and storage providers, replications of the data objects, the infrastructural capability of storage hosts, size of data objects, encryption, and signing data objects. Hence, the efficiency of these processes affects the overall efficiency and performance of DSSs.

A survey on recent empirical performance evaluations of DLs is presented by Dabbagh et al. [54] which identifies the current challenges and achievements of such evaluations. It has been shown in this survey that the size of the DL network and the TXs submitted to the BC, the size of the TXs, and even a DL's client application version (e.g. Geth or Parity of Ethereum), are all impacting the performance of the examined DLs such as Ethereum, Hyperledger, and Libra.

A key performance indicator of DSSs is their throughput, that is, their ability to receive data objects from users and storing them reliably within a time unit calculated by "TX s^{-1}." Performance of DLs and BCs regarding their throughput and latency have been evaluated in different studies [26, 27, 54]. As shown in Table 2.2, the latency and throughput of some DLs such as Bitcoin and Ethereum can be quite limiting as a DSS, while other ones like EOS are offering a higher number of TX handling.

From users' perspective, input/output (I/O) speed, i.e. write/read speed, is a key performance indicator of DSSs. User experience in I/O interactions with DSSs with IPFS has been evaluated in [56] via an imperial approach. Shen et al. [56] sets a network of 8 IPFS storage nodes hosted on Amazon EC2 machines distributed in different countries. Outcomes of this study show how user requests, in terms of size, protocol, and distance, affect the throughput and latency of IPFS. For instance, these outcomes prove that the I/O latency increases with the request size since IPFS divides a data object into multiple blocks, which incurs high disk I/O overhead when storing these blocks to a local storage device. Moreover, IPFS shows almost the same throughput as HTTP for small requests (i.e. request sizes of 1, 4, 16, 64, and 256 KB), whereas for larger requests (i.e. 1, 4, 16, 64 MB), HTTP outperforms IPFS with exponential growth scale. IPFS suffers from higher latency for write and read operations in comparison to HTTP, up to 100% in some cases. Furthermore, this study shows that the geographical distance of storage hosts and the client is also affecting (up to 800 times) the latency of I/O, even for small data

Table 2.2 Comparison of throughput and latency of selected DLs based on [55].

DL	Bitcoin	Ethereum	Litecoin	Monero	Zcash	EOS	Cardano
Throughput (TX s^{-1})	7	15	28	30	27	4000	257
Latency (min)	10	0.25	2.3	2	2	0.5	0.33

Table 2.3 Comparison of throughput and latency of remote read operations in IPFS based on [56].

Data Object Size	16 KB	64 KB	256 KB	1 MB	4 MB	16 MB	64 MB
Throughput ($MB\,s^{-1}$)	0.05	0.1	0.3	0.5	1	1.2	1
Latency (ms)	400	400	600	1	$3 * 10^3$	$8 * 10^3$	$80 * 10^3$

objects. Table 2.3 summarizes the performance evaluation of IPFS for different data sizes.

2.5 The Future of DSS

Each DSS is developed further by its community to cover additional use cases and to offer a competitive advantages above other technologies. Within the context of the DL ecosystems, energy efficiency and consensus optimization for higher scalability and security determine the key goals to reach. Thus, while different and new consensus mechanisms can be expected, at the same time, new and more efficient variants of existing consensus mechanisms, such as PoS and BFT, most likely will be developed, including HotStuff as introduced by Libra [20], or optimized sharding techniques as used by DLIT [21] will be refined. It can be foreseen that this path will be followed by DSS instances that will employ mining nodes, too.

Another course of requirements DSS ecosystems will need to comply to are emerging regulations like GDPR. In this regard, DSSs will need to apply special design and cryptographic algorithms that guarantee data integrity and ownership, while providing user privacy and the right of deletion of the users' personal data. To achieve these goals, DSS developers see valuable synergy toward enabled user privacy, as shown by 0chain [57]. Furthermore, advanced cryptographic primitives may be developed and employed by DSSs to reach energy-efficient operations, especially for IoT-oriented use cases, and to provide controlled data modifications like Chameleon hashes as proposed in [58].

Due to the fact that storage nodes in DSSs are in frequent interaction with their local DataBase (DB), on one hand the efficiency of DBs employed is an imperative factor for higher throughout. These DBs are designed for de-/centralized management of read/write orders. Several proposals such as BigchainDB [59], ChainifyDB [60], IBM HyperLedger Composer [22], and Veritas [61] have started the research and improvement of distributed DBs.

On the other hand, consensus-free DSSs like IPFS follow different goals covering the following ones: storage and transmission of large files, facilitating a decentralized and encrypted Web, and self-archiving based on a WebOS [62].

The key research areas for the IPFS community to achieve these goals include, (i) IPLD (InterPlanetary Linked Data), (ii) the networking layer, (iii) data orchestration through the IPFS Cluster, and (iv) linking service and content directly from DNS by DNSLink.

2.6 Concluding Considerations

The key concerns of centralized storage systems, especially tamper-proofness, time stamping, reliability, and scalability, have been addressed by various DSS by distributing the data objects' storage location and management over decentralized autonomous systems. DSSs' design followed selected notions and approaches taken by DFS to improve the efficiency of large data storage and processing. The set of key requirements and characteristics of DSSs identified include *(a)* trust, *(b)* transparency, *(c)* decentralization, and *(d)* tamper-proofness. These have led to BCs as being highly suitable for a DSS or as an underlying infrastructure for DSSs. However, it is essential to note that current PoW-based BCs cannot be utilized in the same way as distributed databases. This is mainly due to BCs' high data storage costs, delay, low scalability, and restricted privacy considerations. Thus, either DL and DHT-based solutions have been adopted by DSSs or proprietary approaches were developed to replace such BCs.

The adoption of DSSs has been fostered by the emerging Web 3 paradigm. Thus, various DSSs have developed competitive approaches to offer reliable, decentralized, and scalable data storage for a variety of applications. Since different metrics impact the performance of DSSs, such as the underlying peer-to-peer (P2P) overlay network, the employed DHT, consensus mechanisms, security, and privacy, along with measurable technical differences, monetizing strategies proposed by DSSs play a crucial role in their success. First, DSSs are in competition with Cloud Storage systems, which offer highly reliable storage services with low costs. Second, DSSs need to engage storage hosts to guarantee a minimal level of data availability for DSS users, which is the key for their existence.[1]

Acronyms

BC	Blockchain
DL	distributed ledger

1 This work was supported partially by (i) the University of Zürich UZH, Switzerland, and (ii) the European Union Horizon 2020 Research and Innovation Program under Grant Agreement No. 830927, namely the CONCORDIA Project.

DSS	distributed storage system
DeFi	Decentralized Finance
ISP	Internet Service Provider
IoT	Internet-of-Things
GDPR	General Data Protection Regulation
BFT	Byzantine Fault Tolerance
P2P	peer-to-peer
PoW	Proof-of-Work
SC	Smart Contract
Tx	transaction

References

1 Blockchainhub Berlin (2021). Tokenized Networks: Web3, The Stateful Web. https://blockchainhub.net/web3-decentralized-web (accessed 17 May 2023).

2 Niya, S.R., Jeffrey, B., and Stiller, B. (2020). KYoT: self-sovereign IoT identification with a physically unclonable function. *IEEE 45th Conference on Local Computer Networks (LCN 2020)*, 485–490.

3 Rani, L.S., Sudhakar, K., and Kumar, S.V. (2014). Distributed file systems: a survey. *International Journal of Computer Science and Information Technologies (IJCSIT)* 5 (3): 3716–3721.

4 Blomer, J. (2015). A survey on distributed file system technology. *Journal of Physics: Conference Series* 608 (1): 012039.

5 Hac, A. (1985). Distributed file systems-a survey. *ACM SIGOPS Operating Systems Review* 19 (1): 15–18.

6 IPFS (2020). How IPFS works. https://docs.ipfs.io/concepts/how-ipfs-works (accessed 17 May 2023).

7 Sia (2021). Sia combines a peer-to-peer network with blockchain technology to create the world's first decentralized storage platform. https://docs.sia.tech/.

8 Vorick, D. and Champine, L. (2014). *Sia: Simple Decentralized Storage*. Nebulous Inc.

9 AWS (2020). Object storage built to store and retrieve any amount of data from anywhere. https://aws.amazon.com/s3/ (accessed 17 May 2023).

10 Google Cloud (2020). Cloud storage. https://cloud.google.com/storage (accessed 17 May 2023).

11 Ekaba, B. (2019). An overview of google cloud platform services. In: *Building Machine Learning and Deep Learning Models on Google Cloud Platform*, 7–10. Berkeley, CA: Apress. https://doi.org/10.1007/978-1-4842-4470-8_2.

12 GitHub, Inc. (2020). Google/crc32c. https://github.com/google/crc32c (accessed 17 May 2023).

13 Storj Labs Inc. (2018). Storj: a decentralized cloud storage network framework. https://storj.io/whitepaper/ (accessed 17 May 2023).

14 bitcoindeveloper (2020). Block Chain. https://developer.bitcoin.org/reference/block_chain.html (accessed 17 May 2023).

15 McCarthy, N. (2019). Bitcoin Devours More Electricity Than Switzerland. https://www.forbes.com/sites/niallmccarthy/2019/07/08/bitcoin-devours-more-electricity-than-switzerland-infographic/#6f2a0a3321c0 (accessed 17 May 2023).

16 QuantaBytes (2021). A survey of bitcoin transaction types. https://www.quantabytes.com/articles/a-survey-of-bitcoin-transaction-types (accessed 17 May 2023).

17 Bashir, I. (2018). *Mastering Blockchain: Distributed Ledger Technology, Decentralization, and Smart Contracts Explained*. Packt Publishing Ltd.

18 Smart, N.P. (2016). *Cryptography Made Simple*. Springer International Publishing.

19 Monrat, A.A., Schelén, O., and Andersson, K. (2019). A survey of blockchain from the perspectives of applications, challenges, and opportunities. *IEEE Access* 7: 117134–117151.

20 Brühl, V. (2020). Libra- a differentiated view on Facebook's virtual currency project. *Intereconomics* 55 (1): 54–61.

21 Niya, S.R., Beckmann, R., and Stiller, B. (2020). DLIT: a scalable distributed ledger for IoT data. *2nd International Conference on Blockchain Computing and Applications (BCCA 2020)*, 100–107.

22 The Linux Foundation (2020). Hyperledger Wiki. https://wiki.hyperledger.org/ (accessed 17 May 2023).

23 Haber, S. and Stornetta, W.S. (1991). How to time-stamp a digital document. In: vol. 3 (2), 99–111. Berlin, Heidelberg: Springer-Verlag. https://doi.org/10.1007/BF00196791.

24 Mazières, D. and Shasha, D. (2002). Building secure file systems out of Byzantine storage. *Proceedings of the 21st Annual Symposium on Principles of Distributed Computing*, PODC '02, 108–117. New York, NY, USA: Association for Computing Machinery.

25 Niya, S.R., Dordevic, D., Hurschler, M. et al. (2020). A Blockchain-based Supply Chain Tracing for the Swiss Dairy Use Case. *IfI Technical Report No. 2020.07*. Zürich, Switzerland. https://owncloud.csg.uzh.ch/index.php/s/rH6sA25C9JegEHW (accessed 17 May 2023).

26 Fan, C., Ghaemi, S., Khazaei, H., and Musilek, P. (2020). Performance evaluation of blockchain systems: a systematic survey. *IEEE Access* 8: 126927–126950.

27 Hafid, A., Hafid, A.S., and Samih, M. (2020). Scaling blockchains: a comprehensive survey. *IEEE Access* 8: 125244–125262.

28 Dhillon, V., Metcalf, D., and Hooper, M. (2021). *Unpacking Ethereum*, 37–72. Berkeley, CA: Apress. https://doi.org/10.1007/978-1-4842-6534-5_4.

29 Bocek, T. and Stiller, B. (2017). Smart contracts – blockchains in the wings. In: *Digital Marketplaces Unleashed* (ed. C. Linnhoff-Popien, R. Schneider, and M. Zaddach), 169–184. Berlin, Heidelberg: Springer-Verlag.

30 Use Ethereum (2021). A digital future on a global scale. https://ethereum.org/en/eth2/vision/ (accessed 17 May 2023).

31 Benisi, N.Z., Aminian, M., and Javadi, B. (2020). Blockchain-based decentralized storage networks: a survey. *Journal of Network and Computer Applications* 162: 102656.

32 Etherscan (2021). The Ethereum Blockchain Explorer. https://etherscan.io/ (accessed 17 May 2023).

33 Stoica, I., Morris, R., Karger, D. et al. (2001). Chord: a scalable peer-to-peer lookup service for internet applications. *ACM SIGCOMM Computer Communication Review* 31 (4): 149–160.

34 Bocek, T. (2021). TomP2P, A P2P-based High Performance Key-value Pair Storage Library. https://tomp2p.net/ (accessed 17 May 2023).

35 IPFS (2021). Usage ideas and examples. https://docs.ipfs.io/concepts/usage-ideas-examples/ (accessed 17 May 2023).

36 GDPR.EU (2021). What is GDPR, The EU's New Data Protection Law? https://gdpr.eu/what-is-gdpr/ (accessed 17 May 2023).

37 HHS.gov (2021). Summary of the HIPAA security rule. https://www.hhs.gov/hipaa/for-professionals/security/laws-regulations/index.html (accessed 17 May 2023).

38 Gervais, A., Karame, G.O., Wüst, K. et al. (2016). On the security and performance of proof of work blockchains. *ACM SIGSAC Conference on Computer and Communications Security (CCS 2016)*, October 2016, 3–16. Vienna, Austria.

39 Lin, I.-C. and Liao, T.-C. (2017). A survey of blockchain security issues and challenges. *International Journal of Network Security* 19 (5): 653–659.

40 Croman, K., Decker, C., Eyal, I. et al. (2016). on scaling decentralized blockchains. *International Conference on Financial Cryptography and Data Security (FC 2016)*, February 2016, 106–125. Christ Church, Barbados.

41 AWS (2021). Amazon S3 pricing. https://aws.amazon.com/s3/pricing/ (accessed 17 May 2023).

42 GoogleCloud (2021). Cloud storage pricing. https://cloud.google.com/storage/pricing (accessed 17 May 2023).

43 Bhanot, R. and Hans, R. (2015). A review and comparative analysis of various encryption algorithms. *International Journal of Security and Its Applications* 9 (4): 289–306.

44 Tron, V. (2021). The book of swarm. https://gateway.ethswarm.org/bzz/latest.bookofswarm.eth/ (accessed 17 May 2023).

45 Maymounkov, P. and Mazières, D. (2002). Kademlia: a peer-to-peer information system based on the XOR metric. In: *Peer-to-Peer Systems*, IPTPS

2002, Lecture Notes in Computer Science, vol. 2429. Berlin, Heidelberg: Springer-Verlag. https://doi.org/10.1007/3-540-45748-8_5.

46 Filecoin.io (2021). The technology behind IPFS and Filecoin. https://docs.filecoin.io/about-filecoin/ (accessed 17 May 2023).

47 Show, A.K., Kumar, A., Singhal, A. et al. (2020). 6 Blockchain storage. *Blockchain, Big Data and Machine Learning: Trends and Applications*, 141–152.

48 Huang, H., Lin, J., Zheng, B. et al. (2020). When blockchain meets distributed file systems: an overview, challenges, and open issues. *IEEE Access* 8: 50574–50586.

49 Niya, S.R., Allemann, S., Gabay, A., and Stiller, B. (2019). TradeMap: a FINMA-compliant anonymous management of an end-2-end trading market place. *15th International Conference on Network and Service Management (CNSM)*, October 2019, 1–4. Halifax, Canada: IEEE.

50 Niya, S.R., Jha, S.S., Bocek, T., and Stiller, B. (2018). Design and implementation of an automated and decentralized pollution monitoring system with blockchains, smart contracts, and LoRaWAN. *IEEE/IFIP Network Operations and Management Symposium (NOMS 2018)*, April 2018, 1–4. Taipei, Taiwan.

51 Niya, S.R., Schiller, E., Cepilov, I., and Stiller, B. (2020). Standardization of blockchain-based I2oT systems in the I4 era. *IEEE/IFIP Network Operations and Management Symposium (NOMS 2020)*, April 2020, 1–9. Budapest, Hungary.

52 Niya, S.R., Dordevic, D., and Stiller, B. (2020). ITrade: a blockchain-based, self-sovereign, and scalable marketplace for IoT data streams. *IFIP/IEEE International Symposium on Integrated Network Management (IM 2021)*, May 2020. Bordeaux, France: IFIP.

53 Niya, S.R., Dordevic, D., Nabi, A.G. et al. (2019). A platform-independent, generic-purpose, and blockchain-based supply chain tracking. *IEEE International Conference on Blockchain and Cryptocurrency (ICBC 2019)*, May 2019, 11–12.

54 Dabbagh, M., Choo, K.-K.R., Beheshti, A. et al. (2021). A survey of empirical performance evaluation of permissioned blockchain platforms: challenges and opportunities. *Computers & Security* 100: 102078.

55 Bamakan, S.M.H., Motavali, A., and Bondarti, A.B. (2020). A survey of blockchain consensus algorithms performance evaluation criteria. *Expert Systems with Applications* 154: 113385.

56 Shen, J., Li, Y., Zhou, Y., and Wang, X. (2019). Understanding I/O performance of IPFS storage: a client's perspective. *Proceedings of the International Symposium on Quality of Service*, IWQoS '19. New York, NY, USA: Association for Computing Machinery. https://doi.org/10.1145/3326285.3329052.

57 0chain (2021). 0chain protocols. https://zus.network/whitepapers/.

58 Niya, S.R., Willems, J., and Stiller, B. (2021). On-Chain IoT data modification in blockchains. https://arxiv.org/abs/2103.10756.

59 McConaghy, T., Marques, R., Müller, A. et al. (2016). BigchainDB: a scalable blockchain database. *White Paper, BigChainDB*.

60 Schuhknecht, F.M., Sharma, A., Dittrich, J., and Agrawal, D. (2019). ChainifyDB: how to blockchainify any data management system. *CoRR* Vol. abs/1912.04820. http://arxiv.org/abs/1912.04820.

61 Allen, L., Antonopoulos, P., Arasu, A. et al. (2019). Veritas: shared verifiable databases and tables in the cloud. *9th Biennial Conference on Innovative Data Systems Research (CIDR)*. http://cidrdb.org/ (accessed 17 May 2023).

62 GitHub, Inc. (2021). IPFS Project & Working Group Roadmaps Repo. https://github.com/ipfs/roadmap (accessed 17 May 2023).

3

Managing Consensus in Distributed Transaction Systems

Hans Walter Behrens, Kasim Selçuk Candan, and Dragan Boscovic

School of Computing, Informatics, and Decision Systems Engineering, Arizona State University, Tempe, AZ, USA

3.1 Ledgers and Consensus

3.1.1 Distributed Ledgers

A distributed ledger is simply a shared data store that is maintained in a synchronized fashion across multiple sites. While distributed ledgers do not necessarily correspond to financial resources (as in the original meaning of the term "ledger"), several characteristics carry over: in a conventional ledger, the concept of double-entry bookkeeping requires that for any transaction, two accounts (the funding source and the recipient account) must be effected; similarly, in a distributed ledger system, double-spending is disallowed and the injection of new resources into the system is carefully controlled. These desiderata require not just that the ledger's contents be accurate, but that any participating nodes contain the *same* ledger contents. If the data is outdated or out of order, then transaction consistency across the network is at risk. Ensuring that all participants' ledgers match requires that they agree on a common value – that is, that they *come to consensus*.

3.1.2 Consensus

"Consensus" involves multiple processes, which may or may not include faulty and/or malicious parties, needing to agree on some value. While being an apparently simple concept, consensus is also hard to pin down. Merriam-Webster dictionary defines "consensus" simultaneously as "general agreement" (in the sense of "unanimity") as well as "the judgment arrived at by most of those concerned" [1]. These two meanings of consensus (categorized under the same

Blockchains: Empowering Technologies and Industrial Applications, First Edition.
Edited by Anwer Al-Dulaimi, Octavia A. Dobre, and Chih-Lin I.

sense of the word by Merriam-Webster) are clearly distinct from each other. This points to the possibly different formalizations of consensus in different contexts, as well as variations among the protocols to reach consensus. Nevertheless, consensus protocols typically satisfy three characteristics [2]:

- **Termination**: Every correct node eventually decides a value.
- **Integrity**: If every correct node proposes x, they should also decide x.
- **Agreement**: All correct nodes decide the same value.

Here, the term *propose* refers to offering the local value prior to agreement, *decide* refers to updating the local value, and *correct* refers to non-adversarial nodes. Correctness is used in this context because the motivation behind incorrect behavior has no relevance; a node that sends improper readings maliciously or inadvertently are both equally incorrect. Note that proposing and deciding, while useful in a decentralized context, are not the only routes toward agreement. For example, in many systems agreement can be solved dictatorially, using a single source of truth. When two nodes' local copies conflict, they each check with the authority to resolve the disagreement. While this can be efficient, such a trusted source cannot exist in decentralized approaches by definition.

Given the role played by trust (or the lack thereof) in most distributed ledger technologies, it is usual to use the decentralized definition for this reason. However, some distributed ledger technologies, such as those based on private or permissioned ledgers, do not fall under this paradigm. Since they represent an easier subset of agreement, we primarily restrict our investigation to the decentralized case.

In this chapter, we will evaluate the various roles that consensus plays in distributed transaction systems, and especially in those which rely on distributed ledger technologies. We next briefly consider these roles.

3.1.2.1 Consensus for Consistent Data Storage

In distributed ledger systems, each transaction against the ledger may include any participant as either source or recipient. A naive approach might be to treat each transaction individually and to agree upon the ledger state after each transaction is validated. This ensures the correctness and ordering of our transactions but is impractically inefficient since each transaction would take multiple minutes to confirm.

In practice, the consensus and transaction steps are loosely coupled: transactions are packaged into blocks, and then the consensus process agrees on block validity. Indeed, this is the origin of the term blockchain. Data storage systems, especially transaction-oriented ones like databases, must provide additional characteristics than simple agreement.

The most well-known transaction properties offered by databases are referred to as ACID [3]:

- **Atomicity**: A transaction either completely succeeds, or completely fails.
- **Consistency**: A transaction cannot cause the ledger to enter an invalid or corrupt state.
- **Isolation**: Transactions executed concurrently should produce the same output as transactions executed sequentially.
- **Durability**: Once a transaction succeeds, its effect is permanent.

It is easy to see why these properties would be desirable in a distributed ledger – transactions should not be able to deposit funds without withdrawing them from other accounts (atomicity), and the ledger should not forget transactions that have been previously agreed-upon (durability). The ledger should be robust to poorly formed transactions (consistency), and the processing of a block of transactions should be deterministic (isolation).

However, since the ledger is a distributed data store, an additional set of principles common to distributed databases (CAP) must also be considered:

- **Consistency**: Every node's local ledger agrees upon the same values.
- **Availability**: Every transaction can be processed, although not necessarily against the correct ledger value.
- **Partition tolerance**: The network continues to operate even if subsets of nodes cannot communicate with each other.

In the well-known CAP theorem [4], Brewer shows that only two of these three properties can be satisfied at one time (a trilemma). To prevent the ledger from forking, where subsets of nodes disagree on the content and consensus breaks down, most distributed ledgers choose partition tolerance as one of their desiderata. Since connections may drop outside the control of the protocol, this property derives from the mechanisms for resolving such forks when they occur rather than preventing them in the first place.

The other choice varies depending on implementation, but availability is the more common choice – this allows transactions to continue to be processed during the consensus process but also permits (temporary) local divergence from the global ledger. Since these inconsistencies will eventually be resolved during consensus (sometimes called *eventual* consistency), this violates the durability characteristic of ACID. To address this challenge, distributed ledgers include additional mechanisms to resolve this situation, which are discussed in Section 3.8.

In systems that choose consistency, nodes' local ledgers only change during the consensus process. However, since the consensus process can fail if large portions

of the network are unreachable, it is possible for the network to become unable to process new transactions while waiting for consensus to resolve.

3.1.2.2 Consensus for Transaction Ordering

Distributed ledgers, like databases, require their transactions to be linearly serializable to ensure correctness. However, these transactions must be totally ordered across a diverse, distributed system without robust time synchronization, so traditional transaction ordering mechanisms can be ineffective.

Rather than ordering all transactions in the system, the block-based consensus mechanism of most distributed ledgers can be used to address this issue. The transactions within the block can be unordered or ordered heuristically, and the ordering of the blocks themselves comes from the consensus process. That is, when consensus is reached, a transaction ordering emerges implicitly from the blocks.

3.1.2.3 Consensus as a Defense Against Bad Actors

While there are many challenges that can arise by chance in a distributed ledger, they also consider malicious interference as well. Byzantine consensus (described in more detail in Section 3.2.2) explicitly assumes that individual nodes may deviate from the protocol arbitrarily and in coordination with other nodes. Since the Byzantine assumption forms a behavioral superset that also protects against cyberattacks, exploited software vulnerabilities, and inadvertent or unexpected failures, it serves a strong protective foundation.

One of the most well-known uses, cryptocurrencies, provides direct incentivization for adversarial behavior – if transactions can be tampered with or falsified, then value can be transferred to the bad actor as a reward for their actions.

However, the low barrier to entry in public distributed ledgers encourages a diverse mix of participants. As long as malicious actors do not exceed a threshold of total participants, the consensus system serves to enforce normative behavior.

3.1.3 Industrial Case Study

To better illustrate the roles that distributed ledgers can play in real-world industrial contexts, it is useful to consider a motivating practical example. A hotel would like to deploy IoT systems in their parking garage, to automate billing, logging, and actuation [5].

Their system uses autonomous platform-to-platform (M2M) parking reservations and payments without involving human users or intermediaries. It requires a blockchain platform like the one described in [6], and exploits a hybrid blockchain architecture incorporating a private Hyperledger Fabric blockchain and a public blockchain like DASH. The case study encompasses private and

public blockchains and smart access and actuation to facilitate automated parking reservations and payments. More specifically, the Blockchain-Assisted Real-time Transaction Execution and Repository (BARTER) blockchain solution automates, validates, and secures reservations and payments services between the two IoT platforms. In this particular case, Smart Contracts on BARTER enable the hotel IoT platform (SmartHotel) to secure and pay for parking reservations for their guests to the SmartGarage IoT platform each time a new guest is checked-in in the hotel. Similarly, Smart Contracts enable SmartGarage IoT to validate reservations and confirm payments received from the SmartHotel IoT platform. The parking reservations are verified by the Practical Byzantine Fault Tolerance (PBFT) consensus protocol used by HyperLedger Fabric, while payment transactions are verified through a Proof-of-Work (PoW) protocol on the DASH network.

In addition to reservations and payments, the BARTER blockchain solution provides a repository service to store real-time access logs to the SmartGarage. These records are made immutable through PoW consensus protocol.

Further details and resources needed to build and deploy a variant of BARTER M2M services between different IoT platforms are published and available for reuse [7, 8].

3.2 Consensus Protocols, Then and Now

Blockchain systems are only the latest beneficiary of consensus protocols, which first appeared to address the agreement problem [9]. In the intervening time, many improvements, discoveries, and characterizations have proliferated in the literature. In this section, we first briefly summarize several key developments that enabled real-world usage of fault-tolerant consensus. We then follow up with more recent developments that unlocked the concept of an untrusted blockchain. In Figure 3.1, a taxonomy of distributed consensus mechanisms provides a road map for discussion.

3.2.1 State Machine Replication

An important step in consensus protocols came from their structuring as state machines [10, 11]. In this context, each participant maintains a local state that can be permuted by a sequence of commands. Knowledge of these commands (and their ordering) allows for reconstruction of the underlying state. Thus, rather than agreeing upon the underlying data itself, participating nodes must agree upon the ordering of these state transformations, and by doing so may arrive at a common underlying state.

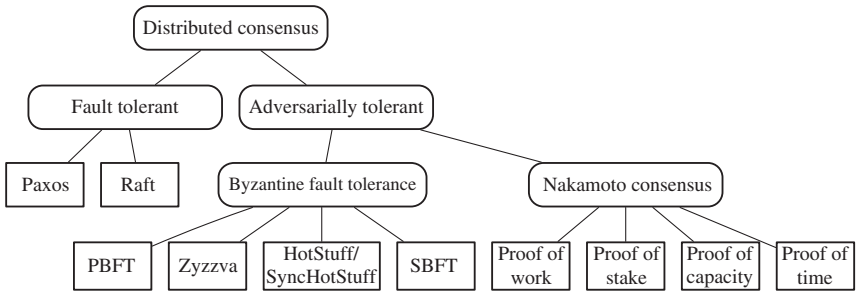


Figure 3.1 A hierarchical taxonomy of distributed consensus approaches discussed. Note that the proof types under Nakamoto consensus represent broad classes of approaches, with many nuanced implementations within each category.

In its most general sense, provable consensus is unsolvable deterministically in asynchronous systems, even with one faulty process [12]. Nevertheless, several practical solutions to the consensus problem have been proposed, relying on various synchrony assumptions [13–15]. One of the first of these was Lamport's Paxos algorithm [14], which relies on a leader-election process to "emulate" synchrony. The leader is initially responsible for proposing a single state transition. The processes communicate among themselves to come to agreement. If the leader fails, then a new election is held to replace the leader. Nevertheless, if an operation is committed by a leader in timestamp t, then all leaders of that come after time t will also agree to the same operation – the safety (i.e. "if a server has applied an entry at a given index to its state machine, no other server will ever apply a different entry for the same index") and liveness of Paxos are based on quorums[1] (subsets of the participants such that any two such subsets share at least one member) that help to ensure that at least some surviving participant retains knowledge of the results in the presence of failures. Using this single-transformation agreement as a building block, participants may produce an ordered set of transactions reasonably efficiently. A similar approach is proposed in Raft [13], which addresses the same requirements and constraints. An important differentiation is Raft's decision to shift from incremental construction of orderings through singly proposed transactions toward a system in which the leader proposes orderings directly. In particular, while Paxos allows any server to be leader, Raft only allows servers with up-to-date entries to become leaders; thus, Raft does not require entries to be exchanged during leader election, leading to improved efficiency in the overall process.

1 While the original Paxos algorithm requires that all quorums intersect, *flexible Paxos (FPaxos)* requires only that quorums from different phases intersect; this enables the protocol to allow improved availability and better efficiency [16].

Yet, while Paxos and Raft are able to preserve the safety property necessary for state machine replication even with no leader or with multiple leaders and even in the presence of asynchrony and crashes, it can guarantee progress only when the leader is unique and can communicate with sufficiently many acceptors in a timely manner. Moreover, these approaches consider only cases exhibiting stochastic failures. A leader may drop out at any time, but no adversarial behavior or collusion is expected or considered. Under this assumption, Paxos uses $2f + 1$ processes to tolerate the benign failure of any f of them.

3.2.2 Byzantine Fault Tolerance

Since they do not consider maliciously faulty (or Byzantine) failures, basic consensus protocols, such as Paxos and Raft, cannot provide the adversarially resilient foundation necessary for a blockchain-based system.

The literature for Byzantine fault tolerance, or BFT, goes back to the original Byzantine Generals problem [17] which introduced the concept of adversarial faults, building on earlier literature addressing stochastic failures. While these initial approaches provided good protection, they relied upon limiting assumptions and did not address the practical scalability of the concomitant communications (Figure 3.2).

A benchmark improvement in Byzantine consensus was PBFT, the 1999 work by Castro and Liskov [18]. This approach does not require synchrony or a trusted proposer, making it suitable for many applications. However, it uses an all-to-all scheme that requires $O(n^2)$ messages in the best case, and $O(n^4)$ in the worst case. This communication complexity means that its scalability is highly dependent on the size of the network, though it is quite robust – the algorithm tolerates f Byzantine processes for $3f + 1$ processes. In 2011, Lamport proposed [19] a Byzantine variant of the ordinary Paxos algorithm, which leverages $2f + 1$ nonfaulty processes emulate the ordinary Paxos algorithm despite the presence of f malicious processes, to derive a Byzantine consensus algorithm with a similar, $3f + 1$, performance.

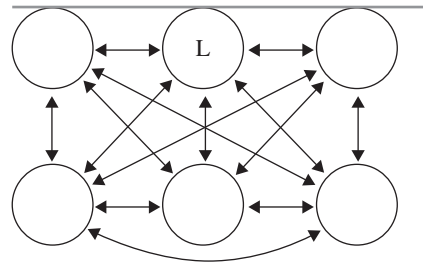


Figure 3.2 Under BFT, a known number of nodes (with a leader L) communicate extensively to double-check their responses and minimize the influence of adversarial collusion, including the leader node themselves.

As consensus became more commonly used in blockchain systems, new research explored methods to improve its scalability. Rather than requiring every node to communicate with every other node through the three-phase commit process used in PBFT, Zyzzyva [20] proposed the concept of intermediate collection nodes to allow for fan-in and fan-out, improving efficiency especially in cases of low contention.

SBFT [21], implicitly Scalable BFT, extends this approach by recognizing the state of the network, and choosing either a "fast" or "slow" path. In cases where there is little contention or asynchrony, the network can come to consensus much more quickly. Then, if the network becomes desynchronized or messages begin to disagree more frequently, a slower but more robust approach to consensus is adopted.

One key drawback of BFT approaches is their reliance on a leader. The process of selecting a node (or replacing a misbehaving node) as a leader can add substantially to the delay in consensus. In fact, even though a malicious leader cannot cause inconsistency, it is not obvious that a malicious leader cannot prevent progress [22]. Hotstuff [23, 24] addresses these limitations by simplifying the leader selection algorithm, at the expense of introducing potential backtracking in the consensus process. In [25], Lamport proposes a leaderless Byzantine agreement algorithm, which replaces the leaders in an ordinary Byzantine Paxos algorithm with a virtual leader that is implemented itself using a synchronous Byzantine agreement algorithm, thereby adding to the cost of Byzantine Paxos algorithm also the cost of the leader agreement process. Note that if the system does not behave synchronously, different servers may choose different virtual-leaders messages – this may prevent progress, but would not cause inconsistency as Byzantine Paxos can tolerate malicious leaders.

A final consideration lies in the assumptions made by most BFT systems. Since these assumptions are well-known, they can be subverted by adversaries in ways that can amplify their impact f above the critical $3f + 1$ threshold. For example, by assuming the communication graph is complete, then a network partition would cause the nodes in the smaller subset to appear (and be treated) as misbehaving or adversarial. Therefore, security claims hold only to the extent that any underlying assumptions can be enforced.

3.2.3 Nakamoto Consensus

Many, if not most, distributed ledgers cannot make the fixed-participant assumption (Section 3.4.3). Since any node can join the network at any time, it cannot prevent Sybil nodes from disrupting the consensus process through the simple expedient of outnumbering correct participants. To address this problem, blockchains rely on a proof mechanism. Consensus based on these types of

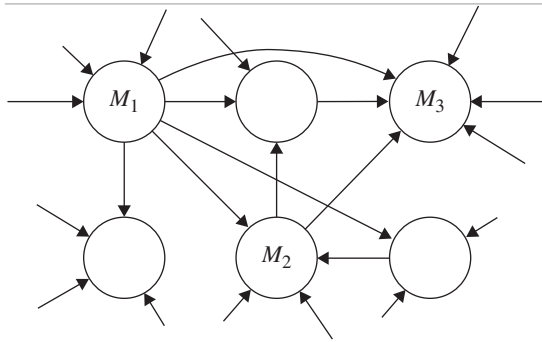


Figure 3.3 In Nakamoto consensus, the borders of the graph are not well-defined, and transactions and blocks may propagate quite widely. In this example, miner M_1 discovered a valid hash for a block of transactions, and broadcasts it to the nodes it knows.

approaches is commonly called permissionless or Nakamoto consensus, named for the proposer of the Bitcoin blockchain.

The role of these proofs is to make the creation of Sybil nodes computationally or economically infeasible and thus secure the consensus through a stochastic mechanism. For example, this might include executing a hash function until the output contains a certain sequence of characters, or randomly choosing a set of nodes from the participant pool (Figure 3.3).

Since proof mechanisms rely on randomness, enforced through cryptographic primitives, creating large numbers of Sybil nodes cannot destabilize the consensus mechanism because their role in the network will be minimal without concrete investment to back up their role.

To read more about the alternative approaches used to meet this goal, please see Section 3.3.

3.2.4 Hybrid Consensus

In recent work, hybrids of BFT and Nakamoto consensus have been explored to combine the advantages of each system. For example, rather than evaluating consensus over the entire network, choosing a smaller subset at random [26] using verifiable random functions allows for fast agreement when rates of adversarial compromise are low.

As we have seen, many emerging systems allow for algorithmic switching between a robust, pessimistic Nakamoto-style consensus mechanism that is relatively slow, and an optimistic approach based on BFT that comes to consensus quickly in favorable conditions.

Alternatively, some approaches such as HotDAG [27] leverage the structure of the network itself to change not only the consensus process but the data being

shared as well. We will discuss the implications of these kinds of structural approaches in Section 3.6.

3.3 Cryptographic Nakamoto Proofs

3.3.1 Proof of Work

Proof of work represents the first and most commonly deployed approach for achieving distributed consensus. Fundamentally, it is based on the concept that reversing one-way functions is computationally difficult. First introduced in 2002 [28], it was initially used as a way to prove a computational investment as a way to combat the rise of spam emails. By taking the content of the email and hashing it, then finding a seed that produces the same hash through brute-force guessing, the email could be signed with the seed. Then, recipients could quickly validate the hashes of the email and the signature to ensure that they match, without having to go through the computationally expensive guessing process needed to generate the signature. This asymmetric difficulty is the core concept of proof of work and is why most proofs-of-work schemes use a collision-resistant hash function [29] as their work function.

The initial application of such schemes to secure a distributed consensus algorithm was proposed by Nakamoto [30], generating the Bitcoin digital currency. In this implementation, the inputs to the hash function consist of the signature of the previous block, as well as an unknown nonce. The length of the target hash can be varied to adjust the difficulty of the work algorithm, with longer hashes taking commensurately more work. The outputs of the hash function are determined based on the number of leading zeros. Participants must then "guess" the nonce to produce the desired output, by simply trying a large number of randomly chosen nonces until one is found that produces the desired output. The node which finds this nonce then receives a reward once it publishes the block nonce, effectively committing the block to the ledger. This guessing process is commonly referred to as "mining" in the distributed ledger domain, as it requires the input of work to produce a financial reward.

However, a wide variety of other implementations also exist, either through the use of a variety of alternative hash algorithms, or by the addition of augmenting data structures such as directed acyclic graphs [31], which produce variations in the computational difficulty, suitability for different hardware, implementation complexity, and ancillary computation features.

One of the main drawbacks of proof-of-work algorithms is the feature that they provide: they require a lot of computation as the tool for validation. Therefore, as the network grows and the number of transactions increases, the computational

demands of the network grow as well. Power draw likewise increases; current estimates place the power usage of the Bitcoin network at the gigawatt-scale [32].

Since the work being done has no practical purpose outside of the validation of the ledger itself, the future scalability and environmental friendliness of such approaches is now in doubt, producing concern among some observers that proof-of-work algorithms may not be suitable as a long-term choice. Some large distributed ledger communities have already begun exploring alternative approaches as candidates for future validation [33, 34].

3.3.2 Proof of Stake

If proof of work serves as the most common choice for distributed ledger consensus, proof of stake is a strong second place. Initially proposed in [35], rather than searching for the solution to a mathematical puzzle that can only be determined through brute force and random chance, the question of validation is instead decided solely through random chance, modified by some weight function [36]. Hybrid approaches with proof of work have also been proposed [37].

The "stake" in proof of stake refers to the digital asset holdings of the participating validation nodes. Individual nodes may choose a portion of their assets to commit to a validation; this is their "stake." Then, when a block is ready to be validated, one of the nodes from the pool of nodes that have committed stakes is chosen randomly to be the validator, weighted by the size of their stake. That is, if five nodes contribute equal stakes, their respective chances of being selected are all equal at 20%, while if four nodes contribute a stake of 10, and the fifth node contributes a stake of 60, then that fifth node has a 60% chance of being selected.

If the selected node behaves correctly and validates the block according to the rules of the network, they keep their stake as well as the block reward. However, if they violate network rules and this violation is detected, their stake is forfeit. Since the size of their stake is directly proportional to their likelihood of selection, miners validating blocks will therefore have the most to lose by subverting the rules.

Computationally, this approach requires almost no power relative to proof-of-work schemes, since the random selection process is quite lightweight. A major motivating factor for the adoption of proof-of-stake algorithms is this benefit, which addresses the environmental and scalability concerns posed by proof-of-work algorithms. It also lowers the barrier to entry to the system, since no special hardware is required to participate in validation. However, since rewards are probabilistically allocated to those who have (or risk) the most assets already, there is some concern that proof-of-stake systems may lead to asset inequality.

Another related issue facing proof-of-stake systems is referred to as the "monopoly" problem. Since the only requirement to participate in the network

is holding digital assets, any user may buy into the system by purchasing those assets using fiat currency. This introduces a risk that a large investor, or group of investors, may be able to pool sufficient resources to control more than half the assets in the system. By doing so, they would then be able to stake their resources and improperly validate transactions, but since they represent more than half the system, consensus would still be reached on the invalid transaction, so they would be able to retain their stake. A prevailing argument against this weakness is that the risk is high to the controlling investor, since detecting such an event through human oversight is simple, and external valuations of the compromised asset would drop rapidly. However, more recent work [38] attempts to formalize and refute these arguments. Other approaches have been proposed [26] to leverage randomness to mitigate these drawbacks.

3.3.2.1 Chain-Based Proof of Stake

An important variant of blockchain-based proof-of-stake algorithms are the chain-based approaches similar to those adopted by Casper [39] and discussed in more detail in Section 3.8. This variant prefers to emphasize availability over consistency by permitting the protocol to generate arbitrarily many forks at each block. This makes it extremely likely that at least one block is proposed during every validity window, but also necessitates additional fork resolution logic when multiple valid blocks are chosen in parallel. Under this approach, the algorithm proceeds as follows:

1. The protocol chooses a foundational block to use as a basis for the upcoming block.
2. The protocol then selects a block producer based on their proportional economic stake in the protocol, and specifies a validity period.
3. The chosen block producer must produce a valid block pointing to the chosen basis block before the validity period expires.
4. If this task is accomplished, the block producer receives the rewards associated with the creation of the valid block.

3.3.3 Proof of Capacity

Proof of capacity, also known as proof of space, requires nodes to set aside a certain quantity of local storage, usually non-volatile storage, for the validation of blocks. This is accomplished through the use of hard-to-pebble graphs [40], a special type of hierarchical graph with some similarities to the well-known Merkle tree [41]. These graphs, known as superconcentrator graphs (see Fig. 3.4 for a very small example), have a large number of hierarchical connections.

However, rather than having a single root and many leaves, a hard-to-pebble graph has many roots and leaves, and the relationships between parents and

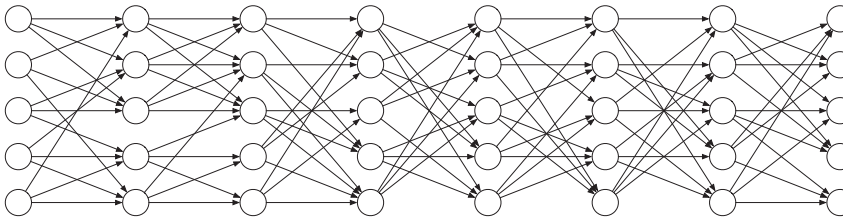


Figure 3.4 A hard-to-pebble superconcentrator graph.

children are determined randomly. Therefore, a given node may have a variable number of children, and the hash value of a given node is equal to the hash of all of its parents' hashes. Since the relationships are varied but still hierarchical, the hashes at the bottom of the graph may require that the majority, or sometimes even the entirety of the graph, must have been previously hashed. This ensures that validating the hash of a given node is computationally infeasible in a short amount of time.

Consequently, upon being provided a random, un-pebbled graph by the network, the prospective verifier must first compute the hashes for all of the constituent nodes. As this may include thousands or even millions of relationships, made even more complex through these hierarchical linkages, such computation cannot be feasibly stored in-memory. Instead, intermediate results must be recorded on-disk for use in later computations as a key-value store, permitting the verifier to look up upstream results much more quickly than they could be computed from the base graph.

However, these intermediate results consume quite a large portion of local, non-volatile storage, which provides the "space" component of the proof of space. When the verifier node has completed the hashing of the entire hard-to-pebble graph, it informs the network that it is complete. To validate the node's claim, the network requests a series of root hashes, as well as several intermediate hashes for randomly chosen nodes from the body of the graph. The responses to these queries must come within a specified timeout period, which ensures that the node maintains the intermediate results store, rather than computing the hashes on the fly (since such computations cannot be practically computed within the timeout period).

This validation process is used to ensure that the signing node has set aside sufficient space for the graph; by reserving this storage, the verifying node shows that it has invested sufficient hardware resources in the validation process, and by scaling the graphs up or down, the amount of storage (equivalent to task difficulty) can be adjusted.

One of the primary strengths of this method is that, except when first building the graph, and when validating the blocks, the computational requirements

of this scheme are quite low. Power consumption of storage, even when actively reading the data, is much lower than the computational load imposed by proof of work, for example. However, by requiring a real-world, hardware-based investment in network validation, the likelihood of successful network monopolization by an adversary is decreased. Recent work [42] has even explored the use of mobile devices in the proof-of-capacity context.

However, the storage set aside for this purpose cannot be usefully purposed; that is, it must be used to store the graph itself, and not some other practical purpose (such as cloud backups and redundant data storage). Therefore, the resources allocated are still wasted, and efficiency is sub-optimal. Furthermore, to validate the computational effects of a node, there must be a bootstrapping process, where at least one node must know the correct answer prior to checking the results of a downstream node. This ensures that each graph must be stored by more than one node in the network, decreasing overall efficiency and creating a chicken-or-egg problem during network bootstrap, where no such nodes exist.

3.3.4 Proof of Time

Proof of time, or more formally proof of *elapsed* time, or PoET, is a protocol that relies on randomness with regards to the passage of time to determine node authority. Specifically, this describes a system proposed by Intel [43] in which dedicated hardware is used to secure a predefined randomness algorithm. This algorithm is used to determine a random period that each node must wait; whichever node awakens first when a block is waiting for approval may approve that block.

Computationally, this system is efficient, as each node must only wait for the next available slot, providing highly efficiency approval of blocks. Additionally, the ability to tune the waiting period provides a great degree of control over the performance of the network as a whole.

However, this system also brings several crucial drawbacks. First, it requires the deployment of a specific piece of hardware on every device that wishes to participate in the network. Since this hardware is only provided by Intel, and its details are not open source, this restricts how widely the network may be deployed, as not every machine has the hardware suitable to act as a node. Recent developments for lightweight systems [44] may be weakening this requirement, however.

Additionally, the entire security of the network relies on the implementation of this piece of secure hardware. If the hardware or software implementation of this system is compromised in any way, the entire network's security will be disrupted. This means that the system contains a single point of failure, and crucially, this point is controlled by a single organization with no external oversight. Furthermore, this means that if Intel wished, they could take control

of any network based on this approach by exploiting insider knowledge to bypass the protections of their implementation.

Due to these drawbacks, this approach has not garnered significant buy-in from the distributed ledger community. However, it has been incorporated into the Hyperledger project [45], and therefore could be used to facilitate private ledger implementations for intra-organizational applications, which are also more likely to be less price-sensitive toward the requirement to deploy specialized hardware.

3.4 Challenges to Scalability

Although consensus plays a pivotal role in enabling distributed transaction systems, it also imposes assumptions that bring their own challenges. Explicitly, using the term "consensus" declares that all nodes in the system agree on the same ordering of committed transactions. *How* and *when* consensus is achieved is left as an implementation detail, but as they say, the devil is in the details.

In distributed transaction systems, some common assumptions appear repeatedly, reflecting practical real-world challenges:

1. Participant communication is not a complete graph.
2. Networks rely on asynchronous messaging.
3. Participants may join or leave at any time.
4. Malicious actors are present, active, and outnumbered by benevolent participants.

What impacts do these assumptions have in the distributed transaction system context, and how can those impacts be ameliorated? These can be summarized in two main classes, communication complexity and asynchrony, and together, they lead to the *blockchain scalability problem* which plagues distributed transaction systems [46].

3.4.1 Communication Complexity

Traditional consensus mechanisms, such as the PBFT algorithm from Section 3.2.2, often use an all-to-all communication paradigm to effect consensus. Intuitively, the common $O(n^2)$ complexity bounds the scalability of systems in terms of the number of nodes that can participate. Additionally, since communication cannot rely on a complete graph, misbehavior during intermediate message routing must also be considered. Since retransmission may be required, possibly over different routes, communication complexity increases even further.

This consensus communication must be repeated each time new information must be agreed upon. One initial instinct to reduce these communication costs would be to come to consensus less frequently but on larger quantities of data. As the number of transactions resolved in one round of consensus increases, so too does the size of the transmitted data. While malicious interference in the network may be rare, normal packet loss and transmission problems may lead to other types of probabilistic failures. As message sizes increase, these faults become increasingly more common, undermining the benefits of moving to larger and less frequent consensus rounds. We discuss the impacts of these tradeoffs in Section 3.5.

An alternative approach might be to limit the number of participants involved in consensus, either to small groups or even to trusted pairs.

3.4.2 Asynchronous Context

In real-world decentralized systems, especially those using commodity hardware or in the presence of highly motivated adversaries, generating a canonical serialization of transactions poses a significant barrier. Likewise, enforcing synchronous timekeeping across participating nodes is impractical, to put it mildly. Fischer et al. famously illustrated the impossibility of distributed consensus in an asynchronous environment [12].

Yet, despite these challenges, distributed ledgers manage to operate in practice. To resolve this seeming paradox, a shift from guaranteed, theoretical bounds to probabilistic ones is necessary. For example, it may not be the case that all possible sequences of messages induce progress toward the consensus goal. However, if the likelihood of persistent deadlock is sufficiently small, such relaxation may be acceptable.

Even under a relaxed model, asynchronous communication also drastically complicates the construction of a total ordering of transactions, since messages may arrive at any time in any order, decoupled from their transmission time. While these challenges are not directly related to consensus, it can be important in resolving disagreements between partitioned subnets of participant nodes. Therefore, we touch on this topic in Section 3.8 to clarify points related to consensus protocols, and provide additional information in Section 3.6, which addresses this challenge more directly.

3.4.3 Participant Churn

A further challenge for distributed transaction systems lies in the assumption that nodes may join or leave the network at any time, a characteristic sometimes called *churn*. In order to provide strong theoretical guarantees, nearly all existing

"strong" consensus approaches assume a fixed number of participants in order to set an upper bound on the proportion of adversarial nodes. However, in a system with non-zero churn, these bounds can no longer be proven.

Instead, the network yet again relies on (negligible) probabilistic bounds to limit the likelihood of adversarial success. The churn assumption also disallows some traditional consensus mechanisms from being used directly without adaptation. We discuss the advantages and disadvantages of such adaption later in this chapter.

3.4.4 The Blockchain Scalability Problem

Collectively, the slowdowns in transaction processing resulting from the above challenges are described as the *blockchain scalability problem* (BSP). While distributed transaction systems aim to supplant existing centralized approaches, including through beneficial relaxation of assumptions, practical considerations cannot be ignored. For example, initial implementations of distributed transaction systems reflected a decrease in transactions per second (TPS) by roughly three to four orders of magnitude.

Since each challenge gives rise to its own aspect of this overall complexity, they can sometimes be addressed individually. More commonly, improvements in one aspect will require a (smaller) decrease in another area, leading to incremental overall improvements. For example, increasing the number of transactions in a consensus transaction increases TPS throughput, but it also extends propagation time, which in turn can undermine transaction ordering to such an extent that consensus on a unified ordering becomes impossible.

Note that the challenges discussed here arise from specific assumptions – modifications to the application context can drastically affect performance. In private, permissioned systems with fixed participants, consensus becomes drastically simpler and TPS performance increases accordingly, for example. The public, permissionless approach remains the most difficult environment for maximizing performance without sacrificing correctness.

In the remainder of this chapter, we address different aspects of the BSP in more detail, evaluating the advantages and disadvantages of different commonly applied mitigations meant to improve network transaction throughput.

3.5 Block Size and Propagation

Although consensus allows nodes to (eventually) agree upon which block to adopt, the details of its implementation have drastic impacts on the effectiveness and scalability of the system. As we've seen, performing consensus after each transaction is not feasible or practical, but nodes must still agree on when to do so.

These are usually structured as *rounds*, where consensus occurs when some condition is met, such as:

- A new, valid block is proposed (proven) under consensus
- A static or dynamic period of time has passed

In practice, these conditions can be interrelated – by adjusting the difficulty of the proving process, the interval between consensus can be tuned toward the desired duration. However, as long as nodes can deterministically agree on the speed with which consensus takes place, then it is reasonable for each round to be semi-synchronous.

In both cases, however, the scalability of the system is measured in the number of transactions that the network can process over time. To increase throughput, networks can include more transactions in each round of consensus, or the frequency of the rounds can be increased.

Each of these approaches has drawbacks [47]. As more transactions are included in a block, simply transmitting the data to participating nodes across the network takes longer. As the frequency of rounds increases, the window of time to propagate the proposed blocks narrows. In either case, if consensus begins before a block has had a chance to propagate to more than half the network, then the consensus process will degenerate.

3.5.1 Larger Blocks

When a block is proposed, such as by being selected by a chosen leader or mined in a proof-of-work scheme, its validity can be checked by the nodes which receive it. For example, its hash can be checked against the transactions it contains, and if it does not match it can be discarded. However, these checks cannot be completed until the entire block is visible to the validator.

As blocks become larger through the inclusion of more transactions, these validations must be delayed as the blocks propagate. Additionally, the stability of the underlying network becomes more relevant as dropped packets or broken pipes require the retransmission of larger quantities of data.

These problems collectively induce increased latency for block propagation within the network. This is directly undesirable because it reduces transaction throughput, but it also leads to other knock-on effects. For example, if proposals take too long to propagate, then new counter-proposals might arise, leading to forks in the chains, and even potentially preventing consensus in extreme cases.

To address these challenges, blocks may be encoded to require less data transmission. For example, rather than encoding the entire blocks, Graphene [48] uses invertible Bloom lookup tables to only transmit the information required by a

node. This can drastically increase propagation throughput when the network is already mostly synchronized.

Other approaches address the transmission problem directly. Velocity [49] uses erasure coding to allow nodes to emit symbols corresponding with a block. This allows for block transmission to scale with the number of nodes, reducing bottlenecks and increasing robustness in propagation. Further improvements based on pipelining and chunking optimizations have also been proposed [50].

3.5.2 Shorter Rounds

Rather than increasing the amount of data per round, other approaches such as FastCoin [51] use shorter rounds to increase throughput. However, the reduced time for consensus can pose challenges if the proposal cannot propagate sufficiently in time. As with larger blocks, these delays can cause problems reaching consensus, or to the creation of competing forks which must be resolved. Other work aims to reduce latency directly [52], though such approaches can run the risk of being dominates by proof generation times.

If consensus can be reached very quickly, such as through the use of a traditional BFT model on a small subset of nodes, then these short rounds do not pose a challenge – the authoritative result can be subsequently delivered to the remaining nodes as time and bandwidth permit. We discuss these committee-based approaches in more detail in Section 3.6.

Other approaches allow for the formation of forks, and the subsequent resolution of those forks using alternative mechanisms. For more information on fork resolution, checkpoint, and finality gadgets, please see Section 3.8.

3.6 Committees, Groups, and Sharding

One of the challenges facing traditional BFT approaches comes from scaling factors in terms of the number of participating nodes. In permissionless, Nakamoto-style blockchains, this number is both impossible to determine and arbitrarily scalable. However, the added complexity from Nakamoto consensus can introduce its own challenges. Fortunately, there are methods that aim to combine the best of both approaches.

3.6.1 Committees

Committees align closely with the core ideas of traditional BFT consensus. Under BFT, a leader is selected and proposes the value to agree upon. As previously

discussed, the identities of the participants are fixed, and the consistency of behavior determines outcome correctness. Consequently, the proof mechanisms required for Nakamoto consensus can be ignored, streamlining the computational workload.

However, the communications explosion remains a problem. To prevent intractable explosions, smaller subsets (called "committees") of participants are chosen for consensus. Their smaller size reduces communication, and by virtue of being a subset of a larger, potentially undefined group, all participants need not be predetermined.

As an example, Augustine et al. [53] use this approach to provide an efficient implementation of agreement with the explicit goal of limiting communications between nodes. Efficient, committee-driven consensus protocols may then be used as a subcomponent of a broader, permissionless system.

3.6.2 Groups

Groups operate in a similar manner, but differ from committees in that participants are partitioned in a way that does not necessarily include the leader (or proposer) of the block. This is the approach taken by Algorand [54]. A small subset of participants is selected randomly using a verifiable random function, and these "selected verifiers" come to consensus on the outcome of the proposed block. They collectively publish their results, which then rapidly propagate through the network. Since the smaller subset of nodes can reach consensus more quickly, and the result is disseminated from several sources in parallel, this approach can reduce the per-round duration.

3.6.3 Sharding

Sharding builds on these principles but leverages them to solve a slightly different problem. As we have seen, when large numbers of transactions are processed, the size of the consensus group had a substantial influence on various scaling factors of the network such as communication overhead.

However, these transactions must also be stored, using a (typically) append-only ledger. At some point, the consensus process is not the only bottleneck – individual participants simply become incapable of storing the entire ledger history locally, complicating verification and bootstrapping. In this section, we will discuss approaches to this challenge; for information about bootstrapping, see Section 3.9.

Like other subdivision strategies, sharding approaches [55, 56] benefit from the smaller partitions of participating nodes (see Figure 3.5). However, rather than composing these local decisions continuously to create cohesive global agreement, sharding allows for these local groups to maintain authoritative control over

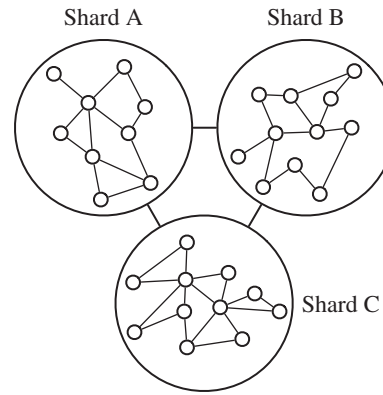


Figure 3.5 In sharded systems, subgroups of participants reach consensus, then cross-shard transactions are resolved separately.

subsets of transactions, with group membership determined by various methods [57, 58]. Then, a second level of agreement can take place between shards, when cross-shard transaction settlement is required.

This is the approach proposed by RapidChain [59]. A primary advantage of a sharding paradigm, in addition to the reduced storage requirements, lies in its scalability. Each shard comes to consensus on its own transactions in parallel, increasing throughput when cross-shard agreement is not required.

Dang et al. propose another shard-based ledger [60] which leverages trusted execution environments (TEEs) to further improve scalability while broadening applicability beyond transaction-oriented workloads. In this work, shards are re-formed regularly to prevent the accumulation of adversarial influence within specific shards. Even with this overhead, throughput remains high, although local storage optimizations are not considered.

3.7 Transaction Channels

A major complication to transaction throughput in blockchains stems from the untrusted nature of the network. Since nodes must validate every transaction to check for malicious behavior, significant computational resources are expended even when most or all transactions are innocuous.

In most consensus protocols, the real-world identity of a node plays no part in its role in the network. However, in practice, node activity tends to follow Zipf's Law – a small number of nodes produce a large proportion of the overall transactions. For example, in cryptocurrencies, well-known exchanges' internal and intra-exchange transfers dominate independent transactions.

By introducing elements of trust and identity to otherwise-untrusted networks, operators and nodes can observe substantial benefits.

3.7.1 Trust-Weighted Agreement

Consider the example of cryptocurrency exchanges. These businesses must satisfy significant legal requirements to conduct operations within a legal jurisdiction. Like other businesses, failure to adhere to contractual and regulatory requirements results in legal action against the business – a process typically ignored in anonymous, untrusted, and permissionless networks.

However, cryptographically authenticated identities may be used to establish trusted communication channels between participating nodes. Two businesses may wish to build a trust-based relationship between their respective nodes, partially based on the assurance that detected malfeasance would be punished through traditional channels. These relationships can be leveraged to establish a trust-influenced agreement protocol [61].

Rather than performing traditional validation and consensus processes, nodes might take their counterparts' balances at face value, for example by assuming that they will not double-spend their tokens. Transfers can be completed more quickly, and agreement uncertainties can be addressed through private comparisons with trusted partners. However, with some exceptions [62], most blockchains prefer to avoid trust as a design criteria.

3.7.2 Off-Chain Transactions

However, it may be impractical to establish business relationships with every node in the network. An alternative approach relies on the creation of *off-chain* transactions [63]. Unlike standard transactions, which are published to the ledger and visible to (and validated by) all participants, off-chain transactions may only involve the sender and recipient.

In cases when there is a disagreement or conflict, then the parties can turn to the ledger as an intermediary in some implementations. For example, the hashed time-locked contracts used in Bitcoin [64] can be used to enforce correct behavior by penalizing misbehaving participants, without putting a load on the ledger in nonadversarial cases. We discuss this type of approach in more detail in Section 3.7.3.

Another consideration with off-chain networks comes from the interactions of these partnerships. Consider two companies, Alice Financial and Charlie Enterprises, who wish to exchange information but do not have a preexisting relationship. However, both companies trust Bob Holdings and have existing off-chain channels with them. Thus, Alice and Charlie can interact with Bob serving as the intermediary. Multiple hops are typically possible as well – as the number

of participant relationships increases, then routing between any two given nodes becomes more likely and easier as their degrees of separation decrease.

However, navigating this potentially complex relationship network introduces new challenges in routing. Flash [65] proposes a mechanism for evaluating the topological properties of this graph to establish max-flow routing for transactions based on their size. The increase in intermediate nodes exposes a broader attack surface on a single transaction, so networks will often implement additional protection mechanisms to detect and disincentivize intermediate tampering.

3.7.3 Lightning Network

One of the best-known examples of using off-chain transaction channels to create a settlement layer is the Bitcoin Lightning network [64], though other, more general approaches also exist [66–68]. Here, the first layer corresponds to the standard consensus process used to commit values to the ledger directly, while the second layer is the meta-graph of these nodes' off-chain relationships. Using hashed time-locked contracts (HTLCs) [69], Bitcoin offers a mechanism for penalizing malfeasance by seizing transaction funds and awarding them to the harmed participant. This mechanism underlies the Lightning Network, which is the Layer 2 metagraph of Bitcoin nodes that have established these channels between each other.

In practice, the topology of the Lightning Network plays a significant role in the privacy, efficiency, and security of transaction settlement. Lee and Kim [70] identify the structure of the Lightning Network as approximating that of a scale-free network. Importantly, this indicates that there is relatively less redundancy or robustness in the Layer 2 network when compared with the original consensus-based, peer-to-peer approach. For example, the importance of nodes in the settlement layer is proportional to their centrality, since highly connected nodes are more likely to participate in intermediate routing. By targeting these "important" nodes, adversaries might be able to disrupt the normal functioning of this layer, a much easier task than doing so at Layer 1.

Seres et al. [71] identify the source and mitigation of this effect as the onboarding process for new nodes. Implementations typically connect new peers to important hub nodes to decrease their degrees of separation from the remainder of the network. However, this increases the importance of these hub nodes and undermines network resilience. If nodes are additionally connected to a subset of random other participants, similarly to how the Layer 1 network behaves, then routing around centralized failures becomes easier at the cost of higher transaction routing latency.

3.8 Checkpointing and Finality Gadgets

A major promise of traditional Byzantine consensus is the guarantee it places on behavior. Once a new state has been committed, the correct nodes in the network can rely on the fact that other correct nodes also agree. Additionally, the security properties of protection against adversarial interference are provable under a certain threshold of bad actors. While these strong characteristics can be beneficial, they also incur added costs. As in NP problems, sacrificing guarantees for a probabilistic approximation of optimality can substantially improve performance. In this section, we examine the efficiency, costs, risks, and mitigations of probabilistic finality in the context of blockchains [72].

3.8.1 Probabilistic Finality

Intuitively, a node's confidence in a proposed block increases as it observes additional confirmations from other nodes. In BFT approaches, nodes wait until they observe sufficiently many messages that agree, and then commit the change locally when there is no possibility for later disagreement, which approximates a step function in confidence in the proposed value. As an alternative, nodes may allow their confidence in a value to vary continuously, and then choose a threshold at which to commit.

This is the approach adopted by proof-of-work systems. Since the likelihood of discovering a block hash has been tuned, observing a message with a correct hash is probabilistically likely to be one of only a few valid proposals during a consensus round. Thus, observing only a few other confirmations from fellow nodes means that with a high likelihood the block can be accepted.

In some cases however, competing, valid proposals may be propagating within subgraphs of the overall network, a situation which may arise randomly but can also occur through adversarial pruning of message propagation (as in Figure 3.6). Even when many confirmations are observed for a given block, it may be the case that the opposing subgraph is larger – when these two proposals are resolved, a committed block that seemed quite likely would need to be reversed and replaced with the opposing proposal. This process is called fork resolution and is a key challenge in highly distributed, loosely connected consensus.

These forks can sometimes persist for multiple rounds of consensus, especially when the network is partitioned. As the lengths of the forks grow, the number of "wasted" transactions (and computational validation) increases, since those must be unwound and replaced when the forks rejoin. Additionally, if any real-world goods or services are paid for in the rolled-back transactions, unwinding those effects may be difficult or impossible. Thus, both reducing the likelihood of forking and minimizing the length of these forks are desirable.

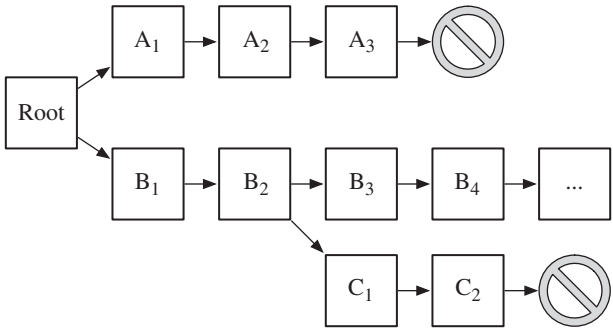


Figure 3.6 Here, the chain forked twice before A and C were pruned, with a maximum fork length of four blocks. Note that these forks can occur either stochastically or maliciously, but resolution methods typically consider both cases.

3.8.2 Checkpointing

One approach for reducing the length of these forks is called checkpointing. Rather than providing a guarantee about committed values after each round of consensus, or eliminating the guarantee in exchange for increased speed, checkpointing provides a middle ground in which guarantees are provided about some fraction of the total blocks.

For example, Das et al. introduce a BFT-based "support service" which uses the slower but guaranteed approach to seal blocks [73]. When nodes receive sealed blocks from the service, they are guaranteed that transactions prior to that cannot be reversed. However, the proposed approach is relatively general and requires careful tuning of the components' parameters to ensure that a balance between security and performance can be achieved. Similar approaches [74] also leverage the problem's topological structure to provide guarantees.

Rather than reducing fork depth, Li et al. [75] attempt to detect malicious behaviors intended to create forks, allowing for penalties to be applied. They also apply TEEs to prevent malicious collusion to create forks at a given chain height. These approaches are particularly useful in Proof-of-Stake systems, where the costs for creating eventually unwound forks are either uncollected or refunded – this is sometimes called the "nothing at stake" attack, because there is no disincentive to sign every fork you observe.

3.8.3 Finality Gadgets

An important component of any checkpointing scheme is the mechanism by which the finalized state is selected. These tools are referred to as "finality gadgets" [39] or "finality layers" [76], because they are usually unrelated to the

underlying consensus mechanism, and are rather used in conjunction with other protocols to introduce the concept of guaranteed finality.

The most well-known finality gadget is Casper [39], the mechanism which adds checkpointing to the Ethereum blockchain. This involves several stages, including when checkpoints are proposed, how conflicting votes are detected, and punitive measures. To prevent adversaries from growing extremely long forks, so-called "long-range attacks," Casper asks validators to extend the fork with the greatest length. If a validator submits conflicting votes, e.g. for two forks in contention for the majority, then once the forks are resolved the staked value of the violator is forfeit.

Another well-known finality gadget is GHOST-based Recursive ANcestor Deriving Prefix Agreement (GRANDPA) [77], used with the Polkadot system. Like other checkpointing mechanisms, it relies on concepts drawn from the BFT literature but makes some changes to improve usability. Rather than requiring all nodes to communicate, which may be impossible in practice due to network issues, it instead introduces a weakly synchronous model in which time windows are used to ensure progress toward consensus can be made. Individual nodes can then tentatively accept proposed blocks while relying on confirmed block values in cases where confirmation is required.

3.9 Bootstrapping

As previously mentioned in Section 3.4.3, participants can join and leave blockchains at any time. The final impact of this design choice arises when onboarding new nodes. Fundamentally, the ledger grows over time as it represents an immutable record of transactions that occurred. Thus, when new nodes join a mature system, huge quantities of data might need to be transferred to bring them up to speed with the current state of the existing node participants. Beyond the data itself, the task of identifying and joining the communication networks poses additional challenges both logistically and defensively.

This "bootstrapping" problem is related to, but slightly different from, the consensus process itself because it does not necessarily require the nodes to come to consensus immediately. Instead, incrementally approaching the current state of the chain may be acceptable or even preferable. While this process has not received as much attention as the throughput problem thus far, as ledger sizes grow, efficient bootstrapping will become increasingly important.

3.9.1 Networking

When determining which networks to join, the fundamentally decentralized nature of most blockchains suggests that centralized coordinators may not be

used. However, in practice, hard-coded IPs are commonly used in many chains, representing an undesirable point of failure when onboarding new nodes. Loe and Quaglia [78] survey the blockchain landscape for these undesirable patterns, and discuss Tor [79] and ZMap [80] as alternative mechanisms for establishing communication with existing participant nodes. While Tor provides a robust and censorship-resistant mechanism for discovery and bootstrapping, it also introduces high latency and can be difficult to configure. Although promising, the authors found ZMap was not able to successfully bootstrap, but could not explain the failure, underlining the complexity of the network discovery phase.

3.9.2 Data

When a new node joins a blockchain network, it must construct the current network state *tabula rasa*, usually requiring data from peers linear to the total number of transactions ever recorded. To address the ledger size problem, Leung et al. propose [81] to decouple the local wallet state from the global transaction history. By doing so, only a much smaller amount of critical intermediary information called "succinct ledger certificates" must be validated. Even still, the number of balances stored is proportionate to the total number of transactions – to mitigate this challenge, encoding mechanisms can be used to improve propagation [82], sampling techniques can reduce the data which must be shared [83], or sharding techniques can subdivide responsibility (see Section 3.6 for more information). In practice, combining these techniques can reduce the data required for bootstrapping by more than 90% [81], drastically simplifying the initial consensus process.

3.10 Future Trends

Distributed ledgers encompass a rapidly evolving landscape, both in their own features and capabilities, as well as the applications that leverage these techniques. In the near future, changing trends in this landscape offer new opportunities for future work.

3.10.1 Private Consensus

Many industrial and commercial environments deal with sensitive data, such as manufacturing output, capital outlays, and other business-critical information. In parallel, the emerging benefits of digital ledger technologies have increased commercial interest in integration. Private and permissioned approaches [45] partially solve this problem, but limit untrusted collaborations. Improvements in

access control, data ownership, and transaction privacy [84] will pave the way for significant new applications in the distributed ledger space.

Data committed to ledgers is both immutable and long-lived by design, and in combination with publicly auditable blockchains, this can pose additional data privacy concerns. Perfect forward secrecy describes a cryptographic principle that if a key is compromised at some point in the future, all past ciphers cannot be retroactively decoded. To extend the idea to a distributed ledger context, identifying a single transaction of a user should not reveal all the transactions of that user, for example. Some approaches have been proposed to prevent this [85], but typically require extensive cryptographic operations that impose additional restrictions, such as limiting throughput, and may still be ineffective [86].

3.10.2 Improved Oracles

Use cases for distributed ledger technologies often replace or augment existing approaches: cryptocurrencies and banking, asset traceability and logistics management, or smart contracts and legal contracts. In many of these cases, connecting the content of the ledger with a real-world asset, event, or outcome can prove difficult. These linkages, called oracles, provide a great opportunity for growth and future research, but the difficulty in addressing latency, accuracy, flexibility, and authenticity concerns, referred to as the "blockchain oracle problem," remains a challenge. Although many existing works have attempted to tackle the problem, reliability issues and inflexible enforcement mechanisms continue to persist [87].

3.10.3 Streaming Consensus

The proliferation of Internet of Things (IoT) devices has produced an influx of new data streams. However, the integrity of such data poses challenges to existing consensus mechanisms, as the speed of both generation and validation must be extremely high. For example, consider a video surveillance system from a network of cameras [88], or a live broadcast [89] – approaches that can validate the authenticity of these streams in an online fashion are only recently appearing in the literature.

Beyond validating the authenticity of these data, additional post-processing and interactions are also possible. Consider a marketplace that authenticates, regulates, and incentivizes the deployment and collection of sensing nodes [90]. Distributed ledgers play a role in collecting, routing, and validating this data from the producing nodes to the correct buyers in a near-real-time way, providing a robust oracle-like layer for real-world sensing.

3.11 Conclusion

In this chapter, we provided an overview of the fault-tolerant and Byzantine consensus, especially within the context of distributed ledger systems and blockchain. We have seen that, through the use of proof mechanisms, Nakamato consensus and its variants prevent Sybil nodes from disrupting consensus by outnumbering correct participants. Yet, the communication and computation costs underlying these consensus protocols cause blockchain-based systems to suffer from significant scalability problems. In this chapter, we also discussed the current proposals to deal with these scalability problems.

References

1 Merriam-Webster (2020). Consensus. In: *Merriam-Webster.com dictionary*. https://www.merriam-webster.com/dictionary/consensus.

2 Coulouris, G.F., Dollimore, J., and Kindberg, T. (2005). *Distributed Systems: Concepts and Design*. Pearson Education.

3 Raikwar, M., Gligoroski, D., and Velinov, G. (2020). Trends in development of databases and blockchain. *2020 7th International Conference on Software Defined Systems (SDS)*, 177–182, April 2020. https://doi.org/10.1109/SDS49854.2020.9143893.

4 Brewer, E. (2012). CAP twelve years later: how the "rules" have changed. *Computer* 45 (2): 23–29. https://doi.org/10.1109/MC.2012.37.

5 Khanna, A. and Anand, R. (2016). IoT based smart parking system. *2016 International Conference on Internet of Things and Applications (IOTA)*, 266–270, January 2016. https://doi.org/10.1109/IOTA.2016.7562735.

6 BARTER Adapter Overview – Vicinity. https://vizlore.com/barter-blockchain-assisted-real-time-transaction-execution-and-repository-framework/.

7 IoT Catalogue. https://www.iot-catalogue.com/search/usecase/5de0ed1ef5e047bc2000c910 (accessed 17 May 2023).

8 VICINITY H2020 (2020). Vicinityh2020/vicinity-adapter-barter, July 2020.

9 Pease, M., Shostak, R., and Lamport, L. (1980). Reaching agreement in the presence of faults. *Journal of the ACM (JACM)* 27 (2): 228–234.

10 Lamport, L. (1978). Time, clocks, and the ordering of events in a distributed system. *ACM Communications* 21 (7): 8.

11 Schneider, F.B. (1990). Implementing fault-tolerant services using the state machine approach: a tutorial. *ACM Computing Surveys* 22 (4): 299–319. https://doi.org/10.1145/98163.98167. https://doi.org/10.1145/98163.98167.

12 Fischer, M.J., Lynch, N.A., and Paterson, M.S. (1985). Impossibility of distributed consensus with one faulty process. *Journal of the ACM* 32 (2): 374–382. https://doi.org/10.1145/3149.214121.

13 Ongaro, D. and Ousterhout, J. (2014). In search of an understandable consensus algorithm. *2014 {USENIX} Annual Technical Conference ({USENIX}{ATC} 14)*, 305–319. ISBN 978-1-931971-10-2. https://www.usenix.org/conference/atc14/technical-sessions/presentation/ongaro (accessed 17 May 2023).

14 Lamport, L. (1998). The part-time parliament. *ACM Transactions on Computer Systems* 16 (2): 133–169. https://doi.org/10.1145/279227.279229.

15 Enes, V., Baquero, C., Rezende, T.F. et al. (2020). State-machine replication for planet-scale systems. *Proceedings of the 15th European Conference on Computer Systems*, EuroSys '20, 1–15, New York, NY, USA, April 2020. Association for Computing Machinery. ISBN 978-1-4503-6882-7. https://doi.org/10.1145/3342195.3387543.

16 Howard, H., Malkhi, D., and Spiegelman, A. (2016). Flexible Paxos: quorum intersection revisited. *arXiv preprint arXiv:1608.06696*.

17 Lamport, L., Shostak, R., and Pease, M. (1982). The Byzantine generals problem. *ACM Transactions on Programming Languages and Systems* 4 (3): 20.

18 Castro, M. and Liskov, B. (1999). Practical Byzantine fault tolerance. *Proceedings of the 3rd Symposium on Operating Systems Design and Implementation*, OSDI '99, 173–186. Berkeley, CA, USA: USENIX Association. ISBN 978-1-880446-39-3.

19 Lamport, L. (2011). Byzantizing Paxos by refinement. In: *Distributed Computing*, *Lecture Notes in Computer Science* (ed. D. Peleg), 211–224. Berlin, Heidelberg: Springer-Verlag. ISBN 978-3-642-24100-0. https://doi.org/10.1007/978-3-642-24100-0_22.

20 Kotla, R., Alvisi, L., Dahlin, M. et al. (2007). Zyzzyva: speculative byzantine fault tolerance. *Proceedings of 21st ACM SIGOPS Symposium on Operating Systems Principles*, 45–58.

21 Gueta, G.G., Abraham, I., Grossman, S. et al. (2019). SBFT: a scalable and decentralized trust infrastructure. *2019 49th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, June 2019, 568–580. https://doi.org/10.1109/DSN.2019.00063.

22 Amir, Y., Coan, B., Kirsch, J., and Lane, J. (2008). Byzantine replication under attack. *2008 IEEE International Conference on Dependable Systems and Networks with FTCS and DCC (DSN)*, 197–206. IEEE.

23 Yin, M., Malkhi, D., Reiter, M.K. et al. (2019). HotStuff: BFT consensus with linearity and responsiveness. *Proceedings of the 2019 ACM Symposium on Principles of Distributed Computing*, PODC '19, July 2019, 347–356. New York, NY, USA: Association for Computing Machinery. ISBN 978-1-4503-6217-7. https://doi.org/10.1145/3293611.3331591.

24 Abraham, I., Malkhi, D., Nayak, K. et al. (2020). Sync HotStuff: simple and practical synchronous state machine replication. *2020 IEEE Symposium on*

Security and Privacy (SP), May 2020, 106–118. https://doi.org/10.1109/SP40000.2020.00044.

25 Lamport, L. (2011). Brief announcement: leaderless Byzantine Paxos. In: *International Symposium on Distributed Computing*, 141–142. Springer-Verlag.

26 Gilad, Y., Hemo, R., Micali, S. et al. (2017). Algorand: scaling byzantine agreements for cryptocurrencies. *Proceedings of the 26th Symposium on Operating Systems Principles*, SOSP '17, October 2017, 51–68. New York, NY, USA: Association for Computing Machinery. ISBN 978-1-4503-5085-3. https://doi.org/10.1145/3132747.3132757.

27 Zhou, C.-X., Hua, Q.-S., and Jin, H. (2020). HotDAG: hybrid consensus via sharding in the permissionless model. In: *Wireless Algorithms, Systems, and Applications*, *Lecture Notes in Computer Science* (ed. D. Yu, F. Dressler, and J. Yu), 807–821. Cham: Springer International Publishing. ISBN 978-3-030-59016-1. https://doi.org/10.1007/978-3-030-59016-1_66.

28 Back, A. (2002). Hashcash - A denial of service counter-measure. *USENIX Technical Conference*.

29 Bellare, M. and Rogaway, P. (1997). Collision-resistant hashing: towards making UOWHFs practical. In: *Annual International Cryptology Conference*, 470–484. Springer-Verlag.

30 Nakamoto, S. (2008). Bitcoin: A Peer-to-Peer Electronic Cash System. *Technical Report*. https://bitcoin.org/bitcoin.pdf.

31 Wood, G. et al (2014). Ethereum: a secure decentralised generalised transaction ledger. *Ethereum project yellow paper* 151 (2014): 1–32.

32 Ullrich, J., Stifter, N., Judmayer, A. et al. (2018). Proof-of-blackouts? How proof-of-work cryptocurrencies could affect power grids. In: *Research in Attacks, Intrusions, and Defenses*, *Lecture Notes in Computer Science* (ed. M. Bailey, T. Holz, M. Stamatogiannakis, and S. Ioannidis), 184–203 Springer International Publishing. ISBN 978-3-030-00470-5.

33 Chanyshev, A. (2018). Cryptocurrencies: fundamentals, developments, and regulation.

34 Kiayias, A., Miller, A., and Zindros, D. (2020). Non-interactive proofs of proof-of-work. In: *Financial Cryptography and Data Security*, Lecture Notes in Computer Science (ed. J. Bonneau and N. Heninger), 505–522. Cham: Springer International Publishing. ISBN 978-3-030-51280-4. https://doi.org/10.1007/978-3-030-51280-4_27.

35 King, S. and Nadal, S. (2012). PPCoin: Peer-to-peer crypto-currency with proof-of-stake. *self-published paper, August*, 19:1.

36 Leonardos, S., Reijsbergen, D., and Piliouras, G. (2020). Weighted voting on the blockchain: improving consensus in proof of stake protocols. *International Journal of Network Management* 30 (5): e2093. https://doi.org/10.1002/nem.2093.

37 Duong, T., Fan, L., Katz, J. et al. (2020). 2-hop blockchain: combining proof-of-work and proof-of-stake securely. In: *Computer Security – ESORICS 2020*, *Lecture Notes in Computer Science* (ed. L Chen, N. Li, K. Liang, and S. Schneider), 697–712. Cham: Springer International Publishing. ISBN 978-3-030-59013-0. https://doi.org/10.1007/978-3-030-59013-0_34.

38 Roşu, I. and Saleh, F. (2020). Evolution of shares in a proof-of-stake cryptocurrency. *Management Science*. https://doi.org/10.1287/mnsc.2020.3791.

39 Buterin, V. and Griffith, V. (2019). Casper the Friendly Finality Gadget. *arXiv:1710.09437 [cs]*.

40 Dziembowski, S., Faust, S., Kolmogorov, V., and Pietrzak, K. (2015). Proofs of space. In: *Annual Cryptology Conference*, 585–605. Springer-Verlag.

41 Merkle, R.C. (1990). A certified digital signature. In: *Advances in Cryptology – CRYPTO' 89 Proceedings*, *Lecture Notes in Computer Science* (ed. G. Brassard), 218–238. New York: Springer. ISBN 978-0-387-34805-6.

42 Jiang, S. and Wu, J. (2020). A game-theoretic approach to storage offloading in PoC-based mobile blockchain mining. *Proceedings of the 21st International Symposium* on Theory, Algorithmic Foundations, and Protocol Design for Mobile Networks and Mobile Computing, Mobihoc '20, October 2020, 171–180. New York, NY, USA: Association for Computing Machinery. ISBN 978-1-4503-8015-7. https://doi.org/10.1145/3397166.3409136.

43 Chen, L., Xu, L., Shah, N. et al. (2017). On security analysis of proof-of-elapsed-time (POET). In: *International Symposium on Stabilization, Safety, and Security of Distributed Systems*, 282–297. Springer.

44 Andola, R.N., Venkatesan, S., and Verma, S. (2020). PoEWAL: a lightweight consensus mechanism for blockchain in IoT. *Pervasive and Mobile Computing* 69: 101291. https://doi.org/10.1016/j.pmcj.2020.101291.

45 Androulaki, E., Barger, A., Bortnikov, V. et al. (2018). Hyperledger fabric: a distributed operating system for permissioned blockchains. *Proceedings of the 13th EuroSys Conference*, EuroSys '18, 30:1–30:15. New York, NY, USA: ACM. ISBN 978-1-4503-5584-1. https://doi.org/10.1145/3190508.3190538.

46 Zhou, Q., Huang, H., Zheng, Z., and Bian, J. (2020). Solutions to scalability of blockchain: a survey. *IEEE Access* 8: 16440–16455. https://doi.org/10.1109/ACCESS.2020.2967218.

47 Rahmadika, S., Noh, S., Lee, K. et al. (2020). The dilemma of parameterizing propagation time in blockchain P2P network. *Journal of Information Processing Systems* 16 (3): 699–717. https://doi.org/10.3745/JIPS.03.0140.

48 Ozisik, A.P., Andresen, G., Levine, B.N. et al. (2019). Graphene: efficient interactive set reconciliation applied to blockchain propagation. In: *Proceedings of the ACM Special Interest Group on Data Communication*, SIGCOMM '19, August 2019, 303–317. New York, NY, USA: Association for Computing Machinery. ISBN 978-1-4503-5956-6. https://doi.org/10.1145/3341302.3342082.

49 Chawla, N., Behrens, H.W., Tapp, D. et al. (2019). Velocity: scalability improvements in block propagation through rateless erasure coding. *2019 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*, May 2019, 447–454. https://doi.org/10.1109/BLOC.2019.8751427.

50 Ayinala, K., Choi, B.-Y., and Song, S. (2020). PiChu: accelerating block broadcasting in blockchain networks with pipelining and chunking. *2020 IEEE International Conference on Blockchain (Blockchain)*, November 2020, 221–228. https://doi.org/10.1109/Blockchain50366.2020.00035.

51 Sompolinsky, Y. and Zohar, A. (2015). Secure high-rate transaction processing in Bitcoin. In: *International Conference on Financial Cryptography and Data Security*, 507–527. Springer-Verlag.

52 Eischer, M. and Distler, T. (2020). Resilient cloud-based replication with low latency. *Proceedings of the 21st International Middleware Conference*, Middleware '20, December 2020. 14–28. New York, NY, USA: Association for Computing Machinery. ISBN 978-1-4503-8153-6. https://doi.org/10.1145/3423211.3425689.

53 Augustine, J., King, V., Molla, A.R. et al. (2020). Scalable and secure computation among strangers: message-competitive byzantine protocols. In: *34th International Symposium on Distributed Computing (DISC 2020)*, *Leibniz International Proceedings in Informatics (LIPIcs)*, vol. 179 (ed. H. Attiya), 31:1–31:19. Dagstuhl, Germany: Schloss Dagstuhl-Leibniz-Zentrum für Informatik. ISBN 978-3-95977-168-9. https://doi.org/10.4230/LIPIcs.DISC.2020.31. https://drops.dagstuhl.de/opus/volltexte/2020/13109. ISSN: 1868-8969.

54 Chen, J. and Micali, S. (2019). Algorand: a secure and efficient distributed ledger. *Theoretical Computer Science* 777: 155–183. https://doi.org/10.1016/j.tcs.2019.02.001.

55 Manuskin, A., Mirkin, M., and Eyal, I. (2020). Ostraka: secure blockchain scaling by node sharding. *2020 IEEE European Symposium on Security and Privacy Workshops (EuroS PW)*, September 2020, 397–406. https://doi.org/10.1109/EuroSPW51379.2020.00060.

56 Huang, C., Wang, Z., Chen, H. et al. (2020). RepChain: a reputation based secure, fast and high incentive blockchain system via sharding. *IEEE Internet of Things Journal* 8 (6): 4291–4304. https://doi.org/10.1109/JIOT.2020.3028449.

57 Zhang, J., Hong, Z., Qiu, X. et al. (2020). SkyChain: a deep reinforcement learning-empowered dynamic blockchain sharding system. *49th International Conference on Parallel Processing - ICPP*, ICPP '20, August 2020, 1–11. New York, NY, USA: Association for Computing Machinery. ISBN 978-1-4503-8816-0. https://doi.org/10.1145/3404397.3404460.

58 Yun, J., Goh, Y., and Chung, J.-M. (2021). DQN-based optimization framework for secure sharded blockchain systems. *IEEE Internet of Things Journal* 8 (2): 708–722. https://doi.org/10.1109/JIOT.2020.3006896.

59 Zamani, M., Movahedi, M., and Raykova, M. (2018). RapidChain: scaling blockchain via full sharding. *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, CCS '18, January 2018, 931–948. New York, NY, USA: Association for Computing Machinery. ISBN 978-1-4503-5693-0. https://doi.org/10.1145/3243734.3243853.

60 Dang, H., Dinh, T.T.A., Loghin, D., Chang, E.-C. et al. (2019). Towards scaling blockchain systems via sharding. *Proceedings of the 2019 International Conference on Management of Data*, SIGMOD '19, June 2019, 123–140. New York, NY, USA: Association for Computing Machinery. ISBN 978-1-4503-5643-5. https://doi.org/10.1145/3299869.3319889.

61 Chase, B. and MacBrough, E. (2018). Analysis of the XRP Ledger Consensus Protocol. *arXiv:1802.07242 [cs]*.

62 Shala, B., Trick, U., Lehmann, A. et al. (2020). Blockchain and trust for secure, end-user-based and decentralized IoT service provision. *IEEE Access* 8: 119961–119979. https://doi.org/10.1109/ACCESS.2020.3005541.

63 Back, A., Corallo, M., Dashjr, L. et al. (2014). Enabling blockchain innovations with pegged sidechains, 72. http://www.opensciencereview.com/papers/123/enablingblockchain-innovations-with-pegged-sidechains (accessed 18 May 2023).

64 Poon, J. and Dryja, T. (2016). The Bitcoin lightning network: scalable off-chain instant payments.

65 Wang, P., Xu, H., Jin, X., and Wang, T. (2019). Flash: efficient dynamic routing for offchain networks. *Proceedings of the 15th International Conference on Emerging Networking Experiments And Technologies*, CoNEXT '19, December 2019, 370–381. New York, NY, USA: Association for Computing Machinery. ISBN 978-1-4503-6998-5. https://doi.org/10.1145/3359989.3365411.

66 Dziembowski, S., Faust, S., and Hostáková, K. (2018). General state channel networks. *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, CCS '18, October 2018, 949–966. New York, NY, USA: Association for Computing Machinery. ISBN 978-1-4503-5693-0. https://doi.org/10.1145/3243734.3243856.

67 Jourenko, M., Larangeira, M., and Tanaka, K. (2020). Lightweight virtual payment channels. In: *Cryptology and Network Security*, Lecture Notes in Computer Science (ed. S. Krenn, H. Shulman, and S. Vaudenay), 365–384. Cham: Springer International Publishing. ISBN 978-3-030-65411-5. https://doi.org/10.1007/978-3-030-65411-5_18.

68 Gudgeon, L., Moreno-Sanchez, P., Roos, S. et al. (2020). SoK: layer-two blockchain protocols. In: *Financial Cryptography and Data Security*, Lecture Notes in Computer Science (ed. J. Bonneau and N. Heninger), 201–226. Cham:

Springer International Publishing. ISBN 978-3-030-51280-4. https://doi.org/10.1007/978-3-030-51280-4_12.

69 Decker, C. and Wattenhofer, R. (2015). A fast and scalable payment network with Bitcoin duplex micropayment channels. In: *Stabilization, Safety, and Security of Distributed Systems*, Lecture Notes in Computer Science (ed. A. Pelc and A.A. Schwarzmann), 3–18. Cham: Springer International Publishing. ISBN 978-3-319-21741-3. https://doi.org/10.1007/978-3-319-21741-3_1.

70 Lee, S. and Kim, H. (2020). On the robustness of lightning network in Bitcoin. *Pervasive and Mobile Computing* 61: 101108. https://doi.org/10.1016/j.pmcj.2019.101108.

71 Seres, I.A., Gulyás, L., Nagy, D.A., and Burcsi, P. (2020). Topological analysis of Bitcoin's lightning network. In: *Mathematical Research for Blockchain Economy*, Springer Proceedings in Business and Economics (ed. P. Pardalos, I. Kotsireas, Y. Guo, and W. Knottenbelt), 1–12. Cham: Springer International Publishing. ISBN 978-3-030-37110-4. https://doi.org/10.1007/978-3-030-37110-4_1.

72 Anceaume, E., Pozzo, A., Rieutord, T., and Tucci-Piergiovanni, S. (2020). On finality in blockchains. *arXiv:2012.10172 [cs]*.

73 Das, R.A., Pahalovi, M.M.S., and Yanhaona, M.N. (2019). Transaction finality through ledger checkpoints. *2019 IEEE 25th International Conference on Parallel and Distributed Systems (ICPADS)*, December 2019, 183–192. https://doi.org/10.1109/ICPADS47876.2019.00036.

74 Tsoulias, K., Palaiokrassas, G., Fragkos, G. et al. (2020). A graph model based blockchain implementation for increasing performance and security in decentralized ledger systems. *IEEE Access* 8: 130952–130965. https://doi.org/10.1109/ACCESS.2020.3006383.

75 Li, W., Andreina, S., Bohli, J.-M., and Karame, G. (2017). Securing proof-of-stake blockchain protocols. In: *Data Privacy Management, Cryptocurrencies and Blockchain Technology*, Lecture Notes in Computer Science (ed. J. Garcia-Alfaro, G. Navarro-Arribas, H. Hartenstein, and J. Herrera-Joancomarti), Cham: 297–315. Springer International Publishing. ISBN 978-3-319-67816-0. https://doi.org/10.1007/978-3-319-67816-0_17.

76 Dinsdale-Young, T., Magri, B., Matt, C. et al. (2020). Afgjort: A Partially Synchronous Finality Layer for Blockchains. In: *Security and Cryptography for Networks*, *Lecture Notes in Computer Science* (ed. C. Galdi and V. Kolesnikov), 24–44. Cham: Springer International Publishing. ISBN 978-3-030-57990-6. https://doi.org/10.1007/978-3-030-57990-6_2.

77 Stewart, A. and Kokoris-Kogia, E. (2020). GRANDPA: a byzantine finality gadget. *arXiv:2007.01560 [cs]*.

78 Loe, A.F. and Quaglia, E.A. (2019). You shall not join: a measurement study of cryptocurrency peer-to-peer bootstrapping techniques. *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, CCS '19, November 2019, 2231–2247. New York, NY, USA: Association for Computing Machinery. ISBN 978-1-4503-6747-9. https://doi.org/10.1145/3319535.3345649.

79 Reed, M.G., Syverson, P.F., and Goldschlag, D.M. (1998). Anonymous connections and onion routing. *IEEE Journal on Selected Areas in Communications* 16 (4): 482–494. https://doi.org/10.1109/49.668972.

80 Durumeric, Z., Wustrow, E., and Halderman, J.A. (2013). ZMap: fast internet-wide scanning and its security applications. *22nd {USENIX} Security Symposium ({USENIX} Security 13)*, 605–620. ISBN 978-1-931971-03-4.

81 Leung, D., Suhl, A., Gilad, Y., and Zeldovich, N. (2019). Vault: fast bootstrapping for the algorand cryptocurrency. *NDSS*, February 2019, 15, San Diego, CA: Internet Society. https://doi.org/10.14722/ndss.2019.23313.

82 Pal, R. (2020). Fountain coding for bootstrapping of the blockchain. *2020 International Conference on COMmunication Systems NETworkS (COMSNETS)*, January 2020, 1–5. https://doi.org/10.1109/COMSNETS48256.2020.9027309.

83 Wang, C., Wang, B., and Fan, X. (2020). EcoBoost: efficient bootstrapping for confidential transactions. *2020 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*, May 2020, 1–3. https://doi.org/10.1109/ICBC48266.2020.9169416.

84 Zhang, Y., Memariani, A., and Bidikar, N. (2020). A review on blockchain-based access control models in IoT applications. *2020 IEEE 16th International Conference on Control Automation (ICCA)*, October 2020, 671–676. https://doi.org/10.1109/ICCA51439.2020.9264499.

85 Möser, M., Soska, K., Heilman, E. et al. (2018). An empirical analysis of traceability in the monero blockchain. *Proceedings on Privacy Enhancing Technologies* 2018 (3): 143–163. https://doi.org/10.1515/popets-2018-0025.

86 Koerhuis, W., Kechadi, T., and Le-Khac, N.-A. (2020). Forensic analysis of privacy-oriented cryptocurrencies. *Forensic Science International: Digital Investigation* 33: 200891. https://doi.org/10.1016/j.fsidi.2019.200891.

87 Lo, S.K., Xu, X., Staples, M., and Yao, L. (2020). Reliability analysis for blockchain oracles. *Computers & Electrical Engineering* 83: 106582. https://doi.org/10.1016/j.compeleceng.2020.106582.

88 Michelin, R.A., Ahmed, N., Kanhere, S.S. et al. (2020). Leveraging lightweight blockchain to establish data integrity for surveillance cameras. *2020 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*, May 2020, 1–3. https://doi.org/10.1109/ICBC48266.2020.9169429.

89 Khalaf, O.I., Abdulsahib, G.M., Kasmaei, H.D., and Ogudo, K.A. (2020). A new algorithm on application of blockchain technology in live stream video transmissions and telecommunications. *International Journal of e-Collaboration* 16 (1): 16–32. https://doi.org/10.4018/IJeC.2020010102.

90 Niya, S.R., Dordevic, D., and Stiller, B. (2021). ITrade: a blockchain-based, self-sovereign, and scalable marketplace for IoT data streams. *IFIP/IEEE International Symposium onIntegrated Network Management*, May 2021, IEEE. https://doi.org/10.5167/UZH-198473.

4

Security, Privacy, and Trust of Distributed Ledgers Technology

Saqib Rasool[1], Muddesar Iqbal[2], Shancang Li[3], Tasos Dagiuklas[2], and Saptarshi Ghosh[2]

[1] *Department of Computing and IT, University of Gujrat, Gujrat, Pakistan*
[2] *Division of Computer Science and Informatics (CSI), School of Engineering, London South Bank University, London, UK*
[3] *Department of Computer Science, School of Computer Science and Informatics, Cardiff University, Abacws, Cardiff, UK*

The distributed ledgers technology (DLT) [1] refers to decentralized technological infrastructure and protocols that allow all participants in the connected system to access, verify, and store updates in an immutable and traceable way across the whole decentralized system. Blockchain technology [2] is a typical example of DLT that can record, validate, and store transactions using cryptographic [3] hash signatures. In DLT, each distributed participant, as a ledger, is able to process and verify every transactional item that can be processed based on a consensus of multiple participants. Unlike the central authority-based ledger systems, which need a central authority to validate the authenticity of transactions recorded in the ledgers, the DLT utilizes cryptography algorithms to automatically access, validate, and record transactions based on a specific consensus algorithm in the decentralized network [4].

As a typical DLT implementation, the blockchain bundles transactions into "blocks" that are "chained" together through their respective cryptographic hashes. Blockchain technologies have attracted much attention across industries and sectors, such as cryptocurrencies, supply chains, finance systems, and banking systems [5]. The DLT has great potential to improve the way of governance, institutions, and corporations work by offering a way to securely and efficiently create a tamper-proof record of sensitive actions and activities. In recent, DLT has been widely researched and several distributed ledger solutions have been developed, such as Hyperledger Fabric, Ethereum, Quorum, and R3 Corda.

Blockchains: Empowering Technologies and Industrial Applications, First Edition.
Edited by Anwer Al-Dulaimi, Octavia A. Dobre, and Chih-Lin I.

The DLT is based on cryptography algorithms [6], decentralization networks [7], and consensus protocols [8], which ensure trust among participants through fair execution of transactions. Data in DLT is structured into chained blocks inherent security properties. Each new block is chained to all the blocks before using a cryptographic chain in such a way that it is nearly impossible to tamper with the data stored in the ledger [9]. All transactions within the blocks are validated and agreed upon through a consensus mechanism, ensuring that each transaction is correct and true. According to the nature of the DLT, there is no single point of failure, and a single user cannot change the record of transactions. However, DLT-based technologies are different with many critical security aspects. In DLT or blockchain, the data is organized by cryptographically connected chained blocks, and each new block connects to the blocks chained before it in a cryptographic chain [10]. In this decentralized system, a single point of failure at each participant cannot change the record of transactions.

This chapter has been divided into three distinct sections. Following is an overview of content covered in each section:

Section 4.1 explains the evolution of distributed databases into blockchain and DLTs. It also elaborates the differences between distributed databases, blockchain, and DLTs. After presenting the idea of the CAP theorem [11], it further elaborates the relationship of three basics pillars of the CAP theorem, viz. (i) Consistency (C), (ii) Availability (A), and (iii) Partition Tolerance (P). Section 4.1 also explains the constraints of DLTs for supporting only two of the three features of the CAP theorem and focuses on the three mechanisms of the PoW (Proof of Work) [12], PBFT (Practical Byzantine Fault Tolerance) [13], and TDAG (Transactions-based Directed Acyclic Graph) [14] for, respectively, achieving the AP, CP, and CA. This chapter will introduce the detailed DLT with respect to the CAP theorem along with its associated security and privacy issues in the DLT.

4.1 CAP Theorem and DLT

Figure 4.1 depicts the blockchain as a subset of the DLTs that is a further subset of the DDBS (distributed database systems). However, the origin of DLTs lies in the blockchain and that of blockchain in the DDBS [15]. Hence, the DDBS initially evolved into the blockchains that further gave rise to the DLTs. Therefore, this section initiates by presenting the evolution of DDBS to the blockchain and then extends that discussion to the transformation of blockchains to the DLTs.

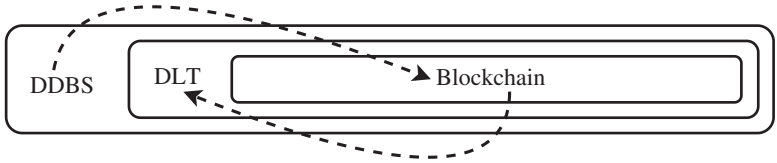


Figure 4.1 The evolution of DLT (distributed ledger technology) to blockchain and distributed database system (DDBS) and relationship of DDBS, DLTs and blockchain.

4.1.1 Distributed Database System (DDBS)

A database is an application that abstracts the operations of data handling of a system. With the tremendous growth in the data generation capabilities, databases encounter the requirement of data and computation scaling at a scale that can be tackled through vertical scaling only. Vertical scaling is a technique that refers to the improvements in the system capabilities that are hosting a database application. Hence, a horizontal scaling approach needs to be adopted for supporting the increasing burden of data management.

The horizontal scaling approach utilizes multiple computing nodes for distributing the computational load across multiple machines. The implementation of a horizontal scaling approach is comparatively simpler for stateless applications that only require the load balancing of the computation across different computational nodes. However, it is quite challenging for stateful applications that require the load balancing of both computation and storage. DDMS (Distributed Database Management System) [16] comes in handy for the stateful application by supporting the horizontal scaling of data management on multiple nodes.

4.1.2 Evolution of DDBS to the Blockchain

A blockchain can be considered as an improved version of a DDBS with some extra constraints. A database offers all four CRUD (Create, Read, Update, and Delete) operations, while a blockchain only supports the create and read operations. We can also achieve the update operation in the blockchains by appending the new values in the ledger of the blockchain. However, the old values will also remain available in the ledger and new updated values will not be able to replace the old values in the ledger.

4.1.3 Public vs Permissioned Blockchains

Figure 4.2 shows three broader categories of blockchain solutions, viz. (i) public, (ii) permissioned, and (iii) hybrid. Public blockchain solutions extend to the

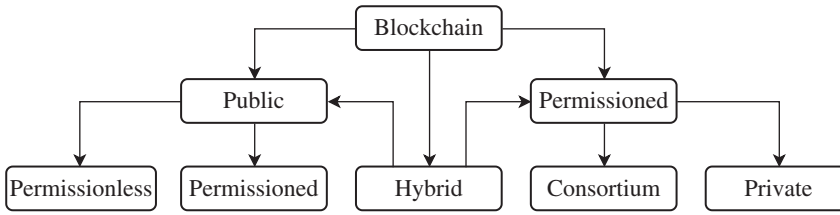


Figure 4.2 Types of blockchains.

permission-less [17] (which is mostly interchangeably used for public blockchain) and public-permissioned blockchain that is a relatively new idea for referring to the **without barrier entry of identifiable nodes** in a public network. Similarly, permissioned blockchain solutions extend to the private (usually referred to as a read-only copy of a distributed database) and a consortium blockchain where a group of known members takes the administrative decisions of a blockchain.

The hybrid blockchain is a relatively new addition to the blockchain arena. It refers to a collection of multiple blockchain ledgers under a single umbrella. According to Figure 4.2, a hybrid blockchain solution may access both public and permissioned blockchain solutions. Hence, a hybrid blockchain can simultaneously act as a public blockchain and a permissioned blockchain, depending upon the operational configurations at any specific time interval.

4.1.4 Evolution of Blockchain to the DLTs

A blockchain is just like a data structure of a link-list where the blocks of data are linked in a chain and are secured through cryptographical hashes. However, more data structures are proposed for offering similar features to a blockchain system. These systems are known as the DLTs and are considered as the superset of the blockchain systems. IOTA is an example of such a system that uses the data structure of a DAG (Directed Acyclic Graph), instead of the link-list, to offer features similar to the blockchain. Section 4.6 covers the DLT solution of IOTA in more detail.

4.2 CAP Theorem

Figure 4.3 presents an overview of the CAP theorem [18] that contains three competing properties of Consistency, Availability, and Partition tolerance. These three properties can be achieved in a centralized system but are not achievable in a distributed system. Only two of the three CAP properties can be strongly achievable

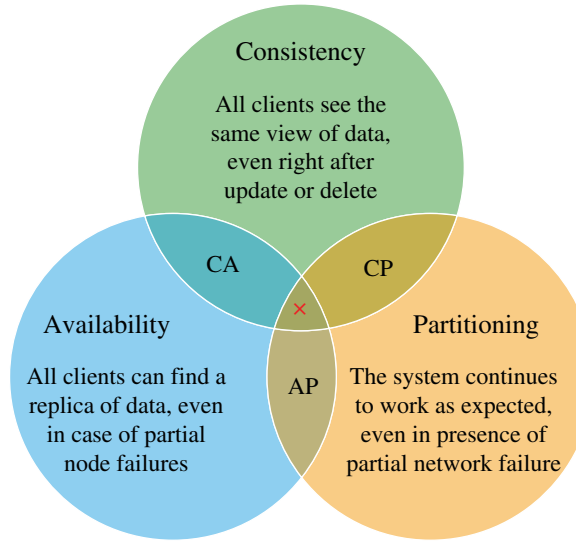


Figure 4.3 An overview of CAP theorem.

in a distributed system. Hence, a trade-off for one of the properties is necessary to achieve for claiming the remaining two properties in one of the three pairs of CA, CP, or CA. The decision of this trade-off drastically affects the overall behavior of a distributed application. Therefore, the particular requirements of an application dictate the decision of this trade-off.

4.2.1 CAP Theorem and Consensus Algorithms

Figure 4.4 shows a triangle with three properties of the CAP theorem. Each of these three properties can be combined in three different pairs that are reflected in the diagram. Each of the three dimensions of the CAP theorem refers to three different types of consensus algorithms. Each consensus algorithm further gives rise to a different set of properties that are covered in detail in Section 4.2.

4.2.2 Availability and Partition Tolerance (AP) Through PoW

The consensus algorithm [19] used in the first generation of blockchain solutions is known as the PoW (Proof of Word). It follows the order-execute model which ensures the ordering of transactions before executing these. It was not only the pioneer consensus algorithm in the DLT world but also the most popular consensus algorithm to date. The consensus algorithm of PoW follows the properties of AP after compromising the property of consistency. More details of this consensus algorithm and associated properties are covered in Section 4.4.

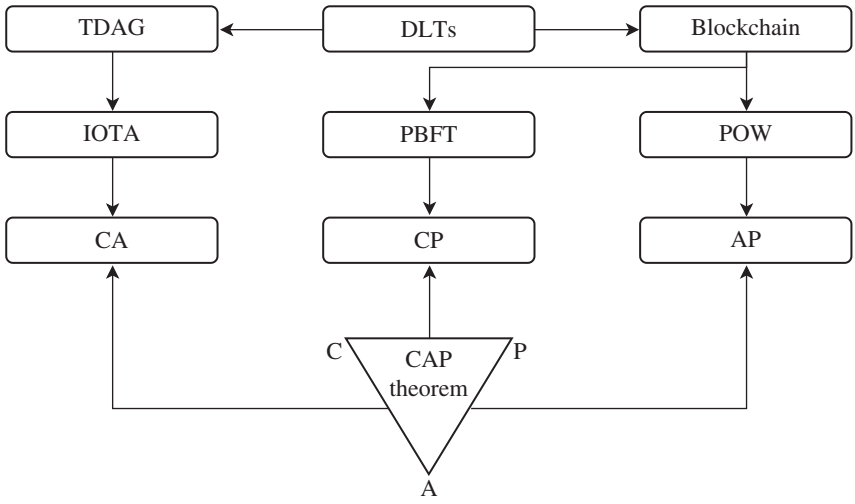


Figure 4.4 DLT solutions against each of the three compatible pairs of the CAP theorem.

4.2.3 Consistency and Partition Tolerance (CP) Through PBFT

The consensus algorithm of PBFT achieves the CP properties after compromising the property of availability. It follows the execute-order-validate approach in which the execution of transactions takes place before the ordering of that transactions. This execute-order-validate approach was introduced in the Hyperledger Fabric first. More details of this consensus algorithm and associated properties of the Hyperledger Fabric 1.x and 2.x are covered in Section 4.5.

4.2.4 Consistency and Availability (CA)

The consensus algorithm used in IOTA [20] is based on a transactional graph known as TDAG or Tangle. It achieves the CA properties after compromising the property of partition tolerance. More details of its consensus algorithm and associated properties are covered in Section 4.6.

4.3 Security and Privacy of DLT

It is important to understand the security requirements of DLT and identify the type of vulnerabilities. This section, will introduce the basic security properties of DLT, specifically focuses on Blockchain.

4.3.1 Security Differs by DLT

As one type of distributed ledger, blockchain networks may be different in the way that each participant access the data. Blockchain networks typically can be categorized into public network or private network depends on the permission and access the network. Public blockchain networks allow anyone to join them and validate transactions, which allow anonymous public participants access, such as bitcoin networks. In private blockchain networks, only confirmed participants are permitted to maintain the transaction ledger and access the network and achieve consensus through a process called "selective endorsement." Both public DLT and private DLT can achieve greater decentralization and distribution.

4.3.2 Security and Requirements for Transactions

The security and privacy requirements for transactions can be categorized into following aspects [21]:

- **Consistency** of Ledger across institution. Different institutions have their own requirement based on the architecture and business processes; however, the inconsistencies between ledgers may cause errors.
- **Integrity** of transaction, the blockchain system should be able to guarantee integrity of transactions and prevent tempering.
- **Prevention of double-spending**. Double spending is one of key challenges in DLT and robust security mechanisms and countermeasures need to be implemented in DLT to prevent spending a coin more than once.
- **Unlikability** of Transaction. In DLT, a participant should require that transactional records related cannot be linked to prevent inferring other information about the specific participant, e.g. account balance, user type, and frequency of transactions, *etc*.

4.3.3 Security Properties of DLT

A DLT system involves many security properties. Table 4.1 summarizes the security properties in DLT. Basically, the security properties in DLT can be classified into consistency, tamper-resistance, DDoS attack resistance, resistance to Double-Spending Attacks, Majority (51%) attack Resistance, Consensus Attack, and Pseudonymity [22].

Main security and privacy properties [23] in DLT can be summarized as

- **Consistency**: In DLT, the consistency denotes to the property that all participants have the same decentralized ledgers when they access the DLT at the same time. The eventual consistency model is proposed to balance between availability (A) and consistency (C), in which the performance (e.g. latency/availability) is a key challenge.

Table 4.1 Security and privacy requirements, properties, and techniques in DLT.

	Requirements	Properties	Corresponding techniques
Blockchain	• Consistency	• Consistency	• Consensus protocols
	• Integrity	• Tamper-resistance	• Hash algorithm
	• Availability	• Resistance to DDoS	• Signature
Needs to be	• Un-linkability	• Unlinkability	• Signature
enhanced	• Confidentiality	• Majority attack	• HE
			• Consensus algorithms

Source: Adapted from Zhang et al. (2019).

- **Tamper-resistance**: It means the resistance to the intentional tampering from the network to an entity by either the participants or the adversaries with access to the DLT entity. The tamper-resistance is usually used to guarantee that transactional data stored in DLT cannot be tampered during/after the process of block creation. Usually, there are two possible tampering ways for transactions: (i) attempts to tamper with information of received transactions; (ii) attempts to tamper with the information stored on the DLT [24].
- **Resistance to DDoS attacks**: Unlike the DoS attack, which refers to denial-of-service attack on a host, the DDoS refers to "distributed" DoS attack to a victim. A DDoS attacker focuses on the availability of DLT and is related to the question of whether a DDoS attacker can make the DLT unavailable by knocking out a partial or the whole network [25]. A cyber-attacker aims at making DLT offline by compromising the availability of computation resources of participants.
- **Resistance to double-spending attacks (DPA)**: In DLT and blockchain, the double-spending is one of key attacks, in which an attacker can create/send a copy of the transaction to make it look legitimate [26]. To prevent Double-Spending Attacks [27], DLT and blockchain systems (e.g. Bitcoin) need to evaluate and verify the authenticity of each transactions using the transaction logs in its blockchain with a consensus protocol, in which all transactions are included in Blockchain and the consensus protocol allows every participant to publicly verify the transactions in a block before committing the block into the global block. By combining transactions signed with digital signatures [28] and public verification, DLT can be resistant to the DPA.
- **Resistance to the Majority Consensus Attack (MCA) [29]**: The MCA, also called 51% Attack, means the risks of cheating in the majority consensus protocol. If a powerful user/group is able to control the whole DLT network, then the consensus protocol will be compromised.

- **Anonymity and pseudonymity**: In DLT systems, transactions are traceable, which may compromise the privacy of users. DLT uses pseudonyms for privacy to shield identity of user as part of self-sovereignty.
- Other security and privacy properties in DLT includes unlinkability, confidentiality of transactions and data privacy, etc.

4.3.4 Challenges and Trends in DLT Security

It is a challenging task to achieve security and privacy protection in a DLT system that needs to meet multiple security and privacy requirements [30]. In this chapter, we summarized three remarks to achieve this:

- To achieve security and privacy of DLT is a complicated task and appropriate techniques should be applied based on the security requirements and the context of applications. The security and privacy protection needs to combine multiple techniques, e.g. HE [31], ABE [32], and SMPC [33].
- The efficiency and security needs to be well trade-off in complicated DLT systems, specifically at the "thin node" and "full node."

4.4 Security in DLT

This section will introduce details of the security in DLT and blockchain. Basically, the DLT security involves the five aspects:

4.4.1 Governance Scenario Security

The DLT ecosystem rules and permission manage onboarding of participants into the network and the roles within the network, which involves following security features

- identity and access management
- key management over physical level security
- security guidelines and policies in organizations
- Security Information and Event Management (SIEM)

4.4.2 DLT Application Security

The DLT widely uses smart contract to conduct automatically agreements between parties to manage the access, application data, third-party apps, etc. over the platform. The related security features include

- DLT application security
- Code security
- Third-party application security and vulnerability assessment

4.4.3 DLT Data Security

Data generated in DLT is stored on the chain that can be encrypted individual and aggregated into the chain blocks, which involve following security features:

- data encryption and key management
- data privacy regulations and guidelines
- off-chain data security

4.4.4 Transactions Security

In DLT, each participant commits transactions to decentralized ledgers with consensus algorithms; to this end, the security considerations include:

- secure and reliable consensus algorithm against double-spending, censorship
- fork management and maintenance

4.4.5 DLT Infrastructure Security

In a DLT ecosystem, participant nodes reside in blockchain networks and systems, and communicate to each other through the public or private connections.

- auditing, monitoring, and logging
- node security and management
- network vulnerability assessment.

4.5 Privacy Issues in DLT

In DLT, privacy is the capability to choose whether information is disclosed to others and determine how it issues. This section raises the privacy questions and focuses on key features associated with DLT. The distribute aspect of DLT means that each participant that processes transactions and builds the blockchain necessarily has access to the data itself, which means the DLT is publicly available and every transaction/event can be traced back to the original genesis block.

Another issue is that the pseudonymous location of data makes it a big concern in terms of it being open for scrutiny by everyone. The public nature of DLT makes the privacy-preserving very challenging. This section summarizes the key data privacy issues in DLT.

- Many DLT applications are based on the mobile/IoT devices, in which sensitive data faces the threats of breaches and compromises by the third-party apps that can collect and control massive amount of sensitive data [34].

- Privacy issues in DLT systems, including smart contract, consensus mechanism, data controller, data processor or service, etc.
- Privacy issues raised in the operation of DLT. The public or permission-less DLT applications allows everyone in any location to access and participate in the network; these activities may cause risks with traditional centrally administered mode.
- Recent data privacy law. Recently, a number of data privacy regulations have been proposed to address a general policy and regulatory concerns. Some key issues between DLT and data privacy requirements have been raised, e.g. how to identify data controllers and processors in DLT implementations, and territorial implications.

Recently, a number of privacy-preserving solutions have been proposed for DLT and its applications: Baskaran et al. [35] introduced an access control moderator and off-blockchain solution to address the decentralizing privacy and trust in a third party. In [36], local private blockchain is used to keep track of transactions and enforce nodes access control policy to address the data privacy in DLT. Kaaniche and Laurent proposed a cryptographic protocol between blockchain and users to preserve the transactional privacy of smart contract [37]. In [38], the secure multi-party computation is used to address the privacy of raw data in DLT.

4.6 Cyberattacks and Fraud

The DLT technology is believed to be a tamper-proof ledger of transactions, while DLT networks are not immune to cyberattacks and fraud. In this section, we will simply introduce the vulnerabilities in Blockchain infrastructures.

- Code exploitation
- Stolen keys
- Employee computer hacked
- Data access and Disclosure.

Cyberattacks have become increasingly targeted and complex due to more sophisticated malware

4.6.1 Challenges

The self-descriptive title of the DLTs depicts their inherent nature of distributed ledgers. Hence, it is crucial to assert strong security and strong privacy-preserving policies for gaining the trust of the stakeholders of a DLT solution. However, the data distribution through the shared ledger makes it challenging to establish the security and privacy of a DLT solution. This chapter focuses on the

stated challenge and presents an overview of the state-of-the-art mechanism of well-established solution DLT solutions in the market. It will be helpful for both academic researchers and industrial practitioners in understanding the current market trends for managing the security, privacy, and trust of a DLT solution.

4.6.2 Key Privacy and Security Techniques in DLT

As mentioned above, it is very important to leverage the security and privacy and the usability in DLT. This section will introduce the techniques to enhance the security and privacy of DLT.

- **Mixing**: The DLT usually does not guarantee anonymity for users (but provides traceability for transactions), in which transactions use pseudonymous addresses that could be publicly verified. Mixing (or tumblers) is a technique of random exchange of information between users to prevent users' addresses from being linked, which is widely used in the crypto-currencies. Typical mixing techniques include *Mixcoin, CoinJoin, etc.*
- **Anonymous signature**: In DLT, anonymous signature schemes (group signature, ring signature) were proposed to provide anonymity for the signer.
- **Homomorphic encryption (HE)**: HE has been significantly improved recently and has becomes a powerful cryptography technique that can perform computations directly on ciphertext without needing decrypt them. The DLT can use HE techniques to deal with data over the chain with no significant changes in the blockchain properties to ensure that the data on the chain will be encrypted. This could address the privacy concerns. The HE techniques provide privacy protection and allow access to encrypted data over public DLT.
- **Attribute-based encryption (ABE)**: The ABE is a public-key encryption method in which the secret key of a user and the ciphertext are dependent upon attribute. In DLT, the decentralized ABE can be employed, in which the permissions could be represented by the ownership of access tokens.
- In DLT, the trusted execution environment (TEE) could provide a privacy-preserving running environment for smart contracts. However, it needs extra software support, such as the Intel software guard extensions (SGX).

4.7 DLT Implementation and Blockchain

In this section, we focus only on the DLT solutions that have been well established in the crypto market. We do not cover the solutions that are in the proposal or research phases. For example, a research project of zkLedger utilizes zero-knowledge proofs for the auditing of the private data stored on the ledger. However, we do not cover this project since it has no industrial footprint yet.

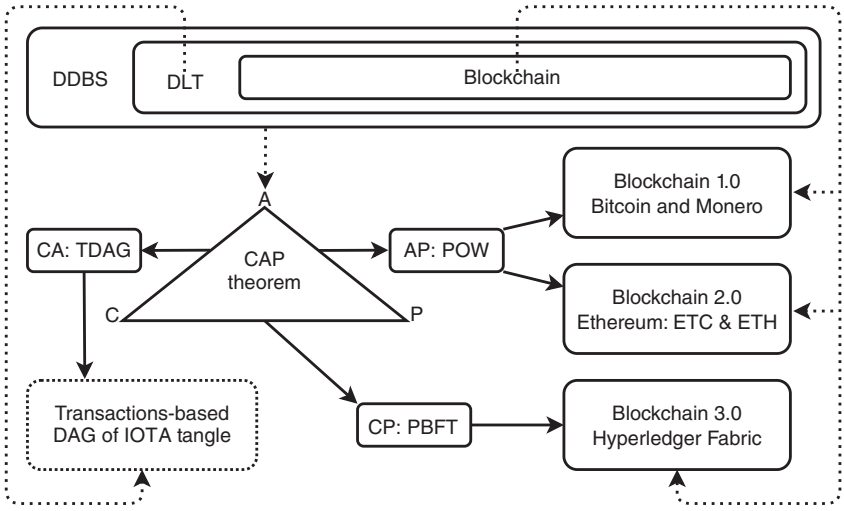


Figure 4.5 Flow of content for the section of implementations of different DLT solutions.

Similarly, from each discussed category of the DLTs, we focus on the most popular and successful DLT solution or framework, as shown in Figure 4.5. For example, there are many projects under the umbrella of hyperledger, but we only cover the most popular option of hyperledger fabric in Section 4.5.

4.7.1 Cryptocurrencies and Bitcoin

The first generation of blockchain solutions was focused only on a single application of financial services through virtual tokens that are known as cryptocurrencies. The consensus algorithm is the main contributing feature of the first generation of blockchain solutions, and it provided the technical foundations for offering public blockchain solutions of cryptocurrencies. Thousands of different cryptocurrencies were launched to date. However, bitcoin was the first and most popular cryptocurrency that we are going to cover in this section. We will also cover the two privacy coins of Monero and zcash in this section.

4.7.1.1 Origin of Blockchain

Bitcoin was the first project to initiate the idea of blockchain in 2009. It operates over the PoW that provides the basis for the collaboration of independent and non-trusted entities to execute the transactions in a distributed manner. Although PoW usually refers to serve as a consensus algorithm in theory, it is just a Sybil control mechanism that combines with the idea of the selection of the longest chain for practically serving as a consensus algorithm.

4.7.1.2 Bitcoin

Although bitcoin is believed to be the most secure blockchain solution, it has the following shortcomings regarding security, privacy, and trust:

- **Security**: Bitcoin is the most secure blockchain solution. However, it operates over a very costly mining algorithm of the nonce (number of ones) finding. Furthermore, this PoW needs to combine with the selection of the longest chain to serve as a consensus mechanism. It results in some issues like selfish mining strategies, withholding, towing, and temporary shutdown tactics. These issues allow the groups with huge mining capabilities to influence the consensus algorithm for their interests. However, the number of minable coins is reducing over time and will eventually result in the depletion of all the minable coins. Hence, the severity of the listed problems will automatically keep on reducing.
- **Privacy**: Bitcoin offers pseudo-anonymity by representing users with unique arbitrary hashes that can be traced by linking multiple transactions listed in the ledger of bitcoin. Furthermore, the privacy of stored data is preservable through the POE (Proof-of-Existence).
- **Trust**: Although bitcoin is the most trusted cryptocurrency, a few of the reported shortcomings may result in an uneven distribution of coins during the mining process that we discussed in the security issues. However, these minor issues are not well known and thus the bitcoin is the most trusted cryptocurrency to date.

4.7.1.3 Monero

Monero [39] is a privacy coin that shines in handling the privacy of the end-users. It is the most popular cryptocurrency that ensures the anonymity of the stack-holders. Dash [40] and z-cash [41] are also the two privacy coins. However, the dash is lagging in popularity, whereas the z-cash offers both open and stealth transactions. Hence, we discuss the Monero and z-cash in this section. Details of Monero are given below:

- **Security**: Monero shifted to the mining algorithm of RandomX [42] in November 2019 that encourages CPU mining by resisting ASIC mining. A couple of vulnerabilities have been reported and successfully patched in Monero's algorithm before that. Monero is lesser mature since it was launched five years after the bitcoin in 2014. Its algorithm is also more complex than the bitcoin. Hence, there are more chances of zero-day vulnerabilities in Monero that makes it lesser secure than Bitcoin.
- **Privacy**: Monero is the most popular privacy coin on earth. It earned this title by offering full anonymity to the end-users performing transactions through this coin. Monero achieves this anonymity through Stealth addresses along with a non-interactive zero-knowledge proof (NIZKP) [43] protocol implemented as a bulletproof algorithm. However, its privacy feature is only limited to the

anonymity of the end-users. For the privacy of stored data, similar to bitcoin, Monero also depends on the POE.

- **Trust**: Monero is well trusted by the end-users. However, the regulatory bodies are not ready to trust Monero due to its non-compliance with AML (Anti-Money Laundering) laws since it resists the notion of KYC (Know Your Customer) by offering full anonymity to the end-users.

4.7.2 Blockchain and Smart Contracts

The second generation of blockchain focuses on multiple applications based on cryptocurrencies. Smart contracts are the main contributing feature of this generation, which provides the technical foundations for offering public/permissioned blockchain solutions for DApps (Distributed Applications) only.

The second generation of blockchain solutions exploits the potential of smart contracts for innovating solutions alongside cryptocurrencies. ETH appeared in 2015 as a pioneer of the second generation of blockchain solutions by offering JavaScript-inspired programming language of solidity for writing smart contracts.

Forbes found more than 100 large American corporations that were actively exploring blockchain technology in 2019, and many of them focused on the ETH network. Since ETH was the first second-generation blockchain solution, it presents few lessons that are adaptable in the feature versions of the second-generation blockchain solution. Hence, it was devised to proceed with two independent versions of ethereum classic (ETC) and ethereum (ETH). This section covers both of these versions of ETC and ETH in detail.

4.7.3 Typical Blockchain Systems

4.7.3.1 Ethereum Classic (ETC)

ETC [44] is an open-source solution that powers the public blockchain network of ETC. It can also be used for establishing private blockchain networks. Next is the discussion on the security, privacy, and trust features of the first version of ETH.

- **Security**: After the first year of its launch, the ETH underwent a 51% percent attack on DAO (Decentralized Autonomous Organization) in 2016 that resulted in the splitting of ETH into ETC and ETH. ETC has also gone through three consecutive 51% attacks in one month of August 2020. It reflects the vulnerability of smart contracts and ETH is shifting to POS in its next version of ETH 2. Unfortunately, ETC is not backward compatible with ETH, so it will not be able to take advantage of ETH's migration to the POS.
- **Privacy**: Same like first-generation blockchain platforms, ETH is pseudonymous and it also depends on the POE for ensuring the privacy of the stored data.

- **Trust**: Due to the hard-fork of ETH from ETC, the community has observed the vulnerability of taking drastic decisions by neglecting the opinion of others. This becomes a more serious concern in the next version of POS-based ETH where larger stakeholders can also influence the future decisions of the ETH that eventually leads to lesser decentralization.

4.7.3.2 Ethereum (ETH)

ETH originated as a result of a hard-fork of ETC. It is better than ETC and therefore more trusted by the industry as compared to the ETC. However, it has privacy controls similar to the ETC.

4.7.3.3 Extensibility of Blockchain and DLT

The third generation of blockchain focuses on the interoperability of multiple applications without being dependent on cryptocurrencies. The flexibility for custom policies and consensus algorithms is the main contributing feature of this generation, which provides the technical foundations for offering permission blockchain solutions that can exist without cryptocurrencies.

4.7.4 Origin of Blockchain 3.0

The blockchain framework of fabric is operated by the Linux Foundation under the umbrella of the hyperledger ecosystem. The initial modular structure of hyperledger fabric [45] was contributed, in 2016, by IBM and digital assets for giving an improved version of blockchain solutions that are not primarily focused on the digital assets of tokens or cryptocurrencies. Hence, it inherently differs from the first two generations of blockchain as the first generation was only based on cryptocurrencies, while the second generation was an extension of the cryptocurrencies with an extra layer of smart contracts. However, the advent of third generation, in the form of hyperledger ecosystem, was not targeting the cryptocurrencies. Hence, end-users can very easily develop tokenless blockchain solutions that can operate without the need for any cryptocurrencies. The hyperledger ecosystem contains many other tools, frameworks, and libraries while we will be targeting its most popular framework of hyperledger fabric.

4.7.5 Overview of Hyperledger Fabric

ETH (a public, permissionless blockchain) and Quorum (private, permissioned blockchain based on ETH code) are based on execute-order architecture. Some of the limitations that this introduces are sequential execution of all transactions which directly affects transaction throughput. The main concept that differentiates Hyperledger Fabric from other blockchains is its execute-order-validate architecture. Transactions in Hyperledger Fabric do need not be executed by each peer.

We can define the endorsement policy that specifies which peer nodes have to execute the transaction and give their endorsement. This means that we can define a subset of peers to execute (endorse) a given transaction and satisfy the transaction's endorsement policy. Therefore, this allows for parallel execution of transactions and directly boosts performances of the system. Hyperledger separates transaction flow in three distinct steps:

- **Transaction execution**: In this initial phase, the client collects the predefined number of endorsements from the nodes that are already designated as the endorser nodes. These nodes execute the corresponding smart contract and return a stamped version to the requesting client.
- **Ordering**: The same client again sends all the collected endorsements to the predefined orderer node that forwards it to random validating nodes.
- **Transaction validation**: These nodes discard or successfully execute the transactions over the distributed ledger after validating that all the requirements of the consensus algorithms have been met. The collection of a specific number of endorsements in the first step is also dictated by the consensus algorithm.

4.8 DLT of IOTA Tangle

All of the previously discussed solutions are both DLTs and blockchain-based solutions. However, in this section, we are going to introduce the DLT solution of IOTA Tangle which is not a blockchain. It is a tailored solution for IoT devices that compromises the partition tolerance for achieving the consistency and availability features of the CAP theorem.

The solution of IOTA is based on a DAG Tangle technology. The word IOTA refers to both the parent organization and the associated token while the tangle is the protocol and the underlying ledger in the form of a graph. In this graph, the nodes at one end are not entirely aware of the state of the other end of the graph. This is in contrast to the blockchain where all the nodes have exactly similar views of the ledger.

In IOTA, the transactions of different sections of a graph are kept on synchronizing but time spent during this transaction results in the emergence of new transactions at the remote ends of the graph. Hence, different nodes view the different states of the graph at a single time interval. Since IOTA compromises the partition tolerance, thousands of transactions can be executed per second at different ends of the graph. IOTA is also a highly scalable solution as it requires every transaction initiation node to validate two other transactions, originated by the different nodes, for showing the PoW. This gives IOTA an option to operate with zero fees as compared to the other blockchain-based DLT solutions.

In terms of security, smart contracts are landed in the IOTA world recently in a pre-alpha release in October 2020. Hence, these will take time in getting mature and gaining the trust of the audience. Similarly, the IOTA is more vulnerable than the other discussed blockchain-based solutions because it only requires 34% of the total hashing power for taking the control of tangle which is 51% for the blockchain-based solutions. This is because each one node is validating the transactions of the two other nodes. Hence, almost one-third of the malicious nodes of the total nodes will be enough for performing the 51% attack in IOTA. In contrast to the blockchain solutions, IOTA claims to be the quantum resistant since it uses the trinary or balanced ternary computations while blockchain only uses the standard binary cryptographic computations.

4.9 Trilemma of Security, Scalability, and Decentralization

According to Vitalik Buterin, the founder of ETH, it is not possible to equally optimize the three crucial attributes of security, decentralization, and scalability in a blockchain system. Hence, more and more blockchain projects (like Cardano and Polkadot) are originating after tweaking different parameters for trying to optimize all of the features of security, decentralization, and scalability at the same time. Here we are going to explain it with the examples of first- and second-generation blockchain solutions.

4.9.1 First-Generation Solutions: BTC/BCH

The Bitcoin (BTC) operates with a block size of 1 MB while a movement was started to increase the block size of BTC for improving its transaction rate. However, due to a disagreement in the bitcoin community, a hard fork of BTC happened with the title of Bitcoin Cash (BCH) at block number 478559. BCH increased the BTC block size from 1 MB to 8 MB which resulted in improving the transaction rate from 7 transactions per second for BTC to 116 transactions per second for BCH. Although BCH has achieved improvements in transaction rate, they compromised the decentralization but collecting more transactions at the same node. The tremendous popularity of BTC proves that decentralization is very important for winning the trust of the audience.

4.9.2 Second-Generation Solutions: ETH/BSC

To reduce the higher gas price of ETH, Binance launched an exact clone of opensource code of ETH project with less gas price by compromising the

decentralization, under the title of BSC (Binance Smart Chain). Again the tremendous popularity of ETH shows that the crypto community trusts decentralized solutions.

4.9.3 Threats in DLT and Blockchain Networks

Like other DLT and blockchain networks, following threats are common [46]:

- **Spoofing**: Malicious attackers pretend to be or impersonate an authentic user. The HLF attempts to mitigate this with having a high-quality CA built in using the highest quality certificates X.509.
- **Tampering**: The HLF uses the built-in encryption like sha-256 or elliptic curve cryptography algorithms.
- **Repudiation**: The HLF uses a built-in strict logs to track events that lead to ledger creation.
- **Replay attacks**: In some case, the replaying of events will corrupt the blockchain itself. The HLF has read/write sets to validate transactions and if transactions fail, read sets invalidate the transaction.

4.10 Security Architecture in DLT and Blockchain

In many DLT application scenarios, the security standards and regulations are still in its infant stage. This section will introduce the security architecture in DLT and blockchain that can help to establish a secure environment by leveraging the cybersecurity risks, best practices, and risk mitigation. Basically, a DLT system requires assessment, authorization, authority to operate processes to determine whether they comply with security regulation and privacy requirements (e.g. GDPR), and security on DLT entities (e.g. Blockchain networks, participants, actors, etc.).

Figure 4.6 shows an example of security architecture in a DLT application, which contains three key components [47]:

- Risk management and scrutinizing
- Threats analysis and model
- Security controls that mitigate the risks and threats

Physically, it can be implemented over a four-layer architecture:

- **DLT network layer**: It consists of data representation and network services planes, which deals with the storage, encoding, and protection of data, while the network services focus on discovery, communication with protocol peers, addressing, naming system, etc.

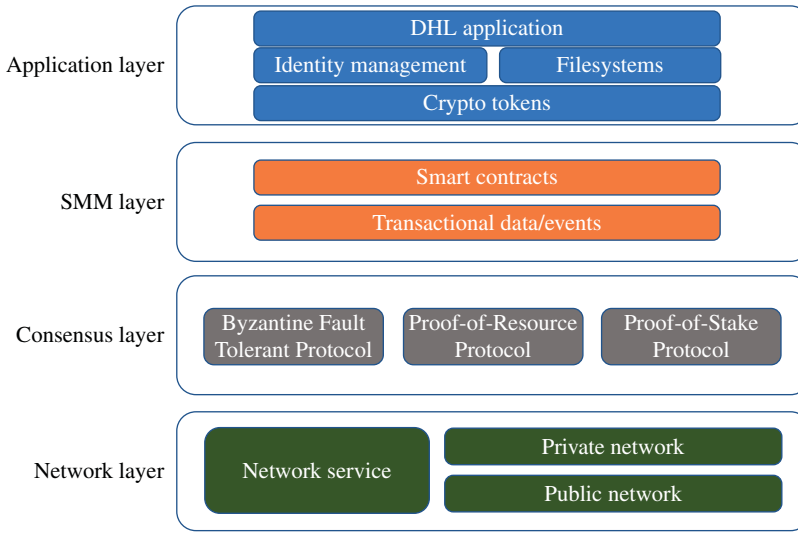


Figure 4.6 DLT security architecture.

- **Consensus layer**: This layer focuses on the dynamic protocol of reaching agreement in a group, which can be classified into three main categories according to the protocols: *Byzantine Fault Tolerant, PoR, POS*.
- **Application layer**: It contains the most common application/services.
- **The state management machine (SMM)**: It deals with the interpretation of transactions.

4.10.1 Threat Model in LDT

Together with the potential benefits, the DLT technologies are also facing a number of potential threats and attack vectors [21], as shown in Figure 4.7. Like other IT systems, the LDT security threats can also be mapped to the Spoofing, Tampering, Repudiation, Information disclosure, Denial of service attacks, and Elevation of privilege (STRIDE) threat model developed by Microsoft.[1] The STRIDE model can be used to address the relationships between entities in LDT, review threats and weakness related to these relationships, and propose appropriate mitigation.

The DLT applications often incorporate with existing IT systems, such as authentication, identity management systems, access control system, regulatory, log and auditing system, and public crypto key system. Aligning with existing system, threats needs to be addressed in the DLT system, as shown in Figure 4.7,[2]

1 https://www.howardposton.com/blog/threat-modeling-for-the-blockchain.
2 https://developer.ibm.com/technologies/blockchain/articles/how-to-secure-blockchain-solutions/.

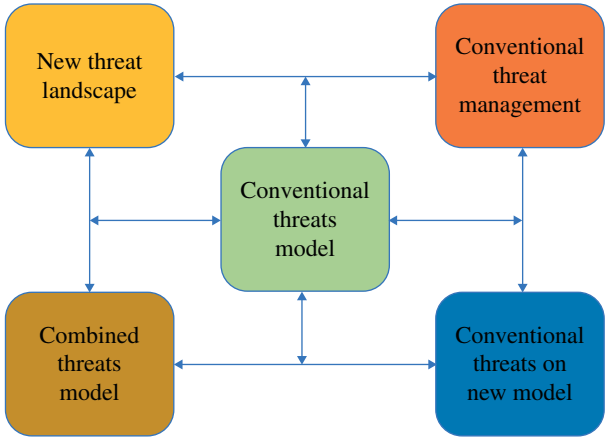


Figure 4.7 Threat model in DLT. Source: Based on [2].

in which it is important to fully understood the *new threat landscape*, new vulnerabilities in DLT infrastructure and tampering with smart contract. For a DLT application, it is not infeasible to build a universal threat model, and specific threats analysis should be conducted based on the application. Also, it is necessary to ensure a secure system environment for a DLT application to use corporate security standards.

4.11 Research Trends and Challenges

One of challenges that the DLT is facing is lack of clarity on the terminology. The DLT has been discussed for a long time; however, there is a big gap between the technical implementation of DLT and businesses model, which makes it difficult to understand how DLT operates in real-world industry. The DLT undoubtedly benefits the existing business processes; however, it is still an open question how to integrate DLT into existing legacy system without disrupting to existing industry practices [48]. Recently, many research efforts focus on the governance of DLT to establish liability among partners in both permissioned and permissionless systems to reduce potential operation failure or compromises.

One of the research trends is to integrate blockchain with an already well-established solutions. For example, DocsChain is a solution that integrates the image processing and blockchain for offering the degree verification [49]. Docs.vet is an improved form of the DocsChain that extends it for the verification of identity documents. Another solution of MultiCoT integrates the blockchain within the osmotic computing to offer the Multi-Cloud of Things

solution [50]. Another project integrates the blockchain in the MEC (Multi-access Edge Computing) to offer the reliable resource sharing for supporting a mobile ad hoc cloud at the edge of the network [51].

References

1 Sunyaev, A. (2020). Distributed ledger technology. In: *Internet Computing*, 265–299. Springer.

2 Nakamoto, S. (2009). Bitcoin: a peer-to-peer electronic cash system. Cryptography Mailing list at https://metzdowd.com (accessed 12 May 2023).

3 Cho, S. and Lee, S. (2019). Survey on the application of blockchain to IoT. *2019 International Conference on Electronics, Information, and Communication (ICEIC)*, 1–2. https://doi.org/10.23919/ELINFOCOM.2019.8706369.

4 Miraz, M.H. and Ali, M. (2018). Applications of blockchain technology beyond cryptocurrency. *Annals of Emerging Technologies in Computing* 2 (1): 1–6. https://doi.org/10.33166/aetic.2018.01.001.

5 Alladi, T., Chamola, V., Parizi, R.M., and Choo, K.-K.R. (2019). Blockchain applications for industry 4.0 and industrial IoT: a review. *IEEE Access* 7: 176935–176951. https://doi.org/10.1109/ACCESS.2019.2956748.

6 Miraz, M.H. and Ali, M. (2018). A survey on cryptography algorithms. *International Journal of Scientific and Research Publications* 8 (7): 495–516. https://doi.org/10.29322/IJSRP.8.7.2018.p7978.

7 Zwitter, A. and Hazenberg, J. (2020). Decentralized network governance: blockchain technology and the future of regulation. *Frontiers in Blockchain* 3: 12. https://doi.org/10.3389/fbloc.2020.00012.

8 Wahab, A. and Mehmood, W. (2018). Survey of consensus protocols.

9 Zhao, S., Li, S., and Yao, Y. (2019). Blockchain enabled industrial Internet of Things technology. *IEEE Transactions on Computational Social Systems* 6 (6): 1442–1453. https://doi.org/10.1109/TCSS.2019.2924054.

10 Li, S., Qin, T., and Min, G. (2019). Blockchain-based digital forensics investigation framework in the Internet of Things and social systems. *IEEE Transactions on Computational Social Systems* 6 (6): 1433–1441. https://doi.org/10.1109/TCSS.2019.2927431.

11 Frank, L., Pedersen, R.U., Havnø Frank, C., and Larsson, N.J. (2014). The cap theorem versus databases with relaxed acid properties. *Proceedings of the 8th International Conference on Ubiquitous Information Management and Communication*, ICUIMC '14, New York, NY, USA: Association for Computing Machinery. ISBN 9781450326445. https://doi.org/10.1145/2557977.2557981.

12 Chepurnoy, A., Duong, T., Fan, L., and Zhou, H.-S. (2017). TwinsCoin: a cryptocurrency via proof-of-work and proof-of-stake. *IACR Cryptology ePrint Archive* 2017: 232.

13 Sukhwani, H., Martínez, J.M., Chang, X. et al. (2017). Performance modeling of PBFT consensus process for permissioned blockchain network (hyperledger fabric). *2017 IEEE 36th Symposium on Reliable Distributed Systems (SRDS)*, 253–255. IEEE.

14 Yeow, K., Gani, A., Ahmad, R.W. et al. (2017). Decentralized consensus for edge-centric Internet of Things: a review, taxonomy, and research issues. *IEEE Access* 6: 1513–1524.

15 IEE Colloquium on 'Distributed Databases' (Digest No.229) (1992). *IEE Colloquium on Distributed Databases*, pp. 0–1.

16 Chen, Y., Xie, H., Lv, K. et al. (2019). DEPLEST: a blockchain-based privacy-preserving distributed database toward user behaviors in social networks. *Information Sciences* 501: 100–117.

17 Helliar, C.V., Crawford, L., Rocca, L. et al. (2020). Permissionless and permissioned blockchain diffusion. *International Journal of Information Management* 54: 102136.

18 De Angelis, S., Aniello, L., Baldoni, R. et al. (2018). PBFT vs proof-of-authority: applying the CAP theorem to permissioned blockchain.

19 Urban, P., Hayashibara, N., Schiper, A., and Katayama, T. (2004). Performance comparison of a rotating coordinator and a leader based consensus algorithm. *Proceedings of the 23rd IEEE International Symposium on Reliable Distributed Systems, 2004*, 4–17. https://doi.org/10.1109/RELDIS.2004.1352999.

20 IOTA (2019). Consensus in the IOTA tangle - FPC. https://blog.iota.org/consensus-in-the-iota-tangle-fpc-b98e0f1e8fa/ (accessed 12 May 2023).

21 Putz, B. and Pernul, G. (2020). Detecting blockchain security threats. *2020 IEEE International Conference on Blockchain (Blockchain)*, 313–320. https://doi.org/10.1109/Blockchain50366.2020.00046.

22 Li, S., Choo, K.R., Sun, Q. et al. (2019). IoT forensics: Amazon echo as a use case. *IEEE Internet of Things Journal* 6 (4): 6487–6497. https://doi.org/10.1109/JIOT.2019.2906946.

23 Zhang, R., Xue, R., and Liu, L. (2019). Security and privacy on blockchain. *ACM Computing Surveys (CSUR)* 52 (3): 1–34.

24 Sun, J., Xiong, H., Zhang, S. et al. (2020). A secure flexible and tampering-resistant data sharing system for vehicular social networks. *IEEE Transactions on Vehicular Technology* 69 (11): 12938–12950. https://doi.org/10.1109/TVT.2020.3015916.

25 Al'aziz, B.A.A., Sukarno, P., and Wardana, A.A. (2020). Blacklisted IP distribution system to handle DDoS attacks on IPS Snort based on blockchain. *2020*

6th Information Technology International Seminar (ITIS), 41–45. https://doi.org/10.1109/ITIS50118.2020.9320996.

26 Sai, K. and Tipper, D. (2019). Disincentivizing double spend attacks across interoperable blockchains. *2019 First IEEE International Conference on Trust, Privacy and Security in Intelligent Systems and Applications (TPS-ISA)*, 36–45. https://doi.org/10.1109/TPS-ISA48467.2019.00014.

27 Zhang, S. and Lee, J.-H. (2019). Double-spending with a sybil attack in the bitcoin decentralized network. *IEEE Transactions on Industrial Informatics* 15 (10): 5715–5722. https://doi.org/10.1109/TII.2019.2921566.

28 Zhu, L. and Zhu, L. (2012). Electronic signature based on digital signature and digital watermarking. *2012 5th International Congress on Image and Signal Processing*, 1644–1647. https://doi.org/10.1109/CISP.2012.6469828.

29 Zamani, M., Khosravian, A., and Ninness, B. (2016). Compensation of attacks on consensus networks. *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 3491–3495. https://doi.org/10.1109/ICASSP.2016.7472326.

30 Li, S., Sun, Q., and Xu, X. (2018). Forensic analysis of digital images over smart devices and online social networks. *2018 IEEE 20th International Conference on High Performance Computing and Communications; IEEE 16th International Conference on Smart City; IEEE 4th International Conference on Data Science and Systems (HPCC/SmartCity/DSS)*, 1015–1021. https://doi.org/10.1109/HPCC/SmartCity/DSS.2018.00168.

31 Tourky, D., ElKawkagy, M., and Keshk, A. (2016). Homomorphic encryption the "holy grail" of cryptography. *2016 2nd IEEE International Conference on Computer and Communications (ICCC)*, 196–201. https://doi.org/10.1109/CompComm.2016.7924692.

32 Qiao, Z., Liang, S., Davis, S., and Jiang, H. (2014). Survey of attribute based encryption. *15th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD)*, 1–6. https://doi.org/10.1109/SNPD.2014.6888687.

33 Shukla, S. and Sadashivappa, G. (2014). Secure multi-party computation protocol using asymmetric encryption. *2014 International Conference on Computing for Sustainable Global Development (INDIACom)*, 780–785. https://doi.org/10.1109/IndiaCom.2014.6828069.

34 Jurcut, A., Niculcea, T., Ranaweera, P., and Le-Khac, N.-A. (2020). Security considerations for Internet of Things: a survey. *SN Computer Science* 1 (4): 193. https://doi.org/10.1007/s42979-020-00201-3.

35 Baskaran, H., Yussof, S., and Rahim, F.A. (2020). A survey on privacy concerns in blockchain applications and current blockchain solutions to preserve data privacy. In: *Advances in Cyber Security* (ed. M. Anbar, N. Abdullah, and S. Manickam), 3–17. Singapore: Springer.

36 Dorri, A., Kanhere, S.S., Jurdak, R., and Gauravaram, P. (2017). Blockchain for IoT security and privacy: the case study of a smart home. *2017 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*, 618–623. IEEE.

37 Kaaniche, N. and Laurent, M. (2017). A blockchain-based data usage auditing architecture with enhanced privacy and availability. *2017 IEEE 16th International Symposium on Network Computing and Applications (NCA)*, 1–5. IEEE.

38 Zyskind, G. and Nathan, O. (2015). Decentralizing privacy: using blockchain to protect personal data. *2015 IEEE Security and Privacy Workshops*, 180–184. IEEE.

39 Wijaya, D.A., Liu, J.K., Steinfeld, R. et al. (2019). On the unforkability of Monero. *Proceedings of the 2019 ACM Asia Conference on Computer and Communications Security*, 621–632.

40 Dash Platform. Dash platform developer documentation. https://dashplatform.readme.io/ (accessed 12 May 2023).

41 Zcash. Z-cash documentation. https://zcash.readthedocs.io/en/latest/ (accessed 12 May 2023).

42 Monero. What is RandomX mining algorithm in Monero? https://academy.bit2me.com/en/which-mining-algorithm-randomx-monero/.

43 Tsai, Y.C., Tso, R., Liu, Z.-Y., and Chen, K. (2019). An improved non-interactive zero-knowledge range proof for decentralized applications. *2019 IEEE International Conference on Decentralized Applications and Infrastructures (DAPPCON)*, 129–134. https://doi.org/10.1109/DAPPCON.2019.00025.

44 di Angelo, M. and Salzer, G. (2019). A survey of tools for analyzing ethereum smart contracts. *2019 IEEE International Conference on Decentralized Applications and Infrastructures (DAPPCON)*, 69–78. https://doi.org/10.1109/DAPPCON.2019.00018.

45 Dabbagh, M., Kakavand, M., Tahir, M., and Amphawan, A. (2020). Performance analysis of blockchain platforms: empirical evaluation of hyperledger fabric and ethereum. *2020 IEEE 2nd International Conference on Artificial Intelligence in Engineering and Technology (IICAIET)*, 1–6. https://doi.org/10.1109/IICAIET49801.2020.9257811.

46 Saad, M., Spaulding, J., Njilla, L. et al. (2020). Exploring the attack surface of blockchain: a comprehensive survey. *IEEE Communications Surveys & Tutorials* 22 (3): 1977–2008. https://doi.org/10.1109/COMST.2020.2975999.

47 Homoliak, I., Venugopalan, S., Hum, Q., and Szalachowski, P. (2019). A security reference architecture for blockchains. *2019 IEEE International Conference on Blockchain (Blockchain)*, 390–397. https://doi.org/10.1109/Blockchain.2019.00060.

48 Wustmans, M., Haubold, T., and Bruens, B. (2021). Bridging trends and patents: combining different data sources for the evaluation of innovation fields in blockchain technology. *IEEE Transactions on Engineering Management* 69 (3): 825–837. https://doi.org/10.1109/TEM.2020.3043478.

49 Rasool, S., Saleem, A., Iqbal, M. et al. (2020). Docschain: blockchain-based IoT solution for verification of degree documents. *IEEE Transactions on Computational Social Systems* 7 (3): 827–837.

50 Rasool, S., Saleem, A., Iqbal, M. et al. (2020). Blockchain-enabled reliable osmotic computing for cloud of things: applications and challenges. *IEEE Internet of Things Magazine* 3 (2): 63–67.

51 Rasool, S., Iqbal, M., Dagiuklas, T. et al. (2020). Reliable data analysis through blockchain based crowdsourcing in mobile ad-hoc cloud. *Mobile Networks and Applications* 25 (1): 153–163.

5

Blockchains for Business – Permissioned Blockchains#

Ziliang Lai and Eric Lo

Department of Computer Science and Engineering, The Chinese University of Hong Kong, China

5.1 Introduction

Bitcoin, the world's first cryptocurrency, is continuing to renew its capitalization of over a recording-breaking of US$ 700 billion in 2021. The technology behind Bitcoin, blockchain, has also been attracting significant attention in both industry and academia because it enables business between *mutually untrusted* parties [1–3]. This fascinating property is especially attractive in business collaborations where multiple parties cooperate for a common goal without putting trust in each other. Without blockchain, an independent third-party is often required to mediate the business and resolve disputes, which incurs significant running costs. Blockchain is regarded to be a promising resolution that could revolutionize the landscape of business collaborations [1].

Consider a typical inter-bank transfer service, where multiple commercial banks are involved. Without putting trust in each other, a Central Bank backed by a government is often the key enabler of such collaborations. While the commercial banks hold accounts of their customers, the Central Bank holds an account for each commercial bank. Say Alice transfers US$ 10k from her Bank-A account to Bob's Bank-B account. To actually transfer the money from Bank-A to Bank-B, Bank-A transfers US$ 10k from its Central Bank account to Bank-B's Central Bank account. The Central Bank has been considered to be a trustworthy mediator, but it charges handling fees for bearing all the risks and maintaining the service.

Blockchain is a potential cheaper replacement to the Central Bank's role in the inter-bank transfer service. Instead of placing trust on a third-party, blockchain

This work is partially supported by Hong Kong General Research Fund (14200817, 15200715, 15204116), Hong Kong AoE/P-404/18, Innovation and Technology Fund ITS/310/18.

Blockchains: Empowering Technologies and Industrial Applications, First Edition.
Edited by Anwer Al-Dulaimi, Octavia A. Dobre, and Chih-Lin I.

establishes trust by the majority. Besides digital signature verification and access control that controls basic privacy and security, any update to the blockchain ledger additionally requires consents from the majority via a consensus protocol, and the update is irreversible because the ledger (data) is tamper-proof. By deploying a smart contract that codes the Central Bank's role (i.e. maintain each commercial bank's account and handle transfer request) as a program, the inter-bank transfer service can be realized without a central bank. Any business collaboration can establish their own blockchain, thereby saving all the handling fees.

Using blockchain in a business setting, however, exerts new requirements that public blockchains fail to support. First, business applications require high throughput. For example, Visa is reported to process transactions at more than 1700 transactions per second (TPS) while public cryptocurrency-based blockchains like Bitcoin and Ethereum only support 7 TPS and 20 TPS, respectively [4]. Second, low tail-latency is as critical as throughput for user experience. Since the proof-of-work (PoW) consensus used in Bitcoin and Ethereum takes a long time to form a block (e.g. 10 s in Ethereum) and a block is finalized only when it has a certain number of blocks that follow it (e.g. 20 blocks in Ethereum), their resulting latency is unacceptable in most business applications. Recently, some new public blockchains claim to offer much higher throughput and lower latency by using some new consensus protocols (e.g. Algorand offers 1000 TPS with 45 s latency [5]), their performance is still way far from useful in business settings.

Private blockchains, as known as *permissioned blockchains*, are blockchains designed for business collaborations. Permissioned blockchains offer higher throughput and lower latency while preserving the ability to enable collaboration without mutual trust and mediators. It is made possible because permissioned blockchains are designed for a less hostile setting than the open Internet setting assumed in public blockchains. Concretely, public blockchains run in peer-to-peer (P2P) networks, where everyone is allowed to join by simply creating a pseudonymous ID (e.g. the public key). Thus, public blockchains have to defend against Sybil attacks using some expensive consensus protocol like PoW [6]. In contrast, permissioned blockchains are maintained in a private network where the peers are business partners. Since each peer exposes its real identity in the network, permissioned blockchains do not have to defend against Sybil attack and can use cheaper consensus protocols (e.g. PBFT [7]). With real identities, permissioned blockchains can even use *detective* instead of *preventive* approaches for security. For example, Hyperledger Fabric [2] does not use Byzantine-fault-tolerant consensus protocols but employs the cheaper non-Byzantine-proof Kafka [8] for better performance. However, if any node that participates in consensus cheats (e.g. by sending different blocks to different peers to diverge the blockchain

states), the peers can post-detect the anomaly by comparing the blocks they received and can issue legal proceedings to the cheater. Therefore, the design choices in permissioned blockchains often lean toward performance.

In what follows, we discuss the details of how permissioned blockchains are architected and optimized under the main theme of improving the performance for the business setting. After reading this chapter you should be able to understand (i) the design choices of permissioned blockchains; (ii) the major architectures used in permissioned blockchains (Section 5.2); (iii) the cutting-edge development of these architectures (Sections 5.3–5.5); and (iv) the trend of future development (Section 5.6).

5.2 Major Architectures of Permissioned Blockchains

In this section, we introduce two major architectures of permissioned blockchains: Order–Execute and Simulate–Order–Validate. In both architectures, the peers run a consensus protocol to synchronize the updates of the blockchain state. The key difference is that Order–Execute synchronizes the *input transactions* while Simulate–Order–Validate synchronizes the *read-write-sets* of the transactions extracted by the simulation phase. This results in different features and performance trade-offs. For instance, by dedicating the simulation phase to a subset of peers, the semantic of a transaction is hidden to other peers. Thus Simulate–Order–Validate provides a certain degree of confidentiality. However, since the input transactions are much smaller than the read-write-set in size, order–execute is less demanding in network bandwidth.

Details of Order–Execute and Simulate–Order–Validate are provided in Sections 5.2.1 and 5.2.2, respectively. Afterward, we provide a thorough comparison between the two architectures in Section 5.2.3.

5.2.1 Order–Execute

The Order–Execute architecture is inherited from public blockchains and has been adopted by many permissioned blockchain systems, including Tendermint [9], Quorum [10], and BCDB [3]. As illustrated in Figure 5.1, it processes transactions in two phases. i) In the ordering phase, the peers run a consensus protocol (e.g. PBFT) to agree on the order of transactions. The transactions are grouped into a block with a hash pointer to the previous block and then the block is broadcasted to the peers. (ii) In the execution phase, the peers execute the block of transactions *deterministically* to generate identical states.

Despite its similarity with public blockchains, Order–Execute permissioned blockchains often modularize the two phases, while public blockchains cannot.

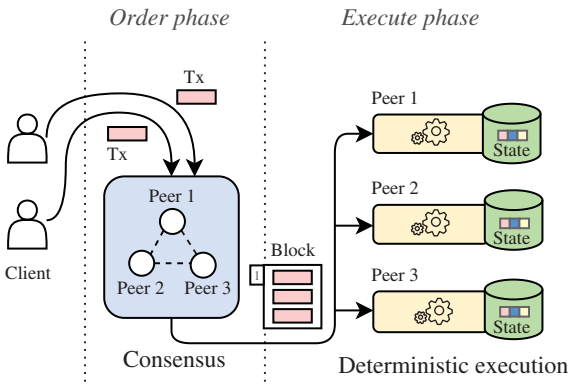


Figure 5.1 The order–execute architecture.

For instance, Ethereum peers have to execute the transaction first and include a digest of the resulting state into the block before solving the consensus (i.e. mining the block's hash puzzle). This design is necessary for public blockchains because that motivates all the peers to execute the transactions. In other words, if the execution is decoupled from the consensus, the peers may not be interested in executing the transactions and simply work on earning the mining reward offered in the ordering phase. Permissioned blockchains do not have that concern because their peers are driven by their business goals instead of the cryptocurrency mining rewards. Hence, permissioned blockchains are able to enjoy the modular order–execute architecture.

Modularity has several merits. First, different consensus algorithms and transaction processing algorithms can be plugged into the ordering module and the execution module, respectively. For instance, BCDB allows developers to choose PBFT or Kafka depending on the requirement. Second, decoupling the two enables pipelining (i.e. the ordering module can order block $i + 1$ while the execution module is working on block i), achieving higher throughput and lower latency.

However, the Order–Execution architecture has some limitations. First, to ensure all the peers result in the same state after executing a block of transactions, early permissioned blockchains of this architecture execute transactions *serially* in order to rule out any non-determinism caused by concurrent transactions. That essentially hurts the throughput and wastes the computation power of multi-core CPUs. Second, to ensure all peers reach the same next state, no non-deterministic operations (e.g. `random()`, `time()`) are allowed in the smart contract. Third, all peers have to execute all the transactions in order to keep in-sync, which limits not only the scalability of the system but also adds odds to the confidentiality (e.g. Bank-A may not want Bank-B to see its transactions with the other banks, say, Bank-C).

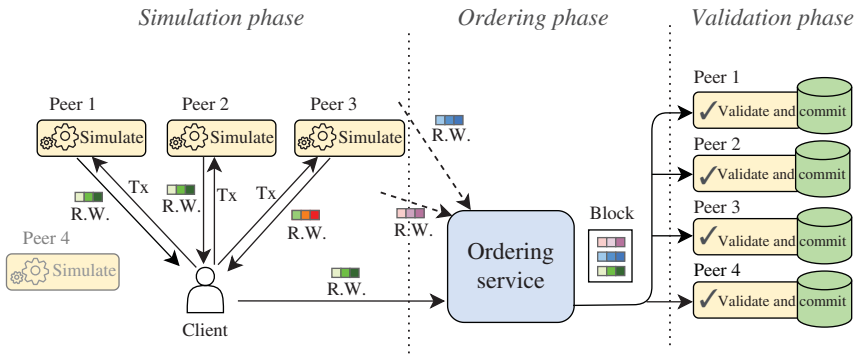


Figure 5.2 The simulate–order–validate architecture.

These limitations motivate the Simulate–Order–Validate architecture, which we are going to introduce next. However, recent research on the Order-Execution architecture is solving those limitations as well. Specifically, deterministic concurrency control (Section 5.3) allows concurrent execution while ensuring the determinism; and sharding (Section 5.6) is a general technique that enables permissioned blockchains (not specific to Order–Execute) to scale out. Nonetheless, the Simulate–Order–Validate architecture still has its merit (e.g. more confidentiality). We provide a thorough comparison of the two architectures in Section 5.2.3.

5.2.2 Simulate–Order–Validate

The simulate–order–validate architecture is advocated by Hyperledger Fabric [2], aiming for scalability and confidentiality. It processes transactions in three phases: simulation, ordering, and validation, as illustrated in Figure 5.2.

5.2.2.1 Simulation Phase

In the simulation phase, a client submits a transaction to a subset of peers called endorsers for simulation. The endorsers are determined by an *endorsement policy* defined when the smart contract is deployed (e.g. the transactions in Figure 5.2 do not require Peer-4 to endorse). Each endorser simulates the transactions against its local blockchain state. However, this process makes no actual updates but only extracts the read-write-set of the transaction, where the write-set records the updates that should be made, and the read-set is for validating serializability later in the validation phase. Afterwards, each endorser signs the discovered read-write-set and sends it back to the client. Notice that the endorsers may produce different read-write-sets even if the same transaction is given because

the endorsers may catch up with the latest blockchain state at different speeds, causing the transaction to be simulated against different states. Nonetheless, the endorsement policy is often designed to tolerate a certain number of diverged read-write-sets (e.g. sufficient when read-write-sets from two out of three endorses are identical). For instance, in Figure 5.2, although Peer-3 generates a read-write-set that diverges from the one generated by Peers-1 and 2, the block is still valid under a two-out-of-three endorsement policy. If enough signatures on the same read-write-set are collected, the client then sends the read-write-set along with the signatures to the ordering service for further processing.

5.2.2.2 Ordering Phase

In the ordering phase, the ordering service establishes a total order among the received transactions' read-write-sets and packs them into blocks (and links the blocks with hash pointers as usual). The blocks are then broadcast to all peers, including non-endorsers (e.g. Peer-4 in Figure 5.2). By default, the ordering service in Fabric is a Kafka cluster that does not tolerate Byzantine fault. However, this component is pluggable and can be replaced by PBFT for Byzantine-fault-tolerance.

5.2.2.3 Validation Phase

In the validation phase, each peer validates the transactions' read-write-sets and updates its local blockchain state by committing the valid transactions. The validation consists of two steps: (i) endorsement policy evaluation and (ii) read-write conflict check. The first step occurs in parallel for all transactions in the block. A transaction passes if the endorsement policy is satisfied. The second step is done by checking transactions sequentially. To ensure serializability, a transaction is aborted if its read-set intersects with the write-set of another transaction of the same block that precedes it based on the agreed order. The transactions that pass the validation then update the blockchain state by applying their write-sets.

5.2.3 Comparison and Analysis

The Simulate–Order–Validate architecture enables several novel features that are not supported by the Order–Execute architecture:

- **Certain degree of confidentiality**: Since the simulation phase is dedicated to a subset of peers, the other peers and the ordering service are not aware of the semantic of the transaction but only see the read-write-set during validation.
- **Tolerance to non-deterministic code**: In the Order–Execute architecture, the executor must be implemented carefully to avoid any non-determinism that would cause divergent states. In contrast, non-deterministic can cause only

a transaction abort in the Simulate–Order–Validate architecture when it fails the endorsement policy (i.e. an insufficient number of signatures on the same read-write-set).

Performance-wise, the Simulate–Order–Validate architecture allows concurrent execution and thus achieves higher throughput, while early Execute-Order permissioned blockchains have to execute transactions sequentially. However, being an optimistic approach, Simulate–Order–Validate may cause extensive transaction aborts in order to uphold serializability. This motivates two schools of research, which aim to (i) enable concurrent execution in the Order–Execute architecture and (ii) reduce the abort rate in the Simulate–Order–Validate architecture. We introduce them in Sections 5.3 and 5.4, respectively. The Simulate–Order–Validate architecture is more scalable because the transactions could be partitioned and simulated on different subsets of peers. However, its scalability is still limited because the validation phase is not partitionable. We introduce the sharding technique that scales both Order–Execute and Simulate–Order–Validate permissioned blockchains in Section 5.5. The performance of Simulate–Order–Validate is limited by the network because of two reasons. First, more communication round-trips are required because it has one more phase than Order–Execute. Second, shipping the read-write-sets requires more network bandwidth, while Order–Execute only ships the input transactions, which could be much smaller than the read-write-sets in size (e.g. imagine a transaction that updates all items). Therefore, the Order–Execute architecture is more suitable to the geo-distributed setting.

5.3 Improving Order–Execute Using Deterministic Concurrency Control

In public blockchains, transaction execution is never a performance bottleneck because the Sybil-proof BFT consensus (e.g. PoW) and the large latency of the P2P network dominate the runtime [11]. However, transaction execution turns out to be a bottleneck for Order–Execute permissioned blockchains because its permissioned environment can use cheap consensus protocols (e.g. PBFT).

Early permissioned blockchains with the Order–Execute architecture suffer limited transactional throughput because each peer has to execute transactions serially in order to reach the same final state. Originated from deterministic databases [3, 12–14], *deterministic concurrency control* offers a performance leap by allowing transactions to be executed in parallel while ensuring the same final state.

In this section, we introduce four deterministic concurrency control algorithms ranging from early initiatives to recent development. These algorithms share the

same ordering phase (i.e. giving each transaction in a block a unique transaction id `TID`). They also assume the transactions contain no non-deterministic functions (e.g. `random()`, `time()`) or fill these operations with constants before execution.

5.3.1 Calvin

Calvin is a deterministic database and it uses *deterministic two-phase-locking* to execute transactions in parallel [12]. Its key idea is to grant locks in a deterministic order so as to eliminate the non-determinism in two-phase-locking. During the execution, if two transactions intend to lock the same record, the one with a smaller `TID` gets the lock first. To achieve that, all the transactions in the block have to pre-declare their read-write-sets in order for the lock manager to determine the access order. Otherwise, when a transaction T_i is acquiring a read-lock on record x, the lock manager cannot determine whether to grant the lock immediately or wait for T_{i-1} to write x first.

Concretely, assuming all transactions' read-write-sets are known before execution, Calvin processes a block in two steps.

1. **Lock pre-acquisition**: Calvin scans all the transactions serially in the ascending order of their `TID`s. When a transaction is scanned, it requests all the locks that it will need according to its read-write-set, but no locks are granted yet. This step essentially builds up a queue on each accessed record where the waiting transactions are ordered according to their `TID`s.
2. **Lock granting and execution**: The locks are granted according to the queue order. A transaction starts execution when it is granted all its locks. After finishing execution, the transaction releases all the locks it acquired.

Prior to execution, Calvin requires to carry out a *static analysis* to extract the transactions' read-write-sets. However, real workloads often contain branches based on some executed query results. Consequently, the static analysis has to make a conservative guess to cover all the possibilities, resulting in overly large read-write-sets that hurts the parallelism because a transaction may have to wait on many unnecessary locks. An alternative is to obtain a rough read-write-set by a trial run. However, the resulting read-write-set may be different during the actual execution because of the non-determinism of the OS scheduling. In this case, Calvin has to abort the unmatched ones. Furthermore, the trial run almost doubles the latency because each transaction is effectively executed twice, and so the throughput is also degraded.

Overall, Calvin is only practical if the transactions' read-write-set can be easily inferred (e.g. the transactions in the blockchains that use the bitcoin's unspent transaction UTXO model). We next introduce another deterministic concurrency

control algorithm that partially solves the problem by requiring only the write-sets (not the read-sets) to be known a priori.

5.3.2 BOHM

BOHM introduced a deterministic concurrency control algorithm based on multi-version concurrency control (MVCC) [13]. In a multi-version database, multiple versions of the same record may coexist. Typical multi-version implementations store the versions of the same record in a linked list, and an update operation simply appends a new version to the list. Similarly, a delete operation appends a tombstone to the list. To improve space utilization, useless versions and the versions installed by the aborted transactions are cleaned by a *garbage collection* process. With multi-versioning, BOHM eliminates the requirement of knowing the transactions' read-sets a priori by only reading the proper versions at runtime. However, the write-sets of the transactions will still be required ahead.

Now assume that all transactions' write-sets are known. BOHM processes a block by following steps.

1. **Inserting placeholders**: For each record in a transaction's write-set, BOHM installs a version to that record along with the transaction's `TID`. However, the version is only a placeholder, which is going to be filled with real value during execution.
2. **Executing transactions and filling the placeholders**: In this step, transactions are executed concurrently and fill the placeholder with real values. However, T_i can only read the version tagged with the largest `TID` smaller than i. If the version is only a placeholder, it waits until the placeholder is filled.

Since the versions read by a transaction are deterministic, BOHM yields deterministic states. The resulting schedule is equivalent to a serial schedule where the transactions are executed in ascending order of their `TID`s, because if T_i's read conflicts with T_{i-1}'s write, T_i has to wait until T_{i-1} to finish writing. Nonetheless, BOHM still requires a static analysis of the transactions' write-set, which limits its applicability.

5.3.3 BCDB

Calvin and BOHM are similar in the sense that they both resolve conflicts *before* the execution by pre-scheduling the access order using locks and placeholders, respectively. Therefore, they require the read-sets and/or write-sets to be known a priori. The rationale behind theirs is that resolving conflicts dynamically during execution may cause a non-deterministic schedule on different peers.

BCDB [3] is the first approach that does not require the extraction of the transactions' read-write-sets ahead. The key idea of BCDB is to defer conflict resolution *after execution*. Consequently, it begins with a simulation where each transaction runs in its local private space. Without interference with other transactions, the simulation can be easily made deterministic by ensuring the transactions read the same initial state (called as snapshot). After the simulation, the read-write-sets of transactions are known, and thus BCDB can deterministically abort only those transactions that violate serializability and commit the rest. Concretely, BCDB processes a block of transactions by a simulation phase and a commit phase:

5.3.3.1 Simulation Phase

In this phase, the transactions are simulated concurrently against the snapshot that captures the final state of the previous block. However, no state changes are made effective but being stored in the transaction's local memory space. When reading a record, the transaction marks that it has read the record by holding a non-exclusive read lock on it. The read lock does not block the other transactions and is served only for dependency detection in the subsequent commit phase. This step is deterministic because transactions are working on a snapshot that is identical for every peer, and concurrent transactions do not interfere with each other.

5.3.3.2 Commit Phase

This phase processes the transactions sequentially based on the ascending order of their `TID`s with each transaction goes through the following steps:

1. **Applying write-set**: In this step, the transaction applies its write-set to make it visible to other transactions. However, when applying a write, if another transaction of the same block has committed and updated the same record, the current transaction has to be aborted (i.e. the *first-committer-wins* rule). Besides, the read-write-dependencies (rw-dependencies in short) are also detected. Concretely, if the record has been read-locked when applying the write entry, a rw-dependency is created and appended to both the dependency list of each reader (transaction) and writer (transaction).
2. **Detecting dangerous structures**: In this step, BCDB checks whether a transaction resides in a "dangerous structure" that may violate serializability and aborts it if so.
   Specifically, let the current transaction be T_x, and we use $T_x \xrightarrow{rw} T_j$ to denote T_x rw-depends on T_j. A dangerous structure is of the form $T_i \xrightarrow{rw} T_x \xrightarrow{rw} T_j$ with T_j be the first to commit. T_x is aborted if such a dangerous structure is found. It is proven that aborting such transactions is sufficient to ensure serializability. In Figure 5.3, T_1 is committed before T_2 and T_3. Thus $T_2 \xrightarrow{rw} T_3 \xrightarrow{rw} T_1$ forms a dangerous structure. BCDB aborts T_3 because they may be some alternate path

from T_1 to T_2 (dashed line) which may form a cycle. However, it is a conservative rule because such a cycle may not exist. BCDB makes a trade-off and aborts T_3 merely by the dangerous structure to save the cost of detecting paths from T_1 to T_2.

3. **Committing the transaction**: The transaction is committed if it is not aborted in the previous steps.

BCDB eliminates the requirement of knowing the transitions' read-write-sets a priori and thus enabling deterministic concurrency control for practical use. Nonetheless, BCDB does not completely parallelize the execution because its commit phase processes transactions serially. The commit phase could be a bottleneck because this is where the transactions invoke I/Os to materialize their writes.

5.3.4 Aria

The commit phase of BCDB is not parallelizable for two reasons: (i) its first-writer-wins rule has to rely on applying the writes sequentially so that a transaction with a transaction id can observe whether it is the first-writer; and (ii) the detection of dangerous structures requires runtime information, e.g. to confirm whether T_3 is in a dangerous structure in Figure 5.3, BCDB needs to know whether T_1 has already committed.

Aria is a recent proposal of the deterministic database that does not require the transactions' read-write-sets to be known a priori. Unlike BCDB, Aria can parallelize all its phases, achieving higher throughput. Aria parallelizes the commit phase by introducing an analysis phase to enforce the first-writer-wins rule by analyzing the intersections of the transactions' write-sets. In addition, Aria devises another dangerous structure that can be checked in parallel. Concretely, Aria processes a block by three phases: (i) simulation, (ii) analysis, and (iii) commit.

5.3.4.1 Simulation Phase

Aria's simulation phase is similar to BCDB, where transactions are simulated concurrently against the same snapshot that captures the final state of the previous

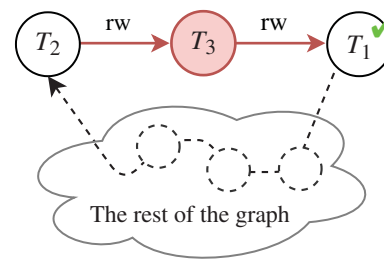


Figure 5.3 T_3 is in dangerous structure.

block. Transactions in Aria also maintain their own local write-sets. However, instead of using read-locks to mark the read-set, Aria explicitly stores each transaction's read-set, but only the primary keys are included instead of the whole record.

5.3.4.2 Analysis Phase

In this phase, Aria analyzes the read-write-sets in two steps.

1. **Write reservation**: In this step, Aria creates a *write reservation table* and each transaction concurrently makes a write reservation for every entry for its local write-set. Concretely, the write reservation table is a hash table mapping the primary key of a record to the reserver's `TID`. When making the reservation for a write, the reserver probes the write reservation table to check whether another transaction has made the reservation on the same record. If not, the reserver stores its `TID` in the corresponding hash entry. Otherwise, the reserver compares the existing `TID` with its own `TID`. If its `TID` is smaller, it overwrites the old `TID` with its own. Otherwise, no reservation is made. In the end, the write reservation table essentially stores the first-writer for each record.
2. **Serializability validation**: In this step, Aria aborts transactions by two rules similar to BCDB. However, transactions are able to check the rules in parallel.
   - **First-writer-wins**: This rule is the same as that in BCDB, but Aria checks it by probing the write reservation table. If all the entries in a transaction's local write-set have been reserved in the write reservation table (i.e. each corresponding hash-entry stores the transaction's `TID`), the transaction satisfies the first-writer-wins rule. Otherwise, it has to be aborted.
   - **The transaction does not have both write-after-read (WAR)-dependencies and read-after-write (RAW)-dependencies**: WAR-dependencies and RAW-dependencies are rw-dependencies with the reader and the writer ordered differently. Specifically, $T_i \xrightarrow{WAR} T_j$ if $T_i \xrightarrow{rw} T_j$ and $i < j$; $T_i \xrightarrow{RAW} T_j$ if $T_i \xrightarrow{rw} T_j$ and $i > j$. If a transaction contains both types of the dependencies, it is aborted. The intuition behind this rule is illustrated in Figures 5.4 and 5.5. Aria aborts none of the transactions in Figure 5.4 because no transaction has both types of dependency, and they are indeed serializable. However, if there is an edge from T_4 to T_1 as shown in Figure 5.5, both T_1 and T_4 are aborted because $T_1 \xrightarrow{rw} T_4$ is a RAW-dependency, causing T_1 and T_4 to violate the rule. Although aborting one of T_1 and T_4 is sufficient, Aria aborts both to break the cycle because T_1 and T_4 are checked independently in parallel.

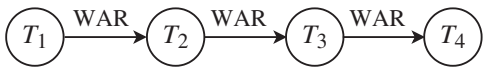


Figure 5.4 No transaction is aborted.

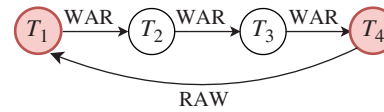


Figure 5.5 T_1 and T_4 are aborted.

Table 5.1 Four deterministic concurrency control algorithms.

Algorithm	Predetermine read-set	Predetermine write-set	Simulation phase	Analysis phase	Commit phase
Calvin	Required	Required	—	—	—
BOHM	No need	Required	—	—	—
BCDB	No need	No need	Parallel	—	Sequential
Aria	No need	No need	Parallel	Parallel	Parallel

5.3.4.3 Commit Phase

In this phase, Aria commits the transactions that are not aborted in the previous phases concurrently.

Although BCDB may abort fewer transactions then Aria (e.g. BCDB does not abort T_1 in Figure 5.5 because $T_1 \xrightarrow{WAR} T_2$ and $T_4 \xrightarrow{RAW} T_1$ are not detected when committing T_1), Aria achieves higher throughput by exploiting parallelism in every phase.

5.3.5 Comparison and Analysis

Table 5.1 compares and contrasts the four deterministic concurrency control algorithms mentioned above. Requiring to predetermine the transactions' read and/or write sets, Calvin and BOHM are more suitable for UTXO models or simple smart contracts where the read-write-sets can be easily inferred. BCDB and Aria support general smart contracts but Aria is superior in terms of performance because all its three phases can be run in parallel.

5.4 Improving Execute–Order–Validate

Similar to the order–execute architecture, the performance bottleneck of the execute–order–validate architecture is not on the consensus protocol. As mentioned in Section 5.2.2, the performance of Fabric is limited by the following two reasons:

1. **Message passing overhead**: The message passing overhead is doubled. First, more round trips are required because Fabric has one phase more

than order–execute permissioned blockchains. Second, Fabric has to synchronize the read-write-sets among peers, and their bigger size is more bandwidth-demanding than synchronizing the input transactions.

2. **Excessive aborts**: It has been reported that Fabric aborts 80% of transactions in an asset-transfer scenario because of the serialization conflicts [15]. This hurts the throughput significantly because the aborted transactions waste both computational resources and network bandwidth.

In this section, we introduce several techniques that optimize Fabric in the above two aspects. Fabric++ [15] and FabricSharp [16] reduce the number of aborted transactions by *transaction reordering*. Fabric++ additionally introduces an *early abort* technique that aborts a transaction that would eventually early, so that they waste less computational resources and network bandwidth. FastFabric [17] introduces several optimization techniques that boost the throughput significantly. For instance, its ordering service only orders `TIDs` of the transactions instead of the raw read-write-sets, which reduces the consumption of the network bandwidth. We next introduce these techniques in detail.

5.4.1 Transaction Reordering

Fabric aborts a transaction in its validation phase if a record in its read-set is updated by another transaction of the same block that is ordered before it. Therefore, it is possible to reorder the transactions before validation to avoid such conflicts. To illustrate, consider a block containing four transactions as shown in Table 5.2, and the transactions are ordered from top to bottom. We use a, b, c, and d to represent the data records with subscripts indicating the versions (e.g. T_1 reads a_1 and updates it to a_2). Since T_1 has updated a_1 to a_2, T_2, T_3, and T_4 has to be aborted because they read a stale version of a (i.e. a_1). To avoid that, we can place T_1 to the last as shown in Table 5.3, such that one transaction is saved from aborting. Interestingly, the opportunity of reordering is less obvious when the dependency among transactions is more complicated. For example, in Table 5.2, one more transaction can actually be saved by reordering them as

Table 5.2 A block of transactions where only T_1 commits.

Transaction	Read-set	Write-set	Valid?
T_1	a_1	$a_1 \rightarrow a_2$	✓
T_2	a_1, b_1, c_1	$b_1 \rightarrow b_2, c_1 \rightarrow c_2$	✗
T_3	a_1, b_1, d_1	$d_1 \rightarrow d_2$	✗
T_4	a_1, c_1, d_1	$c_1 \rightarrow c_3$	✗

Table 5.3 One way of reordering transactions in Table 5.2.

Transaction	Read-set	Write-set	Valid?
T_2	a_1, b_1, c_1	$b_1 \to b_2, c_1 \to c_2$	✓
T_3	a_1, b_1, d_1	$d_1 \to d_2$	✗
T_4	a_1, c_1, d_1	$c_1 \to c_3$	✗
T_1	a_1	$a_1 \to a_2$	✓

Table 5.4 The optimal way of reordering transactions in Table 5.2.

Transaction	Read-set	Write-set	Valid?
T_4	a_1, c_1, d_1	$c_1 \to c_3$	✓
T_3	a_1, b_1, d_1	$d_1 \to d_2$	✓
T_2	a_1, b_1, c_1	$b_1 \to b_2, c_1 \to c_2$	✗
T_1	a_1	$a_1 \to a_2$	✓

shown in Table 5.4. Fabric++ devises an algorithm for reordering by analyzing the dependency graph, which we describe by examples as follows.

1. **Building dependency graph**: In this step, Fabric++ analyzes each transaction's read-write-sets to identify rw-dependencies among them. Concretely, if T_i's read-set intersects T_j's write set, then $T_i \xrightarrow{rw} T_j$. For example, Figure 5.6 shows the dependency graph of the transactions in Table 5.2 (we omit the arrow label since only rw-dependencies are considered in Fabric++). Intuitively, if $T_i \longrightarrow T_j$, T_i should be ordered before T_j otherwise T_i will be aborted during validation. For example, Figure 5.6 indicates that T_1 should be ordered the last. However, when a cycle of the dependencies occurs (e.g. $T_2 \rightleftharpoons T_4$), it is not possible to avoid aborts by reordering. In this case, one of the transactions

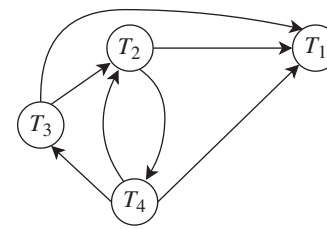


Figure 5.6 The dependency graph of transactions in Table 5.2.

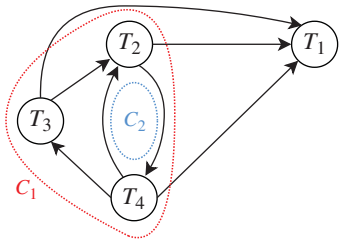


Figure 5.7 The cycles detected in Figure 5.6.

in the cycle must be aborted. Therefore, Fabric++ conducts cycle detection in the next step.

2. **Cycle detection**: In this step, Fabric++ adopts Johnson's algorithm [18] to detect cycles in the dependency graph. Figure 5.7 shows the two cycles detected in the graph of Figure 5.6. To break the cycles, one can randomly abort a transaction for each cycle (e.g. abort T_3 and T_2 for C_1 and C_2, respectively). However, aborting T_2 is sufficient because it participates in both cycles. As a heuristic, the transaction that participates in more cycles should be aborted first so that fewer transactions have to be aborted for the remaining graph. To facilitate aborting transactions by this heuristic, Fabric++ builds a *cycle covering table* as show in Table 5.5. The cycle covering table contains a row for each cycle, and if the transaction participates in the cycle, the corresponding cell is marked 1. The last row sums up the number of cycles each transaction participates in.
3. **Aborting transactions**: In this step, Fabric++ iteratively aborts transactions that participate in cycles starting from the one with the largest number of participation (abort the one with smaller `TID` to break the tie). After aborting a transaction, the cycle covering table is updated by removing the cycles that the transaction participates, and the number of participants is also updated for each transaction. The algorithm terminates until no cycles remain in the table. For example, T_2 is aborted first in Table 5.5, which removes all the cycles, and thus the algorithms terminate.
4. **Generating the order**: After breaking all the cycles, the remaining dependency graph is acyclic. Fabric++ carries out a topological sort to generate the final order.

Table 5.5 The cycle covering table.

Cycle	T_1	T_2	T_3	T_4
C_1	0	1	1	1
C_2	0	1	0	1
No. of participation	0	2	1	2

Fabric++ integrates the reordering algorithm described above into the ordering service so that the transactions that have to be aborted are not broadcasted to the peers, which reduces the communication overhead. Notice that Fabric++ only reorders transactions within the same block, which could limit the opportunities it can find for reordering. FabricSharp extends this idea to allow reordering transactions across blocks [15].

5.4.2 Early Abort

In Fabric, the validity of a transaction is checked in the last phase – the validation phase. This design wastes both computational resources and network bandwidth to process the transactions that are eventually aborted. Integrating the reordering algorithm not only reduces the abort rate but also moves the validation earlier. In this direction, Fabric++ proposes the following.

1. **Early abort in the simulation phase**: Fabric++ observed that stale reads that cause transactions to be aborted could be detected as early as in the simulation phase. For example, consider a peer simulating a transaction T_i, which reads a_1. In the meantime, an ordered block arrives, and the peer starts to validate the block while T_i is being simulated. Assume the a transaction T_j in the block updates a_1 to a_2, then T_i reads a stale value a_1. Observing the stale read during simulation allows Fabric++ to abort T_i directly.
2. **Early abort in the ordering phase**: Besides early abort in the reordering step, the ordering service can observe more stale reads that cannot be resolved by reordering. For example, consider two transactions T_i and T_j from the same block, and T_i reads a_1 but T_j reads a_2. Since T_i and T_j are in the same block, by observing T_j reading a newer version than T_j, we can conclude T_i must stale read. Therefore, Fabric++ can abort T_i in ordering phase.

5.4.3 FastFabric

FastFabric improves Fabric in terms of implementation. We introduce two major non-architectural optimizations of FastFabric here.

- **Ordering transaction header instead of the entire read-write-set**: Fabric employs Kafka to conduct transaction ordering, which specifically involves a consensus protocol (i.e. Paxos [19] in Kafka) and a replication process among Kafka nodes. However, in order to establish a total-order among transactions, ordering the `TID`s is sufficient instead of kilo-bytes of the transactions' read-write-sets. Therefore, upon receiving a transaction, FastFabric ordering service extracts its `TID` and stores the read-write-set separately. After the `TID` is ordered, the read-write-set is retrieved and broadcasted to the peers.

- **Replacing the world state database with a hash table**: Fabric uses LevelDB/CouchDB to store the world state in peers. Being full-fledged key-value stores, these databases provide durability that is, however, duplicated with the functionality of the blockchain because all the world state can be recovered by replaying the blocks. Consequently, the performance of simulation and validation is downgraded because of disk I/Os. To address this problem, FastFabric replaces LevelDB/CouchDB with a lightweight in-memory hash table, boosting the performance significantly.

5.5 Scale-Out by Sharding

Both the Order–Execute and Simulate–Order–Validate architectures are of limited scalability despite the various optimizations discussed. For order–execute, all the peers have to execute all the transactions, and thus the throughput is essentially equivalent to a single machine. Although simulate–order–validate improves it by dividing the work of simulation to different subsets of peers, its scalability is still limited because only the input transactions are sharded to different subsets of endorsers, but neither the state nor the network is sharded. Consequently, all peers have to participate in the validation phase to update the blockchain state.

To further improve the scalability, many permissioned blockchains adopt the techniques in high-performance databases to shard the state, the input transactions, and even the network [20, 21]. Fabric channels and BlockchainDB [20] are two examples.

In Fabric, if some transactions are only of interest for a subset of peers or these peers prefer to hide the transactions from other peers for confidentiality, they can create a *channel*. A channel is a subnet with different membership of peers, maintains a standalone ledger, and runs different smart contracts from the main-net. Thus the channel can be completely disjointed from the main-net, or alternatively, share the ordering service (the ordering service orders the transactions in channels and the main-net separately). However, Fabric does not support cross-channel transactions that modify the state of different channels atomically.

BlockchainDB is a scalable Byzantine-fault-tolerant key-value store built on top of permissioned blockchains (see Figure 5.8). It implements a key-value interface on the client site, which routes the operations to the blockchain storage layer. The data are partitioned across several blockchain networks maintained by different subsets of peers (some peers may participate in multiple networks). Upon receiving a request, the request is routed to the responsible shard. The transaction manager handles different consistency levels of the operation (e.g. for eventual consistency, the get operation is immediately returned regardless of pending writes). In addition, it supports verification queries that generate a certification

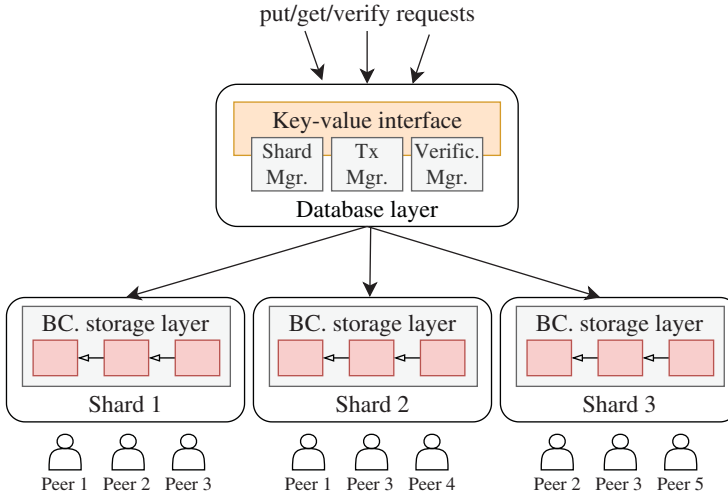


Figure 5.8 The architecture of blockchainDB.

(e.g. a Merkle proof [22]) that proves the value returned by a get request is correct. BlockchainDB is linearly scalable because it is a key-value store that has no transaction across multiple keys.

The major challenge of sharded permissioned blockchains is to support cross-shard transactions, which requires atomically commit/abort a transaction that modifies the state of multiple shards. We next introduce a *coordinator-based* cross-shard protocol that resembles the classic two-phase commit (2PC) in distributed databases [21].

In 2PC, the execution of the cross-shard transaction is facilitated by a coordinator who makes the final decision of abort/commit, ensuring very involved shards take the same action. However, in blockchain setting, the role of the coordinator is done by a *coordinator committee* to protect against Byzantine fault or for resiliency. The client initiates a cross-shard transaction by sending a `BeginTx` request (along with the transaction) to the coordinator committee, and the protocol proceeds in three steps.

1. **Prepare**: In this step, the coordinator committee runs a consensus to include the incoming transaction into its ledger and mark the transaction's state to be `start`. A `PrepareTx` message is then broadcast to every shard that is involved in the transaction. The `PrepareTx` message includes a certificate (e.g. Merkle proof) proving that the state of the transaction is `started` to ensure the message is not forged by the attacker.
2. **Pre-commit**: On receiving the `PrepareTx` message, a shard runs a consensus to include the transaction and locks the states modified by the transaction in its

shard to prevent interference of other transactions. If the locks are successfully granted, it sends a `PrepareOK` message along with a certificate proving that the transaction has been included in the block to the coordinator committee. Otherwise, a `PrepareNotOK` message is sent instead.

3. **Commit**: Upon receiving `PrepareOK` from all the involved shards, the coordinator committee runs a consensus to mark the transaction's state to be `commit` and broadcasts a `commitTx` message to every shard (along with the certificate). The shard then runs another round of consensus to modify the state accordingly and commit the transaction. If the coordinator received a `PrepareNotOK` message of timeout is triggered, it broadcasts an `abortTx` message to abort the transaction.

This protocol closely resembles 2PC, except that every state change has to be performed via consensus, and the messages must be certified for Byzantine fault tolerance. Besides, since both the coordinator and the shards are maintained by a quorum, this protocol does not need to consider the coordination failure or shard failure, and thus three-phase-commit that is used to handle failure in distributed databases is not required.

5.6 Trends of Development

While the topics and directions mentioned above are still active, we discuss two other directions in this section that are increasingly attracting attention from both the industry and the academia.

5.6.1 Trusted Hardware

With the advent of Intel SGX [23], a trusted execution environment can ensure the integrity and confidentiality of the computation. Now, many security concerns of permissioned blockchains can be offloaded to the hardware instead of relying on expensive cryptographic algorithms or complex distributed protocols. SGX provides *enclaves* to host any application code and data that require protection. Along with an *attestation mechanism*, the remote client can verify that the application code is running correctly inside the enclave. By hosting the program of distributed protocols in the enclave, the Byzantine behavior is ruled out because the attestation would fail if a malicious node tries to run another piece of code to deviate from the protocol.

As an example, Dang et al. [21] proposed to run PBFT in the enclaves so that it can tolerate up to $\frac{N}{2} - 1$ failures (although the Byzantine fault is ruled out, the nodes can still fail because of crash fault) instead of $\frac{N}{3} - 1$ in the original PBFT).

A better fault tolerance rate translates into a smaller cluster required for consensus to ensure the same level of resiliency, which results in higher throughput because of the reduced communication overhead.

5.6.2 Chainify DBMSs

Permissioned blockchains have been regarded as a radically novel data management system and thus most of them are built from scratch. In order to improve its performance, researchers have been borrowing ideas from database research, including deterministic concurrency control, transaction reordering, and early abort as mentioned in Sections 5.3 and 5.4. Recently, the community starts to switch from "databasify" the blockchain to "chainify" DBMSs because a permissioned blockchain can be regarded as a secure replicated database in disguise. Building the permissioned blockchains on top of DBMSs comes with at least three benefits:

- **Full-fledged SQL support and well-established relational model**: Current permissioned blockchains mostly build on top of key-value storage and rely on low-level procedural languages to compose transactions. The well-established relational model and SQL support come trivially by building permissioned blockchains on top of DBMSs.
- **Higher query performance**: The database community has been optimizing query performance for decades, and application developer can enjoy the performance speedup through a declarative approach. In contrast, smart contract developers still have to manage data imperatively and optimize the code performance manually.
- **Easier maintenance and integration with the local database**: Besides managing the shared data for business collaboration using permissioned blockchains, each company still has to manage its local data using traditional DBMSs. Maintaining two infrastructures with different architectures and interfaces is troublesome. However, the problem can be mitigated if both of them are DBMSs under the hood, and it becomes possible to integrate the two into one system.

ChainifyDB puts forward this vision by adding a thin blockchain layer on top of PostgreSQL [24]. The blockchain layer only contains a consensus module, a deterministic scheduling module, and signature verification mechanisms. The nodes of ChainifyDB agree on the blocks using the consensus module running PBFT, and then the scheduling module analyzes the dependencies among transactions and deterministically builds a dependency graph accordingly. The scheduling module resembles deterministic concurrency control, but it is separated from transaction processing. ChainifyDB is able to leave PostgreSQL *unmodified* and submit the

transactions to PostgreSQL according to the dependency graph using the standard interface (e.g. if two transactions have no dependency, they can be submitted to PostgreSQL to execute concurrently). Since the serializability is ensured by the dependency graph, ChainifyDB can run PostgreSQL in `Read Committed` isolation level.

Acronyms

2PC	two-phase commit
MVCC	multi-version concurrency control
P2P	peer-to-peer
PoW	proof-of-work
TPS	transactions per second
UTXO	unspent transaction output

References

1 Drescher, D. (2017). *Blockchain Basics*, vol. 276. Springer.

2 Androulaki, E., Barger, A., Bortnikov, V. et al. (2018). Hyperledger fabric: a distributed operating system for permissioned blockchains. *Proceedings of the 13th EuroSys Conference*, 1–15.

3 Nathan, S., Govindarajan, C., Saraf, A. et al. (2019). Blockchain meets database: design and implementation of a blockchain relational database. *Proceedings of the VLDB Endowment* 12 (11): 1539–1552.

4 Fan, C., Ghaemi, S., Khazaei, H., and Musilek, P. (2020). Performance evaluation of blockchain systems: a systematic survey. *IEEE Access* 8: 126927–126950.

5 Gilad, Y., Hemo, R., Micali, S. et al. (2017). Algorand: scaling Byzantine agreements for cryptocurrencies. *Proceedings of the 26th Symposium on Operating Systems Principles*, 51–68.

6 Douceur, J.R. (2002). The sybil attack. In: *Peer-to-Peer Systems. International Workshop on Peer-to-Peer Systems*, *Lecture Notes in Computer Science*, vol. 2429 (ed. P. Druschel, F. Kaashoek, and A. Rowstron), 251–260. Berlin, Heidelberg: Springer-Verlag.

7 Castro, M. and Liskov, B. (1999). Practical Byzantine fault tolerance. *OSDI*, volume 99, 173–186.

8 Kreps, J., Narkhede, N., and Rao, J. (2011). Kafka: a distributed messaging system for log processing. *Proceedings of the NetDB*, volume 11, 1–7.

9 Buchman, E. (2016). Tendermint: Byzantine fault tolerance in the age of blockchains. PhD thesis. University of Guelph.

10 Baliga, A., Subhod, I., Kamat, P., and Chatterjee, S. (2018). Performance evaluation of the quorum blockchain platform. *arXiv preprint arXiv:1809.03421*.

11 Ruan, P., Chen, G., Dinh, T.T.A. et al. (2019). Blockchains and distributed databases: a twin study. *arXiv preprint arXiv:1910.01310*.

12 Thomson, A., Diamond, T., Weng, S.-C. et al. (2012). Calvin: fast distributed transactions for partitioned database systems. *Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data*, 1–12.

13 Faleiro, J.M. and Abadi, D.J. (2015). Rethinking serializable multiversion concurrency control. *Proceedings of the VLDB Endowment* 8 (11): 1190–1201.

14 Lu, Y., Yu, X., Cao, L., and Madden, S. (2020). Aria: a fast and practical deterministic OLTP database. *Proceedings of the VLDB Endowment* 13 (12): 2047–2060.

15 Sharma, A., Schuhknecht, F.M., Agrawal, D., and Dittrich, J. (2019). Blurring the lines between blockchains and database systems: the case of hyperledger fabric. *Proceedings of the 2019 International Conference on Management of Data*, 105–122.

16 Ruan, P., Loghin, D., Ta, Q.-T. et al. (2020). A transactional perspective on execute-order-validate blockchains. *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data*, 543–557.

17 Gorenflo, C., Lee, S., Golab, L., and Keshav, S. (2019). FastFabric: scaling hyperledger fabric to 20,000 transactions per second. *2019 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*, 455–463. IEEE.

18 Johnson, D.B. (1975). Finding all the elementary circuits of a directed graph. *SIAM Journal on Computing* 4 (1): 77–84.

19 Leslie, L. (2001). Paxos made simple. *ACM Sigact News* 32 (4): 18–25.

20 El-Hindi, M., Binnig, C., Arasu, A. et al. (2019). BlockchainDB: a shared database on blockchains. *Proceedings of the VLDB Endowment* 12 (11): 1597–1609.

21 Dang, H., Dinh, T.T.A., Loghin, D. et al. (2019). Towards scaling blockchain systems via sharding. *Proceedings of the 2019 International Conference on Management of Data*, 123–140.

22 Halevi, S., Harnik, D., Pinkas, B., and Shulman-Peleg, A. (2011). Proofs of ownership in remote storage systems. *Proceedings of the 18th ACM Conference on Computer and Communications Security*, 491–500.

23 Costan, V. and Devadas, S. (2016). Intel SGX explained. *IACR Cryptology ePrint Archive* 2016 (86): 1–118.

24 Momjian, B. (2001). *PostgreSQL: Introduction and Concepts*, vol. 192. New York: Addison-Wesley.

6

Attestation Infrastructures for Automotive Cybersecurity and Vehicular Applications of Blockchains

Thomas Hardjono

MIT Connection Science & Engineering, Massachusetts Institute of Technology, Cambridge, MA, USA

6.1 Introduction

The increase in the computerization of the consumer automobile has introduced new challenges to automakers and component supply-chains, namely the need to ensure the trustworthiness the automotive system as a whole. The new features and benefits provided by the modern automotive electronic control units (ECUs) come with a number of cybersecurity-related challenges. These challenges range from unauthorized passive access to user-data stored in vehicles (e.g. user address book, history of destinations, and detailed driver behaviors), to unauthorized active access to vehicle-specific features (e.g. modification of software and firmware, and modification of vehicle odometer and related visuals data), and to active attacks aimed at taking-over (hi-jacking) vehicles in motion.

In the current work, we address the following areas related to automotive cybersecurity and discuss the role of blockchains and distributed ledger technology (DLT) [1, 2]:

- **Trustworthy attestations in automotive cybersecurity**: Attestations technology provides a means to obtain visibility into the integrity status of components within a vehicle with computerized functions. We believe that attestation is core to the value proposition of vehicular safety and security.
- **Supply-chain of component manufacturers' endorsements**: The attestations process requires not only for electronic components of known provenance [3] – such as ECUs in vehicles – to yield trustworthy evidence of their state, but it also requires supply-chain entities to issue endorsements regarding the "known good value" (reference measurements) of their components. Thus, we believe there is a need for a parallel *supply-chain of endorsements* that mirrors the supply-chain of physical components. The

Blockchains: Empowering Technologies and Industrial Applications, First Edition.
Edited by Anwer Al-Dulaimi, Octavia A. Dobre, and Chih-Lin I.

need for this parallel supply-chain of manufacturers' endorsements was first addressed more formally by the Trusted Computing Group (TCG) consortium in the mid-2000s [4–6]. This supply-chain of endorsements is also a core part of the attestation trust infrastructure needed for the future automotive industry. Efforts are underway in the semi-conductors industry to begin addressing this need [7, 8].

- **The role and impact of blockchain and DLT technology**: As vehicles increasingly become computerized and more general compute power is added (e.g. multicore CPUs) for certain applications (e.g. infotainment, passenger game playing), groups or fleets of vehicles may form an interconnected network of nodes as part of some decentralized applications. Another potential impact may be at the vehicle manufacturing and maintenance levels, where blockchains may play a key role in increasing the availability and persistence of endorsements that are relevant for the attestations of components (e.g. ECUs) within a vehicle. That is, blockchains may become a core service within the future supply-chain of component endorsements.

The potential for blockchain and DLT to assist and improve in supply-chains has been widely recognized [9–11], and examples are abound (e.g. container shipping tracking [12, 13]; addressing counterfeit medical goods [14]; fast tracking of food contaminations [15]; tracking of device activation and take-ownerships [16]; and automotive supply-chain tracking generally [17, 18]). However, in the current work we wish to focus on an emerging need related to trust and attestations of devices and components, including automotive electronic and computer components.

In Section 6.2 we briefly discuss some of the cybersecurity challenges facing the modern computerized vehicle and the need for future vehicles to employ of trusted hardware to address some of these challenges. Section 6.3 discusses trusted hardware further, focusing on some fundamental concepts of the trusted computing base (TCB). In Section 6.4 we discuss the TCG attestation framework and discuss the applicability of this framework to the computerized vehicles. The topic of vehicle wallets – needed to protect cryptographic keys and other keying material – is discussed in Section 6.5. Relevant here is the fact that the wallet itself – as part of the vehicle – must be a component covered within the attestation framework. In Section 6.6 we discuss the possible role of blockchain technology as part of trust infrastructures to support the future supply-chain of endorsements. We close the chapter with some conclusions.

6.2 Cybersecurity of Automotive and IoT Systems

Today the modern automobile is a complex system consisting of different vehicular functions implemented using *ECU*, comprising hardware, firmware, and

software. The advent of the ECUs represented a milestone for the industry as a whole because it moved the functional design of automobiles from the "hard-wired" approach to the more modular subsystems approach, namely the ECUs. This modular approach permitted vendors specializing in certain functions (e.g. electric power steering) to innovate within their domain, relying on a common network or "bus" to interconnect the different ECUs. Examples of the common bus include the Controller Area Network or CAN (ISO 11898-1:2003), FlexRay (ISO 17458), and the Automotive Ethernet (e.g. IEEE 802.3bw and IEEE 802.3bp). Currently, the typical modern car may internally utilize dozens of ECUs, which communicate with each other using one of these common bus networks.

The NIST guidelines on the cybersecurity and privacy risks of IoT technology [19] apply notably to the automotive sector due to the increased usage of transducer capabilities within the modern vehicles. The transducer technologies permit computing devices to interact directly with physical entities through the use of *sensors* and *actuators*. Together with transducer capabilities comes the ability to interact with these transducers through application-interface capabilities. Similarly, the network-interface capabilities permit these IoT devices to interact with each other. Viewed in its entirety, the introduction of IoT technology generally – and more specifically to the automotive sector – brings numerous cybersecurity and data privacy challenges that do not exist in traditional conventional IT systems.

6.2.1 Protecting Assets in Smart Cars

The EU report on the cybersecurity and resilience of smart cars [20] views the modern vehicle as being an integration of Internet of Things (IoT) components which bring added value services to both drivers and passengers. Being a collection of IoT components, communications exist (i) between these IoT components intra-vehicle and (ii) with entities and systems outside the vehicle. The report also recognizes the evolving degree of automation that is being introduced into vehicles, ranging from no automation (i.e. human driver monitoring the driving environment), to partial and conditional automation (i.e. human driver assisted by automation) to full automation (i.e. automated driving systems) [21].

The report provides a high-level architecture that views the IoT components and services as grouped according to an "assets" classification (i.e. powertrain control; chassis control; body control; infotainment control; communications control; diagnostic and maintenance systems). Each of these asset groups may employ its own ECU system and sensors that control the mechanical and/or electronic functions relevant to that asset type. The subnetwork typically relies on the CAN protocol, and several CAN buses may exist in a vehicle, interconnected by a gateway that provides a degree of isolation between critical (core) functions (e.g. powertrain management) from the less critical functions (e.g. passenger

multimedia/infotainment). For some critical functions, the use of security hardware (e.g. smart-card core, Trusted Platform Module [TPM], or Hardware Security Modules [HSM]) is advised.

Given the crucial role of many of these assets, there are several security, safety, and privacy concerns related to these assets. Some examples include the following [20]:

- Compromising powertrain or chassis ECUs and networks may result in a vehicle behaving in an unexpected way (e.g. compromise of ignition system, steering, brakes, speed and gear control, or driving support). See, for example, the case reported in [22].
- Compromising body ECUs and networks systems to the point of malfunction may increase harm to the passengers (e.g. airbag or safety belts, door force-lock used for child protection, dashboard display alerts regarding speed and collisions, and headlights disturbing surrounding vehicles).
- Compromising the internal wireless networks (e.g. Tire Pressure Monitoring Systems [TPMS]) may result in a loss of control of a vehicle.
- Compromising ECU firmware may lead to disclosure of the internal design and construction of the firmware, leading to theft of intellectual property (IP).

In general, the various types of threats to the assets of a modern vehicle include physical threats, failure or malfunction of a device or system, unintentional damages or loss of information, network outage in the vehicle, denial of service and manipulation of firmware/software, and so on [23].

Smart vehicles present a unique set of challenges that derives from the nature of the product (i.e. mass-market physical mobile vehicle), the complex Tier-1 and Tier-2 supply-chains in the automotive sector, and the increasing computerization of many of the components of the vehicle [20]:

- **Large attack surface**: Modern vehicles presents an unusually large attack surface (large number of entry points and variety of attack methods).
- **Easy access to mass-market product**: Attackers are able to gain access to a large number of samples across numerous automotive manufacturers.
- **Severe impact to the user**: Compromises may have severe or fatal impact to the driver and passengers. Manufacturers face legal liabilities for their mass-market product.
- **Persistence of threats**: Most vehicles have a long lifetime in the hands of their owners, thus persisting the threats over a long duration of time. Not every vehicle can be recalled with ease and efficiency.
- **Non-essential features as points of attack**: Many features in the modern vehicle (e.g. passenger infotainment) are not core features of the operation of the vehicle. Yet these consumer-centric features are often crucial to the value proposition of the vehicle to the consumers.

- **Cost constraints and increased computerization**: An increasing percentage of manufacturing costs today goes to electronics and software [24]. This is a radical departure from the traditional car manufacturing supply-chain model, and it is poised only to increase in the future. Increase in computerization highlights a number of business challenges for many traditional manufacturers.

6.2.2 Reported Cases

An important milestone reported by Miller and Valasek [22] and which was well publicized pertains to the exploit performed on the head-unit of a Jeep Cherokee. The researchers were able to reprogram the gateway chip in the head-unit, providing them with access and freedom to send arbitrary CAN-related messages. This attack was possible due to the fact that the gateway unit has no ability to validate the source-authenticity of the code used to reprogram it [25]. A similar attack performed on a Tesla Vehicle Model S has also been reported in [26]. Thus, the auto manufacturers in these cases did not use code-signing techniques that were already best practice in the software IT industry.

Code-signing has been standard practice in many Enterprises employing the Microsoft Windows client/server software since at least the early 2000s. Any updates to the operating system or its component software (e.g. drivers) must be digitally signed by the source of the software (e.g. operating system vendor), and the signing-certificate is usually present in the operating system. Support for the TPM trusted hardware (see below) was subsequently added to the Windows operating system [27, 28], in order to provide users and devices with a tamper-detectable hardware to manage keys. This was possible because the PC OEM vendors had already begun to incorporate the TPM v1.2 chip into their PC products starting in the mid-2000s. The availability of the TPM hardware permitted other features to be introduced over time into the PC Client computer market (e.g. secure boot).

6.2.3 Trusted Computing Base for Automotive Cybersecurity

Given the above recognition of assets, threats, and requirements, one of the fundamental questions pertains to the "boundary" of trust that can be accorded to the ECU as a unit of function and service for other ECUs and components in a given vehicle. The challenge of identifying principles which underpin the notion of "trust" was also faced by the designers of the *TPM* hardware (chip) – aimed initially at consumer PC computers – in the late 1990s who sought to embody the *TCB* concept in hardware.

In the following we re-cast some of these principles of *technical-trust* to the domain of automotive cybersecurity. For automotive components that make-up

an ECU, the challenge is identifying or defining the TCB boundary, within which some degree of guarantee can be achieved with regards to the observance of these principles.

We say that a system composed of hardware and software can be considered to exhibit *technical trust* if at least the following properties apply [29–31]:

- **Performs a well-defined function**: A component must be designed to perform a well-defined function, without any ambiguities with respect to possible states (i.e. of its state machine) in which the component may enter. The main idea here is that the function being executed by the component must not harm the component itself and must be computationally bounded so that it does not consume all available resources.
- **Operates unhindered and protected from external interference**: A well-designed component must be able to operate without interference.
  An implementation of a function defined within the TCB boundary must be able to execute until its completion without being hindered in any way (e.g. resources locked or made unavailable) or that its operations are not skewed or influenced in any fashion.
- **Cryptographic identity and truthful reporting**: The TCB instances within the system must be distinguishable from each other. Cryptographic identity ensures each TCB possesses a unique identity and can prove its identity, notably when the internal state of the TCB needs to be truthfully reported.
  The implementation of asymmetric key (public key) cryptographic functions (within a protected area in the TCB) allows the unique public-key to be used to identify the TCB. Furthermore, when the reporting functions within the TCB are used, the matching private-key can be used to digitally sign the report (attestations) regarding the TCB's internal state.
- **Trustworthy TCB dynamism**: For a TCB to be practical, it must be able to be updated, to expand, and to contract [31]. Few instances of code are 100% error free; Thus, updates must be able to be performed in the TCB itself – possibly expanding or contracting the boundary of the TCB.

6.2.4 Special Hardware for Security

Over the last three decades, special cryptographic hardware have been used in different sectors of industry – finance, government, defense – as means to provide the hardening and protection for sensitive cryptographic keys. Historically, the market for *HSM* has been enterprises and large organizations dealing with the management of its own cryptographic keys. The early designs of the HSMs often took a PC-compatible form factor, where an HSM card could be inserted into the PC computer over a standard interface (e.g. SCSI interface) in a manner similar to other pluggable hardware (e.g. network interface cards).

The main purpose of HSMs is to protect the symmetric keys and private-keys that are employed by applications. Thus, the typical HSM would afford physical protection of keys via tamper-detectable physical packaging with some degree of resistance, combined with a high-speed special processor to perform encryption and/or signing of data blocks that are passed into the HSM. Because HSMs are special purpose hardware, they have a high cost (e.g. several thousand dollars to tens of thousands of dollars), and therefore uneconomical for the consumer space.

Seeing the growing need for security features for the consumer PC market, a group of hardware manufacturers formed the Trusted Computing Platform Alliance (TCPA) [30] in the late 1990s, which was subsequently rebranded as the TCG [32]. The goal of the TCG was to develop a trusted hardware specification that permitted the hardware to be manufactured at very low cost (e.g. a few dollars). The cost of the TCG trusted hardware had to be extremely low because generally speaking the PC computer manufacturers (i.e. PC OEMs) are extremely cost-sensitive. At the same time, Smart Cards were under development and targeted primarily for the newly emerging mobile phone market. Thus the TCG trusted hardware must also be below the cost of Smart Cards.

The specifications for trusted hardware from the TCG alliance was called the *TPM*, with the hardware version 1.2 becoming available in the 2004–2005 time-frame. Wide deployment of the TPMv1.2 begun in 2006, notably with the new purchase requirements from the U.S. Army. More specifically, in February 2006 the U.S. Army Small Computer Program published a new Consolidated Buy-2 (CB2) Desktop and Notebook minimum specifications for Army customers. The Army's new specification required desktop and laptop personal computers be equipped with the new TPM (v1.2) hardware. This event represented a milestone in the adoption of trusted computing standards.

Given this wide-scale availability and adoption of the TPM v1.2 in PC computers and given the publication of the TPM v2.0 specifications in 2014, it is reasonable to consider the TPM hardware for addressing the needs of the automotive sector [33]. The various *roots of trust* (RoT) embodied within the TPM hardware provide an attractive starting point for solving the various cybersecurity issues in modern computer-laden vehicles. Some TPM hardware manufacturers – notably Infineon Technologies [34, 35] – have begun efforts to use the TPM hardware in some newer vehicles [36].

6.2.5 Truthful Reporting: The Challenge of Attestations

A key value proposition in trusted computing is the ability of a TCB unit to provide truthful information regarding its current internal computation state to authorized external entities or systems. Without the ability to provide this truthful reporting – performed under the same *technical-trust* principles stated above – there is

no visibility into the current state of the TCB, and therefore no *social-trust* or *business-trust* can be accorded to the functions within the TCB.

In the context of automotive cybersecurity, if the TCB defined as part of an ECU is not able to report unhindered the state of the TCB, then no amount of trust can be accorded to the TCB in the ECU – and by extension, to the ECU itself. This feature of truthful reporting is broadly referred to in the trusted computing literature and discourse as *attestations*.

The concept of attestations originated from the design of the TPM hardware, notably in the need for the registers (called the Platform Configuration Registers [PCR]) in the TPM to store and report its current values or entries. One of the earliest use-cases of the PCR registers in the PC computer platform was to record the cryptographic hash of the boot-code (e.g. BIOS firmware) of the PC computer. Referred to generally as *trusted boot*, the basic idea is to ensure that the firmware loaded during the boot process is the correct version as intended by its manufacturer. Because the TPM had cryptographic identity (i.e. Endorsement Key in TPM v1.2), the key could be used to sign the hash of the loaded firmware as means to accurately report to an external party (e.g. the PC computer owner).

However, this process presumes among others that the TPM hardware manufacturer has implemented the TPM design correctly in its hardware, and that each specific TPM product had a unique cryptographic identity in the form of Endorsement Key pair (i.e. EK public-key pair), where the private-key is not readable or extractable from the TPM hardware. Thus, a core part of establishing trust (technical-trust) is the signed *endorsement manifests* [5, 6] by the manufacturer regarding its implementation of the TCB functions (e.g. such as in the case of the TPM product).

In Section 6.3 we discuss the notion of the TCB, how the TPM hardware embodies the TCB, and how this maps to the automotive cybersecurity situation.

6.3 The TCB and Development of Trusted Hardware

Given the ubiquity of the TPM trusted hardware within the traditional PC computer market, we provide a short review of the notion of the TCB, its embodiment in the TPM hardware, and the efforts related to the scaled-down version ("thin" version) of the TPM addressed specifically for the automotive sector.

6.3.1 The Trusted Computing Base

The need to secure computer systems and to establish trust in systems was recognized from the early days of the commercialization of networked computers. A landmark event in trustworthy computing was the publication of the Trusted

Computer System Evaluation Criteria (TCSEC) by the U.S. Department of Defense in December 1985. The TCSEC was a significant milestone, because among others it recognized and defined the notion of the *TCB*. Broadly speaking, the TCB is the portion of the system that needs to be isolated and which can provide trustworthy behavior. Thus, the notion of the "TCB boundary" means a domain needs to be identified or defined in the system within which security can be guaranteed.

The publication of the TCSEC had positive ramifications for the nascent computer industry in that the concept of the TCB became an important building block in the design of trustworthy computing and the reasoning about system security (e.g. reasoning about protection rings [37]). The TCB portion became the focus of attention of new technical innovations in the following two decades, notably in the design of trustworthy hardware for computers. From the point of view of the operating system, the hardware was thought to be trusted because the operating system has no alternative way to verify that the hardware is behaving correctly. However, increasingly attacks were targeted at the hardware layer, and the threat of hardware vulnerability motivated the computing industry to form the TCG [32] in the late 1990s.

The TCG used the notion of a *hardware root-of-trust* as a means to distinguish the security relevant portions of a hardware platform. Two key building blocks in the definition of hardware trust in the TCG community were *shielded locations* and *protected capabilities*. Shielded locations are "…A place (memory, register, etc.) where it is safe to operate on sensitive data; data locations that can be accessed only by protected capabilities." Protected capabilities are "…the set of commands with exclusive permission to access shielded locations."

6.3.2 The Trusted Platform Module (TPM)

The TCG subsequently published the specifications for the *TPM* hardware (chip), initially version TPM v1.1b, then followed by TPM v1.2 [38]. Given the growing market in PC computers in the late 1990s and early 2000s, the TPMv1.2 was intended to support a "one-size-fits-all" approach that primarily targeted the PC market.

The TPM v1.2 supported unstructured non-volatile (NV) RAM storage and a variety of cryptographic algorithms (e.g. SHA-1, RSA, AES, and 3DES) and a zero-knowledge algorithm called Direct Anonymous Attestation (DAA) [39–41] aimed at preserving the privacy of users of the PC computers containing the TPM v1.2 hardware.

A second-generation TPM (TPM v2.0) has been published [42], and consisted of a library specification which is not intended to be backward compatible with the TPM v1.2. It expanded trusted computing features to better support vertical markets. The TCG introduced platform specific profiles that were designed to use

optional functionality specific to PCs, smart phones, and automotive platforms. Platform-specific profiles allow TPM vendors flexibility in implementing TPM features that accommodates a specific market. Additionally, TPM v2.0 supports four key hierarchies: storage, platform and endorsement as well as a hierarchy for ephemeral keys. Each hierarchy can support multiple keys and cryptographic algorithms. Password-based authorization was added and greater flexibility for policy-controlled use of the other authorization mechanisms. NV-RAM expanded to support monotonic counters, bitmaps, and "extend" operations in addition to unstructured data storage. Support for stronger cryptographic algorithms was added (e.g. SHA256 for hashing and ECC using NIST P-256 curve). The reader is directed to [43, 44] for more detailed and readable discussion regarding the TPM v2.0.

6.3.3 Resource-Constrained Automotive Systems: Thin TPMs

In considering the use of TPMs in the automotive sector, it is worthwhile to review some of the security requirements for ECUs within an automotive system [45]:

- **Firmware integrity**: Support maintaining the firmware integrity of resource-constrained ECUs.
- **Attestation evidence**: Support the storage of ECU firmware measurements (i.e. hashes), the creation of integrity digests as part of attestation evidence, and the signing of these evidence data-structures.
- **Validate digital signatures on firmware updates**: Support the verification of signatures found on firmware and software updates, and report successful (failed) installations of updates.
- **Policy enforcement**: Support the enforcement of policies regarding the updates of firmware, and policies regarding other components in the automotive system.

Although the above requirements may seem reasonable for the traditional PC client computer, they are fairly difficult to achieve in the automotive scenario employing ECUs with constrained computing resources. Thus, on one hand, the modern vehicle must be protected from various attacks. The range of passive, active, and intrusive attacks possible on the modern vehicles points to the need for trusted hardware to be employed. Among others, trusted hardware such as the TPM permits a means to provide root of trusts based on hardware-embedded cryptographic keys – which is considerably more difficult to attack. On the other hand, however, automobile manufacturers are already facing ongoing cost constraints and adding a TPM hardware for each and every ECU in the vehicle (with possibly several dozen ECUs in a typical vehicle) would increase the cost of vehicle production prohibitively.

In order to begin addressing this dilemma of "security versus cost," the TCG developed a profile of the TPM v2.0 library specifications for the automotive sector. The vision put forward by the TCG is based on the understanding that (i) the modern vehicle is in reality a composite industrial control system network with at least one Internet security gateway, and that (ii) the gateway offers the possibility to incorporate a hardware implementation of the TPM. Thus, the assumption is that the vehicle will have at least one hardware TPM that could assist the many resource-constrained ECUs. In order to define the minimal expected capabilities of the resource-constrained ECU, the TCG developed a constrained TPM specification that is referred to as the *Automotive-Thin Profile* [45, 46] – or simply as the *auto-thin* TPMs. Thus, the resource-constrained ECUs would implement the auto-thin TPMv2.0 profile, while the *Security Gateway* would possess a hardware implementation of the full or *auto-rich* TPMv2.0 (see Figure 6.1). The cost of one hardware TPM inside a Security Gateway is assumed to be affordable for the vehicle manufacturer.

Figure 6.1 provides a high-level illustration of the notion of the auto-thin and auto-rich TPMs. In Figure 6.1 the gateway is assumed to be an inter-network processor that aids in the interconnection of networks. It is used as a protocol-bridge in cases where the networks do not employ the same physical and datalink protocols for the purpose of communications. The gateway is assumed to provide Internet connectivity to external entities (e.g. to the manufacturer site for software updates over-the-air). It is important to note the severe limitations of the auto-thin TPM compared to the hardware "full" TPM. For example, the TCG automotive-thin profile of the TPM [45] does not support general purpose signature validations. This means that signatures of firmware-updates received by the gateway must be validated by the auto-rich TPM of the gateway.

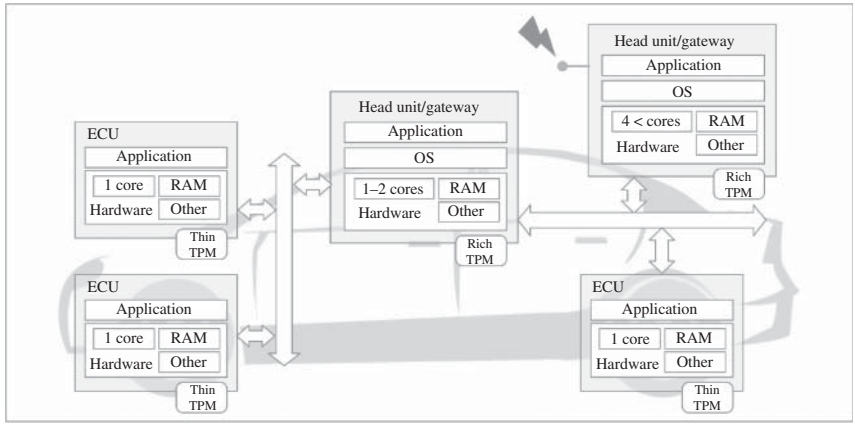


Figure 6.1 The TCG automotive-thin TPMs. Source: Adapted from [45].

The combined use of an auto-rich TPM in the gateway with the auto-thin TPM in the ECUs allows several new features of the TPMv2.0 to be utilized. One key feature is the maintenance by the auto-rich TPM of an internal monotonic counter that provides protection against replays of old firmware by an external attacker. Thus, the auto-rich TPM is able to validate that a new firmware-update is truly an update originating from the remote maintenance center. This feature is notably crucial for *software over-the-air* (SOTA) approaches to software/firmware updating – which is an attractive means for many car manufacturers to reduce the overall costs of maintenance of vehicles, as compared to in-person updates (e.g. performed at auto garages and maintenance centers).

In summary, the auto-thin TPMs [45] is motivated by automotive-specific constraints and technical requirements:

- **Limited physical resources**: Many ECUs have low availability and low speed of ROM, RAM, and NV memory.
- **Unconventional operating system**: Some ECUs have neither a boot-OS nor any conventional OS capable of dispatching distinct application processes. In some cases, an ECU may consist of a single thread of firmware that calls a minimal runtime-library.
- **Long product lifecycle**: Many automotive systems are deployed (e.g. consumer cars) for 20 or more years, far beyond the average PC computer system today.

6.3.4 Virtualized TPMs for ECUs

From an automotive deployment perspective, the TCG automotive-thin profile of the TPM v2.0 consists of a subset of the TPM v2.0 library specification relevant to the vehicle use-cases. The aim is to permit the profile to be deployable for a range of ECUs, including those with scant resources. The Security Gateways – with a more powerful ECU – would be equipped with a Rich TPM with more functionality, and would act as a relay between the other resource-constrained ECUs within the vehicle. More importantly, however, the distinction between rich and thin TPMs permits the auto-thin TPMs to be implemented as *Virtualized TPMs* [46, 47].

In the virtualized TPM approach, the TPM is in effect implemented in firmware using the hypervisor model with the assistance of a boot-loader with secure boot capabilities. In hypervisor-enabled architectures, the secure boot capabilities and hypervisor-provided isolation can establish a protected environment within which a TPM can be run. In such cases, the TPM could even serve a single guest operating system. The benefit here, among others, is that a transitive chain of trust is established, originating from the boot ROM – something core to the notion of attestations. Additionally, approaches like this permit the isolation between the firmware-TPM and the application execution space. In this way, there is a

reasonable guarantee that an application cannot interfere in the operations of the firmware-TPM, providing manufacturers of ECUs employing the firmware-TPM with a clear boundary of responsibility and of liabilities.

A discussion of virtualized TPMs is beyond the scope of the current work, and the reader is directed to [46, 47] for a discussion on the topic.

6.3.5 The DICE Model and Cyber-Resilient Systems

One promising development in the area of IoT security – which may be relevant for automotive ECUs – is the use of a simple hardware-originated storage write-protection "latches" that can protect early boot code [48]. The fundamental concept is to permit early boot code to *write-protect itself* using a storage protection latch, thereby guaranteeing a floor in a layered system below which an attacker cannot mount a successful attack. This guarantee of ongoing resiliency is notably important for low-cost IoT devices which cannot incorporate a TPM hardware or other types of crypto-processors.

Using these storage write-protection latches, IoT device manufacturers are able to provision IoT devices with keys that are accessible to early boot code [49]. In turn, the protected boot code can be used to generate subsequent additional keys that can be used by device firmware to authenticate the device and state. In other words, "layers" of code and keys (associated with the code) can subsequently be built [50], permitting a low-cost root of trust to be established without tamper-resistant shielding in the hardware. Any suspected malware can be ejected from the system by simply performing a reset and re-creating the layers again based on the latches. Because layers may be associated with unique keys, the Device Identity Composition Engine (DICE) model offers an interesting approach to layered attestations [51].

The machinery to protect this early boot code and generate a unique device identity is called the *Device Identity Composition Engine* (DICE) [52]. The engine computes a measurement (hash) of the first mutable code and then combines it with a *Unique Device Secret* (UDS) to produce a *Compound Device Identifier* (CDI). The UDS secret value is provisioned by the manufacturer during the device manufacturing. The combination or mixing of the UDS and the hash of the first mutable code is based on the concatenation of these two values (e.g. using a HMAC function). For simple IoT devices, any modifications to the first mutable code (e.g. infected by malware) will result in a different CDI, something which is detectable by external processes (e.g. organization device monitoring system/service and third-party cloud management services) [53, 54].

We believe there is strong promise for DICE and related cyber-resilient building blocks [55] to be created for automotive-specific uses cases, namely for simple ECUs that exhibit infrequent changes (updates) and must always boot to the same

state as intended by its manufacturer. Readers are directed to [48, 52] for further information regarding this approach.

6.4 Attestations in Automotive Systems

As mentioned previously, attestations by ECUs and Security Gateways in automotive systems represent a core capability that is needed to establish technical trust in a vehicle. In this section we discuss a reference framework for attestations, and how the framework applies to the use-cases related to automotive systems. A strong desire on the part of many manufacturers is the ability to perform updates via *SOTA*, where the gateway establishes a wireless connection to the remote maintenance center to perform the update. As such, we will couch the discussion of attestation in the context of the SOTA general use-case, although the same attestation processes and flows may be used when the vehicle has physical connectivity to a network (e.g. connected to Internet at a vehicle maintenance garage).

6.4.1 A Reference Framework for Attestations

The notion of device attestations is nearing two-decades now. The initial concept of attestations was related to the design principles of the TPM hardware [29, 56], where the TPM hardware needed the ability to attest itself – namely report the internal values of certain registers.

Recently, the notion of attestations has garnered interest within different technical standards organizations and industry consortiums, beyond the TCG alliance (e.g. FIDO Alliance [57], Global-Platform [58], IETF [59, 60]). The concepts around attestations – such as endorsements, validations, and freshness – are just recently coming into wider attention in the broader industry. As such, the development of a reference framework for attestations will permit related standards to be developed around that common framework. This includes the various protocols for relevant industry sectors (routers and network equipment [61, 62], mobile devices [58], and cloud stacks [63]).

The fundamental idea of attestations of a "thing" (e.g. a computing device) is that of the conveyance of truthful information regarding the (internal) state of the thing being attested to [23, 64]. In the related literature on trusted computing the term "measurement" is used to mean the act of collecting (introspecting) claims or assertions about the internal state, and delivering these claims as evidence to an external party or entity for automated review and security assessment.

However, as we know today computing environments can be structurally complex and may consist of multiple elements (e.g. memory, CPU, storage, networking, firmware, software), and computational elements can be linked and composed to

form computational pipelines, arrays, and networks. Thus, the dilemma is that not every computational element can be expected to be capable of attestation. Furthermore, attestation-capable elements may not be capable of attesting every computing element with which it interacts. The attestation capability could in fact be a computing environment itself [31]. The act of monitoring trustworthiness attributes, collecting them into an interoperable format, integrity protecting, authenticating, and conveying them requires a computing environment – one that should be separate from the one being attested.

6.4.2 Entities, Roles, and Actors

The attestation framework of [65, 66] defines of a set of *roles* that implement attestation flows. Roles are hosted by *actors*, where actors are deployment entities. Different deployment models may coalesce or separate various actor components and may call for differing attestation conveyance mechanisms. However, different deployment models do not fundamentally modify attestation roles, the responsibilities of each role, nor the information that flows between them. In the current discussion, we may use the actor and role terminology interchangeably when appropriate in order to simplify discussion (see Figure 6.2).

- **Attester**: The Attester (e.g. target device) provides attestation Evidence to a Verifier. The Attester must have an attestation identity that is used to authenticate the conveyed Evidence and establishes an attestation endpoint context. The attestation identity is often established as part of a manufacturing process that embeds identity credentials in the entity that implements an Attester.
- **Verifier**: The Verifier accepts Endorsements (from Endorsers) and Evidence (from the Attester) then conveys Attestation Results to one or more Relying

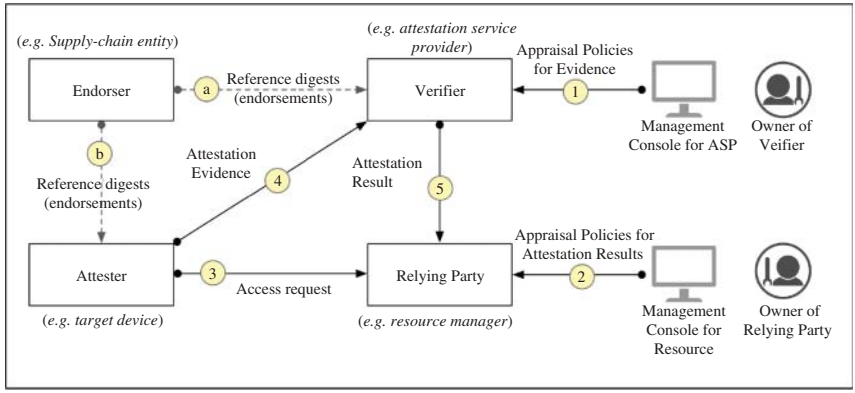


Figure 6.2 Reference architecture for attestations. Source: Adapted from [66, 67].

Parties. The Verifier must evaluate the received Endorsements and Evidences against the internal *appraisal policies* chosen or configured by the owner of the Verifier [68].

- **Relying Party**: The Relying Party (RP) role is implemented by a resource manager that accepts Attestation Results from a Verifier. The Relying Party trusts the Verifier to correctly evaluate attestation Evidence and Policies, and to produce a correct *Attestation Result*. Thus, we assume that the RP and the Verifier have a business relationship or some other basis for trusting one another. The Relying Party may further evaluate Attestation Results according to Policies it may receive from an Owner. The Relying Party may take actions based on the evaluation of the Attestation Results.
- **Endorser**: An Endorser role is typically implemented by a supply-chain entity that creates reference *Endorsements* (i.e. claims, values, or measurements that are known to be authentic). Endorsements contain assertions about the device's intrinsic trustworthiness and correctness properties. Endorsers implement manufacturing, productization, or other techniques that establish the trustworthiness properties of the Attesting Environment. This is shown as flows (a) and (b) in Figure 6.2. At an abstract level, an Endorser can be viewed as a type of *Oracle* in the sense of [69] that truthfully asserts factual information about the component manufacturing provenance.
- **Owner of Verifier**: The Verifier Owner role has policy oversight for the Verifier. It generates Appraisal Policy for Evidence and conveys the policy to the Verifier. The Verifier Owner sets policy for acceptable (or unacceptable) Evidence and Endorsements that may be supplied by Attesters and Endorsers respectively.
- **Owner of Relying Party**: The Relying Party (RP) Owner role has policy oversight for the Relying Party (RP). The RP-Owner sets appraisal policy regarding acceptable (or unacceptable) Attestation Results about an Attester that was produced by a Verifier. The RP-Owner sets appraisal policies on the Relying Party that authorizes use of Attestation Results in the context of the relevant services, management consoles, network equipment, an enforcement policies used by the Relying Party.
- **Evidence**: The Attestation Evidence is a role message containing assertions from the Attester role. Evidence should have freshness and recentness claims that help establish Evidence relevance. For example, a Verifier supplies a nonce that can be included with the Evidence supplied by the Attester. Evidence typically describes the state of the device or entity. Normally, Evidence is collected in response to a request (e.g. challenge from Verifier).
  Evidence may also describe historical device states (e.g. the state of the Attester during initial boot). It may also describe operational states that are dynamic and likely to change from one request to the next. Attestation protocols may

be helpful in providing timing context for correct evaluation of Evidence that is highly dynamic.

- **Endorsements**: Endorsement structures contain reference *Claims* that are signed by an entity performing the Endorser role (e.g. supply-chain entity or manufacturer of the target device). Endorsements are reference values that may be used by Owners to form attestation Policies.
  Some endorsements may be considered "intrinsic" in that they convey static trustworthiness properties relating to a given actor (e.g. device, environment, component, TCB, layer, RoT, or entity). These may exist as part of the design, implementation, Validation, and manufacture of that actor implementation. An Endorser (e.g. manufacturer) may assert immutable and intrinsic claims in its Endorsements, which then allows the Verifier to carry-out appraisal of the Attester (e.g. device) without requiring Attester reporting beyond simple authentication.

Figure 6.2 illustrates the canonical attestation model [66]. When an Attester (e.g. target device) seeks to perform an action at the Relying Party (e.g. access resources or services controlled by the Relying Party) the Attester must first be evaluated by the Verifier. Among its inputs, the Verifier obtains endorsements from the Endorser (e.g. device manufacturer) in flow (a) of Figure 6.2. Prior to allowing any entity to be evaluated by the Verifier, the Owner of the Verifier must first configure a number of appraisal policies into the Verifier for evaluating Evidences. The policies are use-case specific but may require other information about the Attester (or User) to be furnished to the Verifier. This is shown in Step 1 of Figure 6.2. Similarly, in Step 2 the owner of the Relying Party (e.g. resource or service) must configure a number of Appraisal Policies for Attestation Results into the Relying Party.

When the Attester requests access to the resources at the Relying Party (Step 3), it will be redirected to the Verifier (Step 4) – the understanding being that the Attester must deliver attestation Evidence to the Verifier. Included here are the endorsement(s) that the Attester obtained previously from the Endorser (flow (b) of Figure 6.2). The flow represented by Step 4 may be multi-round and may include a nonce challenge that the Attester must include in its computation of the Evidence as a means to establish freshness.

After verification and appraisal of the Attester completes, the Verifier delivers the Attestation Result to the Relying Party in Step 5. The Relying Party in its turn must evaluate the Result against its own policies (set previously in Step 2). If the Relying Party is satisfied with its evaluation of the Attestation Result regarding the Attester, it will provide the Attester with permission to complete the action it seeks to perform (e.g. access resources at the RP).

6.4.3 Variations in Evidence Collations and Deliveries

There are several possible variations in the composition and the conveyance of evidence to the verifier within the framework of [66]:

- **Composite attestations**: An Attestation Evidence yielded by an Attester may in fact consists of other Evidence (e.g. from other local components) collated by that Attester. Here, local components are assumed to have attestation capabilities that generate evidence. This Evidence is conveyed locally to a *Lead Attester* that assembles the various sets of evidences, possibly including evidence that it directly collects as well.
- **Layered attestations**: The layered attestations model is useful for devices and use-cases that preclude the use of trusted hardware or processor such as the TPM (e.g. auto-thin TPMs). The approach is based on the use of a combination of a device-secret that is set by the device manufacturer during production (e.g. fusing during manufacturing). The core idea is to use the device secret and a keyed hash function to derive other secrets (e.g. keys) to be used by the next layer in the boot-sequence. For each "layer" in the sequence, the next layer must be "inspectable" be the current layer, where an attestation evidence can be yielded [50].
- **Evidence delivery flows**: Attestation Evidence can be delivered to the Verifier in several ways, depending on the specific use-cases and type/capabilities of the device. Two general delivery flows have been identified [60]. In the first case the Attester delivers its Evidence to the Verifier as before. However, the Verifier returns the signed Attestation Result to the Attester, which then wields it to the Relying Party. In the second variation, the Attester delivers its Evidence direct to the Relying Party who then forward the Evidence to the Verifier. After appraising the Evidence, the Verifier provides the result to the Relying Party.

6.4.4 Composite Attestations for Automotive Systems

In order to place the discussion of attestations in the context of automotive systems, we consider the scenario of remote maintenance performed on a given vehicle (i.e. SOTA update scenario). We assume that the vehicle possesses a Gateway or Head-Unit (with an auto-rich TPM) that is capable of establishing a secure channel with the Remote Maintenance Center. The attester function in the gateway performs the collation of attestation evidence from the various components in the ECUs, and the gateway delivers the evidence "over the air" to the maintenance center. Figure 6.3 provides an overview of the integration of attestations within the larger software/firmware updates (over-the air) scenario, which is one of the key desirable capabilities for modern automotive systems. The main consideration of the flows in Figure 6.3 is the need for the vehicle as the attester to securely deliver the relevant attestation-evidences to the Remote Maintenance

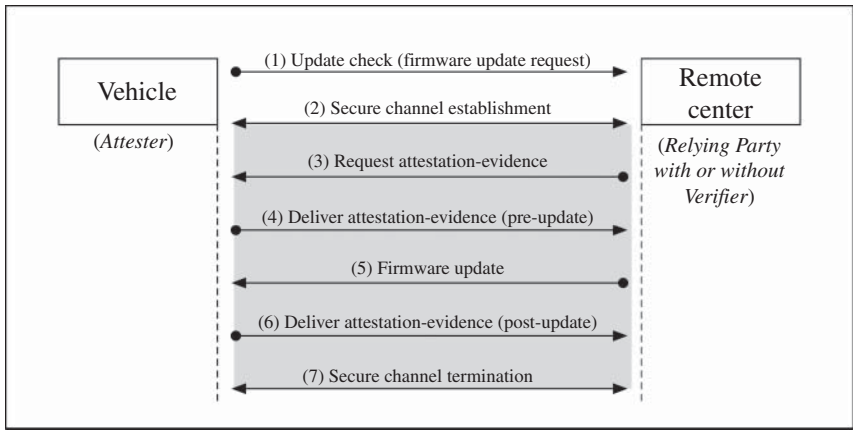


Figure 6.3 Overview of software updates over-the-air (SOTA) incorporating attestations.

Center just prior to an update event (Step 4) and then after update event has completed (Step 6). Although not shown in Figure 6.3, optionally the relevant payloads of the message flows are logged and the hash-values of the payloads recorded onto a blockchain/DLT system. Note that Steps 3 and 4 of Figure 6.3 may consist of a multi-round request/response interaction, where at each round different attestation-evidences may be requested by the Remote Maintenance Center from the gateway in the vehicle.

There are a number of interesting features of the TCG attestation framework when applied to the automotive cybersecurity situation (see Figure 6.4):

- **Local conveyance of component attestation evidence**: Components such as an ECU create attestation evidence "locally" and convey these securely to the

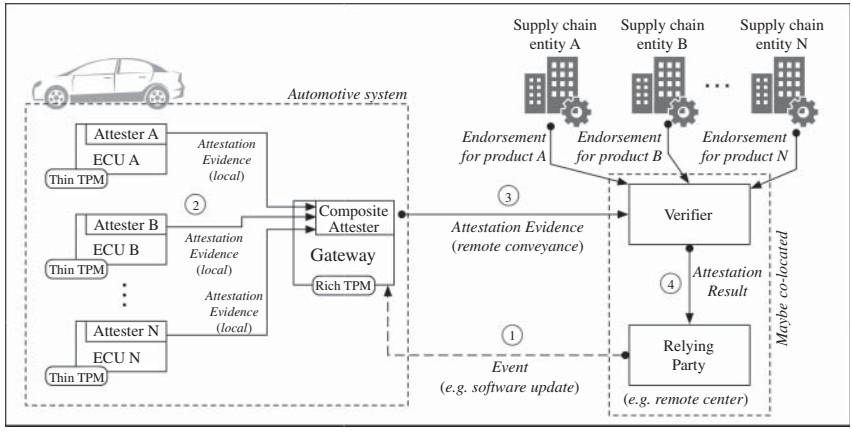


Figure 6.4 Overview of composite attestations in automotive systems.

Lead Attester (Step 2 of Figure 6.4). This approach is consistent with the fact that not all ECUs will have the capability to directly interact with the Remote Maintenance Center, and that it is one of the roles of the Gateway to be the lead attester.

- **Collation and conveyance of composite evidence by lead attester**: The role of the Gateway is also to act as the lead attester that interacts with each attestation-capable ECU and to collate evidences conveyed locally by these ECUs. The resulting attestation evidence is then delivered to the verifier (Step 3 of Figure 6.4).
- **Remote maintenance center as Relying Party**: In the SOTA scenario, the remote maintenance center acts as the relying party consistent with the TCG attestation framework [66]. It is reliant on the Verifier to perform its appraisal of the evidence obtained from the Gateway as the lead attester.
- **Possible colocation of Verifier with Relying Party**: Although the TCG attestation framework [66] distinguishes the Verifier as a separate function or service from the Relying Party, it is reasonable to assume that the remote maintenance center will also incorporate a verifier function.
  This separation of function is useful for cases where other entities may operate its own verifier to other reasons. For example, an automobile insurance company (or a network of insurance companies) may request that insured vehicles periodically (e.g. monthly) submit attestation-evidence regarding the vehicle. Alternatively, a separate attestations service can be made obtained from a third-party (e.g. Microsoft Azure attestation service [54]).
- **Verifier appraisal using supply-chain endorsements**: A core requirement for attestations is that the supply-chain entities make available the endorsements for its products.
  As discussed in [66] an endorsement can be as simple as set of cryptographic hashes of the firmware and software relevant for an ECU. This digest or hash value can be expressed or wrapped in one of many standard formats (e.g. wrapped in an X.509 attribute certificate, SAML2.0 assertion, and JSON claims data structure).

6.4.5 Appraisal Policies

As previously discussed (see previous Figure 6.2), the verification of evidences conveyed by the lead attester (i.e. the Gateway) must be guided by the appraisal policies that have been configured into the verifier system. In general, device attestations and verifications must be part of a broad and comprehensive approach to security and trustworthiness within a given deployment scenario. Organizations (e.g. remote maintenance centers) deploying attestation-capable systems must therefore incorporate attestation mechanisms and corresponding appraisal policies within their lifecycle management policies.

Due to the complexity of the trusted computing ecosystem, it is useful to distinguish between (i) the policies (rules) governing a deployment scenario (i.e. policies at a remote maintenance center) from (ii) the act of a manufacturer endorsing its product (which can be considered as a form of an internal policy on the part of the manufacturer). Endorsements typically include a logical representation of the device or component (e.g. ECU), such as in the form of a *schema* definition. This implies that a Verifier must also have the same logical representation (schema) when appraising the device and component. We refer to the Verifier's set of deployment rules that make use of this device/component logical representation as the *appraisal policies*. Note that the verifier acts as a Policy Decision Point (PDP) in the classical access control model by executing a process that evaluates (compares) its knowledge of the attester, as learned (configured) based on the endorsement from the manufacturer and the evidence conveyed by the attester.

In summary, appraisal policies for a target device or component are established by the verifier-owner (e.g. IT administrator at Remote Management Center) and should incorporate [66]:

- Logical representation of the device or component (e.g. ECU) as expressed through schemas produced by the manufacturer.
- Logical representation of the various possible run-time states of the device or component, as claimed by the manufacturer. The possible states of some components may be determined by the boot sequence of the various components that comprise the device.
- Run time evidence conveyed by the device or component. More specifically, the appraisal policies regarding the evidence conveyed by the device should understand different combinations or aspects of the device topology and at varying levels of information granularity

6.5 Vehicle Wallets for Blockchain Applications

With the emergence of blockchain technology in the past decade there has been considerable attention placed on the possible application of blockchains to the various aspects of the automotive manufacturing and deployment lifecycle. The range of applications of blockchain to the automotive sector include component counterfeit detection, components supply-chain tracking, integrity vehicle maintenance data, insurance management, vehicle energy management, vehicle-based special networks, self-driving vehicles, smart contracts for status reporting [70], and so on.

The blockchain literature employs the term of "wallets" for the technical means (software and/or hardware) used to store and manage private keys. Existing keys that are associated with ECUs should not be overloaded with the additional tasks related to blockchain functions. That is, the cybersecurity features in vehicles

that are designed for the operations and maintenance of the vehicle should not be re-purposed or overloaded for blockchain applications. If additional processing power and keys are required for new and future blockchain applications, then a separate hardware and software should be introduced to support this need.

An important aspect of introducing blockchain-support within vehicles pertains to key management and to the ownership and control of the cryptographic keys used in the various blockchain-related applications. The issue of key-control and legal ownership of keys [71] becomes important when certain blockchain-related tasks have potential impact on the safety and security of the vehicle. Entities who take-on legal liabilities in functions involving keys naturally tend to demand control over the keys. Device attestations can play a role in providing evidence of key-presence and key-status, beyond the basic proof-of-possession of the private key (e.g. by running a challenge-response protocol [72]).

6.5.1 Vehicular Application Scenarios

In considering the landscape of potential applications of blockchain and DLTs to the automotive sector, it is useful to view the vehicle from at least two perspectives in relation to the blockchain network [73]:

- **Vehicle as client to a blockchain network**: In this application scenario, the vehicle utilizes the features of a blockchain network to perform a given task. This includes employing the basic functions of the blockchain, which range from the rudimentary recording of some data onto the shared ledger (e.g. using the blockchain as a notary service), to invoking smart contracts on the blockchain that may indirectly affect other external entities (e.g. other vehicles in the network). For example, the vehicle may record some factual information about itself (e.g. GPS location information and current software stack version) to the ledger of the blockchain.
- **Vehicle as a node within blockchain network**: In this scenario, the vehicle acts as a node that participates in the blockchain network. The assumption here is that sufficient computing power is present in the vehicle to perform the relevant tasks pertaining to the node function.

Beyond the supply-chain tracking of physical components for automotive vehicles, blockchain technology may offer a solution to the problem of the persistent/continuous availability of component endorsements from various automotive components suppliers. As mentioned in Section 6.4.2, these endorsements are core to the value proposition of attestations, as illustrated in Figure 6.4. We discuss this challenge in Section 6.6.

6.5.2 Protection of Keys in Automotive Wallets

Similar to other blockchain-enabled applications, the cryptographic keys used to interact with (e.g. transact to) the shared ledger of the blockchain must be protected from unauthorized access. We use the generic term *wallet* (i.e. crypto-wallet) for the component that holds and utilizes the private-key(s) bound to the vehicle. The following are some general requirements for wallets in vehicles:

- **Support for client and node roles**: The vehicle wallet should support a wide range of blockchain-related applications, including the general cases where (i) the vehicle acts as a client to a blockchain system, and (ii) where the vehicle participates as a node in a given blockchain network.
- **Support for multiple blockchain systems**: A vehicle may interact with multiple blockchain systems simultaneously. At the vehicle plane, the manufacturers may employ a blockchain system to support the maintenance of the vehicle as a whole. At the user plane, the owner of the vehicle may interact with different blockchain networks and decentralized applications (e.g. smart contracts). For example, a blockchain-based payment scheme may be employed by the owner for the vehicle daily functions (e.g. toll-gate payments, and gas- or battery-related payments).
- **Clear separation of key owners**: The vehicle wallet must permit a clear delineation and TCB separation between the "manufacturer space" and the "user space," including the use of separate keys for different blockchain scenarios.

A core feature of some trusted hardware such as the TPM chip is the ability of certain types of cryptographic keys to be generated inside the hardware, and for internal key hierarchies to be established. Using the example of the TPM, certain types of keys can be designated as *non-migratable* at creation time, meaning that the key is bound to that single TPM and that it cannot be migrated or exported from the TPM. The use of non-migratable keys are advantageous when addressing the need to prevent the copying of keys.

A non-migratable key can be used internally to "certify" the application-level keys. This permits some degree of the provenance of an application key to be traced to its "parent" non-migratable key, and therefore to trusted hardware where the parent key is located [74]. This feature may be useful in attestation cases where the user has to prove the origins of an application-level key-pair. Typically, application-level keys can be designated to be *migratable* at creation time, allowing the key-pair to be migrated (or backed-up) to a new compatible trusted hardware using a secure key migration protocol [75]. It is important to note that non-migratable private–public key pairs can be used to uniquely identify the device [38, 76]. Mechanism to provide privacy to these keys has also been created (see [39]).

6.5.3 Types of Evidence from Wallets

Using the TPM concepts and terminology, the following is a non-exhaustive list of some of the possible wallet and key information that can be obtained using attestations:

- **Key creation provenance**: Most (if not all) current generation crypto-processor trusted hardware have the capability to create/generate a new private–public key pairs inside the shielded location of the hardware, and to maintain keys inside its long-term NV protected storage. Furthermore, evidence regarding this process can be yielded by the trusted hardware, allowing the provenance of such keys to be asserted.
- **Key-type evidence and key loss recovery**: Some crypto-processor trusted hardware (e.g. TPMv1.2 and TPM2.0) support the creation of non-migratable and migratable keys. These keys may be used to "certify" other internally generated keys. Information regarding the type of keys and the relationship among these keys (e.g. derivational and certification) may be yielded as attestation evidence.
- **Evidence of signature-origin of transactions on the blockchain**: Related to the key creation provenance and key-type, the use of a hardware-bound private-key to sign transactions permits the wallet-origin of that transaction to be ascertained. This kind of evidence may be important in scenarios needing proof that a set of confirmed transactions on the blockchain originated from the specific wallet within a given vehicle, activated with user-authorization.
- **Evidence of wallet system configurations and diversity**: Device attestations may permit visibility into the wallet-device composition and configuration. This may be useful information with regards to the *diversity* of wallet configurations, something crucial from the perspective of malware aimed at wallet systems [67, 77]. A vehicle manufacturer may choose to diversify the wallet technology used across its vehicle models (e.g. different wallet products), in order to ensure that malware attacks do not cripple entire fleets of vehicles of a specific model.
- **Wallet system health monitoring**: Wallet-device manageability services could include continuous monitoring of the *system health* of the wallets. System health monitoring and reporting has been deployed in the Enterprise networking industry for sometime now [78]. Examples include Microsoft's NAP [79], NAC from Cisco [80], and the TNC from the TCG [81, 82].

6.6 Blockchain Technology for Future Attestation Infrastructures

We believe there is a promising role for blockchain technology to support the development and operations of attestations infrastructures – such as to increase

the availability of endorsements – for manufacturers of automotive vehicles and of computing products generally. The need for manufacturer endorsements has been recognized since the early days of trusted computing [4, 5]. However, there remains a number of challenges related to the issuance and management of the endorsements. In viewing the logistics supply-chain of components and physical products, it is useful to also consider endorsements (i.e. files) as having their own logical supply-chain that parallels the physical supply-chain.

In the following we use the broad term of "endorsement objects" to mean the various files (and packages of files) that represents parts of the endorsement coming from a manufacturer. Examples of the endorsement objects include the RIM files (Reference Integrity Manifest [6]), a manufacturer SWID files (Software Identification tagging [83, 84]), manufacturer usage description (MUD) files [85], a manufacturer's signing-key X.509 certificate file, and so on.

6.6.1 Challenges in the Supply-Chain of Endorsements

We summarize some of the challenges related to the *endorsements supply-chains* as follows:

- **Standardized attestation framework**: In order to achieve accuracy in asserting endorsements and to obtain efficiency in the supply of endorsement objects (i.e. files), manufacturers, service providers are related supply-chain entities in the ecosystem need to operate on a common architecture framework for attestations. That is, they need a shared "mental model" based on a common architecture, standardized evidence-conveyance protocols, and a shared understanding of what it means to "appraise" attestation evidence. Furthermore, a common and standardized attestation framework allows manufacturers and service providers to address different market verticals and industry sectors using the same architecture based on fundamental trusted computing principles. Such a standardized attestation architecture framework is currently under development [60, 66].
- **Discoverability of endorsement-objects**: Verifiers need a clear mechanism to search and discover the manufacturer endorsements pertaining the device being appraised (see Figure 6.5). Ideally, this process of discoverability should be automated as far as possible. Solutions to this search problem may need to rely on persistent keywords and humanly readable identifiers associated with the product (i.e. unique product names, models, SKUs).
- **Accessibility of endorsement objects**: Once the Verifier obtains the location information of endorsement objects, the Verifier must then retrieve the relevant objects (i.e. files). Again, ideally this process of retrieval should be automated as far as possible. However, in seeking to retrieve endorsement objects the Verifier may face accessibility-related issues. For example, the endorsement files may sit behind protected APIs that require human intervention on the part of the Verifier Owner (e.g. user authentication and authorization).

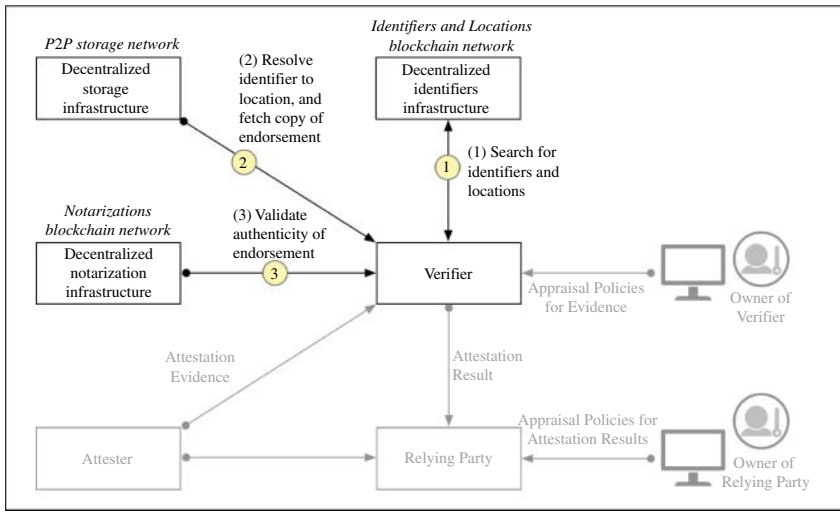


Figure 6.5 Overview of the verifier with decentralized infrastructures.

- **Persistence of endorsement objects**: Signed endorsement objects for a given product need to be available beyond the lifetime of the product and beyond the digital-death of the components manufacturers (e.g. manufacturer goes out of business). In the case of the automotive industry, many vehicles may be on the road for several years or decades. Thus, the use of a publicly readable ultra low-cost decentralized storage becomes attractive as a solution to the persistence problem.
- **Economic cost of endorsements creation**: There is a considerable cost to the component manufacturer in producing endorsements for its products. Historically, this cost has been one of the deterrents for many component manufacturers. On the other side of the equation, however, there is an economic cost to the vehicle maker and its brand should one or more components (e.g. ECU devices and networks) be compromised by cyber-attacks resulting in consumer fatalities. Thus, vehicle makers should incentivize component manufacturers upstream to produce endorsement objects that assist the vehicle maker in ensuring the continuous health of its vehicles once they are deployed on the road by consumers.
- **Confidentiality of endorsement objects**: In some circumstances, manufacturers may create "custom" version of products for certain types of bulk buyers (e.g. governments and auto rental corporations), following the customized technical specifications. In some cases, these customized products may be considered to be "private products" available only to the limited buyers due to the intellectual property (IP) contained in them. In these cases, the manufacturers

must correspondingly treat the endorsement objects as confidential and proprietary, to provide the relevant confidentiality protection.

Efforts are underway to develop industry-based solution to some of these problems [86, 87], focusing on the endorsements related to the TPM hardware.

6.6.2 Decentralized Infrastructures

Blockchain-based infrastructures may provide one or more solutions to the challenge of verifiers searching, obtaining, and validating endorsement objects. More specifically, we consider three (3) interrelated decentralized infrastructures as follows:

- **Decentralized infrastructures for notarizations of endorsement objects**: This is the decentralized infrastructure (e.g. blockchain system) that acts as a *distributed notarization service* for manufacturers to "register and notarize" endorsements objects (endorsement packages). This infrastructure could also be used by Certification Authorities (CAs) to notarize their Root-CA X.509 certificates.
  The notarization blockchain (i.e. its nodes collectively) acts as a "witness notary" that at particular point in time a key-holder entity (i.e. the manufacturer) submitted a hash/digest of a given file as a transaction into the blockchain network (and that the transaction was confirmed or settled). The blockchain is agnostic with regards to semantic meaning of the hashed file, or to the notion of endorsements.
- **Decentralized infrastructures for identifiers and locations**: This is the decentralized identifier (DID) infrastructure that retains the persistent identifier for an endorsement object and the location of a copy of the endorsement object. A manufacturer must have the ability to create a location record within the blockchain ledger. In fact, any entity that claims to possess a copy of a signed non-private endorsement object (with the signature of the manufacturer) should also be able to add location records.
  Since a blockchain is an *append-only* and timestamped system, any entity can add new location records at any time for the same endorsement object (e.g. endorsement files or package of files). Thus, for a given endorsement object there may be multiple location records captured on the blockchain. Some location records may carry inaccurate location information, outdated information, or simply contain false information. In many cases, the only available recourse for a Verifier is to fetch all available copies of an endorsement object in a trial-and-error fashion. A given Verifier could reduce the discover/fetch efforts by focusing initially on location records whose authors have an X.509 certificate issued by a well-known CA.

- **Decentralized infrastructures for storage endorsement objects**: This is the decentralized P2P storage network that holds endorsement-object files or shards of them. Decentralized file storage schemes, such as IPFS [88], may provide a solution to the persistent storage of endorsement objects beyond the manufacturer's business lifetime.

6.6.3 Example of Verifier Tasks

As discussed in Section 6.4 and shown in Figure 6.2, the role of the Verifier is to appraise the evidence conveyed by the device (as attester), against the endorsements issued by the manufacturers of the components making-up the device. In the context of decentralized infrastructures, the workflow for the Verifier is roughly grouped into three main tasks (see Figure 6.5, which extends the previous Figure 6.2):

Step 1: Search for identifiers of a product whose endorsement is known to exist (or known to have existed in the past), and search for location information (e.g. URLs) for the corresponding endorsement objects. The product brand/model and the manufacturer's name should be used as identifier strings within the endorsement objects (e.g. file headers, SWID tags) so that general searches can be made on the objects/files.

Step 2: Fetch a copy of the product's endorsement objects using the location information found in Step 1 (or fetch multiple copies from multiple locations).

Step 3: Validate the endorsement objects, which consists of the following steps:

- **Step 3.1:** *Validation of the source-authenticity* (signatures) of an endorsement object. This includes validating that the X.509 certificate was valid at the time when the endorsement object was signed.
- **Step 3.2:** *Validation of the manufacturer's claim* that it was the source of the endorsement. This implies validating that the Subject stated within the manufacturer's X.509 certificate is indeed the manufacturer as a legal business entity, incorporated/registered within a given jurisdiction. The use of Extended Validation (EV) certificates [89] that includes business information (e.g. incorporation number and LEI number [90]) may provide sufficient business legal information.
- **Step 3.3:** *Validation of the chain of certificates* used to issue the manufacturer's signing-certificate. This implies validating all certificates in the chain, up to the Root-CA certificate. This requirement may become complex if the issuing CA is also defunct (out of business).

The approaches proposed for archiving endorsement objects could also be employed by CAs to publicly archive expired Root-CA certificates.

6.6.4 Notarization Records and Location Records

Since these decentralized infrastructures have different functional goals or purposes, different types of information (relating to endorsement objects) can be recorded on each of these blockchains, respectively. We propose a number of new data structures corresponding to the need to support endorsement objects via blockchains and DLT systems. These data structures are the endorsement *notarization record*, the endorsement *location record*, and the endorsement *certification record*. This is summarized in Figure 6.6.

- **Notarization record**: The endorsement notarization record is used by a manufacturer to register onto a notarizations blockchain the *existence* of an endorsement object corresponding to a product from the manufacturer. In short, the manufacturer creates a self-addressed transaction on notarizations blockchain carrying a hash of the root endorsement manifest file [6]. The transaction is signed by the manufacturer, signifying the act of registration onto the blockchain. Once the transaction has been confirmed (settled) by the blockchain – in which all nodes retain a copy of the blocks in their local ledger – it will be computationally difficult (infeasible) for anyone to subsequently alter (replace) the transaction.
- **Location record**: The location record is used by a manufacturer or other third parties to capture the known locations (e.g. URI/URL) on the Internet of valid duplicate copies of an endorsement object. The first instance of the

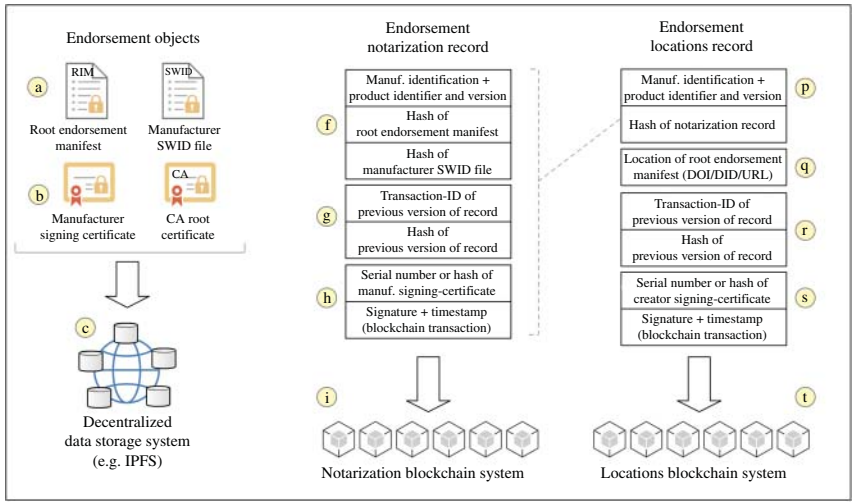


Figure 6.6 Illustration of the logical separation between the (i) endorsement notarization blockchain from (ii) the locations blockchain.

location record should be created by the manufacturer because initially only the manufacturer holds the endorsement object (e.g. the location is the manufacturer's local repository). However, as other copies of the endorsement object are disseminated on the Internet (e.g. customers make copies in other repositories), other entities may add new location records for the same endorsement object.
For the Verifier, in order to find the location of a given endorsement object, the Verifier needs to traverse the blocks of the ledger starting from the most recent, in order to find the appropriate location record containing the relevant product description strings. If the Verifier discovers a location record that was created (signed) by entities other than the manufacturer, the Verifier must decide whether to accept (believe) the information in the location record or to continue its search on the ledger. In the simplest case, the Verifier can simply follow the link found in the location record, fetch the endorsement object, and verify the manufacturer's signature on the object.

Although out-of-scope for the current work, a similar notarization facility could be used by CAs whereby they archive a copy of their public-key (of their Root CA certificate) onto the blockchain. The transaction should be signed by the same key-pair. Ideally, non-digital methods should also be used to archive Root CA certificates (e.g. publishing in a major newspaper, printed on real paper [91]).

6.6.5 Desirable Properties of Blockchain-Based Approaches

There are several desirable properties of the endorsement storage and the archival system employing blockchain technology:

- **Functional separation of notarization from locations management**: The existence information of a signed endorsement object must not be confused or mixed with the location information regarding duplicate copies of that endorsement object.
  Thus, in "registering" a notarization record, the goal of the manufacturer is to create a persistent timestamped archive of information about the endorsement of its product. There must be a 1-to-1 correspondence between an endorsement object and its notarization record. Only a manufacturer's notarization record should be believed. Any entity should be able to create a new location record for a given signed endorsement object corresponding to a pre-existing notarization-record. The caller (verifier) must validate the X.509 certificate of the creator of any location record.
- **Support for multiple versions of a product**: For a given product, the manufacturer may create updates (upgrades) or patches of firmware and software for the product. Typically, it is industry best practice to increment the version major–minor number and build number of the product.

Because the hash of a binary file *before* the addition of the patch will be different from the hash of a binary *after* the patch, the manufacturer must also create a new notarization record for each new version of the product. A new notarization record should carry an indicator that it is an update of an existing version of the product, and include a pointer to (hash of) the previous notarization record (of the version it updated).

- **Support for multiple location-resolution mechanisms**: In order to ensure broad availability of endorsement objects, the self-sustaining archival ecosystem must permit for multiple location-resolution mechanisms to be supported, according to the choice made by the creator of the locations-record. That is, if an entity wishes to store copies of endorsement records and make these available publicly, that entity should have the freedom in selecting one or more location-resolution protocols, and express this choice in the locations-record on the blockchain.
  Examples of location-resolution protocols include the Digital Object Identifier (DOI) ISO standard and its Handle Resolution system [92–94], the W3C DID scheme [95] for blockchains, or traditional URL/URI [96] locations based on the DNS resolution system.
- **Independence of storage mechanisms of endorsement objects**: Entities who wish to store copies of signed endorsement objects – and create location-records on the blockchain a means to point to these storage locations – must have the freedom to select the appropriate storage mechanism for these copies.

6.6.6 Information within the Notarization Record

The manufacturer endorsement notarization record is composed of the following groups of related information (see Figure 6.6):

- **Manufacturer identifier and product identifier**: This set of fields captures the manufacturer identification information (e.g. business incorporation number or LEI number [90]) and the product information (e.g. model/version). This is shown as item (f) Figure 6.6. The set of information should (must) match the manufacturer/product information in the root endorsement manifest (item (a) Figure 6.6). A hash of the root endorsement manifest must also be included to ensure a correct 1-to-1 matching between the notarization record and the root endorsement manifest file shown in (a).
- **Pointers to previous version of notarization record**: For a new version of a product, a manufacturer must create a new notarization-record on the blockchain. This is shown as item (g) Figure 6.6. For a new version of product which already has an old notarization-record, the new notarization-record must include a "pointer" to the old notarization-record. This pointer consists of

the transaction-ID of the confirmed block of the old notarization record. This is accompanied by a hash of the old notarization record. For a new product without previous versions, this field is empty (null).

- **Digital signature and timestamp**: The manufacturer must sign the notarization-record – using its blockchain public-private key pair – before transmitting it to the blockchain. This shown as item (h) in Figure 6.6.

6.6.7 Information in the Location Record

The following is a minimal list of items that should be present in the location record:

- **Manufacturer identifier and product identifier**: This information (item (p)) must be identical to the information found on the notarization-record in item (f) in Figure 6.6. It must refer to the same product, version, etc.
- **Hash of notarization-record**: This is the hash of the notarization-record that declares the endorsement objects. This allows for 1-to-Many matching between the notarization-record (of the product) with the stated location of the endorsement objects of the product.
- **Identifier and file location-resolution protocol**: The item (q) in Figure 6.6 details the identifier type, identifier namespace, and resolution service to be employed. Thus, for example, if the creator of the location-record employs a DOI and the Handle resolution service [92–94], then this field must contain at least (i) the DOI string and (ii) the end-point address of the resolution service to which the DOI string can be inputted. Alternatively, if the creator of the location-record is employing the W3C DID scheme [95], then this field must instead contain at least (i) the namespace of the DID owner, and (ii) the endpoint URL/URI for the location of the endorsement object. Note that the hash of the endorsement file (item (f) in Figure 6.6) can also be used to resolve locations in decentralized file systems, such as IPFS [88]. A correct resolution must bring the querier to the location of the endorsement object (item (c) in Figure 6.6).
- **Pointers to the previous version of the location record**: Item (r) in Figure 6.6 pertains to updates of new location-information (for the same endorsement object) as stated by the same signatory (i.e. same entity creating the previous location-record). An entity that holds a copy of an endorsement object may move the files to a new location endpoint. That entity may use a new location-record (containing the same hash of the notarization-record) to indicate this move. This may help callers/verifiers to search and retrieve endorsement objects faster.
- **Digital signature and timestamp**: The creator of the location-record must sign the record before transmitting it to the blockchain. This is shown as item (s) in Figure 6.6.

6.6.8 The Compliance Certifications Record

Similar to notarization records pertaining to endorsement for a product, a *compliance certification* entity may also wish to store or archive certification documentations pertaining to the product. This is because certification organizations may also go out of business.

This entails the certification entity creating a *certifications record* on the blockchain. The certifications record contains similar information fields to the notarization record and must carry a hash of the notarization-record of the product in question. This topic of product certification is beyond the scope of the current work.

6.7 Areas for Innovation and Future Research

The area of blockchains and DLT is still nascent, and thus more research and innovations are needed in the broader context of automotive and IoT applications:

- **Decentralized coordination of fleets of vehicles**: Shared ledger systems, such as those embodied within current blockchain and DLT networks, may provide the basis for future designs and architectures to address challenges around the decentralized coordination of fleets of vehicles [97]. Early efforts around a goal-driven coordination of a swarm of robots based on a shared ledger has been reported in [98].
  In certain scenarios, such as future smart cities, fully autonomous vehicles (e.g. autonomous trash collectors) may physically meet and even interact with passenger vehicles. Shared ledger systems and blockchains, combined with mesh network technologies, may permit these vehicles to increase the efficiency of processes in achieving their stated tasks (e.g. on part of a planned route, the passenger vehicle intercepts a trash-collector vehicle/robot and piggybacks compacted-trash box to the next trash-collector on its route).
- **Take-ownership of vehicles and registering modifications**: Blockchains and DLTs may be used to record take-ownership proof of a vehicle by its new owner, thereby providing clear boundary of legal transfer of the vehicle from the manufacturer/dealer to the owner [16]. This is notably important when modifications (e.g. to firmware of ECU) are subsequently done by the owner, and thereby causing ambiguity as to the legal obligations of the manufacturer in case of vehicle malfunctions caused by those modifications. Innovations are needed to address the various changes of ownerships of a vehicle throughout its lifetime in a transparent and privacy-preserving manner. Blockchain-based ownership proofs should be closely tied to the blockchains employed to record the endorsements of the components of the vehicle.

- **Group secure computation among autonomous vehicles**: Group-oriented cryptography and secure computation has been a subject of research for over two decades now [99, 100]. Groups of communicating vehicles when equipped with the proper computing power may be the basis for group-oriented secure computations for addressing specific use-cases (e.g. military battlefield). In some cases the availability of a shared ledger may assist group-oriented computations in reaching completion faster with higher efficiency. The role of blockchains and DLTs generally in assisting group-oriented secure computations needs further attention and research. Coupled with the emergence of *confidential computing* based on secure enclaves and trusted execution environments [101, 102], the combination presents interesting and promising opportunities.
- **Trust infrastructures for smart cars and smart cities**: Implementing smart cars that are environment-aware requires viewing the challenges related to smart cars in the broader context of smart cities for better societies. Future road networks and physical infrastructures in smart cities may employ numerous IoT devices (e.g. sensors) that may interact with vehicles (e.g. see [103]).
  The potential for a proliferation of IoT devices in physical infrastructures that interact with those in smart cars points to the need for attestations verification services – which in turn should be part of the design of smart cities. The overall goal should be a coherent trust infrastructure that ensures cyber-resilience across these domains, with the primary goal of ensuring human safety, security, and privacy.

6.8 Conclusion

As automotive vehicles increasingly become computerized, various cybersecurity challenges will increasingly emerge – similar to those faced by other industries reliant on computer technology. In the current work we have discussed a number of key aspects of automotive cybersecurity and cyber-resilience, and placed blockchain technology in the context of potentially supporting the solutions to these challenges.

The ability for the computerized components of a vehicle to provide attestations regarding its internal state represents a core building block for the solutions. Hand in hand with attestations capabilities is the role for trusted hardware in providing not only protection for cryptographic keys but also roots of trust for many functions. However, this requires supply-chain entities to issue endorsements of their components – a task whose value is only beginning to be appreciated recently. Finally, blockchains may also play a key role in increasing the availability and persistence of endorsements, something that is important given that many vehicles are “on the road” for many years and decades.

Acknowledgments

We thank Prof. Sandy Pentland and Stephen Buckley (MIT) for support for the current work. Special thanks to Ned Smith at Intel and Steve Hanna at Infineon for various discussions and inputs. We also thank the various members of the TCG, notably those working in the Embedded Systems and Automotive areas.

References

1 Haber, S. and Stornetta, W. (1991). How to time-stamp a digital document. *Advances in Cryptology – CRYPTO'90 (LNCS 537)*, 437–455.

2 Bayer, D., Haber, S., and Stornetta, W. (1993). Improving the efficiency and reliability of digital time-stamping. In: *Sequences II: Methods in Communication, Security and Computer Science* (ed. R. Capocelli, A. De Santis, and U. Vaccaro), 329–334. Springer-Verlag.

3 Robertson, J. and Riley, M. (2018). The Big Hack: How China Used a Tiny Chip to Infiltrate U.S. Companies. *Bloomberg Businessweek*. https://www.bloomberg.com/news/features/2018-10-04/the-big-hack-how-china-used-a-tiny-chip-to-infiltrate-america-s-top-companies (accessed 18 May 2023).

4 Hardjono, T. and Smith, N. (ed.) (2005). *TCG Infrastructure Reference Architecture for Interoperability (Part 1) – Specification Version 1.0 Rev 1.0*. Trusted Computing Group, TCG Published Specification. https://trustedcomputinggroup.org/wp-content/uploads/IWG_Architecture_v1_0_r1.pdf (accessed 18 May 2023).

5 Hardjono, T. and Smith, N. (ed.) (2006). *TCG Infrastructure Working Group architecture (Part 2) – Integrity Management – Specification Version 1.0 Rev 1.0*. Trusted Computing Group, TCG Published Specification. http://www.trustedcomputinggroup.org/resources (accessed 18 May 2023).

6 TCG (2020). *TCG Reference Integrity Manifests (RIM) Information Model Version 1.00, Rev. 0.16*. Trusted Computing Group, TCG Specifications. https://trustedcomputinggroup.org/resource/tcg-reference-integrity-manifest-rim-information-model/ (accessed 18 May 2023).

7 Dodson, T. (2017). Intel Transparent Supply Chain Process. NIST, Winter 2017 Software and Supply Chain Assurance Forum. https://csrc.nist.gov/CSRC/media/Projects/Supply-Chain-Risk-Management/documents/ssca/2017-winter/TuePM1_3_%20Intel.pdf (accessed 18 May 2023).

8 Dodson, T. and Cabre, E. (2019). Blockchain Augmentation of the Trusted Supply Chain. Intel Corporation, RSA2019 Conference. https://published-prd.lanyonevents.com/published/rsaus19/sessionsFiles/13424/PDAC-F02-

Blockchain-Augmentation-of-the-Trusted-Supply-Chain.pdf (accessed 18 May 2023).

9 Hardjono, T. (ed.) (2018). The Impact of Blockchain for Government: Insights on Identity, Payments, and Supply Chains – Report from the Congressional Blockchain Caucus. http://www.businessofgovernment.org/report/impact-blockchain-government-insights-identity-payments-and-supply-chain (accessed 18 May 2023).

10 Choudary, S.P., Van Alstyne, M.W., and Parker, G.G. (2019). Platforms and blockchain will transform logistics. *Harvard Business Review* (2–6 June).

11 Gaur, V. and Gaiha, A. (2020). Building a transparent supply chain. *Harvard Business Review*, 94–103.

12 Miller, R. (2018). IBM teams with Maersk on new blockchain shipping solution. *Tech Crunch*. https://techcrunch.com/2018/08/09/ibm-teams-with-maersk-on-new-blockchain-shipping-solution/ (accessed 18 May 2023).

13 Miller, R. (2019). IBM-Maersk blockchain shipping consortium expands to include other major shipping companies. *Tech Crunch*. https://techcrunch.com/2019/05/28/ibm-maersk-blockchain-shipping-consortium-expands-to-include-other-major-shipping-companies/ (accessed 18 May 2023).

14 Morris, N. (2020). 12 global pharmaceutical firms join EU blockchain consortium PharmaLedger. *Ledger Insights*. https://www.ledgerinsights.com/pharmaledger-pharmaceutical-blockchain-eu/ (accessed 18 May 2023).

15 Hackett, R. (2017). Walmart and 9 food giants team up on IBM blockchain plans. *Fortune*. https://fortune.com/2017/08/22/walmart-blockchain-ibm-food-nestle-unilever-tyson-dole/ (accessed 18 May 2023).

16 Hardjono, T. and Smith, N. (2016). Cloud-based commissioning of constrained devices using permissioned blockchains. *Proceedings of the 2nd ACM International Workshop on IoT Privacy, Trust, and Security (IoTPTS 2016)*, 29–36. New York, USA: ACM. https://doi.org/10.1145/2899007.2899012.

17 Fraga-Lamas, P. and Fernández-Caramés, T.M. (2019). A review on blockchain technologies for an advanced and cyber-resilient automotive industry. *IEEE Access* 7: 17578–17598. https://doi.org/10.1109/ACCESS.2019.2895302.

18 Reimers, T., Leber, F., and Lechner, U. (2019). Integration of blockchain and internet of things in a car supply chain. *2019 IEEE International Conference on Decentralized Applications and Infrastructures (DAPPCON)*, 146–151. New York: IEEE. https://doi.org/10.1109/DAPPCON.2019.00028.

19 Boeckl, K., Fagan, M., Fisher, W. et al. (2019). Considerations for managing Internet of Things (IoT) cybersecurity and privacy risks. National Institute of Standards and Technology, NISTR 8228. https://doi.org/10.6028/NIST.IR.8228.

20 ENISA (2016). Cyber Security and Resilience of Smart Cars. European Union Agency for Network and Information Security (ENISA), Good Practices and

Recommendations. https://www.enisa.europa.eu/publications/cyber-security-and-resilience-of-smart-cars (accessed 18 May 2023).

21 SAE (2016). *Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles*. SAE International, Technical Standards J3016.

22 Miller, C. and Valasek, C. (2015). Remote exploitation of an unaltered passenger vehicle – BlackHat 2015 Conference.

23 Regenscheid, A. (2018). Platform Firmware Resiliency Guidelines. National Institute of Standards and Technology, NIST Publication SP 800-193. https://csrc.nist.gov/publications/detail/sp/800-193/final (accessed 18 May 2023).

24 Schmittner, C. and Romanovsky, A. (2019). Automotive cybersecurity standards – relation and overview. In: *Proceedings of 8th International Workshop on Next Generation of System Assurance Approaches for Safety-Critical Systems (SASSUR 2019) LNCS11699* (ed. A. Romanovsky, E. Troubitsyna, I. Gashi et al.), 153–165. Singapore: Springer International Publishing. https://doi.org/10.1007/978-3-030-26250-1_12.

25 Miller, C. (2019). Lessons learned from hacking a car. *IEEE Design & Test* 36 (6): 7–9. https://doi.org/10.1109/MDAT.2018.2863106.

26 Nie, S., Liu, L., and Du, Y. (2017). Free-Fall: hacking Tesla from wireless to CAN bus – BlackHat USA 2017 Conference. https://www.blackhat.com/docs/us-17/thursday/us-17-Nie-Free-Fall-Hacking-Tesla-From-Wireless-To-CAN-Bus-wp.pdf (accessed 18 May 2023).

27 Microsoft Corp. Trusted platform module and bitlocker drive encryption. https://msdn.microsoft.com/en-us/library/windows/hardware/dn653315 (accessed 18 May 2023).

28 Microsoft Corp (2018). BitLocker high-level overview. https://docs.microsoft.com/en-us/windows/security/information-protection/bitlocker/bitlocker-overview (accessed 18 May 2023).

29 Trusted Computing Group (2003). *TPM Main – Part 1 Design Principles – Specification Version 1.2*. Trusted Computing Group, TCG Published Specification. http://www.trustedcomputinggroup.org/resources/tpm_main_specification (accessed 18 May 2023).

30 Balacheff, B., Chen, L., Pearson, S. et al. (2002). *Trusted Computing Platforms: TCPA Technology in Context*. New York: Prentice Hall.

31 Hardjono, T. and Smith, N. (2019). Decentralized trusted computing base for blockchain infrastructure security. *Frontiers Journal – Special Issue on Finance, Money & Blockchains* 2. https://doi.org/10.3389/fbloc.2019.00024.

32 TCG. Trusted Computing Group. http://www.trustedcomputinggroup.org (accessed 18 May 2023).

33 Petri, R., Springer, M., Zelle, D. et al. (2016). Evaluation of lightweight TPMs for automotive software updates over the air. *Proceedings 4th Embedded Security in Cars (ESCAR) USA*.

34 Infineon Technologies AG (2018). World's first TPM for cybersecurity in the connected car. https://www.infineon.com/cms/en/about-infineon/press/press-releases/2018/INFDSS201810-004.html (accessed 18 May 2023).

35 Infineon Technologies AG (2019). OPTIGA TPM SLI 9670 trusted platform module (TPM) for special use in automotive. https://www.infineon.com/cms/en/product/security-smart-card-solutions/optiga-embedded-security-solutions/optiga-tpm/sli-9670/ (accessed 18 May 2023).

36 Automotive World (2019). A safe for sensitive data in the car: Volkswagen relies on TPM from Infineon: Volkswagen is one of the first car makers to deploy the OPTIGA Trusted Platform Module (TPM) 2.0 from Infineon Technologies AG as a security solution for the connected car. *Automotive World*. https://www.automotiveworld.com/news-releases/a-safe-for-sensitive-data-in-the-car-volkswagen-relies-on-tpm-from-infineon/ (accessed 18 May 2023).

37 Saltzer, J.H. (1974). Protection and the control of information sharing in MULTICS. *Communications of the ACM* 17 (7): 388–402.

38 Trusted Computing Group (2003). *TPM Main – Specification Version 1.2*. Trusted Computing Group, TCG Published Specification. http://www.trustedcomputinggroup.org/resources/tpm_main_specification (accessed 18 May 2023).

39 Brickell, E., Camenisch, J., and Chen, L. (2004). Direct anonymous attestation. *Proceedings of the 11th ACM Conference on Computer and Communications Security CCS2004*, 132–145. ACM.

40 Brickell, E. and Li, J. (2012). Enhanced Privacy ID: a direct anonymous attestation scheme with enhanced revocation capabilities. *IEEE Transactions on Dependable and Secure Computing* 9 (3): 345–360.

41 Camenisch, J., Chen, L., Drijvers, M. et al. (2017). One TPM to bind them all: fixing TPM 2.0 for provably secure anonymous attestation. *Proceedings of the 2017 IEEE Symposium on Security and Privacy*, May 2017, 901–920. IEEE. https://doi.org/10.1109/SP.2017.22.

42 Trusted Computing Group (2014). Trusted Platform Module Library Part 1: Architecture – Specification Family 2.0. Trusted Computing Group, TCG Published Specification. https://trustedcomputinggroup.org/wp-content/uploads/TPM-Rev-2.0-Part-1-Architecture-01.07-2014-03-13.pdf (accessed 18 May 2023).

43 Proudler, G., Chen, L., and Dalton, C. (2014). *Trusted Computing Platforms: TPM2.0 in Context*. New York: Springer.

44 Arthur, W. and Challener, D. (2015). *A Practical Guide to TPM2.0 – Using the Trusted Platform Module in the New Age of Security*. New York: Apress Media.

45 Trusted Computing Group (2018). *TCG TPM 2.0 Automotive Thin Profile – For TPM Family 2.0 (Specification Version 1.01)*. Trusted Computing Group, TCG Published Specification. https://trustedcomputinggroup.org/wp-content/uploads/TCG_TPM_2.0_Automotive_Thin_Profile_v1.1-r15.pdf (accessed 18 May 2023).

46 Trusted Computing Group (2014). *TPM 2.0 Mobile Reference Architecture*. Trusted Computing Group, TCG Published Specification. https://trustedcomputinggroup.org/wp-content/uploads/TPM-2-0-Mobile-Reference-Architecture-v2-r142-Specification_FINAL2.pdf (accessed 18 May 2023).

47 Berger, S., Caceres, R., Goldman, K. A. et al. (2006). vTPM: virtualizing the trusted platform module. *Security'06: 15th USENIX Security Symposium*, Vancouver, Canada, July–Aug 2006. www.usenix.org.

48 England, P., Marochko, A., Mattoon, D. et al. (2016). RIoT – A Foundation for Trust in the Internet of Things. Microsoft Research, *Tech. Rep. MSR-TR-2016-18*. https://www.microsoft.com/en-us/research/publication/riot-a-foundation-for-trust-in-the-internet-of-things/ (accessed 18 May 2023).

49 England, P., Aigner, R., Marochko, A. et al. (2017). Cyber-resilient platforms overview (MSR-TR-2017-40). Microsoft Corp, Whitepaper MSR-TR-2017-40. https://www.microsoft.com/en-us/research/publication/cyber-resilient-platforms-overview/ (accessed 18 May 2023).

50 TCG (2020). *DICE Layering Architecture – Version 1.0*. Trusted Computing Group, TCG Published Specifications. https://trustedcomputinggroup.org/wp-content/uploads/DICE-Layering-Architecture-r19_pub.pdf (accessed 18 May 2023).

51 TCG (2018). *TCG Implicit Identity Based Device Attestation Version 1.0, Rev. 0.93*. Trusted Computing Group, TCG Published Specifications. https://trustedcomputinggroup.org/wp-content/uploads/TCG-DICE-Arch-Implicit-Identity-Based-Device-Attestation-v1-rev93.pdf (accessed 18 May 2023).

52 TCG (2018). *Hardware Requirements for a Device Identifier Composition Engine*. Trusted Computing Group, TCG Published Specifications – Family 2.0. https://trustedcomputinggroup.org/wp-content/uploads/Hardware-Requirements-for-Device-Identifier-Composition-Engine-r78_For-Publication.pdf (accessed 18 May 2023).

53 Zic, J. and Hardjono, T. (2013). Towards a cloud-based integrity measurement service. *Journal of Cloud Computing* 2 (4). https://doi.org/10.1186/2192-113X-2-4.

54 Microsoft (2020). Microsoft Azure attestation. https://docs.microsoft.com/en-us/azure/attestation/overview (accessed 18 May 2023).

55 TCG (2020). *TCG Cyber Resilient Module and Building Block Requirements Version 1.00, Rev. 0.08*. Trusted Computing Group, TCG Specifications. https://trustedcomputinggroup.org/resource/ (accessed 18 May 2023).

56 US DoD (1985). Department of Defense Trusted Computer System Evaluation Criteria (DoD 5200.28-STD). US Department of Defense, Department of Defense Standard DoD 5200.28-STD. https://csrc.nist.gov/csrc/media/publications/conference-paper/1998/10/08/proceedings-of-the-21st-nissc-1998/documents/early-cs-papers/dod85.pdf (accessed 18 May 2023).

57 Lindemann, R. and Jones, M.B. (2015). FIDO 2.0: key attestation format. FIDO Alliance, FIDO Alliance Proposed Standard. https://fidoalliance.org/specs/fido-v2.0-ps-20150904/fido-key-attestation-v2.0-ps-20150904.html (accessed 18 May 2023).

58 GlobalPlatform (2012). GlobalPlatform and the trusted computing group form work group to drive mobile security standards and solutions. https://globalplatform.org (accessed 18 May 2023).

59 IETF (2019). Remote ATtestation ProcedureS (RATS) Working Group – Approved Charter. Internet Engineering task Force. https://datatracker.ietf.org/wg/rats/about/ (accessed 18 May 2023).

60 Birkholz, H., Thaler, D., Richardson, M. et al. (2020). Remote attestation procedures architecture. IETF, Internet-Draft draft-ietf-rats-architecture-08. https://datatracker.ietf.org/doc/draft-ietf-rats-architecture/ (accessed 18 May 2023).

61 TCG (2019). *TCG Remote Integrity Verification (RIV): Network Equipment Remote Attestation System Version 1.0, Rev. 0.9b*. Trusted Computing Group, TCG Draft Specifications. https://trustedcomputinggroup.org/wp-content/uploads/TCG-NetEq-Attestation-Workflow-Outline_v1r9b_pubrev.pdf (accessed 18 May 2023).

62 Fedorkow, G., Voit, E., and Fitzgerald-McKay, J. (2020). TPM-based network device remote integrity verification. IETF, Internet-Draft draft-fedorkow-rats-network-device-attestation-05. https://datatracker.ietf.org/doc/draft-fedorkow-rats-network-device-attestation/ (accessed 18 May 2023).

63 OCP (2020). Open compute project. https://www.opencompute.org (accessed 18 May 2023).

64 Trusted Computing Group (2017). *TCG Glossary Version 1.1 Revision 1.0*. Trusted Computing Group, TCG Published Specification. https://trustedcomputinggroup.org/wp-content/uploads/TCG-Glossary-V1.1-Rev-1.0.pdf (accessed 18 May 2023).

65 TCG (2020). *Attestations Working Group*. Trusted Computing Group. https://members.trustedcomputinggroup.org (accessed 18 May 2023).

66 Smith, N. (ed.) (2020). *TCG Attestation Framework*. Trusted Computing Group, TCG Draft Specification – Version 1.0.

67 Hardjono, T. and Smith, N. (2021). Towards an attestation architecture for blockchain networks. *World Wide Web Journal – Special Issue on Emerging*

Blockchain Applications and Technology. https://doi.org/10.1007/s11280-021-00869-4.

68 Coker, G., Guttman, J., Loscocco, P. et al. (2011). Principles of remote attestation. *International Journal of Information Security* 10: 63–81. https://doi.org/10.1007/s10207-011-0124-7.

69 Bellare, M. and Rogaway, P. (1993). Random oracles are practical: a paradigm for designing efficient protocols. *Proceedings of the 1st ACM Conference on Computer and Communications Security (CCS'93)*. New York: ACM, 62–73. https://doi.org/10.1145/168588.168596.

70 Sharma, P.K., Kumar, N., and Park, J.H. (2019). Blockchain-based distributed framework for automotive industry in a smart city. *IEEE Transactions on Industrial Informatics* 15 (7): 4197–4205. https://doi.org/10.1109/TII.2018.2887101.

71 Hardjono, T., Lipton, A., and Pentland, A. (2020). Towards a public key management framework for virtual assets and virtual asset service providers. *Journal of FinTech* 1 (1). https://arxiv.org/pdf/1909.08607; https://doi.org/10.1142/S2705109920500017.

72 Simpson, W. (1996). PPP challenge handshake authentication protocol (CHAP). IETF Standard RFC1994. https://tools.ietf.org/html/rfc1994 (accessed 18 May 2023).

73 Berger, C., Penzenstadler, B., and Drogehorn, O. (2018). On using blockchains for safety-critical systems. *2018 ACM/IEEE 4th International Workshop on Software Engineering for Smart Cyber-Physical Systems (SEsCPS'18)*, 30–36. New York: ACM.

74 Hardjono, T. and Kazmierczak, G. (2008). Overview of the TPM key management standard. https://trustedcomputinggroup.org/wp-content/uploads/Kazmierczak20Greg20-20TPM_Key_Management_KMS2008_v003.pdf (accessed 18 May 2023).

75 Trusted Computing Group (2005). *TCG Interoperability Specifications for Backup and Migration Services (v1.0)*. Trusted Computing Group, TCG Published Specification. http://www.trustedcomputinggroup.org/resources (accessed 18 May 2023).

76 Seaman, M. (2018). IEEE Std. 802.1AR-2018 – Secure Device Identity. IEEE, IEEE Standard for Local and Metropolitan Area Networks.

77 Yaga, D., Mell, P., Roby, N., and Scarfone, K. (2018). Blockchain Technology Overview. National Institute of Standards and Technology Internal Report 8202. https://doi.org/10.6028/NIST.IR.8202 (accessed 18 May 2023).

78 Snyder, J. (2006). The competition for NAC: mapping Cisco, Juniper, Microsoft and TCG's access-control schemes. *Network World*. https://www.networkworld.com/article/2310209/the-competition-for-nac.html (accessed 18 May 2023).

79 Goel, M. (2006). Providing 802.1X enforcement for network access protection. presentation at WinHEC 2006. http://download.microsoft.com/download/5/b/9/5b97017b-e28a-4bae-ba48-174cf47d23cd/NET078_WH06.ppt (accessed 18 May 2023).

80 Heary, J., Lin, J., Sullivan, C., and Agrawal, A. (2007). *Cisco NAC Appliance: Enforcing Host Security with Clean Access*. Hoboken, NJ: Cisco Press.

81 Hardjono, T. and Smith, N. (ed.) (2006). *TCG Trusted Network Connect (TNC) Architecture for Interoperability*. Trusted Computing Group, TCG Published Specification – Version 1.1. http://www.trustedcomputinggroup.org (accessed 18 May 2023).

82 TCG (2017). *TCG Trusted Network Communications (TNC) Architecture for Interoperability*. Trusted Computing Group, TCG Published Specification – Version 2.0 Revision 13. https://trustedcomputinggroup.org/wp-content/uploads/TCG-TNC-Architecture-for-Interoperability-Version-2.0-Revision-13-.pdf (accessed 18 May 2023).

83 Waltermire, D., Cheikes, B.A., Feldman, L., and Witte, G. (2016). Guidelines for the Creation of Interoperable Software Identification (SWID) Tags. National Institute of Standards and Technology, NIST Report. http://dx.doi.org/10.6028/NIST.IR.8060.

84 ISO ISO/IEC 19770-2:2015 (2015). *Information technology – Software Asset Management – Part 2: Software Identification Tag*. International Organization for Standardization.

85 Lear, E., Droms, R., and Romascanu, D. (2019). Manufacturer Usage Description (MUD) Specification (RFC8520). https://tools.ietf.org/html/rfc8520 (accessed 18 May 2023).

86 Benton, D. (2019). NSA, Trusted Computing Group and Intel collaborate to standardize supply chain risk management. *Supply Chain Digital*. https://www.supplychaindigital.com/technology/nsa-trusted-computing-group-and-intel-collaborate-standardise-supply-chain-risk (accessed 18 May 2023).

87 Townsend, K. (2019). Intel announces compute lifecycle assurance to protect platform supply chains. *Security Week*. https://www.securityweek.com/intel-announces-compute-lifecycle-assurance-protect-platform-supply-chains (accessed 18 May 2023).

88 Protocol Labs (2019). Inter planetary file system (IPFS). https://docs.ipfs.io (accessed 23 September 2019).

89 CAB-Forum (2020). Guidelines for the Issuance and Management of Extended Validation Certificates. CA Browser Forum, Specification Version 1.7.2.

90 GLEIF (2018). LEI in KYC: A New Future for Legal Entity Identification. Global Legal Entity Identifier Foundation (GLEIF), GLEIF Research Report? A New Future for Legal Entity Identification. https://www.gleif.org/en/lei-solutions/lei-in-kyc-a-new-future-for-legal-entity-identification (accessed 18 May 2023).

91 Kohnfelder, L. (1978). Towards a practical public-key cryptosystem. BS thesis. MIT. http://hdl.handle.net/1721.1/15993 (accessed 18 May 2023).

92 ISO ISO 26324:2012 (2012). *Digital Object Identifier System – Information and Documentation*. International Organization for Standardization. http://www.iso.org/iso/catalogue_detail?csnumber=43506 (accessed 18 May 2023).

93 Sun, S., Lannom, L., and Boesch, B. (2003). Handle system overview. RFC3650. http://tools.ietf.org/rfc/rfc3650.txt (accessed 18 May 2023).

94 Sun, S., Reilly, S., and Lannom, L. (2003). Handle System Namespace and Service Definition. RFC3651. http://tools.ietf.org/rfc/rfc3651.txt (accessed 18 May 2023).

95 Reed, D. and Sporny, M. (2018). Decentralized Identifiers (DIDs) v0.11. W3C, Draft Community Group Report 09 July 2018. https://w3c-ccg.github.io/did-spec/ (accessed 18 May 2023).

96 Berners-Lee, T., Fielding, R., and Masinter, L. (2005). Uniform Resource Identifier (URI): Generic Syntax. RFC3986. http://tools.ietf.org/rfc/rfc3986.txt (accessed 18 May 2023).

97 Ferrer, E.C., Hardjono, T., and Pentland, A. (2019). Editorial: proceedings of the first symposium on blockchain and robotics, MIT Media Lab, 5 December 2018. *Ledger* 4. https://doi.org/10.5195/ledger.2019.179.

98 Ferrer, E.C., Hardjono, T., Pentland, A., and Dorigo, M. (2021). Secure and secret cooperation in robot swarms (to appear). *Science Robotics*. https://arxiv.org/abs/1904.09266.

99 Frankel, Y. (1989). A practical protocol for large group oriented networks. In: *Advances in Cryptology – Proceedings of EUROCRYPT '89*, *Lecture Notes in Computer Science*, vol. 434 (ed. J. Quisquater and J. Vandewalle), 56–61. Berlin, Heidelberg: Springer-Verlag.

100 Desmedt, Y. (1987). Society and group oriented cryptography: a new concept. In: *Advances in Cryptology – Proceedings of CRYPTO '87*, *Lecture Notes in Computer Science*, vol. 293 (ed. C. Pomerance), 120–127. Santa Barbara, CA: Springer-Verlag.

101 Mckeen, F., Alexandrovich, I., Berenzon, A. et al. (2013). Innovative instructions and software model for isolated execution. *Proceedings of the 2nd Workshop on Hardware and Architectural Support for Security and Privacy*

HASP2013, Tel-Aviv. Junehttps://sites.google.com/site/haspworkshop2013/workshop-program (accessed 18 May 2023).

102 Müller, C., Brandenburger, M., Cachin, C. et al. (2020). TZ4Fabric: executing smart contracts with ARM TrustZone. https://arxiv.org/pdf/2008.11601.pdf (accessed 18 May 2023).

103 Zanella, A., Bui, N., Castellani, A. et al. (2014). Internet of Things for smart cities. *IEEE Internet of Things Journal* 1 (1): 22–32. https://doi.org/10.1109/JIOT.2014.2306328.

7

Blockchain for Mobile Networks

Xavier Costa-Pérez[1,2,3], Vincenzo Sciancalepore[1], Lanfranco Zanzi[1,4], and Antonio Albanese[5,6]

[1] *NEC Laboratories Europe GmbH, 6G R&D Group, Heidelberg, Germany*
[2] *i2CAT, AI-DS R&D Group, Barcelona, Spain*
[3] *ICREA, Engineering Branch, Barcelona, Spain*
[4] *Intelligent Networks, Technische Universität Kaiserslautern, Kaiserslautern, Germany*
[5] *Telematic Engineering Department, Universidad Carlos III de Madrid, Leganés, Madrid, Spain*
[6] *Flyhound Co., New York, NY, USA*

7.1 Introduction

The traditional telecommunication business market is being challenged by decreasing revenues and increasing costs related to the deployment and operation of next-generation mobile networks. Therefore, in contrast to previous generations, the design and development of the fifth generation (5G) of mobile telecommunication networks has been driven by the need to accommodate, in a cost-efficient manner, a wider set of heterogeneous mobile services expanding from the classical broadband users toward novel Over-the-Top (OTT) services and *vertical industries*, e.g. automotive, smart factories, smart administrations, public safety, and e-health. In this way, new sources of revenue will be enabled by the continuously expanding *digital transformation of society*. Thus, next-generation mobile networks are expected not only to introduce remarkable performance enhancements in terms of latency, speed, and reliability as a means to support the growing mobile data traffic volumes but also to simultaneously accommodate very different stringent needs of the newly introduced industry verticals.

In order to do so, the latest technological developments in the field of network function virtualization (NFV) and software-defined networking (SDN) need to be exploited for supporting the seamless provisioning of multiple service-tailored virtual network instances with very diverse requirements sharing a common physical infrastructure. Such virtualization and cloudification trends are deeply

Blockchains: Empowering Technologies and Industrial Applications, First Edition.
Edited by Anwer Al-Dulaimi, Octavia A. Dobre, and Chih-Lin I.

affecting the telecommunication industry by pushing the mobile infrastructure to evolve from a relatively complex monolithic architecture, composed of dedicated hardware and network functionalities, into a pool of commoditized networking, computing and radio resources able to be dynamically *orchestrated by software*. However, while this effort may provide significant advantages in terms of flexibility and efficiency during the management and network orchestration (MANO) process, it also introduces novel *security challenges* when considering aspects like data privacy, service reliability, authentication, and monitoring of resource utilization [1]. At the same time, the compelling need to share both physical and virtual networking equipment as a way to reduce costs, exacerbate the requirements for solutions able to keep track of leasing transactions among the multiple business entities as well as to enable efficient management of all the operational aspects within the mobile network infrastructure.

In this context, the *Blockchain* technology represents a promising solution to enable storing and sharing of information in a distributed manner, without the need of a central authority. While maintaining the high security standards required by the enterprise market, the distributed algorithms and asymmetric cryptographic functions that characterize the blockchain solution well suit within the decentralized settings requirements foreseen in the 5G context and beyond, therefore paving the road for novel intra-domain communication schemes able to provide secure and reliable data exchange from and toward external business markets.

7.2 Next-Generation Mobile Networks: Technology Enablers and Challenges

As previously discussed, the next generation of mobile networks is designed not only to address the limitations of previous cellular standards bring remarkable performance enhancements in mobile communication, but also to accommodate, in a cost-efficient manner, the set heterogeneous mobile services targeting advanced use-cases that are gaining more and more attraction due to significant business implications, e.g. automotive, massive mobile broadband, etc. [2]. The hype around SDN and NFV will deeply change the mobile network infrastructure. Networking resources and functionalities traditionally running on top of dedicated hardware will now be executed, as virtualized functions, in a commoditized hardware. This enables *unprecedented levels of flexibility and programmability* into the mobile network management and planning with the associated increased security risks. Such innovations, deriving from the computing industry, yield profound changes in the architecture design of telecommunication systems, where the overall availability of cloudified networking, computing, and radio

resource pools can now be *orchestrated* or *sliced* into separate logical networks of the same physical infrastructure, each one operated independently and targeting specific services [3]. Clearly, this evolution has a strong business motivation, and meets the need of mobile operators to augment revenues and reduce operational costs but introduces new challenges that need to be carefully addressed.

7.2.1 Mobile Networks: Technology Enablers

5G and beyond technologies introduce new features that will enable a very diverse set of novel use cases for the digital transformation of society by incorporating many vertical industries which so far did not take advantage of the capabilities of mobile networks. In the following we briefly describe these new technologies along with the associated challenges to be addressed.

7.2.1.1 Software-Defined Networking (SDN)

SDN is one of the main evolutions in the mobile network ecosystem in the last years. Networking devices were traditionally designed to perform only a limited set of fixed functions, therefore limiting the options to update the functions of a network and increasing the management costs. SDN instead introduced the separation of control plane decision-making functionalities from the underlying network devices. The software-based logic is easier to maintain, as new configurations can be remotely enforced with clear advantages in terms of security and capital expenditures, specially in case of large-scale networks where, through programmable interfaces, centralized network controllers provide scalable solutions for traffic and packet flow management.

7.2.1.2 Network Function Virtualization (NFV)

NFV allows, by means of virtualization techniques, to deploy network functionalities traditionally running on dedicated hardware (e.g. firewall, VPN, router, switches) as virtual machines (VMs) or containers on top of commodity platforms. These virtualized instances, referred as virtualized network functions (VNFs), can be remotely orchestrated with significant advantages in terms of efficiency and flexibility.

7.2.1.3 Cloud Computing (CC)

Cloud computing (CC) provides the means to sustain the increasing demand for computational and storage platforms to host 5G mobile services and core network functionalities. By means of virtualization techniques a common pool of physical resources, including servers, networks, and storage can be efficiently shared, therefore allowing end-users to access running services regardless of their location. At the same time, software-based networking functionalities can be easily upgraded with significant costs savings for mobile operators.

7.2.1.4 Multi-access Edge Computing (MEC)

Multi-access edge computing (MEC) is envisioned as a key technology in the 5G landscape. The possibility to bring cloud computing capabilities to the edge of the network, geographically closer to the end users, enables the provisioning of latency-constrained and bandwidth-demanding services otherwise impossible in legacy mobile network architectures.

7.2.1.5 5G-New Radio (5G-NR) and Millimeter Wave (mmWave)

5G-new radio (5G-NR) and mmWave are two of the main enhancements introduced in the 5G radio access network (RAN). The possibility to exploit higher spectrum frequencies (up to 60 GHz) allows for wider communication bandwidth, and provides the radio access network with the necessary capacity to accommodate the growing traffic volumes. Smaller radio beams would allow dense RAN deployments to simultaneously accommodate an unprecedent number of connected devices in a more reliable way, ensuring latency and bandwidth requirements even in case of concurrent heterogeneous use-cases. From the mobile network perspective, a deep revision of the wireless protocol stack, referred as 5G-new radio (5G-NR), introduce significant novelties in terms of radio resource management and scheduling with respect to previous mobile network generations. Among the wide spectrum availability, only some bandwidth parts, e.g. those with better channel statistics, may be adopted to match the real-time traffic loads finally improving the overall quality of experience and energy consumption.

7.2.2 Mobile Networks: Technology Challenges

In this section we discuss about the main challenges and business motivations behind the paradigm shift in next-generation networks from centralized approaches toward distributed ones.

7.2.2.1 Scalability in Massive Communication Scenarios

While the centralized approach characterizing the past generations of cellular networks allows for easier security management, it often introduces a limitation which hardly matches with the foreseen mobile market evolution. The distributed Internet of Things (IoT) deployments envisioned in the 5G ecosystem demand for ubiquitous radio connectivity and secure data collection schemes to store raw monitoring data coming from heterogeneous sensing sources. In smart cities or transportation scenarios for example, this is fundamental to allow the extraction meaningful information from environmental sensing and improve the situation awareness of safety platforms. In this context, a massive number of connected devices together with the adoption of a centralized architecture

introduces significant challenges in the fulfillment of throughput and latency requirements. Additionally to performance concerns, a centralized approach may raise security issues, e.g. fail in promptly detecting malicious attacks due to communication delay.

7.2.2.2 Efficient Resource Sharing

In order to provide end-to-end services, the setup of network slices implies the creation of a logical chain linking virtual networking functionalities distributed over multiple network domains and cloud providers [4]. Within this scenario, it is fundamental to ensure secure data transmission as well as to avoid any data leakage, not only to preserve end-users' privacy, but also to ensure the correctness of all the orchestration activities involving multi-domain resource allocation processes. Given the shared nature of the networking equipment, denial of service in a single point of the network may have serious impact on multiple services. The last aspect is particularly challenging in the radio access domain, where scarce spectrum resource availability meets a wide number of distinct mobile services to be accommodated [5].

7.2.2.3 Network Slicing and Multi-tenancy

The novel network slicing paradigm allows mobile operators to offer virtualized instances of their physical infrastructure to 3rd-party service providers, or tenants, external at the mobile ecosystem. The resource allocation of each slice must be tailored to satisfy the particular service requirements, e.g. in terms of latency and bandwidth. Tenants can be entitled to directly manage the allotted subset of resources, therefore evolving the traditional resource sharing concept in favor of multi-tenancy management schemes [6]. Clearly, mobility and wireless communication aspects make it hard to find a durable allocation strategy able to satisfy all the business entities. This scenario calls for novel solutions to allow secure multi-domain resource exchange in an efficient manner [3], not only to enable flexible inter-operator resource sharing, but also to support the evolution of a multi-tenant network slicing resource market.

7.2.2.4 Security

Given the key role that mobile networks are gaining within today's society, it is not a surprise that security aspects are regarded as one of the main topics in the design of future 5G networks. In principle, a unique and centralized service platform represents a single-point of failure that implies severe threats for service reliability. On the other side however, a fully distributed approach enlarges the network surface vulnerable to security attacks [7].

Confidentiality among communications involving distributed network domains, data integrity and immutability, authentication and access control are

only some of the security enhancements demanded by the adoption of multiple technology enablers composing the 5G ecosystem.

7.3 Blockchain Applicability to Mobile Networks and Services

Despite becoming famous for the hype around cryptocurrencies, the blockchain technology applicability is not limited to that scope, and its adoption in other business scenarios is currently investigated. In the following, we provide a brief introduction of the main concepts and components of the blockchain technology, as a means to fully realize the potential impact that this solution may have on future mobile network developments [8].

7.3.1 Background and Definitions

In its simplest definition, a blockchain is a distributed data structure shared among the members of the network. As the name suggests, it can be seen as a chain of data blocks. Each block stores a record of the last transactions occurred within the network, a set of meta-parameters to guarantee the correct evolution of the chain e.g. timestamp, amount of good exchanged, partners involved, and most importantly, a reference to the previous block of the chain (usually the hash of its content). The blockchain technology makes use of a distributed database, or ledger, to store new and past transactions. Rather than being controlled by a central authority, the ledger is copied and shared across multiple nodes and made available to all users through an internet connection, being the absence of a centralized control an advantage for data transparency. All members own the same management rights, and act as validators of the state of the ledger making the blockchain a fully distributed system. The absence of a centralized entity controlling the overall operations implies synchronization issues and demands for alternative solutions to manage the insertion of new information into the ledger, and to maintain the transaction history coherent throughout the network. To solve this issue, a *consensus* algorithm must be in place. Its main objective is to ensure that all peers reach a common acceptance about the real-time state of the distributed ledger. Several algorithms are available in the literature each one showing its own advantages and drawbacks, e.g. proof of elapsed time, proof of work, and so on [9]. Unlike centralized systems where trust in the managing authority is a necessary condition, thanks to consensus algorithms blockchain users can still

operate even in absence of mutual trust. Nevertheless, we can identify two types of blockchain which differentiate among the set of management operations granted to peer users:

- *Permissionless* blockchains allow anyone to read, write, and actively participate in the creation of new blocks as well as to update the ledger.
- *Permissioned* blockchains pose restriction on who is allowed to participate in the network activities, e.g. by means of offline invitation processes. Moreover, specific users may be limited in the kind of transactions they can perform, or in their role within the network, i.e. ledger validation only, transaction proposition only, etc.

Considering the enterprise facility represented by a mobile network infrastructure, permissioned access is preferable to maintain high security levels. To this aim, permissioned blockchains often exploit trusted execution environments (TEE) to securely onboard participants and assist with the establishment of the consortium that composes the blockchain network. Such scheme also avoids the need of energy-consuming activities related to block validation process, which has been identified as one of the main drawbacks of public blockchain systems [10]. In this case, we can assume that peer nodes admitted in the system are not malicious and rational, i.e. profit driven. To further guarantee a secure environment for the transaction exchange in absence of mutual trust, smart contracts (SCs) can be optionally used to automatize the exchange of goods in reply to trigger events. A smart contract can be defined as an agent that translates, in an automatic way, contractual clauses into self-enforcing software that minimizes the need of trusted intermediaries. SCs can be stored in the blockchain itself and provided with a unique address, making it easy to be reached from all the peers in the network and inheriting useful security features like distributed consensus agreements to prevent fraudulent usages. We depict the above-described blockchain basic operations in Figure 7.1.

The implementation of smart contracts often implies the usage of high-level programming languages, which are then compiled into low-level byte-coded languages and loaded into the blockchain to ensure immutability, as no other user may change the rules defined within the smart contract. All together, these characteristics ensure high tolerance against security attacks and are at the basis of the wide adoption of the blockchain technology in supply chain management and money transfer contexts.

The distributed framework introduced by the blockchain technology enables transactions exchange without the need of centralized management entities.

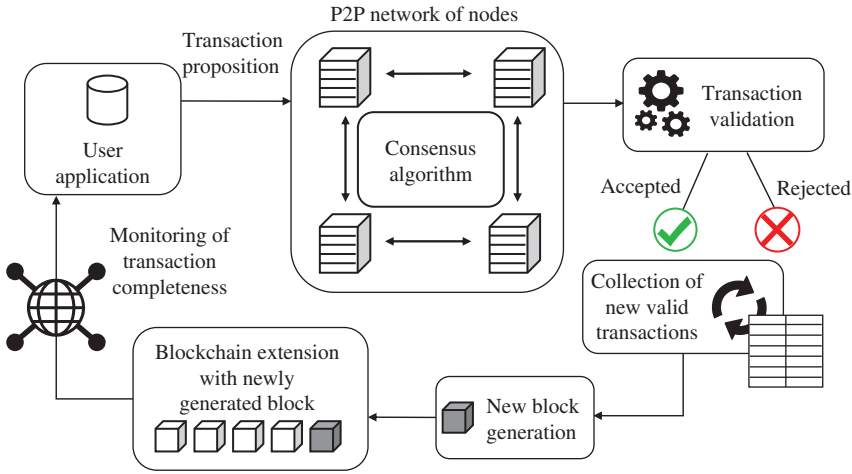


Figure 7.1 Blockchain basic operations.

In [11] the authors present a study on the leasing ledger concept proposing the blockchain technology as a means to overcome absence of trust in data management and satisfy the need for automated solutions in industrial network facilities. One of the key features inherited by the blockchain technology is indeed the capability of providing trust in a distributed way. In a similar way, the authors of [12] address the scalability weakness inherited by the adoption of proof-of-work consensus algorithms, proposing an advanced neighbor selection scheme to reduce the block propagation time along the peer-to-peer network. The mining nodes would consider real-time bandwidth conditions and favor nodes with higher bandwidth availability thus increasing the overall transaction throughput. Differently from the previous approach, with focus on the IoT ecosystem the authors of [13] introduce a service-oriented permissioned blockchain, where different consensus protocols are deployed according to users' Quality of Service (QoS) requirements, and validation entities are dynamically elected according to traffic loads.

7.3.2 Blockchain for Radio Access Networks

Aiming at dynamic scaling and versatile network deployment, flexible resource management is required in every network domain. In this respect, virtualization plays a major role even in the Radio Access Network domain, enabling the implementation of RAN functionalities over general-purpose platforms rather than specialized hardware. The novel cloud-radio access network (CRAN) paradigm represents a promising application scenario for the blockchain technology.

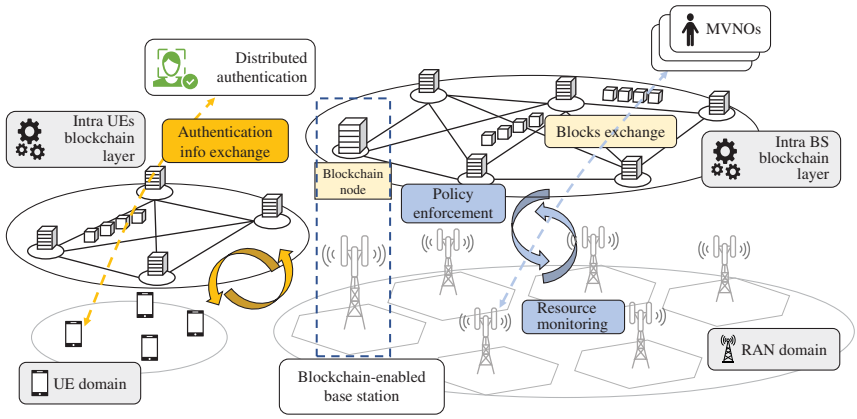


Figure 7.2 Example of blockchain applications in the RAN domain.

The introduction of blockchains into the RAN architecture provides a decentralized, secure, and efficient platform for user access and authentication [14], allowing to propagate sensitive end-users' information in an anonymous and untraceable manner [15]. As depicted in Figure 7.2, user equipments (UEs) and 5G base stations (gNBs) take part in a public blockchain and proceed with the negotiation of an agreement in terms of network traffic payment rate and assigned radio resources. Successful agreements are recorded in smart contracts that are in turn verified by miners so as to ensure that both parties have enough assets to satisfy their mutual agreement. Several smart contracts make up a block of the main ledger while their policies are enforced in the RAN. This approach realizes a trustworthy solution among UEs and gNBs, sparing the overhead costs and the additional security threats originated by centralized authentication schemes, and enacts an open market among users and network providers. Moreover, third parties interested in trading network resources can easily join the market as it is self-regulated by means of the verification process put in place by miners. For instance, mobile virtual network operators (MVNOs) and Infrastructure Providers (InPs) may use the blockchain as a secure platform for negotiating service-level agreements (SLAs) and radio frequency channels assignments while minimizing over-committing issues from both entities [16].

As far as network management is concerned, blockchain allows to execute RAN management operations in a completely decentralized manner [17]. A dedicated blockchain running among gNBs lays the foundations of a self-organized RAN environment that can achieve optimal radio resource utilization without the need of the costly signaling procedures required by traditional approaches. Indeed, the densification of gNBs and the coexistence of different numerologies (i.e. physical layer configurations) in the 5G-and-beyond era exacerbate the

need of effective interference management solutions [18] able to synchronize the run-time operations and settings of neighboring gNBs. Along these lines, blockchain-based cooperation among gNBs can be leveraged to improve mobility management procedures (especially radio handovers), which are prone to security flaws and service disruption, e.g. in case their execution time exceeds a required latency threshold. In particular, being part of a common blockchain network, the set of eNBs/gNBs involved in a handover procedure may exchange user-related encryption keys in a secure manner, inheriting the constituent security features of blockchains [19]. Additionally, by choosing a suitably lightweight consensus protocol, the overall operation can meet the stringent latency requirements [20]. Besides the conventional mobile broadband (MBB) scenarios, 5G brings up new challenging use cases, among which vehicular networks (VANETs) and IoT are notable examples. Blockchains enhance the security and performances of 5G in support to VANETs [21] and provide a mean to check the provenance of data in 5G-IoT networks [22].

7.3.3 Blockchain for Core, Cloud, and Edge Computing

The rise of network-intensive services enabled by the next mobile network generations puts the current cloud computing infrastructure to the test. Indeed, more and more services (e.g. e-health, autonomous driving, tactile Internet) are expected to offload big volumes of data to remote servers, while calling for stringent end-to-end performance requirements in terms of guaranteed bandwidth and overall communication latency. Such requirements, together with the ever-increasing number of security challenges, cannot be met by a centralized architecture, which shows a unique point of failure and poses unacceptable flexibility and scalability issues for the highly dynamic 5G-and-beyond era [23]. Blockchain technology enters the scene as a promising solution to deal with the above-mentioned problems while providing the tools to easily migrate toward a decentralized architecture [24], as shown in Figure 7.3. Specifically, on top of its enhanced security, it provides means to continuously and autonomously update a distributed ledger and quickly recover from failures as several nodes keep a (partial or complete) copy of the chain. The massive data sent to the cloud is vulnerable to information modifications or attacks by third parties, which may also be internal to the network, in the attempt to obtain valuable user-related information (e.g. real-time user positions).

7.3.3.1 Data Provenance

Keeping track of data provenance can protect user data from malicious attacks as well as enhance cloud computing safety against communication vulnerabilities [25], allowing easier detection of threats and quicker reaction to security

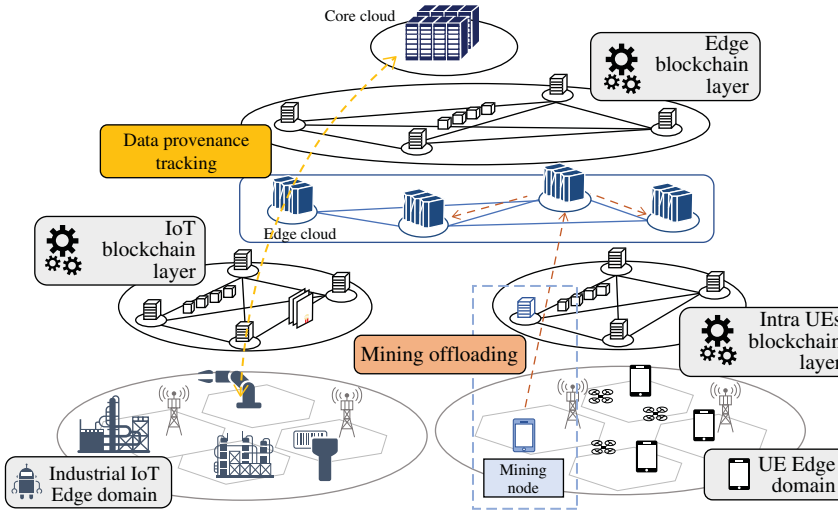


Figure 7.3 Example of blockchain applications in the Edge and Cloud domain.

issues. This reminds of an intrinsic property of blockchains, which record every transaction and involved entities into data blocks that are immutable by design. By means of smart contracts, blockchains provide an automatic system to seamlessly verify data provenance (through distributed signature schemes and relying on auditors for verification) while storing sensitive data either on-chain or off-chain [26].

7.3.3.2 Encrypted Data Indexing

Privacy concerns push users to perform data encryption while uploading contents into cloud platforms, thus limiting data indexing capabilities of the shared infrastructure. To meet the need of searchable encrypted data in the cloud and avoid users the burden of decrypting data before executing the search, searchable encryption techniques have been introduced since the late 1990s [27]. In this regard, blockchains can be used to store index files while relying on the cloud platform just for storage [28].

7.3.3.3 Mobile Network Orchestration

Within the novel slicing paradigm, Rebello et al. [29] propose to extend the network function virtualization-management and network orchestration (NFV-MANO) architecture to account for a dedicated application programming interface (API) through which network slices can be configured and orchestrated according to the negotiated transactions. As future work, Rebello et al. [29] highlight the need for a consensus algorithm able to manage, in an efficient

manner, the huge number of interactions expected in slicing systems. In [30] the authors tackle this issue proposing a hierarchical combinatorial auction approach involving mobile edge and cloud computing platforms to offload expensive mining processes to alleviate mobile terminal battery consumption, while [31] propose to evolve proof-of-work-based consensus algorithms toward the definition of multiple blockchains with sharded consensus as a means to improve throughput and scalability exploiting parallel computing. In a similar way, the authors of [12] address the scalability weakness inherited by the adoption of proof-of-work consensus algorithms, proposing an advanced neighbor selection scheme to reduce the block propagation time along the peer-to-peer network. The mining nodes would consider real-time bandwidth conditions and favor nodes with higher bandwidth availability thus increasing the overall transaction throughput.

7.3.3.4 Mobile Task Offloading

Resource-intensive mobile applications often call for support from the cloud infrastructure to satisfy their execution requirements under energy-constrained scenarios and terminals, e.g. UEs or IoT devices [32]. Although the current paradigm allows mobile users to offload such heavy tasks to remote servers, the massive amount of exchanged data and the unreliable cellular network availability may lead to unaffordable execution delay or, in the worst-case scenario, to service disruption. MEC effectively tackles these criticalities [33] while providing a powerful platform to offload heavy blockchain-related tasks (e.g. the Proof of Work computational effort) at the edge of the network. In this regard, several techniques have been investigated considering heterogeneous and contrasting objectives such as egoistic user's profit, overall system fairness, and the cloud provider's revenue maximization. In the last case for instance, cloud operators can dynamically set the resource prices by means of game-theoretic model, which would maximize their profit at the equilibrium [34].

7.3.3.5 Service Automation

The agile cloud-based environments envisioned in future 5G deployments enable automatic network infrastructure management through software-centric orchestrators supported by real-time analytics. In this context, blockchain smart contracts may be used to negotiate and define partnership agreements among multiple administrative domains resulting in complex end-to-end service deployments covering various and heterogeneous network assets. This will speed up the deployments of novel services, avoiding the adoption of costly intra-domain manual configurations based on infrastructure footprints [35].

7.4 Blockchain for Network Slicing

The novel network slicing paradigm, made available by the latest developments on virtualization and softwarization technologies, enables advanced and dynamic resource allocation schemes built on top of modular mobile architectures and commoditized platforms. Such advanced resource allocation mechanisms must deal with a heterogeneous and wide set of vertical requirements to satisfy per-slice performance guarantees. In this context, the figure of the *Network Slicing Broker (NSB)*, firstly introduced in [36], acts as an entity in charge of mediating between industry verticals' slice requests (SRs) and the mobile infrastructure resource orchestrator.

Thus, the NSB concept can be extended toward further dividing the value chain and allowing the entrance of new players in a similar manner as MVNOs did in telecom networks. MVNOs allowed InPs to address specific market niches, which they did not manage to tap into due to the *subscriber acquisition costs*. The new challenge here is that, while the number of MVNOs is rather small in established mobile markets, network slicing is expected to accommodate hundreds to thousands of new industry vertical tenants, ranging from full coverage connected car platforms to localized IoT deployments.

In order to achieve this, as reported in our previous work [37], the figure of an *intermediate broker (IB)* can be introduced, which leverages on *Blockchain* technology to develop a *network slicing brokering* solution (hereafter referred as NSBchain), enabling *InPs* to allocate network resources to IBs through smart contracts and IBs to allocate and re-distribute their resources among tenants in a *secure, automated, and scalable* manner. While MVNO agreements with InPs have to go through a regular *offline contract signature* process, NSBchain enables a much faster, scalable, and cost-efficient *secure online digital signature* process for the resource allocation transactions.

7.4.1 The Network Slice Broker (NSB)

From a business perspective, the network slicing economy revolves around three main entities [36]:

- The *InP*, which is the owner of the mobile network physical infrastructure and responsible for its maintenance.
- the *Network Slice Tenants*, which are those business entities, e.g. OTT service providers or 3rd-party vertical industries, interested in renting a slice of the mobile network from the InP to provide tailored services to their customers

through allocation of dedicated resources. These entities are usually provided with specific privileges and access rights to the pool of shared networking resources.
- The *NSB*, which is in charge of mediating between tenants' requirements and network resource availability, and instructing the physical infrastructure to accommodate requests.

In more detail, upon slice requests arrivals, the NSB is in charge of running an admission control mechanism, and if granted, enforcing the deployment of the new slices in the system. Such admission control mechanism involves the evaluation of the slice resource requirements against the resource availability over the different network domains, Radio Access Network (RAN), transport, and core. Keeping running slices SLAs isolated from newcomers is of paramount importance in this scenario as it shall avoid resource shortage that might impact the service delivery. As different tenants may require a diverse set of network resources, the admissibility of each slice request depends on an elaborated multi-domain optimization problem, see for instance [6]. To ease this task, a common solution accounts for the usage of a predefined set of network slice templates (NSTs) [38]. Each template specifies static parameters and functional components of different network slice types as well as the relevant attribute's value in terms of resource allocation requirements necessary to satisfy the service provisioning. An illustration of the workflow is depicted in Figure 7.4. Clearly, the negotiation of network resources remains transparent to the slice end-users and involves only 3^{rd}-party business entities and the InP. Nevertheless, the outcome of such negotiation process may have a deep impact on the data plane functionalities, finally influencing the possibility to reach the target levels of QoS and end-users' quality of experience.

7.4.2 NSB Blockchain Architecture (NSBchain)

The network slice ecosystem is envisioned to support dynamic and real-time resource allocation over the mobile network. In such a fast-changing scenario, tenant requirements may vary as a result of external causes, e.g. end-users' mobility, possibly leaving tenants with under- or over-provisioned network slices and the need of acquiring/releasing resources.

In this context, the roles of the InP and the wholesaler can be comparable. From its perspective, it is preferable to deal with the exchange of big quantities of goods to intermediate retailers rather than trading, with a significant increase of management costs, small quantities directly with the end-users. Thus, this opens up to new marketing opportunities for 3rd-party entities willing to play the role of retailers, e.g. MVNOs, municipalities in case of public events, highway operators

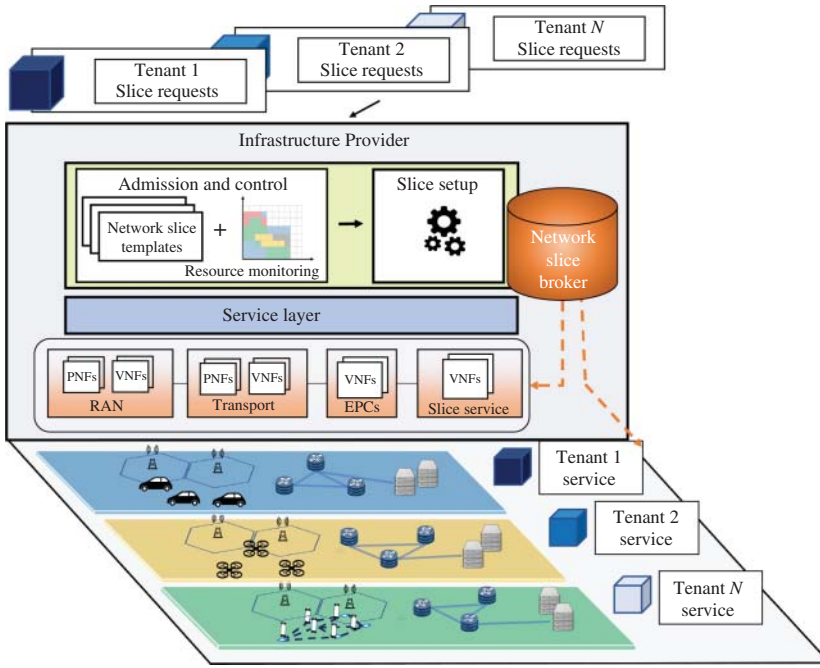


Figure 7.4 Network slice brokerage overview

and factories, which may buy a quota of network resources from the InP and re-sell it to final tenants. We define such business entities as *IBs*.

We envision the network slicing economy as an open market where tenants can select the IB that best suits their requirements, e.g. better price, thus leading to the creation of consortia of tenants under the management of the same IB. The proposed architecture is depicted in Figure 7.5.

7.4.2.1 Technical Challenges

In order to support the hierarchical structure above-described as well as the additional management and security complexity inherited by this enhanced business model, several challenges must be considered: flexibility and scalability are key features for next-generation mobile networks.

The assignment of network resources to tenants is highly affected by mobility and interference management aspects. In such cases, the resource allocation process requires to evolve dynamically following tenant demand variations. At the same time, the chain of network resource loans must be negotiated in a secure, transparent, and fast way [39, 40], such that the lifecycle of each slice is not affected. Current mobile network sharing solutions require long negotiation

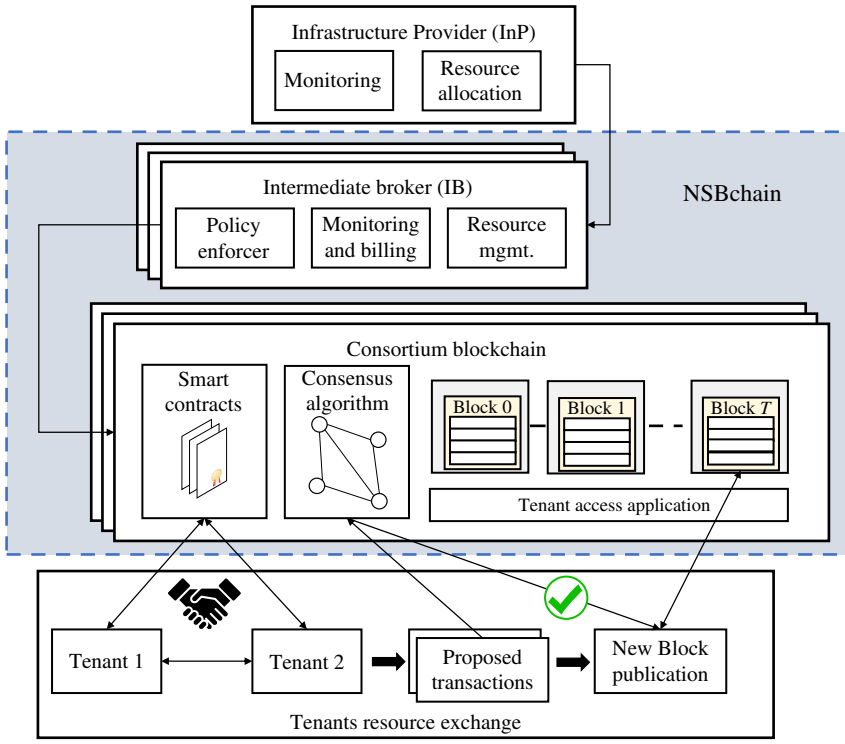


Figure 7.5 A distributed hierarchical architecture for network slicing.

processes that hardly fit within short time-to-market deployments of the 5G use-cases.

Instead, due to its decentralized nature, the blockchain technology well suits these requirements. The distributed ledger allows all members of the system to be aware of the current (and past) network resource availability as well as to be informed, in real-time, about the dynamic exchange of resources through a public hash-chain of blocks provided with valid transactions. A secure resource exchange is guaranteed by smart contracts and distributed consensus algorithms, allowing the system to evolve autonomously without the need of centralized authorities. At the same time, the distributed approach eases the identification of misbehaving members and mitigates the effects of malicious attacks on the system.

SCs are crucial to guarantee reliable auditing and enforce IB-specific policies in the management of requests. For example, one IB may decide to auction his share of resources in different ways [30, 41, 42] or simply sell them to the first coming tenant. Peer nodes can invoke a SC by sending transactions to its address. In more detail, if a new transaction is proposed in the system, the contract address can be

inserted as recipient address of the transaction. To validate the resource exchange, all the peer entities execute the code using, among the others, transaction payloads and current system state as input arguments of the call [43]. The participation in the consensus protocol finally assures that the new output ledger comes from valid transactions.

7.4.3 NSBchain Modeling

Let us introduce $\mathcal{B} = \{b_1, b_k, \dots, b_K\}$ as the set of IBs allowed to trade network resources, and $\mathcal{T}_k = \{\tau_1, \tau_t, \dots, \tau_T\}$ as the set of tenants admitted within the consortium of IB b_k. Being a permission-based system, this framework requires an invitation for participation.[1] To guarantee secure message exchange, each entity is provided with a cryptographic key pair $\{K_{priv}, K_{pub}\}$. The usage of group signature schemes and the generation of new key pair for every message exchange is preferable to avoid reply attacks [44].

We detail in the following the main steps involved in the creation and management network slices on a blockchain-based platform providing a mathematical background for the consensus process and the overall revenue maximization.

7.4.3.1 System Setup

In order to enable dynamic resource exchange among tenants, a dedicated blockchain must be set up for each consortium of tenants. Each IB b_k deploys the first block of the chain and loads a registry of resources $\mathcal{R}_k = \{r_1, r_i, \dots, r_I\}$ into such a block, which reflects the amount of i-type resources, with $i \in \mathcal{I}$, originally assigned by the InP. This step is required to avoid over-selling so as to limit the availability of resources in the blockchain. Each IB b_k can define leasing policies and code them into a set of SCs, which are then available to all tenants in the consortium. Finally, each IB b_k is in charge of assigning the initial share of resources to admitted tenants.

7.4.3.2 Message Exchange

Upon private exchange domain creation, network slice requests can be dispatched among the network of peers. According to their real-time requirements, tenants may decide to publish a resource advertisement or a resource request message. In the former case, the current owner of resources decides to release some of his shares making them available on the market. In the latter case, the tenant broadcasts its need to other tenants, which may be interested in providing their quota. To guarantee authentication, each message is signed with the sender

1 While the admission procedure is out of the scope of this work, it is assumed that such a mechanism is in place and managed by the InP to guarantee that only trustworthy entities are admitted.

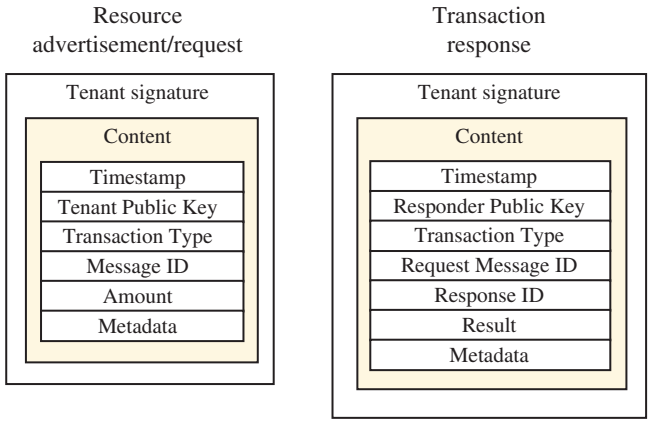


Figure 7.6 Example of transaction message exchange within NSBchain.

private key and uniquely identified by an ID number. A simplified message structure is depicted in Figure 7.6.

The network slicing brokerage must deal with multi-domain resource allocation problems. In its simplest definition, a resource request from tenant τ can be defined as a tuple $\Psi_\tau = [\pi_1^{(\tau)}, \pi_i^{(\tau)}, \ldots, \pi_I^{(\tau)} | \theta_1^{(\tau)}, \ldots, \theta_I^{(\tau)}]$, where $\pi_i^{(\tau)}$ represents the required amount of i-type resources, and $\theta_i^{(\tau)}$ is the price to be paid. There are no limitations on the nature of exchanged resources, as the proposed resource request scheme easily accommodates heterogeneous resource specifications. For example, a tenant could be more interested in trading only Radio Access Network (RAN) resources at the edge of the network, e.g. for delay-sensitive applications, while others may be more interested in cloud resources, e.g. storage and processing power for data analytic applications in the context of the IoT.

7.4.3.3 Billing Management

Interestingly, a blockchain can be viewed as a transaction-based state machine, wherein its state is updated every time consensus is reached on a set of transactions. To this aim, orderer nodes can be introduced and exploited to collect and sort proposed transactions by arrival time. Such nodes are usually not involved in the validation process; however, they may allow decoupling and parallel processing of ordering and validation functionalities thereby improving the overall system efficiency [29].

We show the blockchain architecture with involved players supporting network slicing in Figure 7.7. Specifically, the IB may join blockchain activities, i.e. it might read the blockchain results, participate to validation and consensus phases as an active member of the blockchain consortium. This implies that the IB can recursively apply confirmed (validated) transactions onto resource scheduling

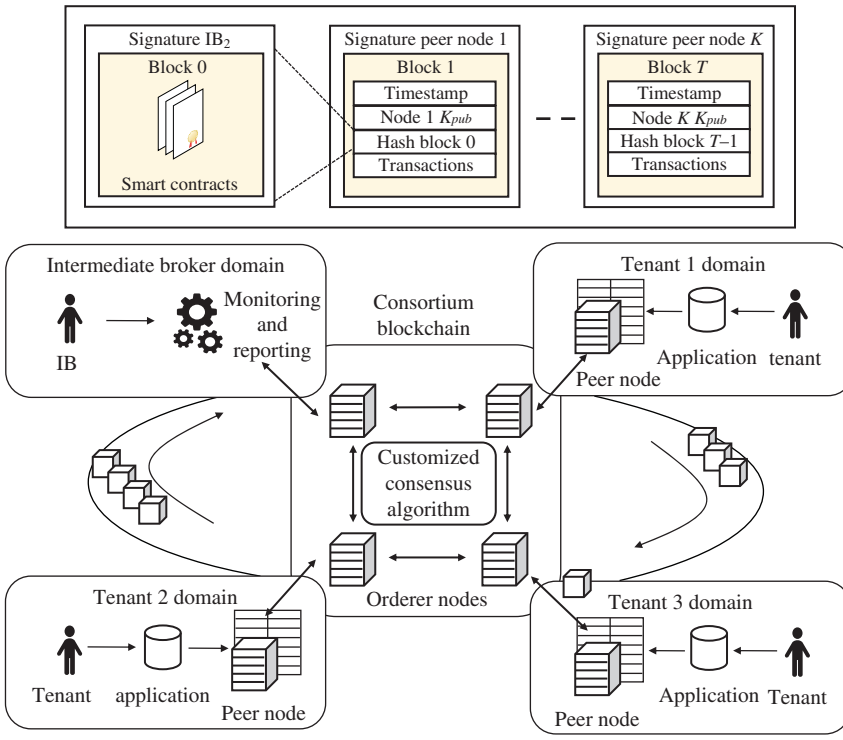


Figure 7.7 Private blockchain architecture supporting network slicing.

policies that might include (not limited to) RAN/transport and computational resources.

Despite enabling a more dynamic resource trading market, the blockchain technology would easily allow to keep track of the different resource exchange over time. From the InP perspective, this also simplifies the billing management as each block of transactions stores precise information about the nature of the exchanged resources and the corresponding time window utilization. Moreover, tenants are directly responsible for the management of their requests: once issued, they could not be withdrawn. Clearly, each IB shows interest in managing properly its resource share with the objective of maximizing the overall final revenue while parsing and processing upcoming slice requests.

Let us assume that each tenant τ can issue multiple slice requests so that the IB can collect all coming slice requests Ψ_j with $j \in \mathcal{J}$. A slice request is accommodated only if all types of demanded resources can be assigned to the tenant thereby guaranteeing a correct end-to-end slice instantiation.

We define transaction throughput as the number of transactions that the system can handle per unit of time. In realistic scenarios, this number can range over a wide range depending on the study use-case. For example, public BitCoin's network supports 7 transactions per second, while the financial networks of MasterCard and Visa handle up to 60 000 [45]. Obviously, different consensus algorithms provide different latency [46]. For this reason, we let each IB b_k choose the preferred method according to its service requirements. In general, being NSBchain a permissioned framework, we suggest the use of relatively light mechanisms, like Practical Byzantine Fault Tolerance (PBFT) consensus protocol [47], Kafka [48], or Raft [49] to allow fast convergence to a common agreement and speed up the resource exchange process.

7.4.4 NSBchain Evaluation

NSBchain can be implemented on top of Hyperledger Fabric [50], an open-source framework for developing permissioned blockchains within private enterprises, which provides a benchmarking tool, namely Hyperledger Caliper, to evaluate the blockchain performance in network slicing scenarios.

7.4.4.1 Experimental Setup

The Proof-of-Concept (PoC) architecture consists of 3 IBs and a variable number of orderer nodes that depends on the adopted consensus algorithm. Such entities run as Docker containers on an Intel Xeon CPU E5-2630v3 32-Core @2.4 GHz 64 GB RAM shared platform.

The definition of dedicated and encrypted communication channels guarantees the isolation among consortia. Moreover, we set the maximum number of entries per block to 20 and the block timeout[2] to 300 ms. This last metric specifies the amount of time (after receiving the first transaction) each orderer waits before publishing a new set of proposed transactions to other peer nodes. It is worth noting that the choice of those parameters may strongly affect the blockchain performance. In particular, although decreasing the block timeout improves the latency, setting it to low values may decrease the overall throughput as new blocks would not be filled up to their maximum capacity [51]. To limit the impact of this trade-off on our results, we do not modify these settings throughout this section.

The benchmark process consists of two phases, dubbed as opening and transfer. In the initial phase, we create tenant instances and assign them with an equal amount of resources such that all available resources at IB side are assigned. Once assigned, each tenant might decide to free or seek additional resources

2 We select such values as they maximize the throughput at a minimum latency cost as proved hereafter in the section.

based on a random value drawn from a uniform distribution between 0% and 30% of the initially assigned amount. During the transfer phase, tenants issue slice requests (SRs), modeled as tuples $\Psi_\tau = \{\rho, \eta, \gamma\}$, where $\rho, \eta, \gamma \in \mathcal{R}_k$ represent the percentage of required radio access, transport, and core cloud resources, respectively. In case the SR does not fit the availability or the need of the involved tenants (SR collision), it is automatically rejected and the respective transaction is dropped.

7.4.4.2 Full-Scale Evaluation

With the first experiment we evaluate the performance of our framework in terms of slice request throughput and latency. We compare two popular consensus algorithms (Kafka and Raft) against a single-orderer configuration (Solo) that does not require any consensus process. The top of Figure 7.8 shows the average SR throughput of the platform in the transfer phase for an increasing consortium size and fixed SR rate of 150 SRs s^{-1} to emulate high load conditions. In these settings, especially for a small consortium size, the limiting factor of the blockchain performance throughput is the multiversion concurrency control (MVCC) process. As we issue SRs at a very high rate, the same database entry, e.g. the resources assigned to a specific tenant, may be edited by a new request before the completion of the validation process involving it. This raises a database

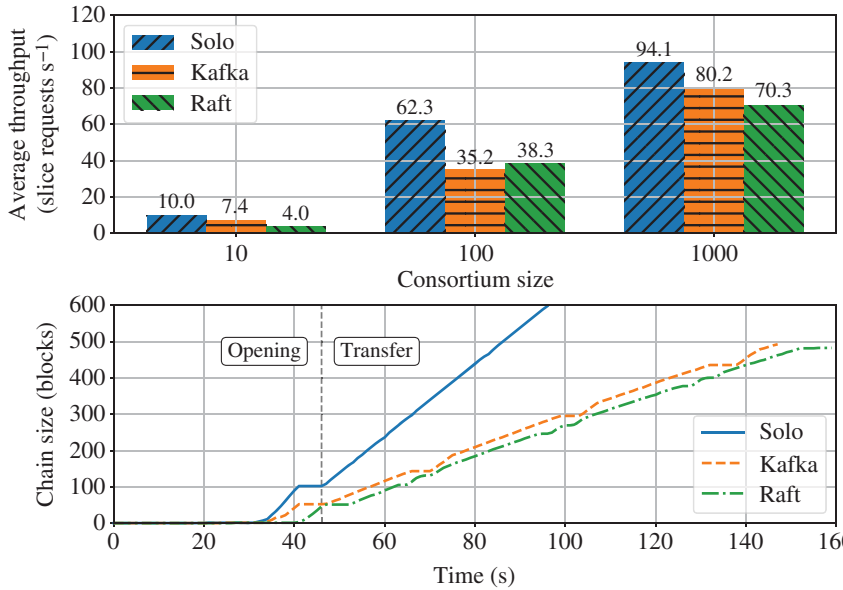


Figure 7.8 Slice request throughput and blockchain size growth for different consensus algorithms and consortium size.

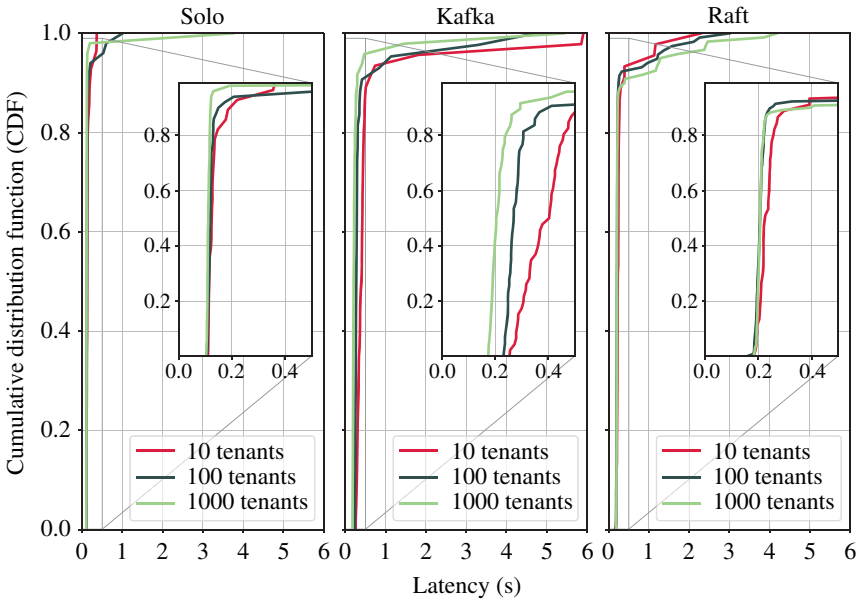


Figure 7.9 CDF of the slice request validation latency experienced by tenants for different consensus algorithms and consortium size.

inconsistency, dubbed as read/write (RW) conflict, which prevents the current transaction to be successful. As shown in the figure, this problem is mitigated by an increasing consortium size.

Figure 7.9 depicts the cumulative distribution function (CDF) of the experienced latency by the successful SRs. As expected, the best latency performance is obtained when no distributed consensus mechanism is in place, i.e. Solo. However, despite being the fastest scheme, this single-node approach is not fault tolerant. It can be noticed that the transaction exchange and validation process introduce a small time overhead for the Kafka and Raft cases, which however has negligible impact, especially when compared to the onboarding time required e.g. by virtualized infrastructures to setup virtual services [40]. The blockchain growth rate is also affected by the different consensus scheme, as shown at the bottom of Figure 7.8, which refers to the consortium size case of 1000 tenants. We plot the evolution of the chain size over time and mark the beginning of the transfer phase with a dashed vertical line. It can be noticed that the blockchain grows at a rate proportional to the average throughput since blocks are filled up to their maximum capacity.

7.4.4.3 Brokering Scenario Evaluation

The second experiment focuses on evaluating the capabilities of the system when dealing with the brokering scenario. To this aim, we consider 3 IBs managing a consortium of 1000 tenants, correspondingly. In light of the performances shown above, we select Kafka as consensus algorithm for its high fault-tolerance and scalability [52]. We assume that resource request values ρ, η, γ are drawn from a right-skewed distribution over a positive interval as resource requests must be non-negative. Such distributions are depicted at the top of Figure 7.10 for different demand ranges, spanning from 0.1% to 4% of the tenant initial resources. Note that since we assume the same distribution for all resource requests within the same slice, it is dubbed as SR probability density function (PDF). The bottom of Figure 7.10 illustrates the system behavior for a constant submission rate of $50\,\mathrm{SRs\,s^{-1}}$ so as to keep RW Conflicts to a minimum (around 2% of the submitted SRs). In such operational conditions, errors raise only in case of SR collisions. It is worth noting that SR collision rate increases along with the SR variance. Specifically, SR distributions with high variance leads to tenant satisfaction more quickly than with a lower variance. Additionally, the closer to tenant satisfaction, the lower the resource

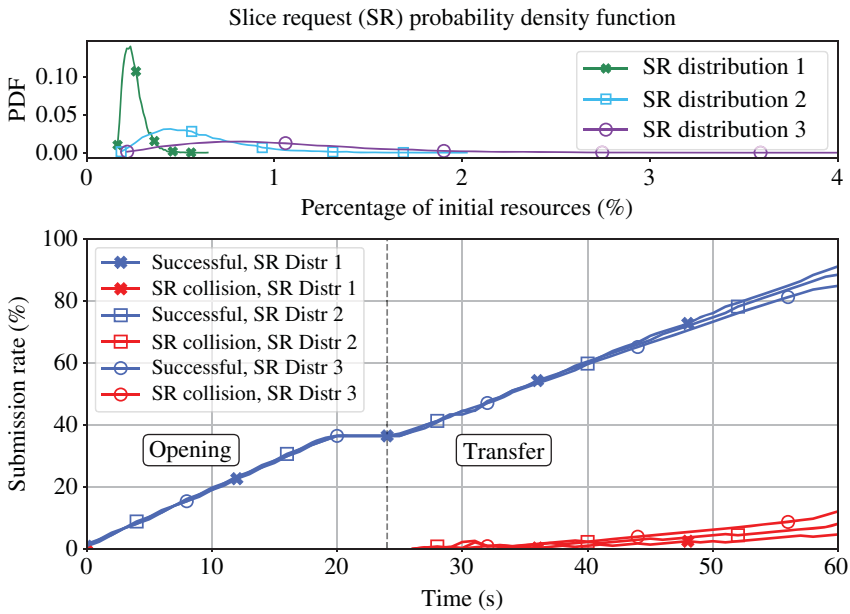


Figure 7.10 Transaction acceptance and error rates for different scenarios.

availability and, in turn, the smaller the likelihood of a request to be accepted by the system.

7.5 Concluding Remarks and Future Work

The blockchain technology has been identified as a key enabler for the development of novel business models. In this chapter, we investigated its applicability in the context of mobile networks focusing on resource management and network slicing use-cases, highlighting the main benefits that this technology can bring into both operational and management aspects of a mobile network infrastructure. As a case study, we introduced NSBchain, a blockchain-based brokering solution to interface the InP and network tenants willing to pay for acquiring, exchanging and managing network and computational slice resources within the domain of an IB. We remark that the applicability of blockchain in realistic mobile network scenarios must firstly overcome several inefficiencies and challenges. General blockchain solutions require the overall information contained into the chain to validate a new set of transactions. Over time, the size of the historical data may become too big therefore introducing significant communication and storage overheads into the system. At the same time, the high energy consumption imposed by complex validation process adopted by public blockchains limits their applicability to scenarios involving a limited number of nodes. Additionally, latency and scalability performances suffer significant communication overhead related to message broadcast and specific consensus algorithm implementations. Conversely from our analysis, we highlight that private blockchain solutions present better performances thanks to lighter validation approaches and to preliminary user authentication phases. However, they are not designed to provide the same levels of security as in the public blockchain case, nor to accommodate a massive number of users as those expected for 5G and beyond mobile networking.

Acronyms

API	application programming interface
CC	cloud computing
CDF	cumulative distribution function
CRAN	cloud radio access network
eNB	evolved node B
IB	intermediate broker
InP	infrastructure provider
IoT	Internet of Things

gNB	5G Node B
MANO	management and network orchestration
MBB	mobile broadband
MNO	mobile network operator
MEC	multi-access edge computing
mmWave	millimeter wave
MVCC	multi-version concurrency control
MVNO	mobile virtual network operator
NFV	network function virtualization
NF	network function
NR	new radio
NS	network slicing
NSB	network slice broker
NST	network slice template
OTT	Over-the-Top
PBFT	Practical Byzantine Fault Tolerance
PRB	physical resource block
PoC	Proof of Concept
QoS	Quality of Service
RAN	radio access network
RAT	radio access technology
ReLU	rectified linear unit
RW	read/write
SC	smart contract
SDN	software-defined networking
SLA	service-level agreement
SR	slice request
TEE	trusted execution environment
UE	user equipment
URLLC	ultra reliable low latency communication
VANET	vehicular networks
VM	virtual machine
VNF	virtual network function
VPF	virtual private network

References

1 Hewa, T., Gür, G., Kalla, A. et al. (2020). The role of blockchain in 6G: challenges, opportunities and research directions. *2nd 6G Wireless Summit*, March 2020, 1–5.

2 NGMN Alliance (2015). 5G White Paper. Public Deliverable.

3 Bega, D., Gramaglia, M., Banchs, A. et al. (2017). Optimising 5G infrastructure markets: the business of network slicing. *IEEE Conference on Computer Communications -INFOCOM*, May 2017, 1–9.

4 Rost, P. (2017). Network slicing to enable scalability and flexibility in 5G mobile networks. *IEEE Communications Magazine* 55 (5): 72–79.

5 Caballero, P., Banchs, A., de Veciana, G., and Costa-Pérez, X. (2017). Multi-tenant radio access network slicing: statistical multiplexing of spatial loads. *IEEE/ACM Transactions on Networking* 25 (5): 3044–3058.

6 Salvat, J.X., Zanzi, L., Garcia-Saavedra, A. et al. (2018). Overbooking network slices through yield-driven end-to-end orchestration. *Proceedings of the 14th International Conference on emerging Networking EXperiments and Technologies*, November 2018, 353–365. ACM CoNEXT.

7 Martini, B., Mori, P., Marino, F. et al. (2020). Pushing forward security in network slicing by leveraging continuous usage control. *IEEE Communications Magazine* 58 (7): 65–71.

8 Nguyen, D.C., Pathirana, P.N., Ding, M., and Seneviratne, A. (2020). Blockchain for 5G and beyond networks: a state of the art survey. *Journal of Network and Computer Applications* 166 (7): 32–39.

9 Sankar, L.S., Sindhu, M., and Sethumadhavan, M. (2017). Survey of consensus protocols on blockchain applications. *International Conference on Advanced Computing and Communication Systems*, January 2017.

10 de Vries, A. (2018). Bitcoin's growing energy problem. *Joule* 2 (5): 801–805.

11 Backman, J., Yrjölä, S., Valtanen, K., and Mämmelä, O. (2017). Blockchain network slice broker in 5G: slice leasing in factory of the future use case. *Internet of Things Business Models, Users, and Networks*, November 2017.

12 Wang, K. and Kim, H.S. (2019). FastChain: scaling blockchain system with informed neighbor selection. *IEEE International Conference on Blockchain*, July 2019, 376–383.

13 Qiu, C., Yao, H., Yu, F.R. et al. (2020). A service-oriented permissioned blockchain for the Internet of Things. *IEEE Transactions on Services Computing* 13 (2): 203–215.

14 Yang, H., Zheng, H., Zhang, J. et al. (2017). Blockchain-based trusted authentication in cloud radio over fiber network for 5G. *International Conference on Optical Communications and Networks*, 1–3.

15 Zhu, S., Hu, H., Li, Y., and Li, W. (2019). Hybrid blockchain design for privacy preserving crowdsourcing platform. *IEEE International Conference on Blockchain*, July 2019, 26–33.

16 Rawat, D.B. and Alshaikhi, A. (2018). Leveraging distributed blockchain-based scheme for wireless network virtualization with security and QoS constraints.

International Conference on Computing, Networking and Communications, March 2018, 332–336.

17 Ling, X., Wang, J., Bouchoucha, T. et al. (2019). Blockchain radio access network (B-RAN): towards decentralized secure radio access paradigm. *IEEE Access* 7: 9714–9723.

18 Zambianco, M. and Verticale, G. (2020). Interference minimization in 5G physical-layer network slicing. *IEEE Transactions on Communications* 68 (7): 4554–4564.

19 Gervais, A., Karame, G., Wüst, K. et al. (2016). On the security and performance of proof of work blockchains. *Proceedings of the ACM Conference on Computer and Communications Security*, October 2016, 3–16. ACM.

20 Lee, H. and Maode, M. (2020). Blockchain-based mobility management for 5G. *Future Generation Computer Systems* 110: 638–646.

21 Ortega, V., Bouchmal, F., and Monserrat, J.F. (2018). Trusted 5G vehicular networks: blockchains and content-centric networking. *IEEE Vehicular Technology Magazine* 13 (2): 121–127.

22 Ali, S., Wang, G., Bhuiyan, M.Z.A., and Jiang, H. (2018). Secure data provenance in cloud-centric Internet of Things via blockchain smart contracts. *IEEE SmartWorld, Ubiquitous Intelligence Computing, Advanced Trusted Computing, Scalable Computing Communications, Cloud Big Data Computing, Internet of People and Smart City Innovation*, 991–998.

23 Zhou, J., Cao, Z., Dong, X., and Vasilakos, A.V. (2017). Security and privacy for cloud-based IoT: challenges. *IEEE Communications Magazine* 55 (1): 26–33.

24 Gai, K., Guo, J., Zhu, L., and Yu, S. (2020). Blockchain meets cloud computing: a survey. *IEEE Communications Surveys & Tutorials* 22 (3): 2009–2030.

25 Oliveira, W., De Oliveira, D., and Braganholo, V. (2018). Provenance analytics for workflow-based computational experiments: a survey. *ACM Computing Survey* 51 (3): 1–25.

26 Ramachandran, A. and Kantarcioglu, M. (2018). Smartprovenance: a distributed, blockchain based dataprovenance system. *Proceedings of the 8th ACM Conference on Data and Application Security and Privacy*, 35–42.

27 Song, D.X., Wagner, D., and Perrig, A. (2000). Practical techniques for searches on encrypted data. *Proceeding 2000 IEEE Symposium on Security and Privacy*, 44–55.

28 Hu, S., Cai, C., Wang, Q. et al. (2018). Searching an encrypted cloud meets blockchain: a decentralized, reliable and fair realization. *IEEE Conference on Computer Communications*, 792–800.

29 Rebello, G.A.F., Camilo, G.F., Silva, L.G.C. et al. (2019). Providing a sliced, secure, and isolated software infrastructure of virtual functions through blockchain technology. *IEEE 20th International Conference on High Performance Switching and Routing*, May 2019.

30 Li, S., Zhu, K., Xu, Y. et al. (2019). Resource allocation for mobile blockchain: a hierarchical combinatorial auction approach. *IEEE Global Communications Conference*, December 2019, 1–6.

31 Ni, Z., Wang, W., Kim, D.I. et al. (2019). Evolutionary game for consensus provision in permissionless blockchain networks with shards. *IEEE International Conference on Communications*, May 2019.

32 Gai, K., Choo, K.R., and Zhu, L. (2018). Blockchain-enabled reengineering of cloud datacenters. *IEEE Cloud Computing* 5 (6): 21–25.

33 Zhao, G., Xu, H., Zhao, Y. et al. (2020). Offloading dependent tasks in mobile edge computing with service caching. *IEEE Conference on Computer Communications*, 1997–2006.

34 Guo, S., Dai, Y., Guo, S. et al. (2020). Blockchain meets edge computing: stackelberg game and double auction based task offloading for mobile blockchain. *IEEE Transactions on Vehicular Technology* 69 (5): 5549–5561.

35 Rosa, R.V. and Rothenberg, C.E. (2018). Blockchain-based decentralized applications for multiple administrative domain networking. *IEEE Communications Standards Magazine* 2 (3): 29–37.

36 Samdanis, K., Costa-Pérez, X., and Sciancalepore, V. (2016). From network sharing to multi-tenancy: the 5G network slice broker. *IEEE Communications Magazine* 54 (7): 32–39.

37 Zanzi, L., Albanese, A., Sciancalepore, V., and Costa-Pérez, X. (2020). NSBchain: a secure blockchain framework for network slicing brokerage. *IEEE International Conference on Communications*, June 2020, 1–7.

38 3GPP (2019). Management and orchestration; provisioning. TS 28.531, 3rd Generation Partnership Project, September 2019.

39 Neudecker, T. and Hartenstein, H. (2019). Network layer aspects of permissionless blockchains. *IEEE Communications Surveys & Tutorials* 21 (1): 838–857.

40 Rebello, G.A.F., Alvarenga, I.D., Sanz, I.J., and Duarte, O.C.M.B. (2019). BSec-NFVO: a blockchain-based security for network function virtualization orchestration. *IEEE International Conference on Communications*, May 2019.

41 Liang, L., Wu, Y., Feng, G. et al. (2019). Online auction-based resource allocation for service-oriented network slicing. *IEEE Transactions on Vehicular Technology* 68 (8): 8063–8074.

42 Jiang, M., Condoluci, M., and Mahmoodi, T. (2017). Network slicing in 5G: an auction-based model. *IEEE International Conference on Communications*, May 2017, 1–6.

43 Luu, L., Chu, D., Olickel, H. et al. (2016). Making smart contracts smarter. *ACM SIGSAC Conference on Computer and Communications Security*, 254–269.

44 Lin, C. (2019). HomeChain: a blockchain-based secure mutual authentication system for smart homes. *IEEE Internet of Things Journal* 7 (2): 818–829.

45 Luu, L., Narayanan, V., Zheng, C. et al. (2016). A secure sharding protocol for open blockchains. *ACM SIGSAC Conference on Computer and Communications Security*.

46 Wan, L., Eyers, D., and Zhang, H. (2019). Evaluating the impact of network latency on the safety of blockchain transactions. *IEEE International Conference on Blockchain*, July 2019, 194–201.

47 Castro, M. and Liskov, B. (1999). Practical Byzantine fault tolerance. *Proceedings of the 3rd Symposium on Operating Systems Design and Implementation*, 173–186.

48 Kreps, J., Narkhede, N., and Rao, J. (2011). Kafka: a distributed messaging system for log processing. *International Workshop on Networking Meets Databases*.

49 Ongaro, D. and Ousterhout, J. (2014). In search of an understandable consensus algorithm. *Proceedings of the USENIX Annual Technical Conference*, June 2014, 305–320.

50 Androulaki, E., Barger, A., Bortnikov, V. et al. (2018). Hyperledger fabric: a distributed operating system for permissioned blockchains. *Proceedings of the ACM 13th EuroSys Conference*, April 2018, 1–15.

51 Kuzlu, M., Pipattanasomporn, M., Gurses, L., and Rahman, S. (2019). Performance analysis of a hyperledger fabric blockchain framework: throughput, latency and scalability. *IEEE International Conference on Blockchain*, July 2019, 536–540.

52 Thakkar, P., Nathan, S., and Viswanathan, B. (2018). Performance benchmarking and optimizing hyperledger fabric blockchain platform. *IEEE International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems*, September 2018, 264–276.

8

Blockchains for Cybersecurity and AI Systems

Dragan Boscovic[1], Kasim Selçuk Candan[1], Petar Jevtic[2], Nicolas Lanchier[2], Sasa Pesic[2], and Axel La Salle[2]

[1] *School of Computing, Informatics, and Decision Systems Engineering, Arizona State University, Tempe, AZ, USA*
[2] *School of Mathematical and Statistical Sciences, Arizona State University, Tempe, AZ, USA*

8.1 Introduction

Cybercrime is such a vast and flourishing underground industry and makes it be a very real and disruptive threat to every profession, every industry, every company in the world. The U.S. defense pioneered the Internet as a medium to share valuable and detailed information across spatially dispersed groups. It also built the global positioning system (GPS) to enable location positioning service, and now Blockchain has entered the defense industry through a big door. According to Accenture, 86% of defense companies interviewed have developed plans to integrate blockchain solutions within the next three years, especially in cybersecurity [1]. Blockchain's adoption has been compared to the early rise of the Internet and has the potential to disrupt multiple industries, including healthcare, the public sector, energy, manufacturing, and financial services. Anticipation for Blockchain is to be the beating heart and the ultimate provider of a trusted data fabric for the new industrial revolution.

Blockchain established a reputation as a legitimate data safeguard for critical operations dependent on critical data exchange and processing. These companies use Blockchain's encryption and decentralization methods to improve data security and maximize privacy. The Blockchain-based solutions are gone beyond proof of concept to production pilots, with business cases being built to demonstrate the technology's benefits. Although present blockchain designs are not always capable of processing and recording a high volume of transactions, blockchain protocols are far more efficient in establishing the global state of the system. This is important because when all the distributed nodes in the blockchain solutions work

Blockchains: Empowering Technologies and Industrial Applications, First Edition.
Edited by Anwer Al-Dulaimi, Octavia A. Dobre, and Chih-Lin I.

with the same level of information, underlying protocols avoid making irrelevant or wrong decisions, and consensus builds up in Blockchain's trustless operations.

The decentralized nature of Blockchain makes it the perfect technology to use for cybersecurity applications. By design, the distributed nature of Blockchain provides no single point "hackable" entrance or point of failure that detrimentally exposes entire datasets.

Blockchain value lies in the fact that it shifts some of the trust in people and institutions to trust in technology. Blockchain adopters have faith in the cryptography, the protocols, the software, the computers, and the network. Nevertheless, this trust must be embedded in larger governance systems. Blockchain-based contracts that are an engine for many blockchain applications, including cybersecurity, require viable dispute resolution mechanisms, which will most likely be a combination of traditional legal recourse and arbitration, on the one hand, and novel decentralized approaches on the other. Consequently, blockchain solutions don't eliminate the need for institutions that act as sources of institutional trust that can't be replaced by technology alone [2].

Different trust models can be mapped onto four different architectures as the following [3]: (i) *peer-to-peer trust* arises when individuals trust each other based on societal morals and reputational systems; (ii) *leviathan trust*, named after Thomas Hobbes's conception of the state, equals to institutional trust that uses the government legal system to resolve disputes; (iii) *intermediary trust* exists when parties trust the intermediary and do not necessarily trust each other (a good example is the credit card system); and finally (iv) *distributed trust* refers to the trust in a decentralized system such as Blockchain that doesn't require trust in any of its individual components. It is that the distributed trust that should be considered by companies as an addition to their existing technology infrastructure to provide data availability, integrity, and confidentiality within their cyber-operations.

A fundamental test for cybersecurity protection capabilities of the distributed trust is its capability to offer additional tools to protect the security and privacy of users, and these must be addressed and tested if Blockchain is to become the real catalyst for social and industrial change that so many think it can be. The objective should be to determine to what extent does blockchain technology help or hinders cybersecurity. While still nascent, there is an ongoing innovation in Blockchain to enable enterprises to effectively deploy Blockchain to manage digital identities and maintain data integrity. Blockchains could potentially help improve cyber defense as the platform can secure, prevent fraudulent activities through consensus mechanisms, and detect data tampering based on its underlying characteristics of immutability, transparency, auditability, data encryption, and operational resilience (including no single point of failure). However, Blockchain's characteristics do not provide an impenetrable panacea to all cyber ills; to think the same would be naïve at best. Instead, as with other technologies,

blockchain implementations and roll-outs must include typical system and network cybersecurity controls, due diligence, practice, and procedures [4].

As an example, the Blockchain has the potential to assign and immutably record unique hashes to all downloads and software updates. In turn, this feature will allow users to compare the hash on their would-be download with the developer's hash and consequently reduce the chance of infecting their systems with fraudulent, well-disguised malware. Moreover, in cases where the system has been compromised, blockchain technology can be used to protect or shield cybersecurity traffic logs from surreptitious modification. Although Blockchain's underlying capabilities natively provide data confidentiality, integrity, and availability, its cyber defense or information system cannot be regarded as 100% secure, and classical cybersecurity controls and measures need to be deployed by organizations using blockchains within their technical infrastructure to protect their organizations from external attacks. Even more, whatever cybersecurity tools are deemed safe today won't be tomorrow, given the lucrative nature of cybercrime and the criminal's ingenuity to seek new methods of attack. As such, and any blockchain cybersecurity approach should follow what is called a "secure, vigilant, and resilient" (SVR) cyber-approach that not only supports entities to remain secure but also become more vigilant and resilient to evolving cyber threats.

Blockchain-based techniques are important components in a cybersecurity tool-box since they provide more security when compared with various classical database-driven transactional structures [5]. Blockchain supports **authentication** of devices and users without the need for a password, which in turn eliminates human intervention from the process of authentication, thereby eliminating it as a potential attack vector. A given organization can leverage a blockchain-based distributed public key infrastructure (PKI) for authenticating devices and users. Records on the issued certificates are managed on the blockchain, and this makes it virtually impossible for malicious players to utilize fake certificates. **Decentralized storage** is an inherent blockchain characteristic. User data is dispersed across computer nodes in the blockchain network, and any change to the data will need to satisfy certain consensus requirements among the participants rendering the data false. Finally, **traceability** feature gives companies the ability to trace back to a particular time period for every transaction and locate the corresponding party that initiated/signed the transaction. In turn, this enables the audit and transparency over every transaction recorded on Blockchain and offers companies a level of security through reassurance that the data hasn't been tampered with and is authentic.

This chapter will shed additional light on the cybersecurity vulnerabilities of both public and private blockchain networks. We go one step further and provide a mathematical model based on graph theory, which will enable practitioners in the field to assess cyber vulnerabilities for a given blockchain design.

As stated earlier, Blockchain's underlying capabilities natively provide data confidentiality, integrity, and availability. However, the information system based on the Blockchain cannot be regarded as 100% secure, and further in this chapter, we will talk about classical cybersecurity controls and standards needed to protect organizations using blockchains within their information management infrastructure. Internet of Things (IoT) is a fast-growing class of information technology (IT) solutions and is expected to scale to millions of connected devices deployed in a single network. This chapter dedicates a subsection to addressing the use of Blockchain to provide security, privacy, data protection, and accountability in favor of IoT systems. Because Blockchain is based on cryptographically secured, immutable distributed ledger technology and consensus-based decision making, it may enhance IoT ecosystems with more automated resource optimization and innate security. The last section in this chapter will look into the intersection between machine learning, artificial intelligence, and Blockchain, specifically, the use of blockchain trust fabric to enable federated machine learning (FML).

8.2 Securing Blockchains and Traditional IT Architectures

Blockchain's underlying capabilities natively provide data confidentiality, integrity, and availability. However, just like other systems, its cyber-defense cannot be regarded as 100% secure, and classical IT cybersecurity controls and standards need to be applied on underlining IT resources to protect blockchain infrastructure from external attacks.

Cybersecurity and Information Security are two terms that are often used interchangeably. They are so closely linked and focused on protecting the computer system from threats and information breaches that they may seem synonymous. The main difference between the two is that "not every data can be considered information." Only data that can be interpreted within a given context becomes meaningful and valuable information. As an example, *12202020* is numeric data and becomes information if it also represents the date in someone's calendar. Thus information means data that has some meaning. In short, cybersecurity deals with danger against cyberspace, while information security deals with the protection of a wide range of information from any form of threat that might alter the meaning or change its ownership. While information security is the foundation of data security, Cybersecurity, on the other hand, is the measure to protect digital communication infrastructure and the flow of information from attacks and damages. Nevertheless, often cyberspace can be hampered by inherent vulnerabilities present in the system by design that cannot be easily removed. A graphical representation of how information security and cybersecurity relate to each

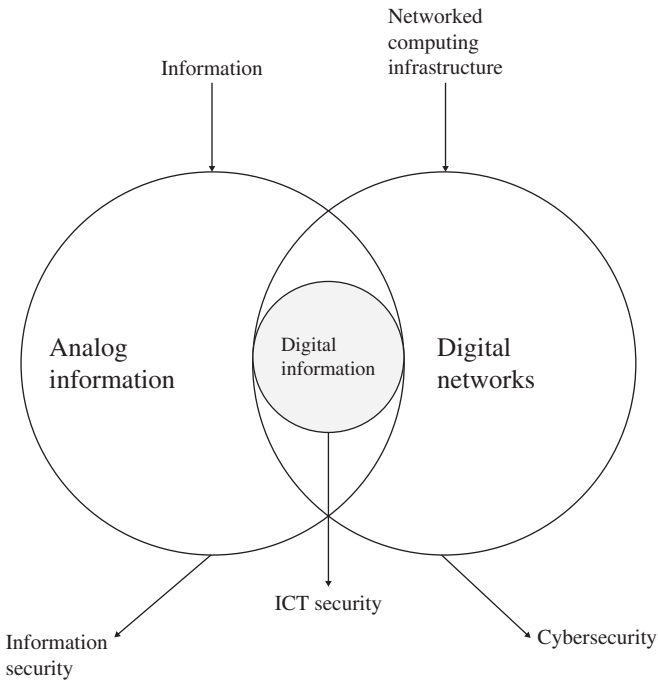


Figure 8.1 Information security vs cybersecurity.

other is displayed in Figure 8.1. ICT stands for information and communications technology, which is an extensional term for IT that defines the role of unified communications and the integration of telecommunications (basically digital communication security). Network security is a subset of cybersecurity, and it protects the data flowing over the network.

8.2.1 On Securing a Blockchain Platform

The blockchain is a peer-to-peer (P2P) framework with the capability to dis-intermediate central entities or processes within a given business workflow, hence improving efficiencies, and creating an immutable audit trail of transactions. It will transform business models from a human-based trust model to an algorithm-based trust model, which might expose participants to risks that they may have not accustomed previously [4]. To respond to such risks, participating entities should consider establishing a robust risk management strategy, governance, and controls framework.

Business logic in blockchains is captured in smart contracts that are self-executing code on the blockchain, and eliminate the need for manual

intervention to execute transactions. Often smart contracts need data from outside data sources referred to as "Oracles." Smart contracts can be used to facilitate, verify, or enforce rules between parties, allowing for straight-through processing and interactions with other smart contracts. Such software provides a large surface area for attack, so an attack on one smart contract could have a domino effect on other parts of the platform (i.e. the language itself or implementation of contracts).

Understanding blockchain and its associated risks may change and evolve as this technology continues to mature. It is therefore imperative for all organizations to continue to monitor the development of this technology and its application to various use cases. While blockchain technology promises to drive efficiency or reduce costs, it has certain inherent cybersecurity risks that can come from outside (via oracles) or from inside (e.g. distributed networking and distributed computing – smart contracts). Understanding these risks and implementing the appropriate safeguards is critical to fully benefit from this technology. These blockchain risks can be broadly classified into three categories: (i) **Standard risks** – blockchain technologies expose users to risks that are similar to those associated with current IT operations but introduce slight nuances for which users need to account; (ii) **Value transfer risks** – blockchain enables P2P transfer of value without the need for a central intermediary. The value transferred could be assets, identity, or information. This new business model exposes the interacting parties to new risks that were previously managed by central intermediaries; and (iii) **Smart contract risks**: Smart contracts can potentially encode complex business, financial and legal arrangements on the blockchain and could result in risk associated with the one-to-one mapping of these arrangements from the physical to the digital framework.

In public blockchains there is no network access control because the protocols used do allow anyone to access and participate in the network, with devices running adequate applications. In contrast, private blockchains require appropriate security controls to protect network access. In the context of the public blockchain, it would be tempting to assume that firewalls, virtual private networks (VPNs), virtual local area networks (VLANs), intrusion detection and prevention systems, and other safeguards in place in local, private networks provide adequate protection through the adoption of a so-called "defense in depth" strategy. However, experience teaches us that relying solely on the effectiveness of network-level cybersecurity controls is clearly insufficient. For this reason, best practices recommend that security controls (such as access controls) should also be implemented directly at the application level because that is the first and most important line of defense against an attacker gaining access to the local network or where a malicious insider is already present [4]. To address the security challenges, blockchain infrastructure operators should implement an overall cybersecurity program that

includes a proper governance framework with roles, processes, accountability measures, and performance metrics. In line with these security requirements, blockchain can include advanced security controls. For example, blockchain can leverage the PKI (a set of roles, policies, and procedures required to create, manage, use, store, and revoke digital certificates and manage public-key encryption) to authenticate and authorize parties and also encrypt their communications.

Every transaction on a public or private blockchain is digitally signed and timestamped, allowing participants to trace each transaction to a specific time period and, if necessarily, identify the corresponding party (via their public address) on the blockchain. This feature relates to an important information security property: non-repudiation, which is the assurance that someone cannot duplicate the authenticity of their signature on a file or the authorship of a transaction that they originated. This out-of-the-box native functionality of the blockchain increases the reliability of the system through the detection of tamper attempts or fraudulent transactions because every transaction is cryptographically associated with a user. Any new transaction added to a blockchain will change the global state of the ledger. As such a fully traceable history log is available because every new iteration of the system also stores the previous state. The audit capability of blockchain provides organizations with a level of transparency and security over every interaction. From a cybersecurity perspective, this provides entities with extra assurance that the data is authentic and that no tampering has occurred.

Performance of the blockchain protocols is not only determined by energy efficiency but also by their resilience to practical attacks. Evaluating the resiliency of blockchain solutions against cyberattacks is a complex problem that needs to be answered through formal math modeling and informed risk assessment [6]. In this chapter, we will look into most common and dangerous currently known blockchain attacks and discuss adequate defense mechanisms specific to a given blockchain protocol. Cybersecurity issues and attacks on blockchain protocols are in-depth explained in Sections 8.3 and 8.4.

8.3 Public Blockchains Cybersecurity

Blockchain has prominent applications in many industries, from agriculture to accounting. Activities associated with blockchain technology can be classified into three categories from the standpoint of accessibility and organization: (i) the first-generation public blockchain (1.0), (ii) the second-generation public blockchain (2.0), and (iii) the third-generation private blockchain (3.0) [7]. Blockchain 1.0 deploys cryptocurrencies in applications related to financial transactions, such as currency transfers, settlements, and digital payments. Blockchain 2.0 includes smart contracts for economic markets and financial applications.

This category handles more than simple cash transactions. It includes stocks, loans, mortgages, and smart contracts. The third category applies to applications beyond finance and markets. It includes areas, such as government, health, science and art. In this section 1st- and 2nd-generation public blockchains will be analyzed.

8.3.1 Vulnerabilities Categorization

Figure 8.2 outlines the categories of public blockchain issues and vulnerabilities from security, regulative, and technical standpoints. Thus, this section will inspect 4 categories of such issues: (i) technical limitations, (ii) legal liabilities, (iii) connected 3rd -party apps, and (iv) cybersecurity issues.

8.3.1.1 Technical Limitations, Legal Liabilities, and Connected 3rd-Party Applications

When discussing the technical limitations of public blockchains three important properties of such blockchains need to be discussed: block size trade-off, distributed storage mechanisms, and honesty-based consensus. The size of individual blocks on a blockchain can have a potentially large impact on the speed and capacity of the network, but there are always trade-offs. Larger blocks could not only improve capacity and speed but also push down fees. On the other hand, larger blocks could also lead to greater centralization and network latency.

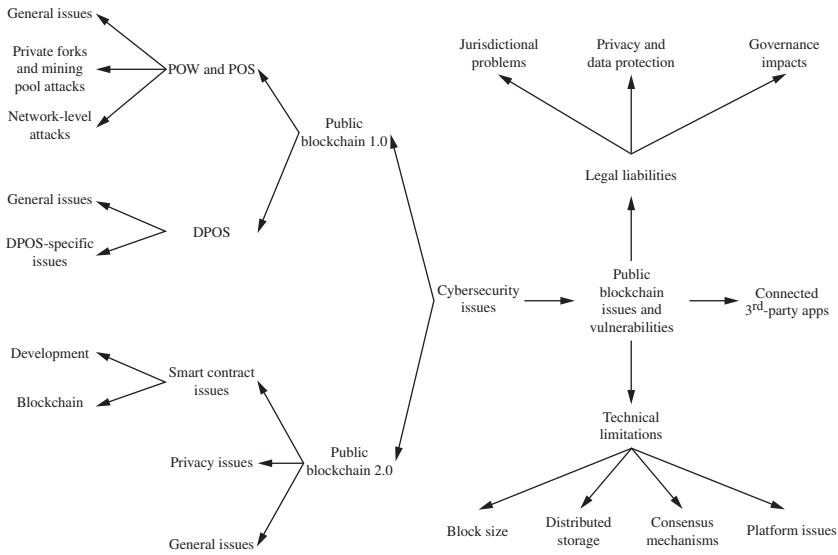


Figure 8.2 Public blockchain issues categorization.

Since blockchain data is stored on every node, a data theft attack can come at a much larger scale and with many alternatives for the attack destination. Another blockchain inherent property is the nature of the consensus mechanisms: they are based on an assumption that the majority of nodes is honest to run and maintain the system. Once a group of attackers has obtained at least 51% of the hashing power inside a blockchain network, their aggregate computing power is sufficient to jointly launch an attack with objective to tamper with the content in blocks and conduct disruptive attacks such as DDoS. Finally, we must not forget about the vulnerabilities of hardware and software platforms that run the blockchain nodes.

Regulations and law around public blockchains as well as observing and deducing legal liabilities are an increasingly important subject as the technology penetrates both the industry and government sectors. We will briefly discuss the most obvious issues: (i) jurisdictional problems, (ii) privacy and data protection, and (iii) governance impacts. Public blockchain nodes can easily span multiple geographical locations around the globe. Because of that it is often challenging to determine which jurisdictions' laws and regulations apply to a given application. We currently see a huge range of opinions from regulators on cryptocurrencies, absolute distrust and bans in some countries to more cautious investor warnings from others, while yet other countries have introduced programs to attract more crypto activity [8]. The challenges of privacy and blockchain technology have been intensely discussed in the past five years, especially since the emergence of privacy regulations around the globe such as the General Data Protection Regulation (GDPR [9]), California Consumer Privacy Act (CCPA) and many more in countries like India, Chile, New Zealand, Japan, and Thailand. The problem with permissionless public blockchains is that no single entity is responsible for the security of the system, while all users can have access to ledger data. These attributes are directly conflicted with thrust of many of the above-mentioned privacy laws which require the party controlling personal data of an individual to safeguard the security and privacy of that data on behalf of the individual as well as enable the *right to be forgotten*. Many organizations will also require their data to be stored in a certain region, or eventually, transfer data for privacy law purposes. Regulators could tackle accountability from different standpoints. Some might follow the approach where all participants of a public blockchain are equally accountable. Others, like France, might suggest that only those actively participating in the system and inputting data, disregarding passive nodes and miners, are accountable [10]. As taxing compliance goes, cryptocurrency transactions might be taxed as assets on a capital gains basis without applying VAT.

When discussing services that connect to the blockchain, one needs to mention two distinctive types: (i) Integration and development services and (ii) External data provider services. Integration and Development services include connected systems' endpoints and external 3rd-party solutions. Endpoints are services

or computers that regular users or enterprises use to access blockchain-based services. External, 3rd-party solutions are often used to move data to/from the blockchain. Weak security on 3rd systems (e.g. flawed code) can expose their clients' blockchain credentials and data to unauthorized parties. Distributed ledgers and smart contracts bring remarkable innovation by eliminating trust-related friction in human-to-human interactions, but until they can trustlessly take off-chain input data, the innovation will be limited. Thus, they absolutely require secure and trustworthy external data provider services. Nowadays, blockchain smart contracts base a fair percent of decision-making on such services called Oracles. Decentralized oracles are gateways for smart contracts to interact with the outside world, while at the same time limiting their reliance on single source of truth [11]. Although allowing blockchains to take inputs from outside the chain is a security risk, it is essential for building robust and far-reaching use-cases. The level of automation that we can achieve with blockchain technology is directly impacted by the level of trust we have in blockchain Oracles.

8.3.1.2 Cybersecurity Issues

Public blockchain cybersecurity issues can be divided into two major categories, as we also discussed this categorization above: (i) blockchain 1.0 (focusing on PoW [Bitcoin] and PoS [Ethereum]) and (ii) blockchain 2.0 (focusing on smart contracts [Ethereum]). Blockchain governance models weigh heavily as some of the most critical factors of a platform such as latency and throughput rest upon their choice. Some of the most popular governance models that every blockchain enthusiast will have heard of include Proof of Work (PoW), Proof of Stake (PoS), and Delegated Proof of Stake (DPoS). These are the most widely used consensus algorithms in the blockchain space. Thus, we will further divide the blockchain 1.0 category based on the governance model standpoint: (i) those running on PoW/PoS and (ii) those running on DPoS (e.g. BitShares, EOS).

8.3.1.3 Public Blockchain 1.0: PoW and PoS

For blockchains using PoW most common **general issues** include the infamous 51% attack, double spending, private forks, and the Sybil attack. For blockchains using PoS these include the nothing-at-stake, grinding, long-range and past-majority attacks. Both PoW and PoS are susceptible to bribery attacks, de-synchronization attacks, transaction denial attacks, end-user vulnerabilities such as cryptocurrency, wallet, or private key theft.

A 51% attack on a blockchain network refers to the success of a single entity or organization in gaining control over the majority of the hash rate, thus causing a network turmoil. These attacks are most likely to come in a form of a mining pool attack: a consolidated and coordinated attack of multiple nodes. Susceptibility to 51% attacks is inherent to most cryptocurrencies. For popular blockchains a

51% attack will be very expensive: about US$550.000 for Bitcoin, US$360.000 for Ethereum, etc. (per hour of attack) [12]. As a result of a 51% attack there can be **private forks** to discredit the main chain and cause a drop in its value and revenue leading to nodes leaving the main chain and joining the private fork. Due to the fact that most of public blockchains are based on PoW for mining, enough computing power allows a node to create a fork and make it the authoritative version of the chain, thus enabling **double-spending**. The objective of the double-spending attack is to issue a transaction, e.g. a payment from an adversarial account holder to a victim recipient, have the transaction confirmed, and then revert the transaction by, e.g. including in the ledger a second conflicting transaction [13]. Double-spending can lead to inflation, thus devaluing the currency relative to other monetary units or goods while diminishing user trust and the retention of the currency.

In **bribery attacks** [14], an adversary deliberately pays miners/shareholders (through cryptocurrencies or fiat money) to work on specific blocks and forks, aiming at generating an arbitrary fork that benefits the adversary (e.g. by supporting a double-spending attack). Miners of PoW-based cryptocurrencies do not have to own any stake in order to mine blocks, which makes this attack strategy feasible. In this setting, the adversary offers a bribe higher than the block-mining reward. **Transaction denial attacks** refer to a situation when the adversary wishes to prevent a certain transaction from becoming confirmed. For instance, the adversary may want to target a specific account and prevent the account holder from issuing an outgoing transaction. In a **desynchronization attack**, a node/shareholder behaves honestly but is nevertheless incapable of synchronizing correctly with the rest of the network. This leads to ill-timed issuing of blocks and being offline during periods when the node is expected to participate. Such an attack can be mounted by preventing the party's access to a time server or any other mechanism that allows the synchronization between P2P parties. Moreover, a desynchronization may also occur due to exceedingly long delays in message delivery.

Cryptocurrency thefts typically involve exploiting vulnerabilities in connected systems. This can happen due to a wallet holding the funds being hacked or a user's private key is stolen. This will allow the attackers to drain specific users' account balances, but the blockchain itself remains intact. These issues are specific because they are caused by bad private key management or wallet control mechanisms by vendors and users themselves. Individuals may lose their private keys, resulting in the loss of blockchain-stored digital assets because private keys are not reproducible by design. Thus, end users must understand and protect the private keys they hold on their systems or other pluggable media (e.g. hardware wallets). Service providers (e.g. digital wallet providers) have emerged to provide key management services to minimize end users' risks. However, these services heavily depend on passwords, device authentication (e.g. SMS or two-factor

authentication), or other similar mechanisms. Because they involve human interaction, these mechanisms are vulnerable unless individuals and organizations take due care. Lastly, by penetrating the above-mentioned authentication mechanisms users can perform attacks of impersonation, phishing, malware, etc. thus infiltrating blockchain applications and their users.

Private forks and mining pool attacks come in multiple shapes: there is selfish mining (the above-described mining pool attack scenario), block withholding, the possibility of a Sybil attack, etc. The types of attack threaten the factor of decentralization inside a public blockchain as well as its trustworthiness and reputation. For example, currently, 3 countries host over 50% of the world's Bitcoin nodes [15]. However, the situation is a bit more alarming: over 50% of Bitcoin network's hashing power rests in a single country: China. The concentration of mining power in countries like China is partially due to cheaper electricity prices. This threatens to subvert the democratic nature of public blockchains. Giant mining pools and the other massive Bitcoin-mining conglomerates can effectively monopolize control over the Bitcoin blockchain. This may lead to network centralization and the possibility of collusion, making the network vulnerable to changes in policy on electricity subsidies. On another hand, in a **Sybil attack**, the attacker debases the reputation system of a network by creating a considerate number of fake identities and while using them manages to gain a disproportionately large influence. In a public blockchain, a Sybil attack can be carried out by well-financed attackers by creating thousands of nodes and inserting them into the network in many places. When an attack takes place, these nodes (Sybil) can concentrate on propagating the attacker's blocks alone (block withholding) or refusing to receive blocks, effectively blocking other users from a network.

The existence of the **nothing-at-stake** problem is due to the nature of PoS. In the event of a fork, whether the fork is accidental or a malicious attempt to rewrite history and reverse a transaction, the optimal strategy for any miner is to mine on every chain, so that the miner gets their reward no matter which fork wins [16]. Stakeholders have an incentive to act properly in a stake on the longest chain in order to preserve the value of their investment. When attempting a **grinding attack**, the adversary tries to influence the slot leader selection process to improve its chances of being selected to generate blocks (which can be used to perform other attacks such as double-spending). The PoS architecture is also vulnerable to a **long-range attack**. An attacker who wishes to double-spend at a later point in time can mount a long-range attack by computing a longer valid chain that starts right after the genesis block where it is the single stakeholder actively participating in the protocol [17]. Even if this attacker owns a small fraction of the total stake, it can locally compute this chain generating only the blocks for slots where it is elected the slot leader and keep generating blocks ahead of current time until its alternative chain has more blocks than the main chain. **Past majority attacks**

can be launched by exploiting old and unused accounts on a PoS blockchain. As stake moves, our assumption is that only the current majority of stakeholders are honest. This means that past account keys (which do not hold any stake at present) may be compromised. This leads to a potential vulnerability for any PoS system since a set of malicious shareholders from the past can build an alternative blockchain exploiting old accounts and the fact that it is effortless to build it.

Network-level attacks in this category include Distributed Denial of Service (DDoS) attack and the Eclipse attack. **DDoS attacks** are one of the most common network bandwidth consumption attacks. It results in slowing down the system by creating numerous transactions to transfer assets through and forth between the attacker's pre-created wallets. The slowdown is caused by the need to process all transactions resulting in increased processing times for legitimate network users. In general, blockchain networks show strong resistance to DDoS attacks. Because of that, DDoS attacks are more popularly launched against cryptocurrency exchanges, gambling operators, wallets, and blockchain-based financial services. An **eclipse attack** aims to obfuscate a participant's view of the P2P network, in order to cause general disruption, or to prepare for more sophisticated attacks. It is generally aimed at a single target for which the malicious actor will ensure that all of the target's connections are made to attacker-controlled nodes. The attacker would first overwhelm the target with its own IP addresses, which the victim will likely connect to upon the restart of their software. A restart can either be forced (i.e. with a DDoS attack on the target), or the attacker can simply wait for it to occur. Eclipse attacks may sound similar to Sybil attacks. However, their end goal is ultimately different – an eclipse attack aims at a single node (for reasons explained in a later section), while a Sybil attack is a network-wide attack designed to trick the reputation system of the protocol.

8.3.1.4 Public Blockchain 1.0: DPoS

DPoS became popular in the cryptocurrency market due to its high scalability [18]. Its functions are similar to those of PoS, but it is different in that it has more democratic features when compared to PoS (e.g. delegated witnesses). DPoS consensus relies on a fixed number of elected parties called block producers (BPs). BPs are nodes selected to create blocks and do so in a round-robin order. BPs are limited in number and voted by the users of the network – number of tokens is proportional to the number of votes the user gets.

In aspects of security, there are some concerns around DPoS. The main reason is its ability to provide a high level of scalability at the cost of **limiting the number of block producers**. In DPoS blockchains one party can participate in the system in a combination of roles (usage, validation, and block production). Therefore, networks running on DPoS will be unable to repel Sybil attacks, where a single user creates multiple identities to take advantage of the network for own profits.

Furthermore, DPoS networks remain at risk of having a group of dishonest nodes colluding to hijack the voting process and following, voters colluding for heinous purposes. Because a DPoS network manages a relatively small number of BPs, theoretically, it is much easier to organize collusion among them. In such a case, an attacker might launch a censorship attack, a double-spending attack or change system parameters.

In a system, such as DPoS, that is naturally designed to promote plutocracy and collusion between BPs, there is little-to-no guarantee for application developers that their transactions (or entire applications) will not be censored due to this collusion. A **censorship attack** against a DPoS network means that BPs might refuse to process valid transactions. This will not cause a problem if a minor group of BPs censors an individual – the next honest majority of BPs will probably validate this transaction in the next block, thus causing simply a small delay in transaction validation. However, if majority of BPs is under the attacker's control the attack will certainly be a success. Since inside a DPoS network all changes must be set in motion by participating stakeholders [18], this allows the attacker to **change important parameters** such as block rewards or fork out certain stakeholders.

A potential problem of **creating voting centralization** can still be observed in DPoS – since for small-stake users a vote may be costly and they will rely on proxy voting, majority of votes will come from high-stake users (whales, wallet providers, exchanges). People's vote strength is determined by how many tokens they have, which means that people who own more tokens will influence the network more than people who own very few.

8.3.1.5 Public Blockchain 2.0: Ethereum Smart Contracts

Smart contract technology is intimately related to blockchain technology. These contracts are merely coded (business) logic. Given the current state of a ledger and external information (usually collected through Oracles), smart contracts effectuate new changes on a blockchain or act through actuators (cyber-physical systems) in the physical world. The oldest definition of smart contracts states they are "a set of promises, specified in digital form, including protocols within which the parties perform on these promises" [19]. In engineering terms, smart contracts are self-executing scripts that can be coded in high-level programming languages and are running on a blockchain platform.

Apart from 51% attacks, smart contracts bring a new level of security weakness to blockchain systems. Smart contracts leave spaces for typical software bugs, misuse, code flaws, etc. Severity of smart contract vulnerabilities increases dramatically in public blockchains due to the inherent cryptocurrencies tied to the blockchain. Exploiting smart contract flaws on public blockchains can lead to cryptocurrency theft as it has in the famous DAO hack on Ethereum [20].

Because of smart contracts, blockchains that leverage them are unique when observed from a vulnerability management point of view since flaws in smart contracts cannot be fixed with a patch of some sort, as in traditional computer systems. As transactions on the blockchain cannot be undone, as is the deployment of smart contracts – once they are up and running it is impossible to dispose of them. Although they cannot be killed, per say, there are ways to mitigate smart contract issues. One would be to deploy additional smart contracts to interact with the flawed ones. Smart contract developers can build *kill switches* in their smart contracts, a certain function that is called when an attack is detected – however, the funds that were stolen are unrecoverable. The only way to recover stolen funds is to rewrite the blockchain, starting from before the attack was launched – if the majority of blockchain community agrees (e.g. Ethereum Classic was created in this manner when not all nodes wanted to rewrite the blockchain after the DAO hack [20]).

On Ethereum, tens of thousands of Solidity smart contracts may contain some kind of vulnerability, while the number of smart contract vulnerabilities reaches 65 000+, according to research conducted in 2017 at AnChain [21]. Smart contract vulnerabilities have been extensively researched, and tools around for detection and impact analysis built from both academic and industrial sides. These analyze either the smart contract source code or the EVM bytecode to perform automated vulnerability detection and optimization [22], extract function call and code execution flow graphs and look for known security vulnerabilities [23] (e.g. re-entrancy, transaction order dependency). In 2019 Perez and Livshits identified that there are roughly 21 000 smart contracts on the Ethereum platform with some kind of vulnerability [24].

Per Figure 8.2, Ethereum smart contract issues will be observed from two points of view: (i) Development and (ii) Blockchain issues. For a deeper technical dive, the most complete and up-to-date overview of Ethereum vulnerabilities is presented by Chen et al. [25].

(i) **Development issues** are formed of coding issues and bad coding practices. These include issues with the high-level programming language at hand (we consider Solidity) and the lower-level EVM bytecode. Solidity security issues directly lead to exploits by a nefarious user smart contract or account. The list of most common vulnerabilities in this category is considerable: unchecked external calls, re-entrance, gasless send, improper transfer of funds, DoS by external contract, costly loops and in-loop transfers, overpowered owner, arithmetic precision and overflow, types conversion, frozen funds, etc. The ones mentioned in the previous sentence can be considered the most dangerous ones. For example, DoS for a smart contract can be caused by an external contract. This can occur if conditional statements inside a calling contract depend on successful execution of an external

function call since: its failure (can be permanent if coded badly) can cause the caller contract function to fail (also permanently). Next, the re-entrance issue is the cause of the infamous DAO hack [20]. Loops have to be used cautiously, since they can be manipulated to become infinite (by changing the array size).

As for **(ii) Blockchain-related issues** when discussing Ethereum smart contracts, there are a few worth underlining. Firstly, since smart contracts run on multiple nodes, and these nodes can produce different timestamps, it is discouraged to use them inside contracts. Instead, environmental variables such as the *block.timestamp* can be used to reference a certain point in time. Theoretically, the timestamp, but also blockhash and other miner-defined values can be manipulated by miners when carrying out an attack on the blockchain on a different level. Thus, writing **timestamp-dependent functions** is discouraged. Secondly (and similarly), generating random numbers inside a contract is a hard problem due to the deterministic nature of the Ethereum (and other) public blockchain. Since a transaction needs to be verified by many nodes, **generating random numbers** must, in a way, be consistent. A possible approach for generating random numbers is using block variables (blockhash, number coinbase, timestamp). However, tread carefully: as explained above, theoretically, these can be manipulated by a malicious miner. Thirdly, there is the issue of **unpredictable states** in contracts. The state of contracts' variables and balance at the time of sending the transaction can indeed change by the time its execution is carried out (other transactions changed its state meanwhile).

8.3.1.6 Public Blockchain 2.0 – Privacy Issues

For public blockchains, the high level of transparency and trust is a direct consequence of low privacy. But for usage in enterprises, a certain level of privacy will most likely be mandatory – entire transactions might need to be kept private for designated users or a part of data inside a transaction needs to be conserved. The lack of transactional privacy is an issue that occurs due to the fact that all public keys are visible by all public blockchain network participants. If privacy is highly regarded for a certain use-case, the system can instead be built using semi-private or private blockchain platforms instead. If it is insisted that the solution is based on Ethereum, industry-grade Ernst and Young's Nightfall protocol [26] or research-grade Phantom protocol [27] for private transactions on Ethereum using zk-snarks are great examples of privacy solutions for Ethereum.

Bitcoin relies on an Unspent Transaction Output (UTXO) model and creates a new address per payment. Ethereum, on the other hand, relies on a user-based model, keeping track of user's Ether. Using such a model has created some unwanted ramifications on privacy. By analyzing user's transactions, an external entity can deduce a user's timezone by analyzing time periods when the user

is most active. Also, since most users rarely change their preferred gas price setting, users with adjusted gas price setting can be easily identified and tracked across the blockchain. Furthermore, by analyzing transactional patterns one's identity can be linked to one's address. Many Ethereum users nowadays use the Ethereum Name Service (ENS) [28]. Ethereum users leverage ENS to customize their addresses, by for example, adding their names, physical addresses, etc. in order to be more easily identifiable on the blockchain. Furthermore, many users publicly reveal their ENS names on their social media profiles, which can again be mined for data by leveraging the social media platform APIs. So, careless usage of a public blockchain platform by its users can indeed cause significant privacy issues.

8.4 Private Blockchains Cybersecurity

In the business context, private or permissioned blockchain solves two main issues when it comes to cross-organizational collaboration. First, the lack of private and shared ledger of mutual transactions that is tamper proof, reliable, and trusted by all the parties. Second, the lack of trust between business organization or entities, in the context where collaboration is desired and resolution should come without a trusted third party. Permissioned blockchain solutions are, at present, available via frameworks such as Hyperledger Fabric, Corda, and Quorum but also Ethereum can be setup as a private network. All those solutions have their own strengths and weaknesses, cyber vulnerabilities, and intended communities of use. Since 2015 the Linux foundation has spearheaded the collaborative open effort to develop enterprise-grade blockchain technology. The effort proved to be a big success – in the business context, the most highly used private blockchain solution today is its Hyperledger Fabric (HLF) [29]. The main feature of HLF is its modular and configurable architecture offering parallel execution of transactions with significant increase of throughput compared to the public blockchains. Currently, more than 20 companies use HLF technology boasting names such as Allianz SE, Amazon, BNP Paribas, Intel, Microsoft, Siemens, State Farm, Visa, and Walmart [30]. This is why, as a case study to categorize private blockchain cybersecurity vulnerabilities, HLF shall be observed.

8.4.1 Hyperledger Fabric Architecture

Given the amount of economic value that currently resides on HLF business solutions, it is apparent that cyber-risk and its potential liabilities for the above companies and other businesses, when deploying HLF technology, are essential

to ascertain. The starting point in this process is understanding potential HLF cyber-vulnerabilities that may arise from HLF's unique architecture comprising key components and protocols.

When it comes to critical roles or components in the HLF framework, the most important are Clients, Membership Service Providers (MSPs), Peers, Ordering service, Channels, and Ledgers [31]. The peers are owned by organizations that use HLF to conduct collaborative business. Through APIs, the peers serve as intermediaries between an organization's client applications and the blockchain ledgers and chaincodes (smart contracts) that peers host. Channels represent logical structures that encapsulate shared information and processes between (subsets) of organizations participating in an HLF application. Within a channel, organizations can share ledgers across multiple peers as well as access to multiple shared business logics implemented via smart contracts. The job of the Ordering service is to order transactions into blocks within a channel. MSPs credential the clients and peers which identify them on an HLF network.

When it comes to transaction flow in HLF, there are three distinct phases: (i) execution phase, (ii) ordering phase, and (iii) validating phase. In the (i) endorsing phase, the client application accesses an endorsing peer, submits a transaction proposal, and waits for a reply. Across a channel, the endorsing peers of participating organizations execute transaction-related chaincode. Along with other information, they cryptographically protect the results of these operations accompanied with their endorsement. Upon receiving sufficient numbers of verified endorsements, the client sends a transaction to the Ordering service, thus beginning the (ii) ordering phase. In turn, when the Ordering service collects a sufficient number of transactions from various clients across a channel, the block of transactions is created and broadcasted to committing peers across a channel for storage. Finally, upon receiving a block for the Ordering service, peers check, verify, and eventually commit blocks to a ledger in the (iii) validating phase.

8.4.2 HLF Vulnerabilities Categorization

HLF vulnerabilities, and by extension other private blockchain vulnerabilities, can be first divided to (i) External attacks and (ii) Internal issues.

HLF **external attacks** can be launched against system endpoints and physical machines hosting the network nodes. Endpoint attacks refer to flaws in libraries, SDKs, or 3rd-party apps that communicate with the blockchain. System-wide penetration tests can reveal some of these issues. Attacks on physical machines hosting HLF nodes come in many forms: attack on the utilized containerization technology (Docker, Kubernetes), network attacks, etc. External attacks are, thus, not in the HLF domain, but, nevertheless, they cannot be avoided.

Internal vulnerabilities are specific to the blockchain protocol, in this case HLF. A good starting point to analyze them are the elements unique to HLF and their inherent vulnerabilities [32]. There are four categories of important internal issues: (i) Configuration, (ii) Consensus and Endorsement, (iii) Membership and access, and (iv) Chaincode.

Every blockchain's lifecycle starts with the **configuration** of network. For HLF this includes specifying organizations, peers, consensus, block creation, system-wide policies (e.g. for channel creation), etc. Poor design of an HLF network architecture can lead to poor performance, lower transaction throughput, etc. ultimately causing the blockchain to deteriorate significantly, making it impractical or impossible to use. HLF supports issuing configuration updates; however, to be accepted a consensus has to be reached within the network, leaving a small window for a malicious organization to invalidate the update.

Consensus ensures the full-circle verification of the correctness of a set of transactions comprising a block. Consensus is the responsibility of Ordering service node. Since Ordering service, like MSP, has an essential and centralizing role in HLF, it is a large part of its cyber-risk surface. Possible attacks on this component are Sabotage attacks, Intentional fork attacks, Block size, Batch ordering, and Transaction reordering attacks [32]. The essence of these attacks lies in the process of block-creation. For instance, in a sabotage attack during block creation, transactions from an organization can be excluded. In intentional fork attacks, a different block can be sent to various parties when the same block would be appropriate. In the block size attack, the change of block size is effectuated by a malicious actor, thus affecting the performance of the blockchain system and ledger formation. Similarly, in a batch time-out attack, a malicious actor changing the time for cut-off in block formation can change the performance of a blockchain system [32]. Also, the network delays can have an adverse impact on block formation and have an effect similar to the above attacks [33]. Finally, in the case when there is an attack on Ordering service and consistency of ledger is violated, important questions are related to attribution of accountability of misbehaving parties [34]. This is not a trivial task and requires formal analysis and moving beyond standard Ordering service protocols.

Endorsement plays a big role in the transaction flow, and by extension, the consensus mechanism. When it comes to endorsing peers, whose identities are known inside a channel, the so-called Malleability attack, where transactions of a particular client are modified or blocked, is possible [35]. Also, in a so-called Wormhole attack, a malicious peer can create a private network with the outside channel's parties and leak information to external parties.

Membership and access control issues occur within a HLF network if the authority providing it is tampered with. When the MSP becomes malicious,

the Sybil, Boycott, and Blacklisting attacks can ensue. In a Sybil attack, newly created malicious endorsing peers can create an endorsing majority for the attacker [32]. This can be mitigated by introducing randomness in endorsing peer selection process [36]. In a Boycotting attack the malicious MSP can prevent certain organizations' certification within the network, thus effectively boycotting them. Similarly in the effect, a malicious MSP can remove a certificate of a node from a trusted list of certificates or add it to a list of certificates that are revoked, thus blacklisting them [32].

When it comes to smart contracts and their HLF-specific vulnerabilities, the docker containers, where HLFs chain codes are run, are an essential point of departure [37]. Specific attacks include code injection, log injection, chaincode sandboxing, and remote inputs. For an attacker whose aim is to gain unauthorized privileges, the crucial aspect of all these vulnerabilities comes from exploiting smart contracts' design and their interaction with and limitations of containers.

Finally, when it comes to cyber vulnerabilities of overall and supporting HLF infrastructure related to storage, cryptographic, and network protocols, a good overview is given by Putz and Pernul [38].

8.5 Modeling Blockchain Vulnerabilities Using Graph Theory

The emerging trends in blockchain vulnerability modeling include the use of Petri nets. When it comes to the modeling of smart contract interactions and loss modeling due to contagious cyber-attacks, a random graph theory coupled with bond percolation can be applied.

8.5.1 Petri Nets

A discrete-state and event-driven system where states change conditional on the arrival of discrete events over time are called Discrete Event Systems (DESs) [39]. When it comes to mathematical modeling languages of discrete-event systems, the Petri Net models take a prominent role. Initially to model systems with interacting concurrent components they were developed by Carl Adam Petri in 1962 [40].

A Petri net is a directed bipartite graph with two main types of vertices: places and transitions [41]. A place represents a passive component model, a state, storage, or a logical condition. Visually, the places are represented via circles or ellipses. Squares or rectangles represent transitions that model active components representing production, the transformation, or consumption of things. The arcs connect places with the transitions and are represented with directed edges.

The arcs can only connect places with transitions or transitions to places, with no other possibilities. Finally, tokens model the objects or values in the real world. Their distribution across Petri nets are called markings. There are many extensions of Petri Net models, including Timed Petri nets, Generalized Stochastic Petri Nets, and Colored Petri nets. The packages implementing various Petri net extensions include GreatSPN, SPNP, CPN Tool, and GPenSIM. When it comes to modeling blockchain systems and their vulnerabilities, the Petri nets play an increasing role.

In their work, the authors [42] explore different blockchain systems' vulnerabilities, which they model by using Petri nets. In particular, they consider the six major categories of vulnerabilities: consensus mechanism vulnerabilities (51% attack, Selfish Mining attack, Double Spending attack, Finney attack, and the GHOST protocol which can be exploited for the Balance attack), mining pool vulnerabilities (51% attack, Double Spending attack, and Block Withholding attack), smart-contract vulnerabilities, design/architectural vulnerabilities, and network vulnerabilities. Additionally, they consider advancements in the quantum computing field and security threats that may pose in blockchain systems.

Specifically, when it comes to smart contracts and modeling or avoiding their vulnerabilities, the authors [43] proceed by using Petri nets. Their approach allows a design of secure smart contract templates with a formal procedure to visualize a model, simulate, and verify business logic/workflows. To minimize the occurrence of errors, thus reducing the potential for vulnerabilities, they use Petri nets to aid in smart contract development. Based on Petri Net model, in their work, the [44] offer an outline for risk modeling of the blockchain ecosystem. Specifically, they propose Petri net models for the analysis of a blockchain platform, and its critical aspects. Among other elements, their model includes formal construction of the individual blocks, formal construction for transformation probability of possible risks, the generator of risk events, and formal metamodel of blockchain operation.

8.5.2 Bond Percolation and Random Graphs

To understand and quantify the loss distribution due to cyber attacks on or contagion failures of Local Area Networks in small- and medium-sized companies, Jevtić and Lanchier [45] introduced a realistic mathematical framework based on random graphs and percolation theory. Their framework is generalized to be applicable to cyber risks in the context of smart contracts [46]. The main novelty is that it accounts for the topology of the network of interactions that encodes all the connections among smart contracts and their users. The model, when dealing with smart contracts, consists of the following stochastic components.

The temporal structure is modeled by a Poisson process with rate μ. The arrival times of this process represent the times at which a cyber attack or a contagion failure occurs, meaning that the time between two consecutive occurrences exceeds t with probability $e^{-\mu t}$.

At the times of the Poisson process, we let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a particular realization of a random graph. In the case of smart contracts, it is also natural to assume that this graph has two different types of vertices: either a smart contract or a user/customer. This graph models the network of interactions in the sense that there is an edge connecting two vertices if these two vertices interact, so the choice of graph (Erdős–Rényi model [47], preferential attachment graph [48], small-world network [49], random tree, etc.) strongly depends on the context.

The contagion process is modeled using bond percolation [50–52], i.e. we assume that each edge is independently open with a fixed probability and that the contagion can only spread through the open edges. Smart contracts networks include two types of edges, so it is natural to consider two bond percolation parameters: the probability p that an edge connecting two smart contracts is open and the probability q that an edge connecting a smart contract to one of its users is open. These parameters measure the vulnerability of the network. The smart contracts and users that are "infected" by the contagion are simply the vertices that can be reached by a path of open edges from the source of the contagion, another vertex of the graph chosen at random.

Finally, to turn the set of infected vertices into a numerical value representing the loss at a given time, the last step is to fix a cost topology, i.e. attribute a dollar amount to each vertex. The most natural assumption is to attribute values that are independent random variables following one of two possible distributions, one for the smart contracts and another one for their users. The loss at a given time is then obtained by adding the values of all the vertices that are infected, and the aggregate loss by adding all the losses that occurred up to the present time.

Due to the inclusion of a network structure, this modeling approach is realistic but often too elaborate to derive an exact expression of the loss distribution. However, the mean and variance of the aggregate loss, two key quantities used by researchers in probability to understand complex random objects and also by companies to estimate potential liabilities, can be computed analytically for simple networks and numerically for more sophisticated networks. In particular, the mean and variance of the loss distribution (or at least good lower and upper bounds) have been derived analytically in [53] for paths, rings, and stars, and in [45] for random trees. Interestingly, these works show that the mean and variance of the loss distribution strongly depend on not only the size of the network but also its topology, meaning that the structure of the network indeed plays an essential role and can hardly be ignored.

8.6 Security: Blockchain for IoT

The term IoT refers to all devices of everyday life that are connected to the Internet and that have some kind of intelligence. It is applicable in domestics, all kinds of appliances that communicate with the user, biomedical control systems, trends in consumer use, automobile industry, etc. Driven by artificial intelligence, cognitive computing, and new solutions for machine-to-machine (M2M) connectivity as well as rising technologies concerning big data and data analytics, the adoption of IoT concept has accelerated rapidly.

8.6.1 IoT Security Vulnerabilities

Securing an IoT platform brings four important benefits over traditional cloud-centric IoT systems: (i) sensitive data can be kept at the edge level at all times without the need for transferring to cloud, thus enhancing privacy; (ii) the edge level can provide more contextual information about security and privacy than traditional cloud-centric IoT systems – information about the operating environment is easy to collect at the edge level; (iii) cloud communication overhead is reduced, making the attacks on that link less probable; (iv) proximity and intelligence at the edge of the network enable real-time interaction, predictable network latency, and clock synchronization. Security frameworks built for IoT systems must leverage these benefits [54].

Considering the high-density distribution of heterogeneous nodes in the edge, security in the system itself must be autonomous. Security features of an IoT system should not depend on any cloud-based, or otherwise centralized system, in an ideal edge environment. Yes, cloud services should be included in the deeper analysis of suspicious node behavior, etc. (and derive new and updated existing security policies), but reactions to malicious actions should happen instantly when detected, on the edge level of the system, regardless of connectivity to a cloud platform, and without the involvement of system administrators. In an IoT architecture, end-to-end security must cover all devices from the cloud level to the edge of the network (gateways, sensors, actuators). However, the security of IoT systems has to start with secure device hardware. If a device is not trustworthy at the time of deployment, the whole infrastructure becomes unsecured and unstable. When and only when trusted edge nodes are deployed, a secure communication layer encompassing all nodes can be built, thus creating a secure end-to-end IoT infrastructure. An IoT system should secure itself by authenticating the identity of all users, data consistency and integrity, and the availability of system services. An edge system should have the ability to revoke connections and handle key management anonymously, as well as

to insert real-time constraints and monitoring them [55]. It is, however, very important that this process is anonymous and stateless to protect sensitive user data (location, identity, etc.) even from the IoT system itself. For a highly efficient IoT system, it is also relevant to aggressively analyze activities of both users and administrators, and adjust topology, bandwidth allocation, and traffic policies real-time [56].

To conclude, coming up with an efficient security framework for an IoT system is much more difficult than handling traditional distributed systems. Cyber-threats may come from many sources, focus on many different layers and technologies, and the major security task here is to leverage the dense distribution of computational resources properly to utilize an efficient distributed security framework.

8.6.2 Blockchain–IoT Convergence

IoT ecosystems are a fast-growing class of IT solutions and are expected to scale to millions of connected devices deployed in a single network. The challenge is how to best provide security, privacy, data protection, and accountability in favor of users. Because blockchain is based on cryptographically secured, immutable distributed ledger technology and consensus-based decision making, it may enhance IoT ecosystems with more automated resource optimization and innate security. Among other advantages, this blockchain-based architecture offers organizations and users: data sharing across a network of key stakeholders via a distributed system of records; automating interactions between nodes with embedded business terms into smart contracts; provenance authentication by means of hash-based security and cryptographic identity verification; and finally detection of bad actors and threat mitigation by consensus and agreement models. Blockchain-enabled IoT deployment will improve overall system health and integrity by allowing devices to register and validate themselves against the network. Actions on a blockchain are validated against the certificates (or set of private/public keys) of participating devices. Autonomous network access management will need capabilities, including a complex and on-demand certificate generation, trust computation and sharing, and certificate revocation and blacklisting. For trusted certificates, trusted nodes are thus required to "propagate" trust to the newly added nodes.

Figure 8.3 shows the dimensions for analyzing the merits Blockchain and IoT integration. While analyzing the security merits of the integration between the two technologies, it is equally important to analyze the merits of the most prominent application domains that benefit largely from it. We shall not go deep into application domains in this chapter but only consider cybersecurity.

Blockchain is a perfectly suitable complement to IoT bringing improved security, privacy and traceability, interoperability, scalability, reliability, and

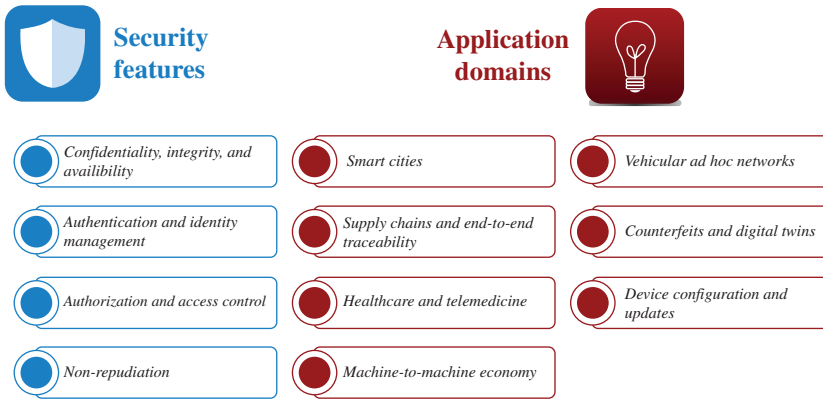


Figure 8.3 Dimensions of analyzing blockchain–IoT convergence.

autonomous interactions. IoT security is bettered through empowering and decentralizing authentication and access control schemes as well as data integrity and identity management. Complex privacy schemes are enabled by utilizing private blockchains, but data traceability is a universal blockchain characteristic that could benefit IoT in many ways: misbehavior discovery and verification, false data tracing to source, etc. IoT universal interoperability with blockchain is achieved because devices will share an overlay blockchain layer (protocol) that unifies data storage and transportation mechanisms. With blockchains, scalability of IoT systems is improved through higher levels of availability and decentralization of data and services. A system is as reliable as are its data and services; thus, blockchains play a major role in creating reliable IoT systems through integrity enforced through cryptographic schemes such as asymmetric encryption, hashing, and digital signatures. Finally, all of the above enable secure, machine-centric, autonomous interactions through smart contracts automatically executing programmed and enforced contract clauses.

When discussing the Blockchain–IoT convergence, one must keep in mind the operational requirements for IoT systems: dynamic but verifiable group membership, devices, data and decision-making processes integrity, lightweight operations in terms of resources, optional encryption, capability for handling device power-off and sleep periods, and handling large degree of resource diversity (data, sensors, aggregators, etc.). Desirable properties that blockchain brings are its distributed protocols with verifiable transaction history as well as dynamic membership and multi-party signatures. As for the undesirable properties: most of public blockchains require PoW and PKI and size of the ledger is an issue for devices lacking storage. These undesirable properties can, however, be tackled. PoW could be replaced by proof of earlier participation using blockchain transaction history.

PKI could be replaced with lightweight hash-based signatures or other Merkle tree-based schemes. The ledger could be pruned and compressed for storage-poor devices: a device could store only hashes of transaction headers or maintain only a device-relevant transaction ledger.

8.6.2.1 Enhancing IoT Security Features

Security features that blockchains bring to the IoT are numerous. Essentially they could be grouped to four larger categories: (i) Confidentiality, Integrity, and Availability, (ii) Authentication and Identity Management, (iii) Authorization and Access Control, and (iv) Non-repudiation. All of these are already built into blockchain technology by design and the purpose of this section is to elaborate on how they can be leveraged to cover some of IoT security concerns.

One of the major security issues for the IoT is handling sensitive, personal data captured by IoT devices. Data **integrity** asserts that data generated inside an IoT ecosystem was not replayed, falsified, or otherwise tampered with (i.e. through an injection attack) on its path from source to destination device. Even if such an event occurs, the data, devices, and processes that took part in the activity shall be *blacklisted* by the network and removed or otherwise penalized. Providing integrity of data and processes is absolutely essential for IoT ecosystems since the vast amount of sensors generate valuable information that often needs to be acted with in real-time. Finally, availability rests upon the requirement that one system has to be available, whenever needed, to provide services to end-users (which is an existing feature of blockchains).

Public blockchains, which are intended to be permissionless, are less quipped to deal with **confidentiality** of data. On the other hand, semi-private (e.g. Quorum) and private (e.g. HLF) blockchains are typically equipped with complex mechanisms to handle confidentiality on different layers: user, transaction, smart contract, node, etc. All blockchains ensure integrity of data through: (i) the immutability and append-only characteristics of the ledger, (ii) storing transaction data in form of Merkle trees, as well as (iii) cryptographic protections between blocks that contain data. Merkle trees not only better data integrity but provide an easy and secure way to verify it as well. By design, blockchains are resistant to data modifications. Additionally, blockchains can be leveraged for IoT to provide proof of data history easing internal and external audits. One of the main advantages of the blockchains is that they are, theoretically, always available. In other words, as the network is distributed there is no single point of failure. In this manner, blockchains can be used to extend IoT systems' **availability**. However, we must consider read and write availability of blockchains separately. While read availability is typically at the highest level, write availability significantly depends on the speed of validating transactions, forming and publishing blocks, etc. [57].

Authentication and Identity Management (AIM) refer to providing users/devices of the system with proper means to prove their identity. Traditionally, AIM relies on human–computer interaction; however, in IoT, the AIM needs to be M2M-based. AIM for IoT should be as decentralized as possible and devices need to have reliable AIM attributes (i.e. sufficient proof of their identity) without contacting a central server. Multiple AIM models coexist in the IoT ecosystem, sometimes all at the same time – for more sensitive operations a more sensitive model will be used or, for example, a distributed model is preferred when the operation is new device on-boarding. Considering implementations, AIM can range from public-key infrastructure, trusted platform modules (TPMs), X.509 certificates, hardware security modules (HSMs), symmetric keys, etc. Anyway, we need to weigh multiple variables, such as energy resources, hardware ability, financial budgets, security specifications, and accessibility, before determining which IoT authentication model fits the use-case at hand the best. Blockchain-based IoT authentication is a good step toward what we call self-sovereign digital identity for both devices and users. Furthermore, this digital identity will be used to sign every microtransaction (e.g. sensor reading, user login) inside an IoT system enabling ultimate traceability throughout the ecosystem. Blockchain as a decentralized AIM provider has been proposed by IBM (Verify Credentials [58]), Accenture [59], etc. Creating and maintaining digital identities is one of the World Economic Forum global initiatives for 2020–2021: the Known Traveller Digital Identity initiative aiming to diminish paper identity documents and enhance security of world travel [60].

Authorization and access-control present system-wide security mechanisms for specifying access rights/privileges to the system's resources as well as handling key management services. The problem with modern authorization schemes is they typically rely too much on a trusted central authority [54]. When the central authority is compromised, hackers can corrupt and impair the authorization policies system-wide. If an authorization scheme is decentralized, network unavailability, devices, and communication links failure are real issue for many highly distributed IoT systems. For IoT systems, authorization schemes need to be flexible – multiple authorization and access-control schemes should be supported; data used for authorization must have the potential to be migrated and aggregated easily; data provenance must be enabled at all times [54]. Authorization rules and policies can be coded into upgradeable smart contracts, immediately creating a dynamic, decentralized, verifiable authorization scheme where all activity and can be easily traced [61]. Authorization activity can be analyzed system-wide, and if misbehavior is detected the ledger can be updated with *blacklisted* devices, users and services immediately, while the smart contract in charge of authorization will have access to this data. Blockchain protocols, such as NuCypher [62], can be used to provide decentralized key management and data access control.

Non-repudiation refers to a system's characteristic that ensures every activity and state it lead to can be verified, and that eventual disputes around them can be easily solved. It is an inherent characteristic of blockchains. This basically means that a cryptographically signed and verified transaction cannot be denied by either the transaction initiator or the blockchain network. The property of non-repudiation is significant for providing dispute resolution services and there are many blockchain-based systems proposed for that purpose: by examining transaction logs [63], arbitrarily invoking a dispute–resolution smart contract that has access to the ledger or transaction history [64], etc.

8.7 Blockchain for Federated AI

Modern artificial intelligence (AI) development and applications face two significant challenges: the first is that, in most cases, data is owned and stored in big and isolated datacenters preventing democratization of AI-related innovation, while the other is the need to strengthen privacy and security of data used by AI applications. These two challenges can be resolved through the disaggregation of data ownership, which in turn creates opportunities for creative and secure techniques of federated learning (FL). Federated learning is an emerging decentralized approach that is particularly cognizant of restrictions concerning privacy and resource constraints [65]. FML works with the distributed dataset and normally performs the model training locally at each data source in the federation of all data sources needed for particular applications. By keeping the data close to where it is generated, FML exploits the on-device processing power and untapped private data by performing the model training in a decentralized manner. After training a local model, each individual learner transfers its local model parameters, instead of the raw training dataset, to an aggregating unit. The aggregator utilizes the local model parameters to update a global model, which is eventually fed back to the individual local learners for their use. As a result, each local learner benefits from the datasets of the other learners only through the global model, shared by the aggregator, without explicitly accessing their privacy-sensitive data.

There are typically two types of FML models. The **centralized federated learning** model refers to executing different steps of the learning algorithms by the distributed participating nodes, while coordinated by a central server. The central authority selects the nodes at the beginning of the training process and is responsible for the aggregation of the received updates. The **decentralized federated learning** model requires no central server and all nodes are able to coordinate themselves to obtain the global model. The model updates happen between the interconnected nodes.

Although federated learning helps preserve data privacy by sharing only the model parameters instead of raw data, the parameter exchange might leak significant information. A cyberattack can recover data from gradients uploaded by individual nodes. Moreover, the federated approach for training the model is susceptible to model poisoning attacks [66]. Blockchain was adopted to address these security concerns. Blockchain is used in federated learning mainly for parameter exchange, data verification, node recovering, data storage. The typical procedure for blockchain-based federated learning has 4 steps [67]. (i) A machine learning model is built at nodes where local training datasets are generated. (ii) Each of the nodes generates shared training parameters based on the local model in a manner that precludes any requirement for the raw data to be accessible by each of the other nodes on the blockchain network. (iii) Shared training parameters are exchanged with other nodes via a blockchain transaction. (iv) In a decentralized approach, each node individually merges the training parameter and continues training till it is ready to output a new set of the training parameters.

Owing to blockchain's intrinsic features, the fundamental limitations of trust, resiliency, and accessibility associated with centralized frameworks can be eliminated. First, the round delineation, selection, and model aggregation occur on-blockchain in a fully decentralized manner. The underlying consensus protocol and smart contracts help ensure transparency, fairness, and impartiality. Moreover, blockchain delivers resiliency in computation while smart contracts perform real-time coordination related to round delineation, selection, and aggregation phases across participating devices. Due to such versatility, blockchain-based federated learning distributes operational costs across the participants and eliminates the complexity of segregated resource requirements present in a cloud-driven (centralized) application [68]. Consequently, a blockchain-based distributed set up lowers the entry barrier for smaller players and improves accessibility significantly.

8.7.1 FML Basic Principles

Machine learning is being used to train the computers to learn from data provided to perform tasks without being explicitly programmed to do so. Nowadays, machine learning has been used more and more in industrial, medical, education, e-commerce, and other data-rich applications, and it shows significant benefit. For traditional machine learning, it is usually required that all the local datasets are uploaded to one server (*traditional centralized machine learning*) or the local data samples have to be uniformly distributed and roughly be of the same size (*classical decentralized machine learning*). On the contrary, FML is a machine learning technique that trains an algorithm across multiple decentralized edge devices or

distributed servers holding local data samples, which are typically heterogeneous and of different sizes, without exchanging them. Federated learning may be unreliable since it is subject to more computational failures or communication outages due to its distributed nature [69]. Traditional distributed learning usually uses data centers which are reliable and connected with fast networks. There are typically two types of FML models. In **centralized federated learning** a central server is used to oversee the different steps of the algorithms execution and to coordinate all the nodes participating in the learning process. The server is responsible for the nodes selection at the beginning of the training process and the aggregation of the received model updates. On the other hand in **decentralized federated learning** there is no central server for decentralized federated learning. All the nodes are able to coordinate themselves to obtain the global model. The model updates happen between the interconnected nodes.

The principle of federated learning consists of training local models on local data samples and exchanging parameters between these local nodes at some frequency to generate a global model shared by all nodes. The benefits of federated learning are numerous. With federated learning, only the machine learning parameters are exchanged. No local or raw data is uploaded or exchanged externally. So the privacy exposure risk is minimized. After the learning process, it is possible to deploy clustering and aggregate IT resources of the nodes that share some similarities. In some cases, it is illegal to transfer data externally. Federated learning brings solutions to train a global model while respecting security constraints.

The first federated-learning framework was proposed by Google [70]: Defining N data owners with their respective data $(D_1, \ldots, D_N)$, a conventional approach is to put all data together and use the resulting dataset $(D = D_1 \cup \cdots \cup D_N)$ to train a model. The idea of the federated-learning is to build a learning process in which the data owners collaboratively train a global model without exposing its data to the other owners. There are several ways to categorize federated learning, based on the distribution characteristics of the data [71]. Firstly, there is (i) the horizontal or sample-based federated learning. It is introduced in the scenarios in which datasets share the same feature space (same sensors, devices) but different space in samples (different measurements). (ii) The *vertical or feature-based federated learning* applies to the cases in which two datasets share the same sample ID (same measurements) space but differ in feature space (different sensors, devices). Thirdly, (iii) we have *Federated Transfer Learning* (FTL). It is used when two datasets vary not only in samples but also feature space (see Figure 8.4) and a transfer function between the features of the two datasets is computed. In this way, it is possible to use this relationship to estimate a private dataset's outputs. This approach can help process and analyze non-homogeneous datasets with traditional machine learning techniques and deal with data privacy, data storage reduction, and personalized models for each data owner.

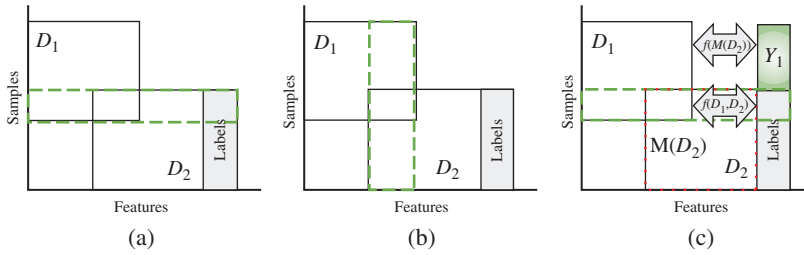


Figure 8.4 A schematic representation of the characterization of federated learning. (a) Horizontal FL, (b) vertical FL, and (c) federated transferred learning.

8.7.2 Case Study: Blockchain-Based FML in Large-Scale Environmental Sensing

Modern tools for monitoring different environmental compartments rely on sensory systems, which are custom-tailored to characterize, track, and model natural processes. Moreover, environmental monitoring has additional challenges that stem from the need to deploy a large-scale data collection apparatus, both in spatial and temporal domains. While the technical challenges could be overcome by the deployment of IoT-based systems and laboratories, other challenges, which originate from inherent business risks associated with data collection and analysis, create data security constraints for stakeholders interested in pursuing environmental monitoring. In these business environments, the majority of critical monitoring applications would benefit from a combination of data generated from multiple transactional points and different stakeholders, which in turn require: (i) FML both in the central cloud and the distributed vertical IoT silos, and (ii) development of trust management and data protection platform.

To enable large-scale environmental sensing, modeling, and natural process characterization using machine learning and AI that involves a heterogeneous constellation of business and data sources, it is desirable to deploy a trust management platform based on private permissioned blockchain networks for managing data access privileges, ensuring integrity and repeatability of analysis results, and enabling real-time audibility of critical transaction frequently required by government agencies and regulators. By contrast to the centralized model, in the federated-learning (FL) architecture, each vertical IoT system will host and train a local model based on the stakeholder's private data, which are then aggregated trustfully by an FL aggregator and protected in a central model governed by the blockchain network. Most time-series data solutions use dedicated IoT systems for sensing, analyzing, and characterizing environmental processes. Resource and process optimization is mostly performed within these separate vertical silos. In contrast to the traditional AI methods, federated learning migrates the models

closer to the data source, or to client IoT device, for training and inferencing. FL authority (usually hosted in the cloud) defines federated-learning principles for the system at hand (global/generic model, performance metrics, update cycles, data preprocessing and preparation, metadata, ontologies, and interfaces). FL utilizes computing and storage resources on the fog and edge systems (server, IoT controllers/gateways, and mobile devices) to perform local model training on locally collected and stored datasets and data streams. Locally trained models send their parameters, weights, and performance metrics to the FL aggregator, which adapts the global model and sends it for the next iteration of training on local systems. After several iterations, the global model for a specific analysis challenge or decision-making task improves performance and is optimized for the local datasets. This approach for ML model training and execution drastically reduces latency, reduces cloud computing costs, saves communication bandwidth, and enforces data privacy and sovereignty of local systems/stakeholders.

Keeping data private is the major additional benefit for each of the entities participating in this federated-learning platform. The data structures and parameters are likely to be similar. However, they do not need to be the same, which necessitates a lot of data preprocessing to standardize model inputs at each client IoT level.

Figure 8.5 depicts a design for large-scale sensing and monitoring of aquatic and soil environments in which ML/AI models are created, trained, and used by different participating entities. The entity that creates data to be used for the model

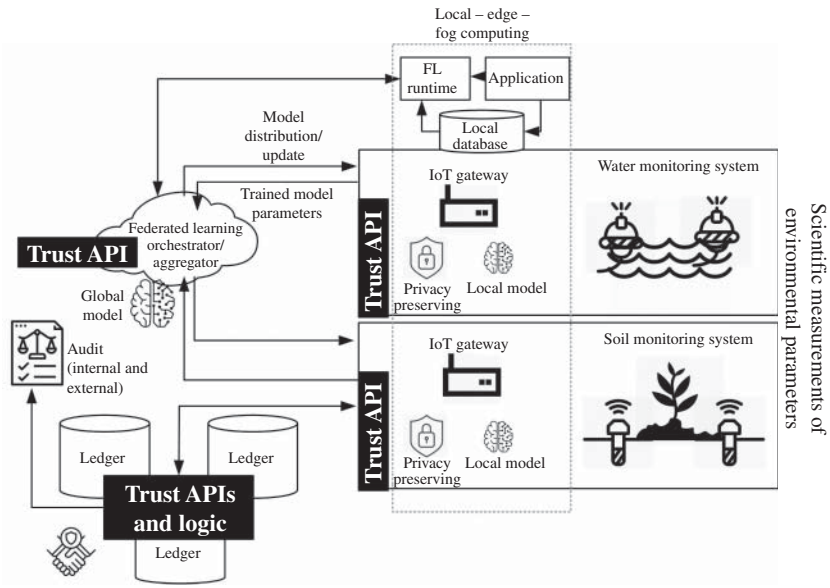


Figure 8.5 Blockchain-based FML for large-scale environmental monitoring.

is often different from the entity that trains the model, which is different yet again from the user of the trained model. The user needs to trust the received ML/AI model, and this requires having the provenance information about how the model was trained and the data the model was trained on.

References

1 Gelle, M. and Schmidt, J. (2020). Innovating for Resilience. *Technical Report*. Technology Vision 2020 for the Aerospace and Defense industry, Accenture.

2 Boscovic, D. (2019). Who can you trust? *Issues in Science and Technology* 35 (3): 94–95.

3 Werbach, K. (2018). *The Blockchain and the NW Architecture of Trust*. MIT Press.

4 Kehoe, L., Piscini, E., and Dalton, D. (2017). Blockchain & Cybersecurity. *Technical Report*. Deloitte: Blockchain Lab.

5 Ravindra, S. (2018). The role of blockchain in cybersecurity. https://www.infosecurity-magazine.com/next-gen-infosec/blockchain-cybersecurity/ (Last accessed 20 December 2020).

6 Croman, K., Decker, C., Eyal, I. et al. (2016). On scaling decentralized blockchains. In: *International Conference on Financial Cryptography and Data Security*, 106–125. Springer.

7 Swan, M. (2015). *Blockchain: Blueprint for a New Economy*. O'Reilly Media Inc.

8 Law Library of Congress (U.S.) (2018). Regulation of cryptocurrency around the world.

9 Goddard, M. (2017). The EU general data protection regulation (GDPR): European regulation that has a global impact. *International Journal of Market Research* 59 (6): 703–705.

10 Chassang, G. and Béranger, J. (2018). La blockchain pour la recherche en santé? Potentiels, enjeux juridique et éthique.

11 Adler, J., Berryhill, R., Veneris, A. et al. (2018). Astraea: a decentralized blockchain oracle. *2018 IEEE International Conference on iThings, GreenCom, CPSCom and SmartData*, 1145–1152. IEEE.

12 Exaing (2020). Pow 51% attack cost. https://www.exaking.com/51 (Last accessed 24 December 2019).

13 Jang, J. and Lee, H.-N. (2020). Profitable double-spending attacks. *Applied Sciences* 10 (23): 8477.

14 Yang, G., Wang, Y., Wang, Z. et al. (2020). IPBSM: an optimal bribery selfish mining in the presence of intelligent and pure attackers. *International Journal of Intelligent Systems* 35: 1735–1748.

15 BitNodes (2020). Global Bitcoin nodes distribution. https://bitnodes.io/ (accessed 24 December 2020).

16 Bach, L.M., Mihaljevic, B., and Zagar, M. (2018). Comparative analysis of blockchain consensus algorithms. *41st International Convention on Information and Communication Technology, Electronics and Microelectronics*, 1545–1550. IEEE.

17 Deirmentzoglou, E., Papakyriakopoulos, G., and Patsakis, C. (2019). A survey on long-range attacks for proof of stake protocols. *IEEE Access* 7: 28712–28725.

18 BitShares Documentation (2020). Delegated proof of stake. https://how.bitshares.works/en/master/technology/dpos.html. (accessed 25 December 2020).

19 Szabo, N. (1996). Smart contracts: building blocks for digital markets. *EXTROPY: The Journal of Transhumanist Thought* (16), 18(2).

20 Mehar, M.I., Shier, C.L., Giambattista, A. et al. (2019). Understanding a revolutionary and flawed grand experiment in blockchain: the DAO attack. *Journal of Cases on Information Technology (JCIT)* 21 (1): 19–32.

21 AnChain.AI (2017). AnChain smart contract auditing. https://www.anchain.ai/smartcontractauditing (Last accessed 09 June 2019).

22 Feist, J., Grieco, G., and Groce, A. (2019). Slither: a static analysis framework for smart contracts. *2019 IEEE/ACM 2nd International Workshop on Emerging Trends in Software Engineering for Blockchain (WETSEB)*, 8–15. IEEE.

23 Tsankov, P., Dan, A., Drachsler-Cohen, D. et al. (2018). Securify: practical security analysis of smart contracts. *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, 67–82. ACM.

24 Perez, D. and Livshits, B. (2019). Smart contract vulnerabilities: does anyone care? *arXiv preprint arXiv:1902.06710*.

25 Chen, H., Pendleton, M., Njilla, L., and Xu, S. (2020). A survey on ethereum systems security: vulnerabilities, attacks, and defenses. *ACM Computing Surveys (CSUR)* 53 (3): 1–43.

26 Konda, C., Connor, M., Westland, D. et al. (2019). Nightfall. *Technical Report*. EY Global Blockchain R&D.

27 Li, X., Zheng, Y., Xia, K. et al. (2020). Phantom: an efficient privacy protocol using zk-SNARKs based on smart contracts. *IACR Cryptology ePrint Archive*, 2020:156.

28 Johnson, N. (2020). Ethereum name service. https://docs.ens.domains (accessed 12 December 2020).

29 Androulaki, E., Barger, A., Bortnikov, V. et al. (2018). Hyperledger fabric: a distributed operating system for permissioned blockchains. *Proceedings of the 13t EuroSys conference*, 1–15.

30 del Castillo, M. (2019). Blockchain 50: billion dollar babies. https://www.forbes.com/sites/michaeldelcastillo/2019/04/16/blockchain-50-billion-dollar-babies/?sh=30b04d5c57cc (Last accessed 01 November 2020).

31 Linux Foundation. Blockchain network. https://hyperledger-fabric.readthedocs.io/en/release-1.4/network/network.html (Last accessed 01 November 2020).

32 Dabholkar, A. and Saraswat, V. (2019). *Ripping the Fabric: Attacks and Mitigations on Hyperledger Fabric*, 300–311. ISBN 978-981-15-0870-7.

33 Nguyen, T.S.L., Jourjon, G., Potop-Butucaru, M., and Thai, K.L. (2019). Impact of network delays on hyperledger fabric. *IEEE INFOCOM 2019-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, 222–227. IEEE.

34 Graf, M., Küsters, R., and Rausch, D. (2020). Accountability in a permissioned blockchain: formal analysis of hyperledger fabric. *2020 IEEE European Symposium on Security and Privacy (EuroS&P)*, 236–255. IEEE.

35 Andola, N., Gogoi, M., Venkatesan, S., and Verma, S. (2019). Vulnerabilities on hyperledger fabric. *Pervasive and Mobile Computing* 59: 101050.

36 Hardjono, T. and Pentland, A. (2019). Verifiable anonymous identities and access control in permissioned blockchains.

37 Hasanova, H., Baek, U.-j., Shin, M.-g. et al. (2019). A survey on blockchain cybersecurity vulnerabilities and possible countermeasures. *International Journal of Network Management* 29 (2): e2060.

38 Putz, B. and Pernul, G. (2019). *Trust Factors and Insider Threats in Permissioned Distributed Ledgers: An Analytical Study and Evaluation of Popular DLT Frameworks*, 25–50. ISBN 978-3-662-60530-1.

39 Choi, B.K. and Kang, D. (2013). *Modeling and Simulation of Discrete Event Systems*. Wiley.

40 Peterson, J.L. (1981). *Petri Net Theory and the Modeling of Systems*. Prentice Hall PTR.

41 Reisig, W. (2013). *Understanding Petri Nets: Modeling Techniques, Analysis Methods, Case Studies*. Springer.

42 Shahriar, M., Bappy, F.H., Hossain, A.K.M. et al. (2020). Modelling attacks in blockchain systems using Petri Nets. *arXiv preprint arXiv:2011.07262*.

43 Zupan, N., Kasinathan, P., Cuellar, J., and Sauer, M. (2020). Secure smart contract generation based on Petri Nets. In: *Blockchain Technology for Industry 4.0*, 73–98. Springer.

44 Kabashkin, I. (2017). Risk modelling of blockchain ecosystem. In: *International Conference on Network and System Security*, 59–70. Springer.

45 Jevtić, P. and Lanchier, N. (2020). Dynamic structural percolation model of loss distribution for cyber risk of small and medium-sized enterprises for tree-based LAN topology. *Insurance: Mathematics and Economics* 91: 209–223.

46 Petar, J. and Nicolas, L. (2021). Probabilistic framework for loss distribution of smart contract risk.

47 Erdős, P. and Rényi, A. (1959). On random graphs. I. *Publicationes Mathematicae Debrecen* 6: 290–297.

48 Barabási, A.-L. and Albert, R. (1999). Emergence of scaling in random networks. *Science* 286 (5439): 509–512.

49 Watts, D.J. and Strogatz, S.H. (1998). Collective dynamics of 'small-world' networks. *Nature* 393: 440–442.

50 Grimmett, G. (1999). *Percolation*, *Grundlehren der Mathematischen Wissenschaften [Fundamental Principles of Mathematical Sciences]*, vol. 321, 2e. Berlin: Springer-Verlag. ISBN 3-540-64902-6.

51 Kesten, H. (1982). *Percolation Theory for Mathematicians*, *Progress in Probability and Statistics*, vol. 2. Boston, MA: Birkhäuser. ISBN 3-7643-3107-0.

52 Lanchier, N. (2017). *Stochastic Modeling* Universitext. Cham: Springer. ISBN 978-3-319-50037-9; 978-3-319-50038-6.

53 Jevtić, P., Lanchier, N., and La Salle, A. (2020). First and second moments of the size distribution of bond percolation clusters on rings, paths and stars. *Statistics & Probability Letters* 161: 108714.

54 Pešić, S., Ivanović, M., Radovanović, M., and Bădică, C. (2020). CAAVI-RICS model for observing the security of distributed IoT and edge computing systems. *Simulation Modelling Practice and Theory* 105: 102125.

55 Khalid, W., Ullah, Z., Ahmed, N. et al. (2018). A taxonomy on misbehaving nodes in delay tolerant networks. *Computers & Security* 77: 442–471.

56 Pešić, S., Tošić, M., Iković, O. et al. (2017). Context aware resource and service provisioning management in fog computing systems. In: *International Symposium on Intelligent and Distributed Computing*, 213–223. Springer.

57 Weber, I., Gramoli, V., Ponomarev, A. et al. (2017). On availability for blockchain-based systems. *2017 IEEE 36th Symposium on Reliable Distributed Systems (SRDS)*, 64–73. IEEE.

58 IBM (2020). IBM Verify Credentials: transforming digital identity into decentralized identity. https://www.ibm.com/blockchain/solutions/identity (accessed 24 December 2020).

59 Accenture (2020). Digital identity. https://www.accenture.com/ch-en/services/blockchain/digital-identity (accessed 24 December 2020).

60 System Initiative on Shaping the Future of Mobility (2018). The known traveller: unlocking the potential of digital identity for secure and seamless travel. World Economic Forum.

61 Siris, V.A., Dimopoulos, D., Fotiou, N. et al. (2019). Trusted D2D-based IoT resource access using smart contracts. *2019 IEEE 20th International Symposium on" A World of Wireless, Mobile and Multimedia Networks"(WoWMoM)*, 1–9. IEEE.

62 Egorov, M., Wilkison, M.L., and Nu nez, D. (2017). NuCypher KMS: decentralized key management system. *arXiv preprint arXiv:1707.06140*.

63 Zou, J., Wang, Y., and Orgun, M.A. (2016). A dispute arbitration protocol based on a peer-to-peer service contract management scheme. *2016 IEEE International Conference on Web Services (ICWS)*, 41–48. IEEE.

64 Klems, M., Eberhardt, J., Tai, S. et al. (2017). Trustless intermediation in blockchain-based decentralized service marketplaces. In: *International Conference on Service-Oriented Computing*, 731–739. Springer.

65 Niknam, S., Dhillon, H., and Reed, J. (2020). Federated learning for wireless communications: motivation, opportunities, and challenges. *IEEE Communications Magazine* 58 (6): 46–51.

66 Konečný, J., McMahan, H., Ramage, D., and Richtárik, P. (2016). Federated optimization: distributed machine learning for on-device intelligence.

67 Konecný, J., McMahan, H.B., Yu, F.X. et al. (2016). Federated learning: strategies for improving communication efficiency. *CoRR*, abs/1610.05492.

68 Khajeh-Hosseini, A., Sommerville, I., and Sriram, I. (2010). Research challenges for enterprise cloud computing.

69 Kairouz, P., McMahan, H.B., Avent, B. et al. (2019). Advances and open problems in federated learning. *arXiv preprint arXiv:1912.04977*.

70 Zhao, Y., Zhao, J., Jiang, L. et al. (2019). Mobile edge computing, blockchain and reputation-based crowdsourcing IoT federated learning: a secure, decentralized and privacy-preserving system. *CoRR*, abs/1906.10893. http://arxiv.org/abs/1906.10893.

71 El Rifai, O., Biotteau, M., De Boissezon, X. et al. (2020). Blockchain-based federated learning in medicine. In: *Artificial Intelligence in Medicine. AIME 2020*, *Lecture Notes in Computer Science*, vol. 12299 (ed. M. Michalowski and R. Moskovitch), 214–224. Cham: Springer.

9

6G Resource Management and Sharing: Blockchain and O-RAN

Hao Xu[1], Paulo Valente Klaine[2], Oluwakayode Onireti[3], and Chih-Lin I[4]

[1] *Shanghai Engineering Research Center for Blockchain Application and Services, China*
[2] *Ericsson Japan K.K., Japan*
[3] *James Watt School of Engineering, University of Glasgow, UK*
[4] *China Mobile Research Institute, China*

9.1 Introduction

The fifth generation of mobile networks, 5G, is already being commercialized in some parts of the world, with the expectation of addressing limitations of current cellular systems and providing an underlying platform for new services to emerge and thrive [1].

5G was envisioned to be not only a faster 4G, but also an enabler for several other applications, such as the IoE, industry automation, intelligent transportation and remote healthcare, to name a few, by providing ultra-high reliability, latency as low as 1 ms, and increased network capacity and data rates [2] with higher energy efficiency [3]. However, despite the emergence of new technologies, such as millimeter waves, massive Multiple-Input–Multiple-Output (MIMO), and the utilization of higher frequency bands, it is clear that 5G is not capable of meeting all of these requirements, albeit improving significantly from its predecessors.

As such, research has already shifted toward the next generation of mobile networks, 6G [2, 4–6] with futuristic vision offered by opening and softening RAN carried out by the O-RAN Alliance based on current 5G mobile networks [7, 8].

Throughout the years, the RAN has experienced architectural transitions from LTE toward 5G New Radio (5G NR) and, more recently, with the Open RAN/O-RAN initiatives [7] that brings several open challenges toward 6G and beyond, marking the transition from physically distributed base stations to centrally managed functions with loosely coupled open interfaces, such as the hybrid architecture of Centralized Units (CU), Distributed Units (DU), and Radio Units (RU). The next-generation RAN is already making the most use of

Blockchains: Empowering Technologies and Industrial Applications, First Edition.
Edited by Anwer Al-Dulaimi, Octavia A. Dobre, and Chih-Lin I.

general computing resource [7], a good example is C-RAN with C standing for Cloud, Centralized features for software-defined RAN evolution [9-11]; vRAN, the virtualized network functions for RAN; and O-RAN, an open initiative pursued by operators and vendors. Thanks to initiatives of centralizing CU and DU into server clusters, the universal computing capacity offers more significant possibilities to manage and optimize the network at its edge. For instance, additional RAN functions can be integrated into the existing system simply by adding a virtual machines or container instances [7, 9]. Such improvement with RAN virtualization and clustering provides blockchain with general computing and communication accessibility, making the integration of blockchain functions into 5G and beyond effortless and low cost.

In addition to changes in system architecture, it is also expected that by 2030 our society will shift toward a more digitized, data-driven and intelligently inspired society that needs near-instant and ubiquitous wireless connectivity [5, 12]. Thus, several novel applications that provide such interaction and integration are bound to emerge in the next decade [5]. As such, some key trends that are foreseen to emerge soon are: virtual and augmented reality, 8K video streaming, holograms, remote surgery, industry 4.0, smart homes, fog computing, artificial intelligence (AI) integrated services, unmanned aerial vehicles (UAVs), and autonomous vehicles, to name a few [5, 6, 13]. These, by their turn, will demand much more from mobile networks in terms of reliability, latency, and data rates than 5G and its improvements can support [2, 5, 6, 8]. In short, several research initiatives around the globe have been working to shape the direction of 6G, and some of its key requirements are already being speculated, as in [2, 4, 5, 8]:

- Provide peak data rates of at least 1 Tb s^{-1} and latency of less than 1ms;
- Support user mobility up to 1000 km h^{-1};
- Operate in GHz to THz frequency range;
- Increase the network spectral efficiency, energy efficiency, and security;
- Harness the power of big data, enabling a self-sustaining RAN;
- Support for a massive number of devices and things, enabling the IoE.

In order to enable all the above and increase the system's total capacity, two different approaches are possible, according to Shannon's information theory: either increase the system bandwidth or improve the spectral efficiency [5, 14, 15].

It is well known that spectrum management is a key to achieve efficient spectrum usage; however, it still has issues. For example, it is known that current fixed paradigms for spectrum assignment and resource management is a major challenge in mobile networks.

This will become even more challenging in 6G, due to the ever-growing number of subscribers and their need of intermittent connectivity as well as the development of more data-hungry applications. Moreover, a number of studies have

shown that fixed spectrum allocation, despite being less complex, produces low spectrum efficiency, since license holders of that spectrum do not utilize it all the time (see [14] and references therein).

As such, several approaches have been proposed to improve spectrum management, such as Opportunistic Spectrum Access (OSA) or auction mechanisms.

Despite the benefits of these approaches, issues in terms of security, high computational power, and convergence are present. On top of that, even if such protocols provide some collaborations at the system level, the collaboration between users is still not considered, hindering the overall performance of those solutions. As 6G is expected to be much more cooperative than its preceding generations, with new technologies, such as wireless power transfer, mobile edge computing, the IoE, and Device-to-Device (D2D) communications, novel approaches that do not rely on a central authority controlling spectrum and resource management, such as the blockchain, are needed [2, 4].

Due to its inherent characteristics, blockchain is being regarded as the next revolution in wireless communications, with even the Federal Communications Commission (FCC) emphasizing the crucial role that it can play in 6G and beyond [16]. The main idea behind blockchain is that of an open and distributed database (ledger), where no single party has control, and transactions[1] are securely recorded in blocks. Each block is chained together to its predecessor in a sequential, verified, and secure manner, without the need of a trusted third party. As such, the blockchain is expected to revolutionize resource and spectrum sharing by eliminating the central authority and replacing it with a distributed one, realizing asset transactions without central authorization, improving network security, and reducing costs [17, 18].

Moreover, this integration between open RAN and the blockchain will allow the network to monitor and manage spectrum and resource utilization in a more efficient manner, reducing its administration costs and improving the speed of spectrum auction. In addition, due to its inherent transparency, the blockchain can also record real-time spectrum utilization and massively improve spectrum efficiency by dynamically allocating spectrum bands according to the dynamic demands of devices [15].

It can also provide the necessary but optional incentive for spectrum and resource sharing between devices, fully enabling new technologies and services that are bound to emerge [18]. Furthermore, with future open RANs shifting toward decentralized solutions, with thousands of cells deployed by operators and billions of devices communicating with each other, fixed spectrum allocation and operator-controlled resource sharing algorithms will not be scalable nor

1 These transactions can mean anything, such as holdings of a digital currency (i.e. Bitcoin), movement of goods across a supply chain, and spectrum and resource allocation in wireless networks [15].

effective in future networks. As such, by designing a communications network coupled with the blockchain as its underlying infrastructure from the beginning, 6G and beyond networks can be more scalable and provide better and more efficient solutions in terms of spectrum sharing and resource management. Moreover, with privacy in mobile networks becoming more and more critical, due to the emergence of novel applications, such as automated vehicles, industry 4.0, and medical applications, where even a minor failure can lead to disastrous consequences, the blockchain can be of great advantage in securing and storing sensitive information. Since all information in a blockchain is verified by all peers and it is immutable, this can allow future mobile networks to have a permanent record of all events with its corresponding timeframe [14].

Based on that, in this chapter, it is envisaged that 6G-enabled blockchain resource management, spectrum sharing and computing, and energy trading can serve as the driving force for future wireless network's use cases [14, 15, 17]. These resources are considered to be in a resource pool, in which spectrum is dynamically allocated, network slices are managed, and hardware is virtualized in order to enable the blockchain resource, and spectrum management. Based on this envisioned framework, a discussion on how the blockchain can enable resource sharing between devices, such as energy, data, spectrum lease, and computing power, is presented. In addition, the motivations of utilizing the blockchain for different use-cases are highlighted, mainly in terms of the Internet of things (IoT), D2D communications, network slicing, and network virtualization. Lastly, some future trends expected in the realm of blockchain-enabled open radio access networks are discussed, and conclusions are drawn.

The remainder of this chapter is organized as follows:

Section 9.2 presents an overview of current spectrum management, and allocation techniques, as well as a link between the blockchain, and spectrum management.

Section 9.3 discusses the motivations behind blockchains and outlines its fundamentals.

Section 9.4 discusses some key applications of blockchain, and how it can transform current open radio access networks. Lastly, Section 9.5 concludes the chapter.

9.2 Spectrum Management

In order to meet the increasing demand of high data rate for 5G and above applications, the capacity of the networks must increase. Hence, there is also an increase in the demand for spectrum. A dynamic policy for the management of

the spectrum license has recently been proposed to manage the spectrum more efficiently [19]. It allows unlicensed secondary users to opportunistically access the licensed spectrum without interfering with the licensed primary user. One of the options for using the new spectrum license is to distribute operation parameters to policy-based radios via a database. Such a model has been established for sharing the Television White Space (TVWS) and the Citizen Broadband Radio Service [20]. Recently, the application of blockchain as a trusted database has emerged [21] where various kinds of information, such as spectrum sensing and data mining outcomes, spectrum auction results, spectrum leasing mappings, and the idle spectrum information, are securely recorded on the blockchain. Blockchain thus brings new opportunities to Dynamic Spectrum Management (DSM) [15, 17, 21], and it has recently been identified as a tool to reduce the administrative expenses associated with DSM [22]. In particular, blockchain features can improve conventional spectrum management approaches, such as spectrum auction [14]. Further, the blockchain can aid in overcoming the security challenges and the lack of incentive associated with DSM [21]. Since the blockchain is a distributed database, it lends this property such that records in the DSM system are recorded in a decentralized manner.

One of the key areas where blockchain finds application in spectrum management is in recording its information. Note that the blockchain can record information as transactions, while spectrum management relies on databases, such as the location-based database for protecting the primary users in the TVWS [23]. With the blockchain, information about spectrum management, such as

(1) the TVWSs;
(2) spectrum auction results;
(3) the spectrum access history; and
(4) the spectrum sensing outcomes, can be made available to the secondary user.

As such, the benefits of recording the spectrum management information with the blockchain are discussed here:

- Contrary to conventional third-party databases, the blockchain enables users to get the direct control of the data in the blockchain, thus guaranteeing the accuracy of the data. In particular, information on TVWS, and other underutilized spectrum can be recorded in a blockchain. Such data could include the usage of the spectrum in frequency, time, and the geo-location of TVWS, and the primary users' interference protection requirement.
- Improved efficiency of spectrum utilization with efficient management of the secondary users' mobility, and the primary users' varying traffic demands. This is supported by the decentralized nature of the blockchain with primary users recording information on the idle spectrum, which can be readily accessed by

unlicensed secondary users. Moreover, secondary users can make their arrival into the network or departure from it known to other users by initiating a transaction.

- Access fairness can be achieved with blockchain-based approaches where the access history is recorded. This is not the case with the traditional Carrier Sensing Multiple Access (CSMA) schemes where their access is not coordinated. Access can be managed in the blockchain via smart contracts, where a threshold is defined, and users can be denied access to a specific band for a specified period when they reach the predefined access threshold.
- Blockchains provide a secure and verifiable approach to record information related to spectrum auction. Spectrum auction has been established as an efficient approach for dynamic allocation of spectrum resources [24]. The benefits of the blockchain-based approach include: (i) it prevents frauds from the primary users by providing transparency; (ii) it guarantees that the auction payments are not rejected because all transactions are verified before they are recorded on the blockchain; and (iii) it prevents unauthorized secondary users from accessing the spectrum since all secondary users can cooperatively/ collaboratively supervise and prevent such unauthorized access.
- Open Radio Access Network offers the necessary open interfaces and general computing compacity to host blockchain services at the RAN side, hence enabling the access of blockchain to wider coverage with versatile service possibility thanks to its open framework and the powerful RIC integration. O-RAN shares the initiatives of being open, soft, and secure with blockchain deployment in 6G and beyond.

In [15], the authors explored the applications of the blockchain in spectrum management, including primary cooperative sharing, secondary cooperative sharing, secondary non-cooperative sharing, and primary non-cooperative sharing. Moreover, in [25], the authors utilized a blockchain verification protocol for enabling and securing spectrum sharing in cognitive radio networks. Spectrum usage based on the blockchain verification protocol was shown to achieve significant gain over the traditional Aloha medium access protocol. The authors in [26] proposed a privacy-preserving secure spectrum trading and sharing scheme based on the blockchain technology, for UAV-assisted cellular networks. Furthermore, in [27], the authors proposed a consortium blockchain-based resource sharing framework for V2X, which couples resource sharing and consensus process together by utilizing the reputation value of each vehicle.

In [17], the authors proposed the integration of the blockchain technology and AI into open radio access networks for flexible and secure resource sharing.

9.3 Benefit of Using the Blockchain

9.3.1 Blockchain Background

Blockchain has seen a tremendous boost in the cryptocurrency and ledger keeping industry, and thanks to the vitality of the community, the technology has gained much attention from policymakers, mobile operators, and infrastructure commissioners [28].

Blockchains are distributed databases organized using a hash tree,[2] which is naturally tamper-proof and irreversible [29].

It has the attribute of adding distributed trust, and it is also built for enabling transaction consistency in a database. Furthermore, the blockchain allows for atomicity, durability, auditability, and data integrity [30].

Besides its chain-link data structure nature, the Consensus Mechanism (CM), which ensures an unambiguous ordering of transactions and guarantees the integrity and consistency of the blockchain across geographically distributed nodes, is of great importance to blockchains.

The CM largely determines the blockchain system performance, as shown in Table 9.1. Comparison of blockchain consensus, such as transaction throughput, delay, node scalability, and security level. As such, depending on application scenarios and performance requirements, different CMs can be considered.

The blockchain opens up transparent and distributed information reformation, which can benefit all aspects of industries, accommodating all range of centralization using different CMs. Regarding the utilization of the blockchain technology in 6G, the massive deployment of blockchain may lead to a major step forward for the communications industry and all other departments of the economy.

The transparent information flows on the blockchain are valuable assets for both users, operators, and service providers and societies. In social practice, the authority has always attempted to grip every detail for every operation and transaction. However, it would never track down every transaction that occurred if they are not born to be recorded.

Blockchain occurs to be the ideal tool for tracking of transactions if the blockchain native transactions are de facto in panoptic scenarios. The blockchain native resources and assets will stimulate a new era of information revolution. Such reformation will significantly improve the system efficiency and security thanks to the better public order [31]. It enables the Infrastructure as a Service (IaaS), Blockchain as a Service (BaaS) [32] to spread out in terms of feasibility, and now the infrastructure can be organized in a distributed way by allowing

2 A hash tree or Merkle tree is a tree in which every leaf node is labeled with the hash of a data block, and every non-leaf node is labeled with the cryptographic hash of the labels of its child nodes [67].

Table 9.1 Comparison of blockchain consensus.

Consensus	Suitable type of blockchain	Latency/TPS	BFT[a)]	Communication complexity[b)]	Security threshold[c)]	Energy usage	Scalability
PBFT	Consortium/private	Low/high [29]	Yes [29]	$O(n^2)$ [29]	33% [29]	Low	Low [29]
RAFT	Consortium/private	Very low/very High [30]	No [30]	$O(n)$ [31]	50% [30]	Low	Medium [31]
PoW/PoS	Public	High/low [32]	Yes [33]	$O(n)$ [32]	50% [32]	High	High [32]
Proof of storage	Public	High/low [34]	Yes [33]	$O(n)$ [34]	50% [34]	Low	High [34]

a) The ability to tackle byzantine fault.
b) n indicates the number of participants.
c) The given percentage stands for the maximum acceptable faulty nodes or attack.

the infrastructure to trade without further efforts to be centrally managed. Later, such an ecosystem incubates the Blockchain as an Infrastructure (BaaI), which provides a solid toolchain for settlements between the producer, the trader, and the consumer, as shown in Figure 9.1.

As seen in Figure 9.1, blockchains can be the information backbone of a locally distributed resource management system that organizes the customers and producers in an open, transparent market, breaking up the information barriers to publicize the resources, and accelerate the flow of transactions.

The blockchain has incubated the new horizon of resource trading for fixed assets, such as licensed spectrum and computing hardware. In our proposed blockchain 6G resource management scheme, trade-able spectrum, and computing resources are integrated parts of resource pools, where spectrum is dynamically allocated, and network slices are managed, and the hardware is virtualized to facilitate blockchain-enabled resource management. The automated blockchain-enabled resource management relies on the programmable blockchain functionality, which in most cases is described as a smart contract[3]. The contract's content is transparent for both public and agreement making parties, making it publicly traceable. The virtual machine concept is used in the smart contract executions, where the code will be executed by a node on the virtual stack, and its results will be stored on the chain as a transaction record. The tamper-proof ability and fully automated process give the contract high immutability against breaches of the contract and misrepresentations.

9.3.2 Impact of Consensus and Security Performance

If the impressive and resistive data structure of the blockchain is the façade of a building, the consensus is its pillars.

Blockchain has various options on the CM. Choosing a suitable consensus for 6G resource management is the most critical step of making a secured and efficient future-proof the blockchain system. As the CM, which ensures an unambiguous ordering of transactions, and guarantees the integrity and consistency of the blockchain across geographically distributed nodes, is of importance to blockchains since it determines its performance in terms of TPS, delay, node scalability, security, etc.

Depending on the access criteria, the chain can be divided into public or private ones. The public chain is permission-less, which uses proof-based consensus to provide a secure, reliable network for every participant without requiring their identities at entry points. In the 6G resource pool, there are potential anonymous

3 The smart contract is essentially an executable program code stored on the chain, representing terms of agreements triggered automatically when certain conditions are met [68].

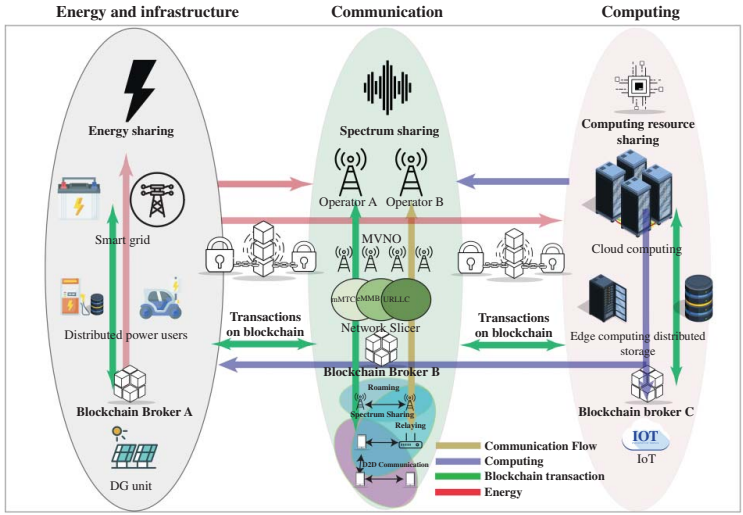


Figure 9.1 Blockchain-enabled resource management framework.

clients, and providers on ad hoc basis [5]. The benefit of adopting a public chain is significant for ad hoc networks, where the barriers of identification and security are broken down for panoptic information exchanges.

As such, public chains can potentially promote the efficiency of the community and regulate the order of participants [31]. However, if participants are concealed, violations and malicious activity are emerging threats to the system. The consortium/private chain, in contrast, is permissioned, meaning that the entry is controlled. It has a rather stable community composition, where the identity of the participant is not kept secret. The network faces fewer threats from unknown attacks but has challenges within the network, for instance, the malicious byzantine node.[4]

Before adapting to any new technologies, security and reliability are always the principal concerns. Blockchain technology is born to outperform existing solutions regarding security performance and robustness. Table 9.1 shows the comparison between widely used CMs of the blockchain regarding six aspects: latency, TPS, complexity, security, energy consumption, and scalability. As it can be seen, private/consortium consensuses show better latency, TPS, and energy consumption performance alongside lower ability to scale up; however, the private/consortium blockchain applications prioritize latency and TPS over scalability. On the other hand, proof-based mechanisms have decent performance on scalability, but sacrifice latency and TPS. In some cases, like proof of work, it also consumes a huge amount of power. However, their good performance in terms of scalability gives them the capability to grow fast in the public network, and it does not suffer from a surge of users, which makes them excellent at mass market trading and distributed file storage system. Regarding security performance, it is worth noting that the non-byzantine consensuses assume non-malicious activities, but byzantine consensus has tolerance not only against inactivity but also against false and erroneous messages. PBFT functions with less than $(n-1)/3$ byzantine nodes, and some variants of PBFT provide higher tolerance with trades-off of latency, such as multi-layer PBFT [33].

Besides the consensus, which secures the blockchain from top-level threats, the communication link should be hardened to prevent external security breaches. The wireless communication is in peril of jamming and spoofing because of open channels. In the practice of wireless blockchain network, the communication failure will result in the node failure, thus lowering the security level. To mitigate the transmission success rate, a collision avoidance mechanism, such as Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA), and physical layer security can be considered.

---

4 A byzantine node is a malicious node that conceals its existence, and tempers the consensus, which tampers with the security of network.

9.4 Application Scenarios

9.4.1 IoT and D2D Communications

The IoT is a paradigm which envisions that all our daily objects and appliances will be connected to each other, collecting and sharing information. This will allow the automation of specific tasks, and enable several other applications to emerge, such as smart homes, smart transportation, wearable devices, smart farming, healthcare [34, 35], and machine-to-machine communications, among others [36].

In order to reach such automation and growth, it is necessary to have proper standards and protocols for IoT devices. However, current solutions still rely on centralized models, which incurs high maintenance costs for manufacturers, while consumers also lack trust in these devices. Combined with the resource constraints of IoT devices, privacy and security concerns, as well as poor interoperability among different vendors make IoT a challenging domain [37, 38]. Similarly, D2D communications, a paradigm that envisions the communications and share of data between devices, also share similar challenges to the IoT [39]. For example, mobile devices are constrained by battery, while security is an ever-present concern in mobile communications. Moreover, in order to fully enable D2D communications, a proper incentive for trading and sharing resources, such as power or data is needed, as current D2D paradigms lack the motivation to do so [39]. Besides, the benefit of making use of existing RAN as a regional service controller without the central intervention is also an appealing feature in the case of D2D IoT deployment, and the potential of standalone RAN operation can be made possible under the scope of BE-RAN [40], a blockchain solution for O-RAN to utilize resources more locally.

In this context, the blockchain is an excellent complement to both IoT and D2D communications, as it can provide the underlying infrastructure with improved interoperability, privacy, reliability, and scalability [38]. For example, in the context of resource management, blockchains can be used to perform spectrum sharing and record all the spectrum utilization and lease requests [15] between IoT devices and the RAN. Moreover, it can provide the incentive needed for devices to share and trade resources, as current protocols lack the incentive to do so. Integrating the blockchain into the IoT and D2D with or without help of local O-RAN elements can provide rewards every time devices share their power or data, allowing for a more cooperative and trusted network environment [28, 38]. Furthermore, this reward mechanism can also be applied in the context of spectrum sharing, in which whenever a user leases spectrum to another, a reward can be assigned, creating a more collaborative environment and improving spectrum efficiency [14, 15]. In addition, blockchains can be utilized in the realm of Vehicular-to-Anything (V2X) communications by encouraging vehicles to

trade energy or information with each other [17]. Last but not least, another key aspect of V2X communications is how to guarantee a secure communication between vehicles and Public Key Infrastructures (PKI) or to run the network without any certification or a centralized PKI. In this context, the blockchain can be utilized as the infrastructure to provide secure and private communications to the PKI or act like a PKI, or also the communication medium between PKIs from different vendors [28]. O-RAN benefits the local distributed systems by providing its RAN Intelligent Controller (RIC) to its RAN elements, and making blockchain functions integration as simple as adding an xAPP [7].

However, despite all of these benefits, the integration of blockchains in the IoT and D2D domain is still challenging [15, 17]. In the case of public chains, for example, the decentralized CMs often require extensive computing power from network nodes (such as PoW-based blockchains). This can be a problem, as most IoT devices are power-constrained. This is especially true for devices powered by cellular IoT, which can be deployed in very remote or difficult to access areas and are expected to have over ten years of battery life [41].

Thus, the utilization of the blockchain in cellular IoT, especially when considering the computation of the consensus algorithm, can significantly reduce the lifetime of cellular IoT devices, limiting their communication capabilities and effectiveness.

As such, it is still unclear how the generation of the PoW could be done when integrating public blockchains with IoT or D2D communications [38]. Hence, other CMs, such as PBFT, are being proposed in the context of IoT applications [38, 42, 43]. One possible solution powered by O-RAN architecture is to have the heavy-duty blockchain nodes deployed at the RIC or MEC servers which are locally accessible to all clients under the same cell/RAN coverage.

Another challenge in integrating the blockchain into small devices comes due to their limited memory capabilities. Since in the blockchain every node needs to have a record of all the current and previous blocks in the chain, it can be infeasible to store such a huge amount of data in IoT devices. Thus, it is still not clear how the blockchain can be fully integrated in IoT. Moreover, the blockchain still has privacy issues, as other studies have shown by other studies that by analyzing transaction patterns, identities of users could be inferred [17].

On top of that, it is also known that the blockchain introduces delay, due to its decentralized approach and its CMs. As such, this additional delay might also impact the performance of certain wireless communication use-cases, such as in V2X, industrial applications, or D2D and it is still an area to be investigated. Moreover, in V2X scenarios, information security and resilience are critical, since any small failure can lead to catastrophic and even fatal consequences. In those cases, the blockchain can provide an additional secure layer for vehicles

to perform key management exchange, as in [44], or even to protect a vehicle's identity and location, in what is known as pseudonym management [45].

Lastly, another important challenge in the realm of wireless blockchain, which has not been largely explored is how the performance of the wireless link affects the performance of the blockchain [18].

Despite recent works investigating the suitability of the CSMA/CA protocol in wireless blockchain networks [46], or the security performance and optimal node deployment of blockchain-enabled IoT systems [47], more researches are needed in this area [48].

9.4.2 Network Slicing

Network slicing is an up-and-coming technology in future cellular architecture and emerging ones, e.g. O-RAN solutions with help of RIC [7], and it is aimed at meeting the diverse requirements of different vertical industry services. Network slicing is a specific form of virtualization that allows multiple logical networks to run on top of a shared physical network infrastructure [7]. A network slice is realized when several Virtualized Network Functions (VNF) are chained-based on well-defined service requirements, such as the massive Machine-Type Communication (mMTC), enhanced Mobile Broadband (eMBB), and the ultra-Reliable Low Latency Communication (uRLLC). The management and orchestration of network slices must be trusted and well secured, in particular for accommodating applications that require high security, such as in the case of remote robotic surgery and V2X communications.

Network slicing also enables Mobile Network Operators (MNOs) to slice a single physical network into multiple virtual networks which are optimized based on specified business and service goals [49]. Hence the term Mobile Virtual Network Operators (MVNOs). The implementation of MVNOs necessitates the integration of a network slice broker into the architecture, as seen in Figure 9.2.

9.4.3 Network Slicing Broker

The aim of a network slice broker is to enable MVNOs, industry vertical market players, and Over-The-Top (OTT) providers to dynamically request and release the network resources from the infrastructure provider entity based on their needs [50]. The network slicing brokering relies on the ability of the MNO/Communication Service Providers (CSP) to automatically and easily negotiate with the requests of the external tenants of the network slice based on the currently available resources with the infrastructure provider. In [50], the authors proposed the concept of 5G network slice broker that could lease network resources on-demand basis; also the proposed feature may perform under the RIC with O-RAN compliance.

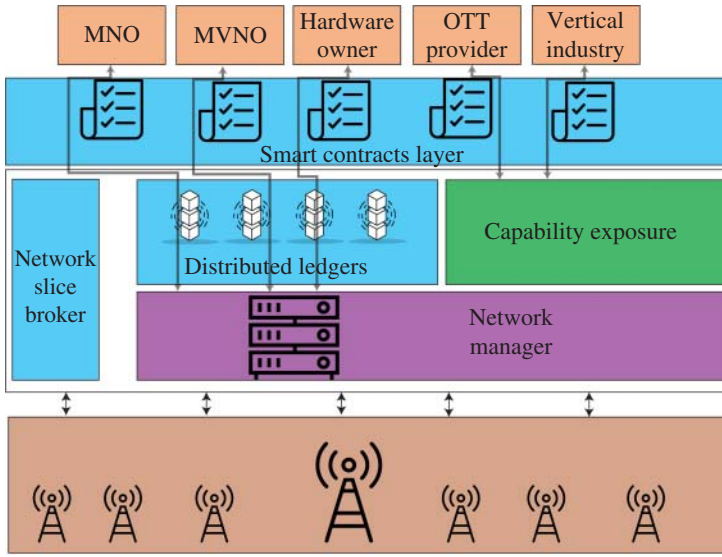


Figure 9.2 Spectrum management using the blockchain and smart contract.

The 3GPP's study on orchestration and management of network slicing for 5G and beyond networks indicated the establishment of mutual trust between the actors (MVNOs, MNOs, OTT providers) as a prerequisite for an effective and efficient multi-operator slice creation [51], and O-RAN architecture has introduced RIC to provide functional support [7]. Hence, trust and security are important factors to be considered in the implementation, coordination, design, and integration of a network work slice broker.

9.4.4 Integration of Blockchain to Network Slicing and Resource Brokerage

A major challenge associated with network slicing and resource brokerage is the need to keep a transparent, fair, and open system within the available number of resources and several suspicious players.

Blockchain and Distributed Ledger Technology (DLT) functionalities can be utilized to address the aforementioned trust and security issues associated with the implementation of network slicing either for the coexistence of various applications and services, or for both the service and operational use-cases of CSPs. The trading of a network slice can be offered by blockchain-based nodes at RIC where the blockchain smart contract orders the slice orchestration based on the agreed SLA from the network slice broker. Blockchain can be integrated for taking the record of how each resource has been used and how each service

provider has performed against the SLA. Blockchain combines a distributed network structure, CM, and advanced cryptography to present promising features which are not available in the existing structures. The key gain that is achieved through the blockchain is the integration of the trust layer which lowers the collaboration/cooperation barrier and enables an effective and efficient ecosystem. Further, the distributed nature of the blockchain prevents the single point of failure problem and thus enhances security.

Figure 9.3 illustrates the provision of the remote surgery/consultation and remote control of drones over a long distance (with network operators in different geographical regions), while leveraging on network slicing and blockchain technologies. Here, a blockchain-based approach is used to automate the reconciliation and the payment between provider in different locations. Without this approach, a more costly manual intervention or the integration of a third party for settlement would be required.

Blockchain can also enable the seamless access of devices to a diverse number of networks. However, this might require the network provider to manage rules, protocols, and transactions at an increasing number of access points. Blockchain can play a reinforcing role, such as in the case of auditing agreement. Once the information is stored on a blockchain, it can be operated through "smart contracts" [29].

In [52], the authors proposed a model where brokering is managed by the 5G network slice broker [50] while the payout, billing, and leasing are managed by the blockchain-based slice leasing ledger which is incorporated in the service layer.

In [53], the authors introduced a blockchain-based hierarchical network slicing framework that can support the recent evolution in the telecommunication sector business model. In particular, the intermediate broker is introduced into the network slicing model in [50]. A blockchain-based brokering solution is developed at the intermediate level which then enables the infrastructure providers to allocate the network resources to the intermediate brokers using smart contracts. The intermediate brokers can automatically redistribute and allocate resources among the tenants in a scalable and secured manner. Note that the agreements between the MVNO and the infrastructure providers are the conventional offline-based signature process. With the approach proposed in [53], the resource allocation transaction can be recorded through a cost-effective online digital signature process which is much faster and highly scalable.

In [54], building on the work in [55], the authors developed a framework that leverages blockchain to allow for network slice providers to build a secured end-to-end network slice while using the network resources from the different 5G network stakeholders. Here, end-to-end slice request is received by the slice provider who then publishes a request for resources for each of the sub-slices that makes up the end-to-end slice in the blockchain. The slice provider then selects

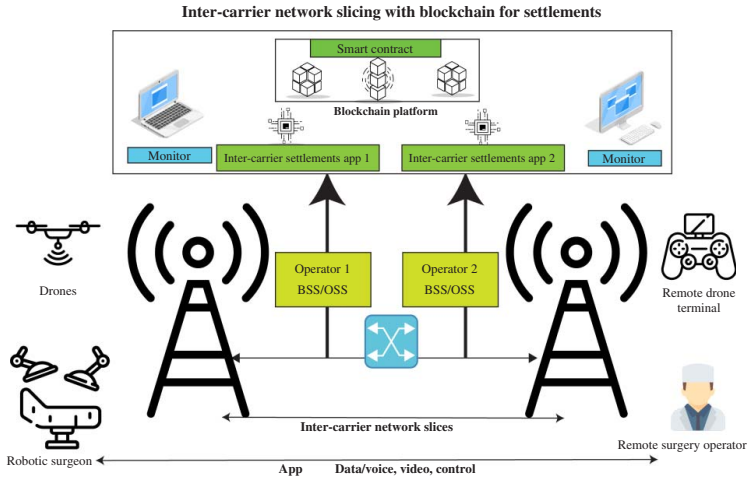


Figure 9.3 Network slicing applied with the use of the blockchain.

from the received offers for each sub-slice with the objective to minimize cost and meet the quality-of-service requirement.

To enable an on-the-fly leasing of resources toward service continuity and meeting the end-to-end quality of service requirement, the authors in [56] proposed a distributed blockchain-enabled network slicing (DBNS). The architecture is both distributed and on-demand based, and it is enabled by global service provision. The latter provides admission control for service requests and a dynamic allocation of resources through a bidding scheme that is blockchain-based.

The performance analysis of a real-world 5G network with blockchain integrated within its operation for both security and transparency is presented in [57]. The latency awareness of the network slices and the user equipment state-based resource allocation by the telecommunication network provider are both accounted for using blockchain. The performance evaluation in [57] showed that the blockchain-based framework can improve the network resource efficiency and the security of the entire system.

Blockchain can enable secure and automated brokerage of network slicing while proving the following gains:

- Significant savings in the operational (transaction and coordination) cost;
- Speed up the slice negotiation process leading to a fall in the cost of slicing agreement;
- Increased efficiency of operation for each network slice [58];
- Increased security of the network slice transactions;
- The creation of a blockchain-enabled contract for MVNOs and MNOs that cannot afford the required network capital investment which could be on the high side. In particular, the frequency spectrum could be leased by large operators or players on a pay-as-you-go basis or in real time.

Blockchain can also enhance the enforcement of quite straightforward agreement which are related to many brokering operations. Furthermore, the negotiation on SLAs can be more efficient when pricing and Quality of Service (QoS) levels are identified as smart contract parameters.

Other opportunities associated with the blockchain in the next-generation networks include:

- The settlement of transactions between multiple carriers, including voice transactions and Call Detail Records (CDRs) of all involved call participants;
- Managing the Service-Level Agreement (SLA);
- Simplification of roaming terms and agreements between multiple operators;
- Managing money transfers across borders and cross-carrier payment platforms;
- Managing user/nodes identity and authentication process;
- Managing Licensed Spectrum Access (LSA) via the blockchain-based carrier marketplace.

9.4.5 Inter-Domain Blockchain Ecosystem

In a blockchain-based ecosystem, we can find various streams of the blockchain transaction, energy, and computing flows using shared communication assets in the resource management scheme, as seen in Figure 9.1. Arrows in Figure 9.1 represent the flow directions and they are started with the provider, through the inter-domain sharing scheme to reach the final consumers at both local level with consortium blockchain and national or global level via public blockchain. The ecosystem is not limited to the scope of energy, communication, and computing as it can expand itself to a wider range through cross-field integration to reach, for instance, automotive, finance, manufacturing, logistic chain, and so on.

Organizations that intend to fuse such resources can be recognized as Virtual Infrastructure Operators (VIOs), since they do not own all the resources but a vendor of combined sets of resources.

An example of VIO can be found in remote regions, where local infrastructure investors tend to have off-grid Distributed Generation (DG) units [59], for instance, solar and wind farms and micro Combined Heat and Power (microCHP) to offer energy and heat to remote users in the form of Distributed Energy Resources (DER) [60]. A local-based integration of such resources, also known as a Virtual Power Plant (VPP), plays the role of the vendors for electricity and heat and buys from or sells to other grids, with unfilled demands and excess electricity.

Since these establishments are far away from the central network and lack a cost-effective way of trading regarding the communication and delivery cost, it is ideal to broke with other local providers and exchange the electricity for other goods.

For example, the communication relay service and computing service of DG sensor are used as the exchange of hardware power supply, to cultivate the ecosystem while the internal demand grows. In addition to the resources owned by the operator, there are many common resource containers among all participants.

However, the blockchain ecosystem has to accommodate the performance and security requirements of the intended application. In terms of the performance and security, the consensus is the major concern in the phase of planning. Different consensuses can be applied to the sharing scheme. For example, a public chain is more suitable for inter-domain transactions on top-level operators like the national grid and first-tier MNO. However, if the resources are local-oriented, the private chain can be hosted for IoT and local/off-grid nodes, where the information from a private chain is kept within the network with confidence for external auditing. An ecosystem may introduce multiple consensuses on different chains to achieve its best results.

Beyond the deployment of blockchains, the actual hardware plays an important role in the ecosystem; as current blockchain applications are designed for

upper-layer applications, it lacks the understanding of the portable solutions for mobile device, such as drones, cars, and IoT. It is worth noting that the wireless capability for the blockchain is essential in 6G deployment. Wireless blockchain-enabled nodes empower the Machine-to-Machine (M2M) trade among distributed and shared resources; therefore, it becomes essential that the remote nodes are wireless enabled. In the near future, the VANET-enabled car equipped with blockchain nodes can recharge the battery from multiple wireless charging points while moving and trade the information it carries, for instance, the Light Detection and Ranging (LiDAR) mapping data, relaying the internet access, edge computing resources, and anything that can be used by the remote DG unit using wireless communication, D2D, and edge computing. The transactions are kept in the blockchain and carried by the vehicular network or local RAN then mined by the local infrastructure or base station blockchain nodes. Later, the mined blocks will be relayed by satellite-linked base stations for a fee [61]. The auction of spectrum and network slices can be found on data relay and short-range Vehicle to Ground (V2G) communication, which requires huge local bandwidth to achieve lower latency. Besides, the wireless blockchain nodes can participate in decision-making process for reliability improvement, where the decision can be made more reliable by contacting multiple agents [62]. This example intends to give an insight of inter-domain blockchain ecosystem, and further additional features are all made possible based on the inter-domain transactions.

9.4.6 Blockchain Introduction on Mutual Authentication, Identities, and Certifications for O-RAN

RAN tends to be more distributed in the 5G new radio in order to provide low latency and flexible on-demanding services. Blockchain has been widely considered as an access approach [40, 63] inspired by the distributed attribute of decentralized network. In the recent effort of integrating blockchain into RAN, B-RAN [63] and C-RAN with blockchain (BC-RAN) [64], showed a comprehensive understanding of blockchain potential on RAN, where the value of trust-free and smart contracts is deeply acknowledged. Taking the BE-RAN core functions: identity management and mutual authentication as an example, the blockchain is taken as a medium of identity exchange, where the management is purely user-centric and zero-trust. With the identification exchanged over blockchain, BE-RAN conducts zero-trust mutual authentication over the unique blockchain-enabled routers and switch functions.

In this section, Blockchain-enabled Radio Access Networks (BE-RAN) is illustrated as a novel decentralized RAN architecture to facilitate enhanced security and privacy on identification and authentication, as shown in Figure 9.4.

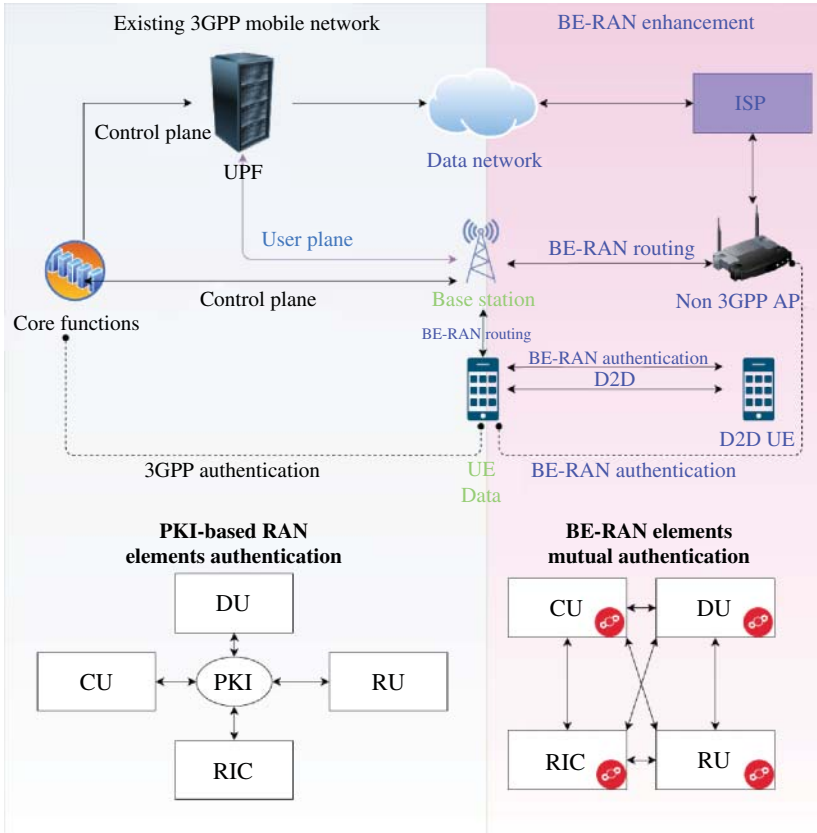


Figure 9.4 Overview of BE-RAN in addition to existing mobile network. Source: [40]/IEEE.

It can offer user-centric identity management for UE and RAN elements and mutual authentication to all entities while enabling on-demand point-to-point communication with accountable billing service add-on to the general public. In the sense of enabling localized RAN operation, resources including D2D, non-3GPP Access like Wi-Fi can be integrated into a local regional network secured by Blockchain to assist cellular network and offload traffic from core network.

The current state of the art RAN opt-in PKI-based encryption and authentication solutions and transferring the trust ground of elements into zero-trust framework by O-RAN alliance. Blockchain kicks in RAN security with its unique features of decentralized authentication protocols, as the conventional PKI solution requires a centralized or federated CA (Certification Authority) as a trusted party to provide

identity notary. As the CA is a third-party service and possesses the credential and privacy of users of PKI, it makes the CA a vulnerable target for malicious activities, and the communication outage with CA will result in catastrophic consequences for identity authentication and communication encryption.

To overcome the single-point outage and privacy concerns, mutual authentication powered by the mechanism of the blockchain network is appealing to users and networks. By enabling blockchain in the network, the UE or element can easily request authentication with the known blockchain address of other side users or elements, and complete authentication by verifying the signature of both private keys. Note that the cellular network security consists of UE security, RAN security, and CN (Core Network) security, and the scope of BE-RAN covers UE and RAN prospects, with potential security aid to CN security.

O-RAN security framework has regulated the impacts to all O-RAN interfaces as per mentioned in O-RAN security analysis [65]. In the adoption of BE-RAN, the added functions and interfaces have additional impacts on existing descriptions for BE-RAN mutual authentication protocol. Mutual authentication indicates that two individuals have authenticated each other at the same time before transmitting any data. BE-RAN mutual authentication takes full advantage of blockchain features, by binding the (global) Blockchain address (BC ADD) with the actual (local and temporary) O-RAN elements addresses (ADD). The mutual authentication is user-centric with their independent identity management, and it is an inclusive solution in addition to traditional mutual authentication, which is through the CN or third-party PKI, e.g. password-based scheme and certificate-based scheme. The distributed mutual authentication facilitates not only authentications at RAN, but also applies to wider application scenarios, with the power of distributed identity management (Figure 9.5).

By completing mutual authentication without a third-party involvement, the privacy of users is kept to their own with easy identity management, as the

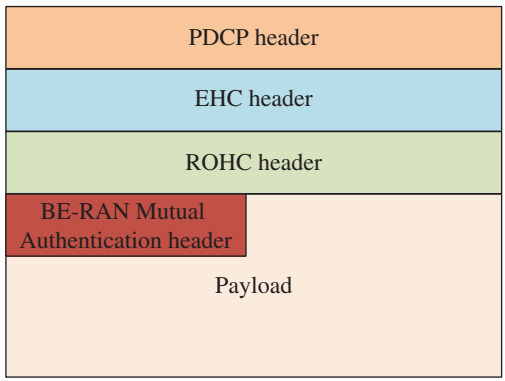


Figure 9.5 A modified PDCP data with BE-RAN integration.

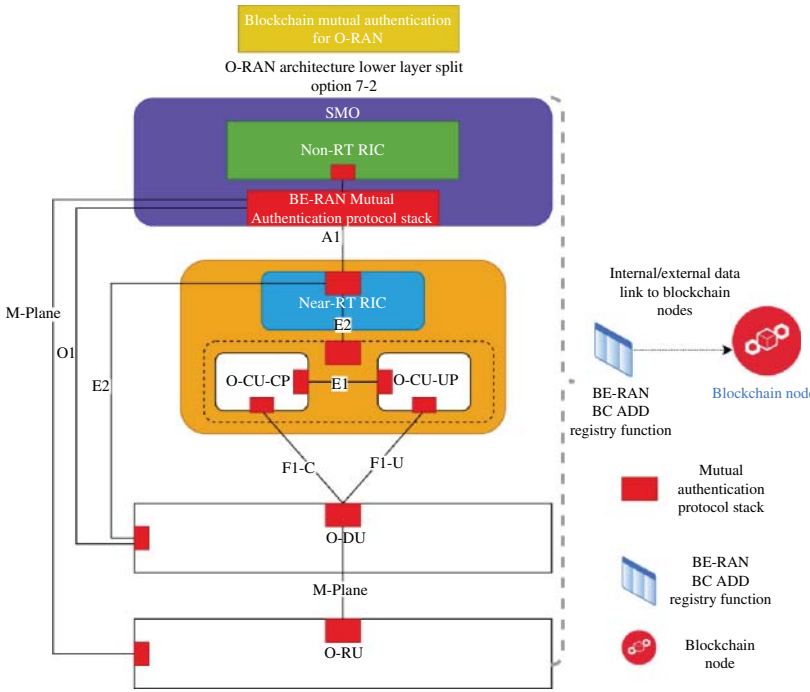


Figure 9.6 O-RAN integration of mutual authentication protocols.

blockchain address can be kept as a mobile phone number or email address in the era of decentralization.

BE-RAN mutual authentication is a great help to solve the identity crisis in cybersecurity, and communication security, with the example of deploying BE-RAN protocol stacks on the top of O-RAN interfaces and elements, as shown in Figure 9.6. RAN elements with BE-RAN can easily verify other RAN elements within or among other RAN networks, as the RAN elements are now globally authenticated. A great use case of it would be stopping fake base stations, for example. Once the UE has a list of trusted base stations from a trusted blockchain ledger maintained by communities and MNOs (the tamper-proof ability and consensus will ensure the correctness of record), it helps counter fake base station issues.

9.4.6.1 O-RAN Common Protocol Stack Integration of PDCP

Since the BE-RAN mutual authentication protocol is designed as an add-on security feature, it has impact on all interfaces that choose to have distributed accessibility. BE-RAN protocol impacts the interfaces by inserting the authentication protocol into their transportation protocols at their headers and preambles, taking the most

front part of original protocols' payload [66]. The control plane and user plane can both benefit from the common BE-RAN mutual authentication integration.

BE-RAN introduces blockchain node into the RAN elements, as either a co-hosted service on the virtual machine or containers via internal data link shared by all logical units, or a standalone blockchain node communicable via communication interfaces, e.g. HTTPS, UDP, and TCP/IP.

In O-RAN architecture, PDCP is a common part of protocol stacks existing in both control plane and user plane, which makes it an ideal candidate for BE-RAN integration. With the compatibility with its existing integrity check and ciphering function, the blockchain address provides the PDCP with mutual authentication from the initiator, instead of upper layer functions.

9.4.6.2 O-RAN Interface Integration Scenario

O-RAN interfaces like E2, E1, O1, F1, A1 share a similar protocol stack in terms of transport network layer. Hence the common protocols including SCTP, IP, and Ethernet can be found in its underlaying data structure. An example of E2 interfaces in Figure 9.7 demonstrates a fairly common protocol stack used not particularly in E2 but also in many other interfaces both 3GPP defined (Xn, E1, F1, etc.) and O-RAN alliance defined (E2, O1, A1, M-plane, etc.). An example of protocol adaption can be found in Figure 9.8, where the BE-RAN mutual authentication protocol is added behind each transport protocols header.

9.4.7 Challenges of Applying the Blockchain Technology in Resource Sharing and Spectrum Management

Though the blockchain has many advantages, some features need to be eliminated when applied to the resource sharing and spectrum management scenarios.

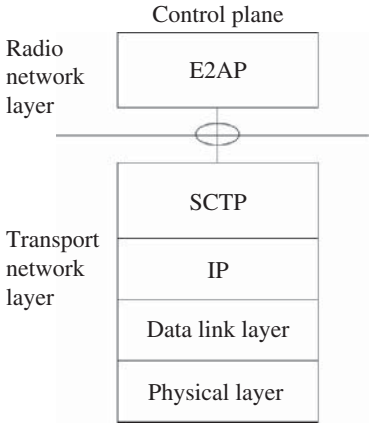


Figure 9.7 E2AP protocol stack [3].

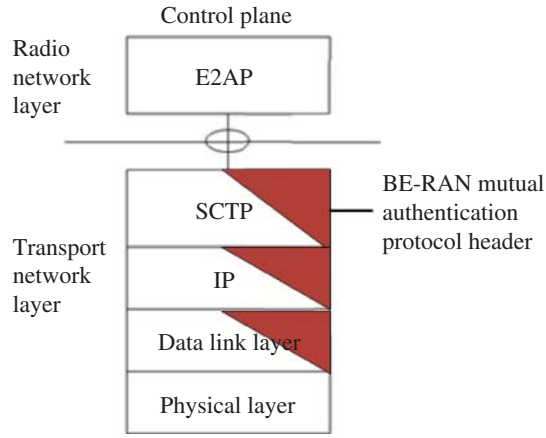


Figure 9.8 E2AP protocol stack with BE-RAN.

Here we highlight some of the challenges of applying the blockchain technology in resource sharing and spectrum management.

- *Storage*: Each replica node in the conventional blockchain network must process and store a copy of the completed transaction data. This can give rise to both storage and computation burden on IoT devices, which are generally resource constrained, thus limiting their participation in the blockchain network.
- *Underlying networking*: Implementing a consensus mechanism within the blockchain is computationally expensive and it also requires significant bandwidth resources. Meanwhile, resources are very limited in future network; thus, meeting the resource requirement for large transaction throughput might be hard to achieve with the current system.
- *Scalability of the blockchain network*: The scalability of the blockchain network is a serious issue in current systems. The number of replicas in the blockchain network relates directly to the throughput (i.e. number of transactions per second) and latency (i.e. the time required to add a transaction to the blockchain). Hence, sustaining the huge volume of transactions expected in blockchain-enabled future networks demands solutions for improving the throughput of the blockchain system. There are solutions of employing multilayered PBFT [43] and DAG [38] to help with throughput issues that are yet to be seen in applications.

9.5 Conclusions

In this chapter, a blockchain-enabled 6G resource management with emerging open radio access network, spectrum sharing, and computing and energy trading was envisioned as an enabler for future use-cases. We first briefly introduced

the current spectrum management and allocation techniques and discussed the link between the blockchain and spectrum management. We have then given the motivation behind the blockchain as well as an overview of its fundamentals. Moreover, we have discussed a set of key applications of the blockchain and the transformation that brings to the current wireless networks and O-RAN. The discussed applications include IoT and D2D communications, network slicing, and the inter-domain blockchain ecosystem.

To enable the full ecosystem and manage the resource for 6G, we have identified the following open problems:

(1) development of lightweight blockchain solutions for low-cost IoT devices;
(2) high-performance blockchain and decentralization for the vertical industries and future networks;
(3) Development of blockchain solutions ecosystem by considering the security and privacy issues;
(4) implementation of blockchain protocols over the wireless channel and evaluating fundamental limits relating to the performance and security.

Acronyms

ADD	address
AI	artificial intelligence
BaaI	blockchain as an infrastructure
BaaS	blockchain as a service
CA	certification authority
CDR	call detail records
CM	consensus mechanism
CN	core network
CSMA	carrier sensing multiple access
CSP	communication service providers
CU	centralized unit
D2D	device-to-device
DAG	directed acyclic graph
DBNS	distributed blockchain-enabled network slicing
DER	distributed energy resources
DG	distributed generation
DLT	distributed ledger technology
DSM	dynamic spectrum management
DU	distributed unit
eMMB	enhanced mobile broadband
FCC	federal communications commission

IaaS	infrastructure as a service
IoE	internet of everything
IoT	Internet of Things
IP	internet protocol
LiDAR	light detection and ranging
LSA	licensed spectrum access
M2M	machine-to-machine
MEC	mobile edge computing
MIMO	multiple input massive output
mMTC	massive machine type communication
MNO	mobile network operator
MVNO	mobile virtual network operator
NR	new radio
OTT	over the top
OSA	opportunistic spectrum access
PBFT	practical byzantine fault tolerance
PDCP	packet data convergence protocol
PKI	public key infrastructure
PoS	proof of stake
PoW	proof of work
QoS	quality of service
RAN	radio access network
RIC	RAN intelligent controller
RU	radio unit
SLA	service-level agreement
SON	self-organizing network
TPS	transactions per second
TVWS	television white space
UAV	unmanned aerial vehicles
UE	user equipment
uRLLC	ultra-reliable low latency communication
V2G	vehicle to ground
V2X	vehicular-to-anything
VIO	virtual infrastructure operator
VPP	virtual power plant

References

1 Andrews, J.G. et al. (2014). What will 5G be? *IEEE Journal on Selected Areas in Communications* 32 (6): 1065–1082. https://doi.org/10.1109/JSAC.2014.2328098.

2 Saad, W., Bennis, M., and Chen, M. (2019). A vision of 6G wireless systems: applications, trends, technologies, and open research problems. *IEEE Network* https://doi.org/10.1109/MNET.001.1900287.

3 Han, S. and Bian, S. (2020). Energy-efficient 5G for a greener future. *Nature Electronics* 3 (4): 182–184. https://doi.org/10.1038/s41928-020-0404-1.

4 Tariq, F., Khandaker, M., Wong, K.-K. et al. (2019). A speculative study on 6G. *IEEE Wireless Communications* 27 (4): 118–125. http://arxiv.org/abs/1902.06700.

5 Zhang, Z., Xiao, Y., Ma, Z., Xiao, M., Ding, Z., Lei, X., Karagiannidis, G.K et al. (2019). 6G wireless networks: vision, requirements, architecture, and key technologies. *IEEE Vehicular Technology Magazine* 14 (3): 28–41.

6 Yang, P., Xiao, Y., Xiao, M., and Li, S. (2019). 6G wireless communications: Vision and potential techniques. *IEEE Network* 33 (4): 70–75.

7 Chih-Lin, I., Kuklinski, S., Chen, T., and Ladid, L. (2020). A perspective of O-RAN integration with MEC, SON, and network slicing in the 5G era. *IEEE Network* 34 (6): 3–4. https://doi.org/10.1109/MNET.2020.9277891.

8 You, X., Wang, C.X., Huang, J. et al. (2021). Towards 6G wireless communication networks: vision, enabling technologies, and new paradigm shifts. *Science China Information Sciences* 64 (1): 110301. https://doi.org/10.1007/s11432-020-2955-6.

9 Chih-Lin, I., Rowell, C., Han, S. et al. (2014). Toward green and soft: a 5G perspective. *IEEE Communications Magazine* 52 (2): 66–73. https://doi.org/10.1109/MCOM.2014.6736745.

10 Chih-Lin, I., Han, S., Xu, Z. et al. (2016). New paradigm of 5G wireless internet. *IEEE Journal on Selected Areas in Communications* 34 (3): 474–482. https://doi.org/10.1109/JSAC.2016.2525739.

11 China Mobile Research Institute (2011). C-RAN: the road towards green RAN.

12 David, K. and Berndt, H. (2018). 6G vision and requirements: is there any need for beyond 5G? *IEEE Vehicular Technology Magazine* 13 (3): 72–80. https://doi.org/10.1109/MVT.2018.2848498.

13 Chen, J., Wei, Z., Li, S., and Cao, B. (2020). Artificial intelligence aided joint bitrate selection and radio resource allocation for adaptive video streaming over F-RAN. *IEEE Wireless Communications* 27: 36–43.

14 Kotobi, K. and Bilen, S.G. (2018). Secure blockchains for dynamic spectrum access: a decentralized database in moving cognitive radio networks enhances security and user access. *IEEE Vehicular Technology Magazine* 13 (1): 32–39. https://doi.org/10.1109/MVT.2017.2740458.

15 Weiss, M.B.H., Werbach, K., Sicker, D.C., and Bastidas, C.E.C. (2019). On the application of blockchains to spectrum management. *IEEE Transactions on Cognitive Communications and Networks* 5 (2): 193–205. https://doi.org/10.1109/TCCN.2019.2914052.

16 Dai, Y., Xu, D., Maharjan, S. et al. (2019). Blockchain and deep reinforcement learning empowered intelligent 5G beyond. *IEEE Network* 33 (3): 10–17. https://doi.org/10.1109/MNET.2019.1800376.

17 Dai, H.N., Zheng, Z., and Zhang, Y. (2019). Blockchain for internet of things: a survey. *IEEE Internet of Things Journal* 6 (5): 8076–8094. https://doi.org/10.1109/JIOT.2019.2920987.

18 Sun, Y., Zhang, L., Feng, G. et al. (2019). Performance analysis for blockchain driven wireless IoT systems based on tempo-spatial model. In: *2019 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC)*, 348–353. IEEE https://doi.org/10.1109/CyberC.2019.00066.

19 Li, A., Han, G., Rodrigues, J.J.P.C., and Chan, S. (2017). Channel hopping protocols for dynamic spectrum management in 5G technology. *IEEE Wireless Communications* 24 (5): 102–109. https://doi.org/10.1109/MWC.2017.1700046.

20 Federal Communications Commision (2013). NPRM: Ammendment of the Commission's Rules with Regard to Commercial Operations in 3550–3650 MHz Band.

21 Liang, Y.C. (2020). Blockchain for dynamic spectrum management. *Signals and Communication Technology* 121–146.

22 Eggerton, J. (2010). FCC's rosenworcel talks up 6G. *Multichannel News* https://www.multichannel.com/news/fccs-rosenworcel-talks-up-6g.

23 Gurney, D., Buchwald, G., Ecklund, L. et al. (2008). Geo-location database techniques for incumbent protection in the TV white space. In: *2008 IEEE Symposium on New Frontiers in Dynamic Spectrum Access Networks, DySPAN*, 232–240. *IEEE* https://doi.org/10.1109/DYSPAN.2008.31.

24 Zhang, Y., Lee, C., Niyato, D., and Wang, P. (2013). Auction approaches for resource allocation in wireless systems: a survey. *IEEE Communication Surveys and Tutorials* 15 (3): 1020–1041. https://doi.org/10.1109/SURV.2012.110112.00125.

25 Kotobi, K. and Bilén, S.G. (2017). Blockchain-enabled spectrum access in cognitive radio networks. In: *2017 Wireless Telecommunications Symposium (WTS)*, 1–6. IEEE https://doi.org/10.1109/WTS.2017.7943523.

26 Qiu, J., Grace, D., Ding, G. et al. (2020). Blockchain-based secure spectrum trading for unmanned-aerial-vehicle-assisted cellular networks: an operator's perspective. *IEEE Internet of Things Journal* 7 (1): 451–466. https://doi.org/10.1109/JIOT.2019.2944213.

27 Chai, H., Leng, S., Zhang, K., and Mao, S. (2019). Proof-of-reputation based-consortium blockchain for trust resource sharing in internet of vehicles. *IEEE Access* 7: 175744–175757. https://doi.org/10.1109/ACCESS.2019.2956955.

28 Liu, H., Zhang, Y., and Yang, T. (2018). Blockchain-enabled security in electric vehicles cloud and edge computing. *IEEE Network* 32 (3): 78–83. https://doi.org/10.1109/MNET.2018.1700344.

29 Underwood, S. (2016). Blockchain beyond bitcoin. *Communications of the ACM* 59 (11): 15–17. https://doi.org/10.1145/2994581.

30 Bhattacharya, P., Tanwar, S., Shah, R., and Ladha, A. (2019). *Mobile Edge Computing - Enabled Blockchain Framework - A Survey*, vol. 597. Cham: *Springer*.

31 Foucault, M. (1977). *Discipline and Punish: the Birth of the Prison*. Pantheon Books.

32 Singh, J. and Michels, J.D. (2018). Blockchain as a service (BaaS): providers and trust. In: *2018 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*, 67–74. IEEE https://doi.org/10.1109/EuroSPW.2018.00015.

33 Li, J. and Guo, X. (2020). COVID-19 Contact-tracing Apps: A Survey on the Global Deployment and Challenges, *arXiv Preprint, arXiv2005.03599*.

34 Klaine, P.V., Zhang, L., Zhou, B. et al. (2020). Privacy-preserving contact tracing and public risk assessment using blockchain for COVID-19 pandemic. *IEEE Internet of Things Magazine* 3 (3): 58–63. https://doi.org/10.1109/iotm.0001.2000078.

35 Xu, H., Zhang, L., Onireti, O. et al. (2020). BeepTrace: blockchain-enabled privacy-preserving contact tracing for COVID-19 pandemic and beyond. *IEEE Internet of Things Journal* 8 (5): 1. https://doi.org/10.1109/JIOT.2020.3025953.

36 Fernández-Caramés, T.M. and Fraga-Lamas, P. (2018). A review on the use of blockchain for the internet of things. *IEEE Access* 6: 32979–33001. https://doi.org/10.1109/ACCESS.2018.2842685.

37 Christidis, K. and Devetsikiotis, M. (2016). Blockchains and smart contracts for the internet of things. *IEEE Access* 4: 2292–2303. https://doi.org/10.1109/ACCESS.2016.2566339.

38 Cao, B. et al. (2019). When internet of things meets blockchain: challenges in distributed consensus. *IEEE Network* 33 (6): 1. https://doi.org/10.1109/MNET.2019.1900002.

39 Fodor, G. et al. (2012). Design aspects of network assisted device-to-device communications. *IEEE Communications Magazine* 50 (3): 170–177. https://doi.org/10.1109/MCOM.2012.6163598.

40 Xu, H., Zhang, L., Sun, E., and Chih-Lin, I. (2021). BE-RAN: Blockchain-enabled Open RAN with Decentralized Identity Management and Privacy-Preserving Communication. http://arxiv.org/abs/2101.10856.

41 Mangalvedhe, N., Ratasuk, R., and Ghosh, A. (2016). NB-IoT deployment study for low power wide area cellular IoT. In: *2016 IEEE 27th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, 1–6. IEEE.

42 Onireti, O., Zhang, L., and Imran, M.A. (2019). On the viable area of wireless practical byzantine fault tolerance (PBFT) blockchain networks. In: *2019 IEEE Global Communications Conference (GLOBECOM)*, 1–6. IEEE https://doi.org/10.1109/GLOBECOM38437.2019.9013778.

43 Li, W., Feng, C., Zhang, L. et al. (2021). A scalable multi-layer PBFT consensus for blockchain. *IEEE Transactions on Parallel and Distributed Systems* 32 (5): 1146–1160. https://doi.org/10.1109/TPDS.2020.3042392.

44 Lei, A., Cruickshank, H., Cao, Y. et al. (2017). Blockchain-based dynamic key management for heterogeneous intelligent transportation systems. *IEEE Internet of Things Journal* 4 (6): 1832–1843.

45 Bao, S. et al. (2019). Pseudonym management through blockchain: cost-efficient privacy preservation on intelligent transportation systems. *IEEE Access* 7: 80390–80403. https://doi.org/10.1109/ACCESS.2019.2921605.

46 Cao, B., Li, M., Zhang, L. et al. (2019). How does CSMA/CA affect the performance and security in wireless blockchain networks. *IEEE Transactions on Industrial Informatics* 1. https://doi.org/10.1109/TII.2019.2943694.

47 Sun, Y., Zhang, L., Feng, G. et al. (2019). Blockchain-enabled wireless internet of things: performance analysis and optimal communication node deployment. *IEEE Internet of Things Journal* 6 (3): 5791–5802. https://doi.org/10.1109/JIOT.2019.2905743.

48 Zhang, L., Xu, H., Onireti, O. et al. (2021). How Much Communication Resource is Needed to Run a Wireless Blockchain Network?. http://arxiv.org/abs/.

49 NGMN (2016). Description of network slicing concept by NGMN alliance. In: *Ngmn 5G P1*, vol. 1, 19. https://www.ngmn.org/uploads/media/160113_Network_Slicing_v1_0.pdf.

50 Samdanis, K., Costa-Perez, X., and Sciancalepore, V. (2016). From network sharing to multi-tenancy: the 5G network slice broker. *IEEE Communications Magazine* 54 (7): 32–39. https://doi.org/10.1109/MCOM.2016.7514161.

51 3GPP (2017). Study on Management and Orchestration of Network Slicing for Next Generation Network.

52 Backman, J., Yrjola, S., Valtanen, K., and Mammela, O. (2017). Blockchain network slice broker in 5G: slice leasing in factory of the future use case. In: *Joint 13th CTTE and 10th CMI Conference on Internet of Things - Business Models, Users, and Networks*, vol. 2018, 1–8. https://doi.org/10.1109/CTTE.2017.8260929.

53 Zanzi, L., Albanese, A., Sciancalepore, V., and Costa-Perez, X. (2020). NSBchain: a secure blockchain framework for network slicing brokerage. In: *ICC 2020 - 2020 IEEE International Conference on Communications (ICC)*, 1–7. IEEE https://doi.org/10.1109/ICC40277.2020.9149414.

54 Nour, B., Ksentini, A., Herbaut, N. et al. (2019). A blockchain-based network slice broker for 5G services. *IEEE Networking Letters* 1 (3): 99–102. https://doi.org/10.1109/LNET.2019.2915117.

55 Samad, A., Hayes, S., French, L., and Dodds, S. (2002). Digital imaging versus conventional contact tracing for the objective measurement of venous leg ulcers. *Journal of Wound Care* 11 (4): 137–140.

56 Togou, M.A. et al. (2020). DBNS: a distributed blockchain-enabled network slicing framework for 5G networks. *IEEE Communications Magazine* 58 (11): 90–96. https://doi.org/10.1109/MCOM.001.2000112.

57 Gorla, P., Chamola, V., Hassija, V., and Niyato, D. (2020). Network slicing for 5G with UE State based allocation and blockchain approach. *IEEE Network* 1–7. https://doi.org/10.1109/MNET.011.2000489.

58 Ridgewell, P. (2019). Blockchain: Where's the Value for Telecoms?, pp. 1–39. https://www.accenture.com/_acnmedia/pdf-101/accenture-blockchain-wheres-the-value-for-telecoms.pdf.

59 Talapur, G.G., Suryawanshi, H.M., Xu, L., and Shitole, A.B. (2018). A reliable microgrid with seamless transition between grid connected and islanded mode for residential community with enhanced power quality. *IEEE Transactions on Industry Applications* 54 (5): 5246–5255. https://doi.org/10.1109/TIA.2018.2808482.

60 Landsbergen, P. (2007). Feasibility, beneficiality, and institutional compatibility of a micro-CHP virtual power plant in the Netherlands. *Energy* 28 (22): 13. https://www.semanticscholar.org/paper/Feasibility%2C-beneficiality%2C-and-institutional-of-a-Landsbergen/df67fad3f4ef94f32fd8feef092f75f3f677e412 (accessed 26 December 2019).

61 Yaacoub, E. and Alouini, M.-S. (2019). A Key 6G Challenge and Opportunity – Connecting the Remaining 4 Billions: A Survey on Rural Connectivity. http://arxiv.org/abs/1906.11541 (accessed 26 December 2019).

62 Yu, D., Li, W., Xu, H., and Zhang, L. (2021). Low reliable and low latency communications for mission critical distributed industrial internet of things. *IEEE Communications Letters* 25 (1): 313–317. https://doi.org/10.1109/LCOMM.2020.3021367.

63 Ling, X., Wang, J., Bouchoucha, T. et al. (2019). Blockchain radio access network (B-RAN): towards decentralized secure radio access paradigm. *IEEE Access* 7: 9714–9723. https://doi.org/10.1109/ACCESS.2018.2890557.

64 Tong, W., Dong, X., Shen, Y., and Zheng, J. (2020). BC-RAN: cloud radio access network enabled by blockchain for 5G. *Computer Communications* 162: 179–186. https://doi.org/10.1016/j.comcom.2020.08.020.

65 O-RAN Alliance (2021). O-RAN Security Analysis.

66 3GPP (2017). TS 136 323 - V14.3.0 - LTE; Evolved Universal Terrestrial Radio Access (E-UTRA); Packet Data Convergence Protocol (PDCP) specification (3GPP TS 36.323 version 14.3.0 Release 14), vol. 0, pp. 0–53.

67 Merkle, R.C. (1990). A certified digital signature. In: *Advances in Cryptology – CRYPTO' 89 Proceedings*, 218–238.

68 Werbach, K. (2019). The blockchain and the new architecture of trust. *The Blockchain and the New Architecture of Trust*. https://doi.org/10.7551/mitpress/11449.001.0001.

10

Blockchain for Smart Healthcare

Dinh C. Nguyen[1], Pubudu N. Pathirana[1], Ming Ding[2], and Aruna Seneviratne[3]

[1]Networked Sensing and Biomedical Engineering Research Group, School of Engineering, Deakin University, Waurn Ponds, VIC, Australia
[2]Data61, CSIRO, Eveleigh, NSW, Australia
[3]School of Electrical Engineering and Telecommunications, University of New South Wales (UNSW), Eveleigh, NSW, Australia

10.1 Introduction

Smart healthcare is an industrial sector where organizations and medical institutions provide healthcare services, medical equipment, medical insurance to facilitate healthcare delivery to patients. Recent advances in blockchain have led to new opportunities for revolutionizing smart healthcare [1], thanks to its decentralization, immutability, and traceability. With its inherent features, blockchain has offered many promising solutions to solve critical issues in smart healthcare in terms of security, privacy, and medical service enhancement, and thus potentially transforms current healthcare systems [2, 3]. For example, blockchain can provide high degrees of security to healthcare operations, e.g. secure electronic medical records (EMRs) storage and safe patient diagnosis [4]. Blockchain has been also used to provide a transparent and reliable therapy management solution for in-home healthcare [5]. Moreover, blockchain was employed in [6] for a secure epidemic discovering and remote monitoring, by allowing distributed data health exchange among healthcare entities under security and quality-of-service (QoS) requirements. The authors in [7] utilized a blockchain-based decentralized scheme with mobile edge computing (MEC) [8] to support secure therapeutic data communication in a peer-to-peer (P2P)

Blockchains: Empowering Technologies and Industrial Applications, First Edition.
Edited by Anwer Al-Dulaimi, Octavia A. Dobre, and Chih-Lin I.

network. The study in [9] introduced a health data management scheme using a single cloud server which tends to incur single-point failures and low QoS (e.g. communication overhead). Another work in [10] employed a decentralized interplanetary file system (IPFS) platform with Ethereum blockchain for EMRs sharing over clouds. Recently, a patient-centric health information exchange architecture was proposed in [11] wherein a distributed blockchain network was deployed in the healthcare setting and interconnect multiple patients together in a P2P manner. Smart contract, a programmable and self-executing application running on a blockchain network [12], is employed to monitor the data exchange and store touchpoints, which contain the metadata of primary diagnosis and treatments of patients. A limitation of this work is the high blockchain transaction latency in the large-scale healthcare ecosystems. Meanwhile, the authors in [13] paid attention to a decentralized authentication framework in a distributed hospital network. By using a decentralized blockchain, the authentication of user access can be done at a hospital without the need to authenticate repeatedly in other hospitals, which thus potentially reduces communication latency. The work in [14] designed a privacy-preserving EMRs sharing in the consortium blockchain environment created by a set of medical institutions. The EMRs data are stored in the cloud while metadata, e.g. patient address, treatment information, are kept in the blockchain for privacy preservation [15]. In another research effort, the proposal in [16] presented a decentralized healthcare management model using MEC to support blockchain-based mechanisms. Also, an access authentication scheme for healthcare has been proposed in [17]. However, this work relies on a global Kademlia distributed hash table (DHT) in traditional IPFS platforms which thus results in excessive communication latency during the data retrieval process [18].

Despite these research efforts, the use of blockchain for collaborative healthcare, including health data sharing among federated hospital organizations, has not been explored fully. In fact, in the digital healthcare era, it is of the utmost importance to share EMRs across healthcare institutions in order to support collaborative health services and achieve universal healthcare [19]. For example, the sharing of cross-institutional EMRs would accelerate healthcare service delivery by enabling instant data exchange among authorized providers, health organizations, and clinics. Based on EMRs data, specialists are able to analyze patients' health information for appropriate clinical decision making, diagnosis, and care, while users can access EMRs to trace their medical history [20]. In such scenarios, ensuring high QoS and security is the key to realize a more effective healthcare system, which can be achieved by using the blockchain

technology. Hence, a more holistic blockchain solution for smart healthcare is needed to facilitate secure and low-latency EMRs data management and sharing among various healthcare domains, and form collaborative healthcare environments among medical institutions for accelerating smart healthcare services and applications.

In this chapter, we investigate the applications of blockchain in smart healthcare and provide a more comprehensive architecture where blockchain is integrated with other technologies such as MEC, smart contracts, and IPFS for facilitating healthcare service delivery. Furthermore, to solve the issues of high network latency caused by the blockchain adoption in smart healthcare systems, we are the first to propose a novel block mining mechanism which is enabled by the distributed Proof-of-Stake (DPoS) [21] concept. The contributions of this chapter are summarized as follows:

1. We propose a new fully *decentralized* healthcare architecture for collaborative smart healthcare among medical institutions with blockchain. To facilitate health data management and EMRs sharing, we design a new decentralized data storage platform using IPFS with blockchain in the MEC network. Particularly, an access authentication mechanism is developed by using blockchain-based smart contracts that realize access verification at the network edge without requiring any central authority or third party.
2. Moreover, to improve the efficiency of blockchain operations in smart healthcare, we propose a new Proof-of-Reputation (PoR) consensus mechanism enabled by a lightweight block verification strategy. Our blockchain design not only reduces the mining latency but also achieves network bandwidth savings.
3. We conducted extensive experiments in a practical hospital setting to investigate the effectiveness and feasibility of the proposed healthcare framework. The results show that our scheme outperforms the baselines in terms of improved data retrieval rate, reduced blockchain cost, and high security.

The remainder of this chapter is organized as follows. Section 10.2 introduces a new health architecture and describes our blockchain network design. In Section 10.3, we present the details of our EMR sharing scheme with two key parts: user authentication and EMRs data retrieval using blockchain, smart contract and MEC. Next, we propose a new PoR blockchain mining scheme in Section 10.4. The experimental results and evaluations are given in Section 10.5 along with security analysis. Finally, Section 10.6 concludes the chapter. A list of key acronyms used in the chapter is summarized in Acronyms section.

10.2 Smart Healthcare Architecture with Blockchain

In this section, we present a new health architecture and then describe our blockchain design.

10.2.1 Blockchain-Based Healthcare Architecture

Here we consider a decentralized health architecture for EMRs data management and sharing as shown in Figure 10.1 with a network of federated medical institutions with blockchain. Each institution is controlled by an edge server (ES) at an access point (AP) which collects data from IoMT devices for storage and communicates with ESs for EMRs sharing via a P2P network. Here, we consider a set of ESs as $\mathcal{M} = \{1, 2, \ldots, M\}$. Each ES $m \in \mathcal{M}$ is located at an institution HP_m to collect EMRs data via health gateways (e.g. smartphones, laptops, or tablets) from a set of patient $\mathcal{PID} = \{1, 2, \ldots, J^m\}$ with IoMTs (e.g. wearable sensors). We also assume that there is a set of healthcare users (HUs) $\mathcal{U} = \{1, 2, \ldots, U\}$ such as medical technicians, doctors who may be situated at any institution and want to access EMRs data in the MEC network for providing healthcare, e.g. medical diagnosis. The details of each network components are explained as follows:

- **Edge servers**: Each ES acts as a coordinator that manages healthcare operations (e.g. EMRs collection, local EMRs sharing) within its institution. An ES also links with other ESs to form a decentralized EMRs sharing without the need for central authority. We consider a realistic scenario that ESs may be semi-trusted parties, which means that an ES may be curious about data of other ESs. Motivated by this, we store sensitive EMRs data in the IPFS, instead of in the ES's hard drive for security guarantee.
- **IPFS storage**: As ESs are semi-trusted, EMRs collected from IoMTs are uploaded to a decentralized off-chain IPFS storage. IPFS introduces low-latency and fast decentralized archiving with reliable P2P content delivery, which has been investigated in distributed healthcare scenarios [18]. Thus it is well suitable for our EMRs sharing scenario. We integrate IPFS with smart contract that supports storage of EMRs' hash values at the network edge without relying on a remote DHT as in traditional IPFS designs [17], which further enhances the data retrieval rate for EMRs sharing.
- **Smart contract**: Here, we design a contract called access control smart contract (ACSC) to implement decentralized access authentication in EMRs

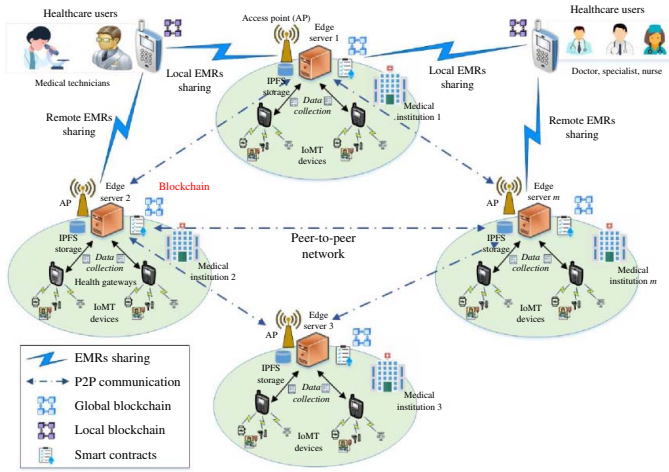


Figure 10.1 Decentralized EMRs sharing with blockchain for smart healthcare. Source: Chanut-is/Adobe Stock.

sharing. Each ES holds a copy of smart contracts and any new events (e.g. EMRs request from users) are updated at other ESs via the global blockchain network. Due to the decentralization of smart contracts, our scheme is able to perform access authentication directly at the network edge without passing a centralized authority like previous works [21–23], which thus significantly reduces authentication latency for our EMRs sharing.

10.2.2 Blockchain Design

Blockchain is the heart of our decentralized healthcare architecture. We here suggest using a permissioned Hyperledger Fabric [24] blockchain platform to implement our EMRs sharing system. The Hyperledger Fabric blockchain only allows authenticated users to join the network, and the validation is performed by only pre-selected nodes with high computing capability, i.e. ESs in our scenario, without mining requirements for lightweight entities like mobile users. This would improve the transaction performance, e.g. low transaction latency, compared to permissionless blockchains such as Ethereum [22]. As shown in Figure 10.1, we consider two types of blockchain for our healthcare architecture: a global blockchain and local blockchains.

- **Global blockchain**: It interconnects all ESs together for EMRs sharing communications under the control of all ESs. Once an EMRs retrieval event occurs (when a mobile user performs a data request to an ES), this ES creates a sharing transaction and broadcasts it to other ESs in the health network for tracing.
- **Local blockchain**: Each medical institution deploys a local blockchain to link the local ES with its mobile users. This local blockchain is controlled by its ES. When a mobile user performs a data request to the ES, he or she creates a sharing transaction and submits it to the local blockchain so that the ES can process the request and return the data. If the ES can look up data locally, the ES will return immediately to the user. Otherwise, it asks the other ESs to find the address of the requested data and then responds the user.

10.3 Blockchain for EMRs Data Sharing in Collaborative Healthcare

We assume that EMRs collected from IoMTs in each institution were already stored in the IPFS storage for EMRs sharing. To ensure security, the EMRs data

are encrypted as

$$C_j^{enc} \leftarrow Enc(C_j, PKM_m), \tag{10.1}$$

where C_j denotes the EMRs data of the patient j and PKM_m is the private key of the ES m at an institution. Moreover, uploading the data to the IPFS storage will automatically return a cryptographic hash of its content by using a hash function H_{IPFS}: $h_j = H_{IPFS}(C_j^{enc}, timestamp)$. Here, we keep the hash value in the smart contract at the ES for fast data look-up later, instead of relying on a global Kademlia DHT in traditional IPFS designs. Now, we are ready to analyze the EMRs sharing scheme as shown in Figure 10.1 with two key parts: user authentication and data retrieval.

10.3.1 User Authentication with Smart Contract

The user authentication process consists of three phases: initialization phase, user registration phase, and data retrieval phase, which are explained as follows.

10.3.1.1 Initialization Phase

Each medical institution is initialized by its ES. In this phase, a key is set up by the ES for data sharing establishment. More specifically, an ES m in each institution implements the following steps to initialize its network:

- The ES m selects an elliptic curve $E_p(a, b)$ over a prime finite field F_p and chooses a base point P over $E_p(a, b)$.
- Then, it chooses a high entropy random integer x_h as a private key and derives $X_h = x_h.P$ as its elliptic curve cryptography public key. Based on that, we can express the private key and the public key of the ES m as $SKM_m = x_h$ and $PKM_m = X_h$, respectively.
- Next, it selects a secure one-way hash function $Hash(.) : 0, 1^* -> Z_p^*$, where $Z_p^* = 1, 2, \ldots, p-1$, for a given prime p, is a finite cyclic group of order (p?1). This group is commutative under multiplication mod p.
- Based on that, each HU u in the institution HP_m generates a high entropy random integer SKU_u as its private key. Afterwards, a HU computes the public key $PKU_u = SKU_u.P$. Besides, each HU also has a unique identity ID_u for identification.

10.3.1.2 Registration Phase

This phase is invoked whenever a HU wants to register to the ES for the first time. To do so, the user joins the local blockchain network and follows the steps as below.

- Each HU u submits a transaction T_{reg_u} to the ES m as T_{reg_u} for registration:

$$T_{reg_u} = (PKU_u||ID_u||timestamp). \tag{10.2}$$

- The ES m decodes the T_{reg_u} to obtain the user's public key as $PKU_u \leftarrow T_{reg_u}.getSenderPublicKey()$.
- Then, the ES decodes the transaction to get the user ID. First, it decodes the transaction T_{reg_u}, then finally obtain the user ID as an unique address ADD_u:

$$deT_u \leftarrow abiDecoder.decodeMethod(T_{reg_u}), \tag{10.3}$$

$$ADD_u \leftarrow web3.eth.getData(deT_u([ID_u])). \tag{10.4}$$

- The ES checks user information and stores $\{PKU_u, ADD_u\}$ as the user identity in contract's database.
- The ES then calculates the hash value of the register transaction T_{reg_u} as:

$$H_{HU_u} \leftarrow Hash(T_{reg_u}, SKM_m), \tag{10.5}$$

which is then published to the local institution blockchain network for tracing. The ES m also broadcasts its public key PKM_m to the user that is necessary for the future user data access.

10.3.1.3 User Authentication Phase

It is supposed that a HU u wants to access the patient's EMRs stored in the MEC network for their medical tasks. To obtain the EMRs of the target patient, the HU u needs to know his patient identity PID_j so that the ES can locate the address of this patient in the health network during EMRs sharing. The data retrieval process is presented by the following key steps.

- A HU u prepares a data retrieval request T_{req_u} involved a target patient ID PID_j and the address of patient's institution HP_w ($w \in \mathcal{M}$). Thus, the target patient address in the health network can be expressed as $P_{addr} =< PID_j, HP_w >$, e.g. the 5th patient in the 3th institution. Then, the request T_{req_u} can be specified by

$$T_{req_u} \leftarrow (PKU_u||ID_u||P_{addr}||timestamp), \tag{10.6}$$

where each component in T_{req_u} is formatted with an index in the array $index = [1-4]$, e.g. the index of PKU_u is 1. This format is necessary for transaction decoding later.

- To ensure privacy, the user request should be encrypted with the ES m's public key PKM_m (obtained from the registration phase) as

$$T_{enc_u} \leftarrow Enc(T_{req_u}, PKM_m), \tag{10.7}$$

which is then submitted to the ES m.
- At the edge side, the ES m decrypts the user request T_{enp_u} as

$$T_{dec_u} \leftarrow Dec(T_{req_u}, SKM_m). \tag{10.8}$$

To provide security for the EMRs sharing, user authentication is highly essential. To do so, the ES extracts the user's public key from the request as

$$Pub_u \leftarrow T_{dec_u}.getSenderPublicKey(). \tag{10.9}$$

It also decodes the transaction T_{dec_u} to obtain the request information $ReqInf_u$:

$$deT_{dec_u} \leftarrow abiDecoder.decodeMethod(T_{dec_u}), \tag{10.10}$$

$$ReqInf_u \leftarrow web3.eth.getData(deT_{dec_u}[DataIndex]), \tag{10.11}$$

and then obtains the user identity $Iden_u$ as

$$Iden_u = ReqInf_u(Index[index_{Iden}]). \tag{10.12}$$

- The ES checks and authenticates the received user identification information $< Pub_u, Iden_u >$, and then puts them into user mapping as

$$UMAP_{PK_u} = Map < Pub_u => PKU_u >, \tag{10.13}$$

$$UMAP_{ID_u} = Map < Iden_u => ID_u >, \tag{10.14}$$

by using the smart contract (see in Algorithm 10.1). If both $UMAP_{PK_u} \rightarrow true$ and $UMAP_{ID_u} \rightarrow true$, the user request is validated successfully, otherwise a penalty is issued for access prevention.
- In the case of successful request validation, the ES m calculates the signature of T_{dec_u} as

$$Sig_u \leftarrow Hash(T_{dec_u}, SKM_m). \tag{10.15}$$

Finally, the ES issues a certificate $Cert_u$ as

$$Cert_u = \{Sig_u, PKU_u, ID_u, timestamp\}, \tag{10.16}$$

which is then sent to the HU u via the local blockchain for successful authentication proof.

10.3.2 Health Data Retrieval with Blockchain

After successful authentication, the ES m locates the requested EMRs based on the patient information $< PID_j, HP_w > \leftarrow ReqInf_u(Index[indexPID_u])$ that is defined in the authentication phase. In fact, the patient and the user may be located in the same institution or in different institutions. For example, a patient may see a particular specialist in one location for a certain condition and may visit a different specialist possibly at a different location for a separate medical condition [13]. Motivated by this realistic scenario, here we consider two cases: (1) The patient and the user are in the same institution and (2) The patient and the user are in different institutions.

Case 1: The patient and the user are in the same institution: In this case, the ES finds that the HU and the patient are in the same institution by checking HP_w information. It is supposed that the HU u communicates with the ES m to request the data record of the patient PID_j in the same institution, then the data retrieval process is implemented by the following steps:

- The ES m first verifies the request information PID_j by referring to the ACSC contract to perform mapping between the patient record stored in the contract PID_j^{sc} and information in the request PID_j^{req}:

$$UMAP_{PID_j} = Map < PID_j^{sc} => PID_j^{req} > . \tag{10.17}$$

  If $UMAP_{PID_j} \rightarrow true$, the request information is verified for ready data retrieval.
- Based on the received patient information, the ACSC contract extracts the hash value h_j that represents the patient j's health record. Then, the contract sends a request to the IPFS storage for data retrieval using the hash by a command: $C_j^{enc} = GET_{IPFS}(h_j)$, i.e.: `ipfs get /ipfs/ Qmd84db7be0690ebb015f1cD9d9491cE18076c`.
- Since the data stored in the IPFS was encrypted (see in Eq. (10.1)), we need to decrypt to obtain the real data as

$$C_j \leftarrow Dec(C_j^{enc}, SKM_m). \tag{10.18}$$

  The ES m then returns the data C_j via a secure channel to the requestor.
- Finally, the ES m adds a conjunction of $(PKU_u, h_j, PKM_m, timestamp)$ as a transaction and broadcasts it to the global blockchain network:

$$ES_m \rightarrow *: (PKU_u, h_j, PKM_m, timestamp). \tag{10.19}$$

Case 2: The patient and the user are in different institutions: In this case, the ES seeks the address of the patient in the MEC network. Due to all patient addresses (PID_j, HP_w) are stored in the ACSC contract replicated across the health network, an ES at an institution can know exactly where the requested patient data is currently located. Hence, an ES only needs to send the data request to the destination ES (using HP_w information) that contains the requested data for data retrieval, without broadcasting the requests to all ESs. This strategy not only saves data lookup time but also saves network bandwidth and potentially reduces the traffic congestion in the global blockchain network. The data retrieval process in this case is summarized as the following steps:

- The ES m first also verifies the request information PID_j from the user PKU_u by referring to the ACSC contract, and then performs mapping to verify that the patient information PID_j is correct. Then, it also identifies which ES is currently storing the requested data by checking HP_w information. Here, we assume that an ES $ES_y, (y! = m, y \in \mathcal{M})$ is holding the requested data.
- After identifying the destination ES y, the ES m sends a transaction for data retrieval request:

$$ES_m \rightarrow ES_y : (PID_j, PKU_u, PKM_m, time). \tag{10.20}$$

- Based on the patient information PID_j, the ACSC contract in the ES y obtains the hash value h_j. Next, the contract sends a request to the IPFS node in the ES y by a command: $C_j^{enc} = GET_{IPFS}(h_j)$, which is then decrypted to obtain the real data:

$$C_j \leftarrow Dec(C_j^{enc}, SKM_y). \tag{10.21}$$

- The ES y then transmits the collected data C_j to the ES m so that the ES m returns it to the requestor. Finally, the ES y adds a conjunction of (PKU_u, $h_j, PKM_y, timestamp$) as a transaction and broadcasts it to the global blockchain network:

$$ES_y \rightarrow *: (PKU_u, h_j, PKM_y, timestamp). \tag{10.22}$$

  Finally, all ESs update the user access events and achieve a synchronization over the data sharing across the healthcare network.

The EMRs retrieval process for two cases is shown in Figure 10.2, and the EMRs sharing is summarized in Algorithm 10.1.

Algorithm 10.1 EMRs sharing with MEC and blockchain

Initialization: *(by the user HU u)*
Encrypt the request: T_{enc_u} and submits it to the ES m
Pre-processing the request *(by ES)*
The ES m decrypts the user request T_{enp_u} as $T_{dec_u} \leftarrow Dec(T_{req_u}, SKM_m)$
Obtain the user's public key: $Pub_u \leftarrow T_{dec_u}.getSenderPublicKey()$
Decode T_{dec_u} and get the user identity $Iden_u$
Authentication *(by the ACSC contract)*
if $msg.sender == ES_m$ **then**
  $PKcheck = policy[EHRresource][action].PKU_u$
  $Idencheck = policy[EHRresource][action].ID_u$
  if $PKcheckIdencheck \rightarrow true$ **then**
    $Auth_u \leftarrow AccessResult(msg.sender, Accepted, true, time)$
  else
    $Auth_u \leftarrow AccessResult(msg.sender, Denied, false, time)$
  end if
end if
while $Auth_u \rightarrow true$ **do**
  if $HP_w == HP_w^{sc}$ **then**
    if $PID_j^{sc} == PID_j$ **then**
      Get the data on IPFS: $C_j^{enc} = GET_{IPFS}(h_j)$
      Decrypt to obtain the real data $C_j \leftarrow Dec(C_j^{enc}, SKM_m)$
    end if
    The ES m returns the data C_j to the HU_u
    The ES m adds a transaction to the global blockchain network: $ES_m \rightarrow *: (PKU_u, h_j, PKM_m, timestamp)$
  else if $HP_w ! = HP_w^{sc}$ **then**
    if $PID_j^{sc} == PID_j$ **then**
      Communicate with the ES y: $ES_m \rightarrow ES_y : (PID_j, PKU_u, PKM_m, timestamp)$
      Get the data on IPFS: $C_j^{enc} = GET_{IPFS}(h_j)$
      Decrypt to obtain the real data $C_j \leftarrow Dec(C_j^{enc}, SKM_y)$
    end if
    The ES y returns data C_j to the ES m and then the HU u
  end if
end while

10.4 Blockchain Mining Design for Smart Healthcare System

In the distributed edge computing-based blockchain environment like our healthcare scenario, latency and scalability are among the most important factors determining the efficiency of a blockchain. Given a consensus algorithm, when the number of transactions to the blockchain increases, the consensus workload to validate and append them into the blockchain also increases significantly. In current consensus algorithms, e.g. DPoS [25], each miner must contact at least more than half of the total nodes in the miner group, which consequently increases

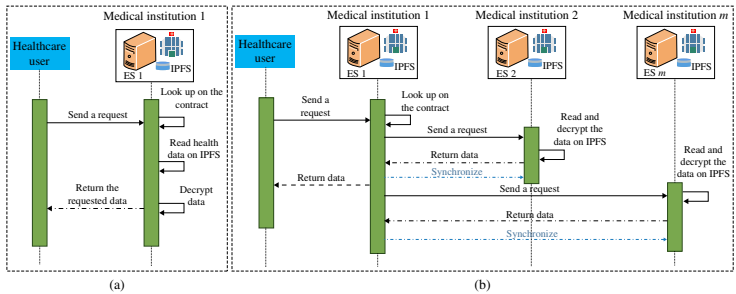


Figure 10.2 The proposed EMRs sharing procedure with MEC and IPFS. (a) Case 1: The patient and the user are in the same medical institution. (b) Case 2: The patient and the user are in different medical institutions.

latency and alleviates scalability of the blockchain system. Moreover, each miner node must implement a repeated verification process across the miner network, which results in unnecessary consensus latency and network bandwidth waste. A possible solution is to reduce the number of miner nodes to mitigate the consensus latency, but it potentially compromises the security of blockchain because of the higher probability of adding compromised transactions from malicious nodes [25]. To solve these mining issues, here we propose a new lightweight PoR consensus mechanism based on DPoS for our blockchain healthcare system. Compared to the DPoS scheme, here we make an improvement in the miner selection based on a reputation score evaluation approach for blockchain-based smart healthcare. Moreover, instead of using a repeated verification among miner nodes, we implement a lightweight block verification solution that allows each miner to only verify once with another node during the consensus process, which would significantly reduce the verification latency. There are two main parts to our PoR consensus, including miner node selection and block verification, as illustrated in Figure 10.3.

10.4.1 Miner Node Selection

In this phase, the HUs first calculate the reputation score of ESs and then select the miner nodes to implement the mining process.

10.4.1.1 Reputation Calculation

In our health scenario, in addition to the EMRs exchange function, HUs also participate in the delegate selection process to vote the mining candidates for performing blockchain consensus. In this regard, each HU votes its preferred ES with the most reputation. Here, the reputation of an ES is measured by its computing capability to mine the block. That is, an ES that allocates more computational resources to the mining tasks will have a higher reputation score to obtain a higher priority for mining the block. To this end, we define a reputation function to determine the score for each ES as follows:

$$\Psi_n = \left[e^{1-\frac{T_n^{PoR}}{\tau_n}} - 1 \right]^+. \tag{10.23}$$

Here, T_n^{PoR} is the mining latency of the ES n, where n is the miner index defined in the mining context with $n \in \{1, \ldots, N\}, N \leq M$. Its detail will be explained in the following section. Also, τ_n denotes the ES n's EMRs retrieval latency constraint to ensure the quality of user experience in the EMRs sharing. $[y]^+ = \max\{y, 0\}$ implies that the reputation of an ES n is set to 0 if the mining latency T_n^{PoR} is exceeded to its maximum EMRs retrieval time τ_n.

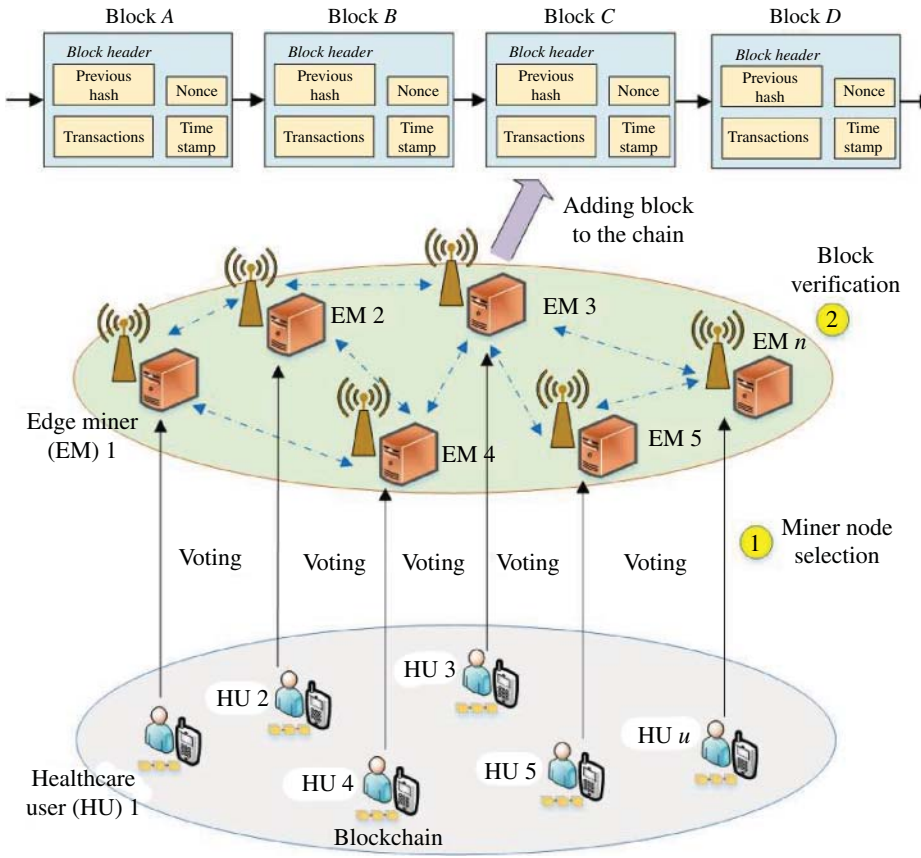


Figure 10.3 The PoR consensus of health blockchain.

10.4.1.2 Miner Selection

Based on the calculated reputation score, each MU votes for ES candidates as the miners based on their reputation ranking as indicated in Figure 10.3. The top ESs with the highest reputation scores are selected to become edge miners (EMs) to perform consensus. Besides, similar to the traditional DPoS framework [6], each of the EMs also acts as a block manager which is responsible for performing block generation, broadcasting blocks to other miners for verification, and block aggregation after being verified, during its time slot of the consensus process.

10.4.2 Lightweight Block Verification

In this phase, the block manager first produces an unverified block that contains several health transactions collected in a given amount of time. Then the

manager broadcasts this created block to all EMs within the miner network for verification. Different from the traditional DPoS scheme which relies on a repeated verification process among miners, here we implement a lightweight PoR-based verification solution that allows each miner only needs to verify once with another node during the consensus process, which significantly reduces the verification latency. Algorithm 10.2 presents how our proposed block verification procedure is performed. In lines (2–7), the block manager divides the block B with the whole transaction into K transaction parts Tr_k ($k = (1, \dots, K)$) that will be assigned to each EM member EM_n within the miner group. Each miner EM_n will be also assigned a unique random number R_n. In lines (9–19), an EM_n selects any miner s ($s! = n$) within the miner group to implement the verification for its assigned transaction part Tr_k. If 51% of the EMs respond positive verification, and the sum of random numbers *Sum* calculated by all EMs is equal to the initial number set *Rnd*, the block manager accepts the verified block B and adds it to the blockchain with signature. For instance, in Figure 10.3, the EM 3 works as a manager to create the block C and appends it into the blockchain. Otherwise, the manager discards it from the network (lines 20–25).

Algorithm 10.2 Procedure of the proposed PoR consensus

1: **Initialization:** Select an unverified block B, group the selected EMs $(EM_1, \dots, EM_N)$ in the list $Array[n], n \in [1, N]$, initiate public key $Array[n].PK$ and block manager BM
2: Divide the block B into K parts Tr_k, $Sum \leftarrow 0$
3: **for** $n = 1, \dots, N$ **do**
4: Set a part of block $Tr_k \rightarrow Array[n].content$
5: Assign a random number $R_n \leftarrow Random()$
6: Calculate a signature as $Sig_n \leftarrow Hash(Array[n].content, Array[n].PK, timestamp)$
7: **end for**
8: Specify the total random number $Rnd = K(1 + \frac{K-1}{2})$
9: **for** $n = 1, \dots, N$ **do**
10: Run a random function $s = Random.randrange(1, N, 1)$
11: **if** $n! = s$ **then**
12: *Select a random different EM within the list*
13: Send the Tr_k to EM_s: $EM_n \rightarrow EM_s : (Tr_k, Array[n].PK, Sig_n, timestamp)$
14: Verify the transaction Tr_k
15: **if** $(Array[n].PK_{EM_s} == EM_s^{EM_n.PK}) \cap (Verify(Sig_n) \leftarrow true)$ **then**
16: $Sum \leftarrow Sum + R_n$
17: **end if**
18: **end if**
19: **end for**
20: **if** $Sum == Rnd$ **then**
21: Accept the block B and send it back to the block manager BM
22: The manager BM appends the block B into the blockchain network: $BM \rightarrow *: (BM_{PK}, B, Sign_B, timestamp)$
23: **else**
24: Discard the block B from the blockchain
25: **end if**

10.4.3 Latency of Block Verification

In this section, we calculate the verification latency incurred by the mining. For simplicity, we assume that the transaction part Tr_k (which also expresses the size) is the same for all EMs. Each EM is willing to contribute their resource $C = \{c_1, \dots, c_N\}$ (in CPU cycles s^{-1}) to execute the verification of the transaction part k. For each EM n, the CPU resource occupied to verify the transaction k is Φ_n. The size of verified transaction result for the Tr_k is denoted by Tr_k^{re}. Hence, the transaction verification task can be expressed as a tuple $(Tr_k, \Phi_n, Tr_k^{re})$.

Conceptually, the block verification process in our proposed PoR mechanism at an EM experiences four steps: (i) unverified block transmission from the block manager to the EMs, (ii) local block verification at the EM, (iii) broadcasting of the verification result among two EMs, and (iv) transmission of verification result feedback from the EMs to the manager. For a miner n, the time required to complete these steps is expressed as:

$$T_n^{PoR} = \frac{Tr_k}{r_n^d} + \frac{\Phi_n}{c_n} + \xi Tr_k |L^2| + \frac{Tr_k^{re}}{r_n^u}, \quad k \in [1, \dots, K], \tag{10.24}$$

where r_n^u and r_n^d are uplink and downlink transmission rates between the miner n and the block manager. Here, the transmission time of an unverified transaction part Tr_k from the block manager to the miner is $\frac{Tr_k}{r_n^d}$, while the local verification time of this transaction is $\frac{\Phi_k}{c_n}$. Moreover, similar to [26], the time for transaction broadcasting among two miners is a function of transaction size Tr_k and network scale $Tr_k|L^2|$ (which means two miners for transaction verification), which is defined as $\xi Tr_k|M^2|$. Here, ξ is a pre-defined parameter of broadcasting verification result and comparison among two miners, which can be acquired from the previous verification records [26]. Besides, $\frac{Tr_k^{re}}{r_n^u}$ is the verification feedback time.

Meanwhile, in the traditional DPoS scheme [6], each miner has to implement a repeated verification process among all miners for the block B, instead of dividing into separate transaction parts like our proposed PoR model. Therefore, the verification latency of the DPoS consensus is expressed as [26]:

$$T_n^{DPoS} = \frac{B}{r_n^d} + \frac{\Phi_n^B}{c_n^B} + \xi B|L^N| + \frac{B^{re}}{r_n^u}, \tag{10.25}$$

where Φ_n^B is the CPU resource occupied to verify the block B under the computation budget c_n^B. B^{re} denotes the size of verified result of the block B. $|L^N|$ expresses the whole miner network which means all miners n join the repeated block verification in each consensus process, instead of two-miner verification in our PoR scheme. By comparison of Eqs. (10.24) and (10.25), it can be seen that the proposed PoR scheme consumes less time in the verification process, compared to the traditional DPoS scheme, for the same block size and number of miners.

Moreover, our mining scheme can save much network bandwidth due to less message exchange during the consensus process. The benefits of our proposed PoR mechanism will be verified in the following section.

10.5 Experimental Results

In this section, we present the numerical experiments to evaluate the feasibility and practicality of our blockchain-based approach for smart healthcare.

10.5.1 Experimental Settings

We implemented a testbed to evaluate the proposed EMRs sharing architecture through a decentralized cooperative healthcare network where each ES represents a medical institution. We considered a three-ES system by employing three computers with Intel core i7 at 3.4 GHz, 8 GB of RAM running Microsoft Windows 10 64-bit version on 500 GB hard drive. Each ES will connect with a network of mobile health users via a Cisco access point to perform EMRs sharing. Here, each user is equipped with a Sony Android phone (with Qualcomm Snapdragon 845 processor, 1 GB memory) that is also installed with a blockchain account to communicate with the ES for EMRs request via blockchain. For the EMRs data generation, we used $BioKin^T$ [27] developed by our lab to act as IoMT devices to collect simultaneously human motion data and store them in separate files on IPFS storage.

For blockchain deployment, two Hyperledger Fabric platforms, version 1.3, were used to build the global blockchain in the edge health system and the local blockchain in the ES-device system. We followed the instructions in the official Hyperledger Fabric tutorial to install required files and docker images. The smart contract was implemented in docker to serve user authentication and data retrieval. We also installed the JavaScript version of the IPFS platform in the edge health system. Each of three ESs holds an IPFS node which is embedded with the Fabric blockchain to perform data storage and data sharing. To highlight the merits of our EMRs sharing scheme, we compare with the state-of-art works from different performance metrics as follows.

10.5.2 Evaluation of EMRs Sharing Performance

We evaluate the proposed EMRs sharing scheme via the authentication cost and data retrieval latency metrics.

10.5.2.1 Authentication Cost

We first compare the authentication latency of our proposed scheme using smart contracts with a centralized scheme [9] with the different number of HUs. In the proposed scheme, we organize the access authentication at the network edge where each ES authenticates its users by the distributed smart contract. Meanwhile, the scheme [9] relies on a central authority to implement its user authentication. As shown in Figure 10.4, our scheme exhibits a lower latency compared to the baseline [9]. This is because the use of decentralized smart contract enables fast authentication at the network edge without passing a remote authority, which thus reduces communication overhead in the authentication process.

10.5.2.2 Data Retrieval Latency

We investigated the data retrieval latency of our proposed model from blockchain design and network design perspectives as shown in Figure 10.5. We used smartphones to send data requests continuously to the ESs to record the results. In terms of blockchain design, we use two existing works for comparison. The first one is a centralized edge-based health sharing scheme without IPFS [16] which used a centralized ES to serve a large hospital network and health data was stored in a traditional database. The second one is an edge blockchain health sharing scheme with traditional IPFS [17] that integrated blockchain and edge computing without IPFS design improvement.

From Figure 10.5a, we can see that when the number of requests increases, the baseline [16] has the highest data retrieval latency due to the queuing latency in the centralized EM. The baseline [17] used a traditional IPFS storage with a global

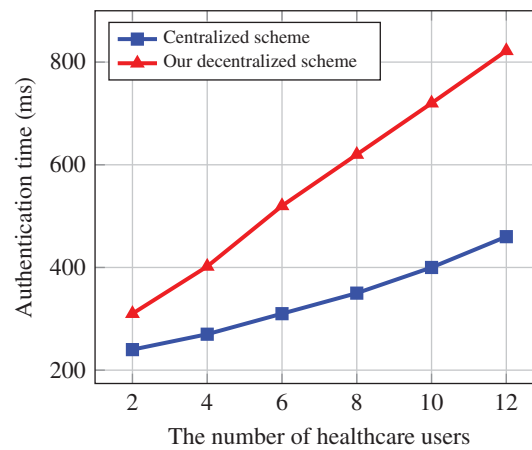


Figure 10.4 Authentication latency with different numbers of healthcare users.

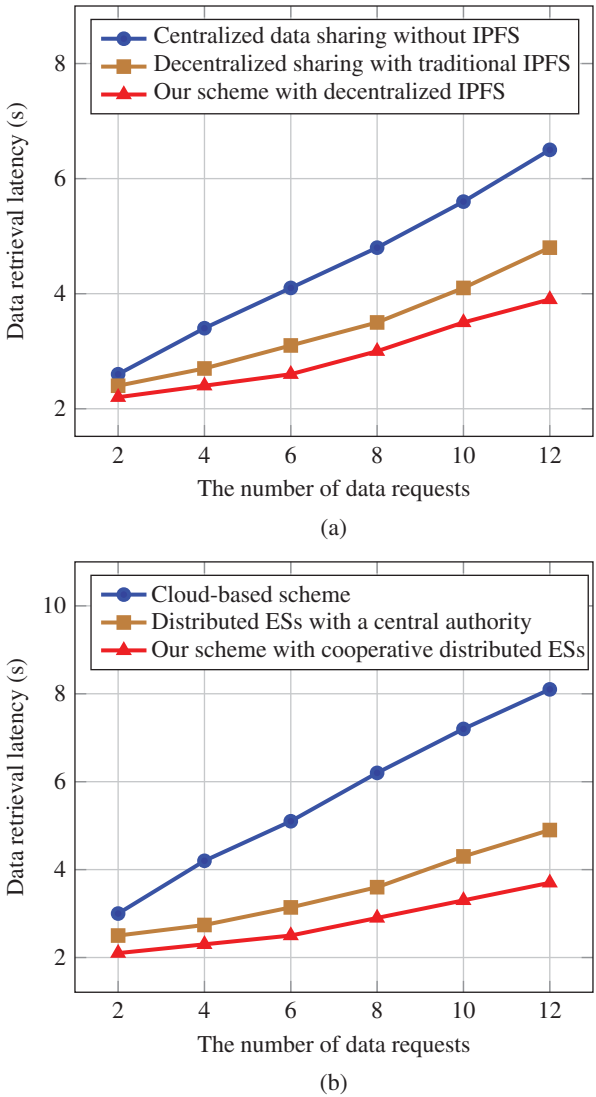


Figure 10.5 Data retrieval latency under different numbers of data requests. (a) Data retrieval latency under blockchain designs. (b) Data retrieval latency under network designs.

DHT look-up solution that results in unnecessary communication overhead. By contrast, our scheme provides a fully decentralized solution with distributed ESs and smart contracts, which allows for implementing request verification and data look-up at the network edge without using the global DHT. As a result, our scheme can achieve a minimum data retrieval latency.

Next, we evaluated the data retrieval latency from a network design perspective as shown in Figure 10.5b. We leveraged a cloud-based scheme [9] and a distributed ES with a central authority [7] as the baselines for comparison. For cloud computing implementation, we employed Amazon cloud services to communicate with smartphones. The results in Figure 10.5b clearly show a significant improvement in our decentralized scheme with a much lower retrieval latency. This is because our scheme combines ES, distributed smart contracts, and decentralized IPFS for fast data retrieval, without passing any external authority during the data sharing. Meanwhile, the work in [9] relies on a remote cloud model which remains high latency due to excessive communication overhead. Also, the work in [7] used a central authority for request verification that consumes a certain overhead for communication between the EMs and the authority in the request verification.

10.5.3 Evaluation of Blockchain Performance

In this section, we evaluate the efficiency of our proposed PoR consensus scheme and then present the security benefits of our health blockchain model.

10.5.3.1 Blockchain Consensus Performance

Here, we evaluate the performance of our proposed PoR consensus scheme via numerical simulations using Python programming and compare it with the traditional DPoS scheme via the verification block latency and bandwidth usage metrics. We set up 10 transactions per block and vary the numbers of mining nodes (i.e. ESs) from 2 to 100. Motivated by Kang et al. [26], the mining parameters are set up as follows: edge computation resources $c_n = [10^3 - 10^6]$ CPU cycles s^{-1}, input/output block data sizes $B = 500$ KB, $B^{re} = 50$ KB, the uplink transmission rate $r_n^u = [100 - 250]$ kbps, the downlink transmission rate $r_n^d = [100 - 250]$ kbps, $\xi = 0.5$, $\tau_n = 1000$ ms.

We first show the block verification latency performance vs different numbers of mining nodes with the block size fixed at 50 KB. As illustrated in Figure 10.6, our proposed PoR scheme requires significantly less time for mining blocks, compared to the DPoS scheme due to the optimized block verification procedure. Although the time required for block verification increases with the increasing number of miners, our solution still achieves a much better performance than that of the DPoS scheme. This result confirms our lightweight consensus design that is thus well suited for large-scale health blockchain networks.

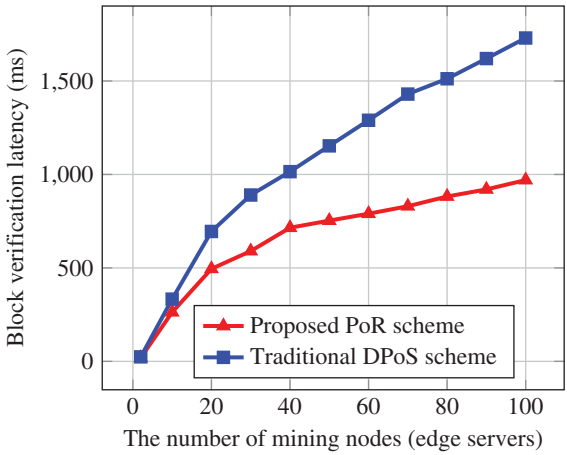


Figure 10.6 Comparison of block verification latency.

Next, we evaluate the block verification time when varying the size of data block from 50 to 500 KB in the health blockchain network with 10 miners. As shown in Figure 10.7, our proposed PoR scheme yields a lower verification latency than the traditional DPoS scheme due to our lightweight verification strategy. In particular, our mechanism shows its good advantage when the size of data block is large (e.g. >400 KB), while the DPoS scheme requires much time to verify the large blocks. The simulation results also imply the block mining analysis in Section 10.4.3. Moreover, Figure 10.8 indicates the simulation result in terms of network bandwidth cost spent by the mining process for different data block sizes. Thanks to our optimized message exchange procedure where each ES

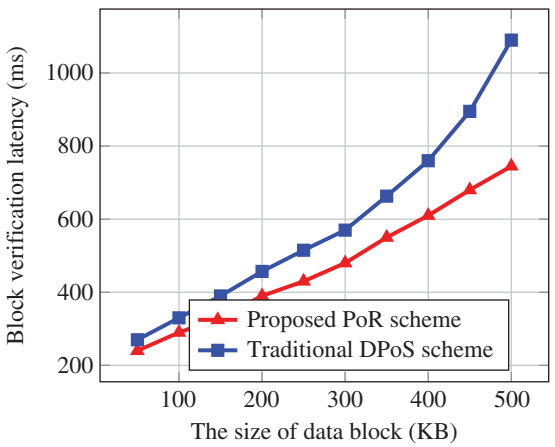


Figure 10.7 Comparison of block verification latency.

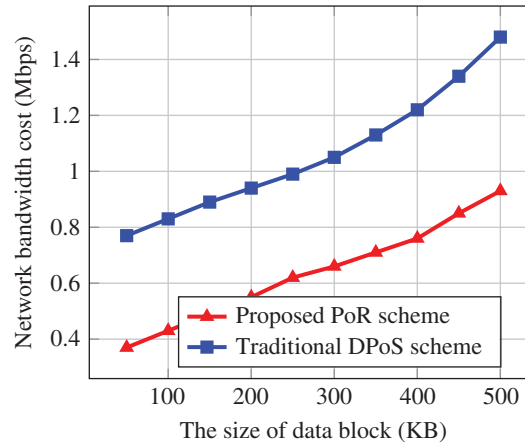


Figure 10.8 Comparison of network bandwidth cost.

only needs to contact with one miner to perform verification, instead of using a repeated process, our PoR scheme can save much network bandwidth resources, compared to the DPoS scheme. The simulation results clearly demonstrate the benefits of our proposed consensus mechanism with low mining latency and bandwidth savings.

10.5.4 Security Analysis

Our blockchain-based approach is able to achieve high security for smart healthcare via three features: data privacy, authentication, and traceability.

10.5.4.1 Data Privacy

Our proposed scheme is able to preserve data privacy. The user request is always encrypted as in Eq. (10.7) so that private user information is protected against threats. Importantly, we store health data on IPFS to avoid data breaches from curious ESs. Unlike the existing works [9, 28] with a centralized IPFS on a third-party cloud which can remain single-point failures, our scheme provides a fully decentralized data storage over the MEC network under the management of all ESs without a third party, which thus provides better data privacy control.

10.5.4.2 Authentication

User authentication in our EMRs sharing is performed by the distributed ACSC contract which works independently with the ES. This means authentication policies cannot be controlled by the curious ES but managed by the global blockchain

network. This would avoid authentication rule modifications caused by network attacks and hence ensure reliable authentication.

10.5.4.3 Traceability

In our smart healthcare system, any data access events and user behaviors are traced by ESs and users. Particularly, the ESs can trace the health data stored in the IPFS network via IPFS hash values. Any modifications on EMRs will lead to a change in its hash value so that ESs can trace and prevent, which cannot be achieved with a centralized storage as designed in [7].

10.6 Conclusions

In this chapter, we have presented the applications of blockchain in smart healthcare and proposed a new fully decentralized healthcare architecture for collaborative healthcare. A decentralized IPFS system is integrated with MEC to build a secure data storage system among the ESs in blockchain. To realize decentralized access management, we have developed a new ACSC contract that enables user authentication at the network edge without relying on any central authority, which ensures authentication reliability and reduce data retrieval latency. Furthermore, we have proposed a new lightweight PoR consensus mechanism for the blockchain-based EMRs sharing which not only reduces the mining latency but also achieves network bandwidth savings. We have implemented various real-world experiments to verify the effectiveness of our blockchain approach for smart healthcare. The implementation results demonstrated the high performance of our blockchain-based method compared to the baseline in terms of a significant QoS improvement with reduced data retrieval latency, enhanced blockchain performance, and security guarantees.

Based on the obtained results, the proposed MEC-based healthcare scheme has the potential to apply in other use cases, such as fog-based computing solutions where fog nodes can collaborate with cloud computing using the proposed blockchain network for future healthcare applications.

Acronyms

EMRs	Electronic medical records
QoS	Quality-of-service
MEC	Mobile edge computing
P2P	Peer-to-peer
IPFS	Interplanetary file system

DHT	Distributed hash table
DPoS	Distributed Proof-of-Stake
PoR	Proof-of-Reputation
AP	Access point
ESs	Edge servers
HUs	Healthcare users
ACSC	Access control smart contract

References

1 Liu, X., Zhou, P., Qiu, T., and Wu, D.O. (2020). Blockchain-enabled contextual online learning under local differential privacy for coronary heart disease diagnosis in mobile edge computing. *IEEE Journal of Biomedical and Health Informatics* 24 (8): 2177–2188.

2 Nguyen, D.C., Pathirana, P.N., Ding, M., and Seneviratne, A. (2020). Blockchain for 5G and beyond networks: a state of the art survey. *Journal of Network and Computer Applications* 166: 102693. https://doi.org/10.1016/j.jnca.2020.102693.

3 Nguyen, D.C., Pathirana, P.N., Ding, M., and Seneviratne, A. (2020). Integration of blockchain and cloud of things: architecture, applications and challenges. *IEEE Communications Surveys & Tutorials* 22 (4): 2521–2549. doi: https://doi.org/10.1109/COMST.2020.3020092.

4 Nguyen, D.C., Pathirana, P.N., Ding, M., and Seneviratne, A. (2019). Blockchain for secure EHRs sharing of mobile cloud based e-health systems. *IEEE Access* 7: 66792–66806.

5 Rahman, M.D.A., Hossain, M.S., Loukas, G. et al. (2018). Blockchain-based mobile edge computing framework for secure therapy applications. *IEEE Access* 6: 72469–72478.

6 Abdellatif, A.A., Al-Marridi, A.Z., Mohamed, A. et al. (2020). ssHealth: toward secure, blockchain-enabled healthcare systems. *IEEE Network* 34 (4): 312–319.

7 Saha, R., Kumar, G., Rai, M.K. et al. (2019). Privacy ensured e-healthcare for fog-enhanced IoT based applications. *IEEE Access* 7: 44536–44543.

8 Nguyen, D.C., Pathirana, P.N., Ding, M., and Seneviratne, A. (2019). Blockchain as a service for multi-access edge computing: a deep reinforcement learning approach. *arXiv:2001.08165 [cs, eess]*, December arXiv: 2001.08165.

9 Liu, J., Li, X., Ye, L. et al. (2018). BPDS: a blockchain based privacy-preserving data sharing for electronic medical records. *2018 IEEE Global Communications Conference (GLOBECOM)*, 1–6.

10 Wang, S., Zhang, Y., and Zhang, Y. (2018). A blockchain-based framework for data sharing with fine-grained access control in decentralized storage systems. *IEEE Access* 6: 38437–38450.

11 Zhuang, Y., Sheets, L., Chen, Y.-W. et al. (2020). A patient-centric health information exchange framework using blockchain technology. *IEEE Journal of Biomedical and Health Informatics* 24 (8): 2169–2176.

12 Nguyen, D.C., Pathirana, P.N., Ding, M., and Seneviratne, A. (2020). Blockchain and edge computing for decentralized EMRs sharing in federated healthcare. *2020 IEEE Global Communications Conference (GLOBECOM).*

13 Yazdinejad, A., Srivastava, G., Parizi, R.M. et al. (2020). Decentralized authentication of distributed patients in hospital networks using blockchain. *IEEE Journal of Biomedical and Health Informatics* 24 (8): 2146–2156.

14 Jiang, S., Wu, H., and Wang, L. (2019). Patients-controlled secure and privacy-preserving EHRs sharing scheme based on consortium blockchain. *2019 IEEE Global Communications Conference (GLOBECOM)*, 1–6.

15 Nguyen, D.C., Pathirana, P.N., Ding, M., and Seneviratne, A. (2020). Privacy-preserved task offloading in mobile blockchain with deep reinforcement learning. *IEEE Transactions on Network and Service Management* 17 (4): 2536–2549. doi: https://doi.org/10.1109/TNSM.2020.3010967.

16 Li, X., Huang, X., Li, C. et al. (2019). EdgeCare: leveraging edge computing for collaborative data management in mobile healthcare systems. *IEEE Access* 7: 22011–22025.

17 Arachchige, P.C.M., Bertok, P., Khalil, I. et al. (2020). A trustworthy privacy preserving framework for machine learning in industrial IoT systems. *IEEE Transactions on Industrial Informatics* 16 (9): 6092–6102.

18 Nguyen, D.C., Pathirana, P.N., Ding, M., and Seneviratne, A. (2021). BEdgeHealth: a decentralized architecture for edge-based IoMT networks using blockchain. *IEEE Internet of Things Journal* 8 (4): 11743–11757.

19 Jin, H., Luo, Y., Li, P., and Mathew, J. (2019). A review of secure and privacy-preserving medical data sharing. *IEEE Access* 7: 61656–61669.

20 Qiu, H., Qiu, M., Memmi, G., and Liu, M. (2020). Secure health data sharing for medical cyber-physical systems for the healthcare 4.0. *IEEE Journal of Biomedical and Health Informatics* 24 (9): 2499–2505.

21 Attia, O., Khoufi, I., Laouiti, A., and Adjih, C. (2019). An IoT-blockchain architecture based on hyperledger framework for healthcare monitoring application. *2019 10th IFIP International Conference on New Technologies, Mobility and Security (NTMS)*, 1–5.

22 Ramani, V., Kumar, T., Bracken, A. et al. (2018). Secure and efficient data accessibility in blockchain based healthcare systems. *2018 IEEE Global Communications Conference (GLOBECOM)*, 206–212.

23 Xiao, Y., Zhang, N., Lou, W., and Hou, Y.T. (2020). A survey of distributed consensus protocols for blockchain networks. *IEEE Communications Surveys & Tutorials* 22 (2): 1432–1465.

24 Pongnumkul, S., Siripanpornchana, C., and Thajchayapong, S. (2017). Performance analysis of private blockchain platforms in varying workloads. *2017 26th International Conference on Computer Communication and Networks (ICCCN)*, 1–6.

25 Yang, F., Zhou, W., Wu, Q.Q. et al. (2019). Delegated proof of stake with downgrade: a secure and efficient blockchain consensus algorithm with downgrade mechanism. *IEEE Access* 7: 118541–118555.

26 Kang, J., Xiong, Z., Niyato, D. et al. (2019). Toward secure blockchain-enabled internet of vehicles: optimizing consensus management using reputation and contract theory. *IEEE Transactions on Vehicular Technology* 68 (3): 2906–2920.

27 Nguyen, D.C., Nguyen, K.D., and Pathirana, P.N. (2019). A mobile cloud based IoMT framework for automated health assessment and management. *2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, 6517–6520.

28 Zheng, X., Mukkamala, R.R., Vatrapu, R., and Ordieres-Mere, J. (2018). Blockchain-based personal health data sharing system using cloud storage. *2018 IEEE 20th International Conference on e-Health Networking, Applications and Services (Healthcom)*, 1–6.

11

Blockchain Standards

Hui Ding[1], Xiaofeng Chen[2], Kyeong Hee Oh[3], Ismael Arribas[4], Jörn Erbguth[5], Alexander Chuburkov[6], Lisa J. Y. Tan[7], and Xiangjuan Jia[8]

[1] *Digital Technologies, Ant Group, Beijing, China*
[2] *Hangzhou Qulian Technology Co., Ltd, Hangzhou, Zhejiang, China*
[3] *TCA services, Seoul, South Korea*
[4] *Kunfud, Spain*
[5] *Independent Consultant, Geneva, Switzerland*
[6] *Technical Committee on Standardization 'Hardware and Software for DLTs' (TC159), Moscow, Russia*
[7] *Economics Design, Singapore*
[8] *Hangzhou Qulian Technology Co., Ltd., Hangzhou, Zhejiang, China*

After reading this chapter you should be able to:

- Obtain an overview of the development and adoption progress of blockchain standards
- Understand the main topics in blockchain standards in regional, national, and industrial applications
- Propose possible next steps in its industrial applications

11.1 Introduction

This chapter covers the blockchain standards. Blockchain, whose definition has not yet reached consensus, is a technology set that incorporates not only information technologies such as peer-to-peer communication, encryption, and distributed networking, but also social science aspects including economics and finance, management and governance, law and legality,. Hence, the standard community of blockchain brings members from various backgrounds and results in multiple levels of gaps in their domain-knowledge. Standards set the basis of bridging those gaps so that people in this community can communicate and

Blockchains: Empowering Technologies and Industrial Applications, First Edition.
Edited by Anwer Al-Dulaimi, Octavia A. Dobre, and Chih-Lin I.

understand one another, set multilateral rules to balance benefits and costs, and hopefully maximize the positive impact while minimizing the risks of innovation.

This chapter is organized as follows. Section 11.2 provides an introduction of standards and an overview of initiatives of blockchain standards. Section 11.3 discusses the progress and plan of blockchain standards in the international, regional, and industrial levels. A summary and future directions in Section 11.4 conclude this chapter.

11.2 The Role of Blockchain Standards

11.2.1 A Brief Introduction to Standards

Standards development may be informally dated back to the eighteenth century with the onset of industrial revolution, when the procurement, production, and trading activities increased and expanded across multiple geographical locations, and evolved toward global supply chain collaboration and increasing division of labor.

Taking the automobile industry as an example, there are thousands of standard specifications in the production of a car giving detailed instruction on the raw, intermediary, and end products' functions, performance, process in production, and transportation. Refer to Figure 11.1 for an illustration of a typical automobile supply chain. The upstream of the supply chain is the production stream including raw material sourcing, production, assembly, and the output are ready-for-sale

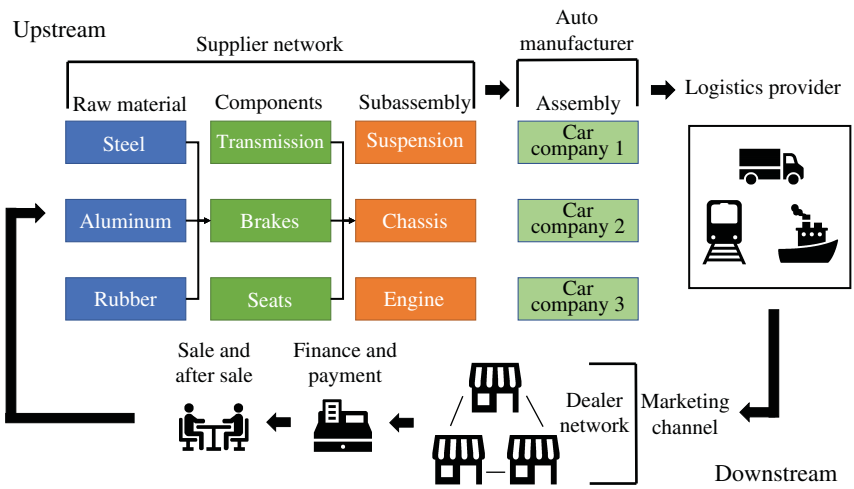


Figure 11.1 Illustration of a typical automobile supply chain.

cars. The raw materials may include metals, glasses, rubber, IoT devices, control systems, communication systems, etc. They go through sub-assembly lines and are sent to the assembly factories. Thanks to the standards, each tier of suppliers can deliver its products to the next tier, and the next tier can perform specification consistency tests to ensure that their order is correctly delivered according to their contract before acceptance and arrange their production plan with their purchases as inputs.

Standards also play an important part in downstream of the supply chain, which includes distribution and sales. Usually, the dealer network promotes this type of car to customers before the car reaches its warehouse. Despite the eye-catching advertisement, the focus of potential buyer will eventually turn to the list of specifications that define the functions and performance. And some of those specifications are defined in standards that are applicable to automobile industry.

Broadly speaking, standards are the linkage among multidisciplinary characteristics: politics, business and economics, science and technology, labor, and culture and ideas, which can include a set of definitions, requirements, measurements, operational steps, or a way of describing the world. Authors in [1] conducted extensive study on innovation and standardization and established that standardization is an essential component of innovation. With the acknowledgment of the importance of standards in production and consuming of innovative products, let's move on to the ways of how to develop a standard.

There are generally three ways to set a standard, namely de facto, de jure, and by negotiation.

- **De facto**: Adapted from its French meaning, a de facto standard is something that is used so widely that it is considered a standard for a given application although it has no official status. With the automobile industry example, the inventors of the car designed three-wheel and four-wheel car models, and finally, a typical modern car with four wheels are accepted by the mainstream market.
- **De jure**: A de jure standard is a technology, method or product that has been officially endorsed for a given application and is enforced by the law. These are mandatory standards and the products falling in these categories generally require certificates issued by government-backed regulatory bodies to enter the market and are subjected to regular check of conformity to related standards. In the automobile industry, there are a wide range of de jure standards enforced by regional or national laws including but not limited to wind shields, lighting systems, brakes, emissions [2, 3].
- **By negotiation**: The third method, by negotiation, is widely adopted by Standard Development Organizations (SDO) around the world. It is a way of reach consensus among stakeholders on a specific work item in a working group (WG). Depending on the enforcement power of stakeholders, such

standards may be adopted by a sovereign government, or an industrial sector across multiple jurisdiction bodies. For SDOs whose member include member states, such as ITU and ISO, the published standards are often adopted by its member states and enforced by a regulatory body in that state. For SDOs whose members are organizations and individuals, such as IEEE, W3C, IETF, the published standards tend to be adopted by its members in the applicable industrial sector. Despite the differences in governance, SDOs establish liaison relationship to coordinate in standards development on mutual interested topics; examples include a number of liaison relationships between TCs and WGs in ISO, ITU, and IEEE on blockchain. Sometimes, one SDO will adopt a completed standard from another SDO. For instance, 3GPP 5G technologies are agreed to become ITU-R IMT-2020 specifications [4].

11.2.2 Initiatives of Blockchain Standards

Blockchain standards can be viewed as a demonstration of a broad standards set that are not restricted to information and communications technology (ICT) but also stretches into related verticals along with its governance structure and economical design, which bring new challenges for standard development. As the driving force of world economy is shifting from capital to innovation and blockchain is regarded as the infrastructure of trust (names including "trust machine and" "internet of value") [5], the role of blockchain standards has moved beyond technical guidance toward a strategic competition to score a leading role in the digital economy era.

International, regional, and national SDOs started investigation and research into blockchain technologies around 2016 and explored blockchain standardization opportunities related to existing standards in their focus area. ISO established TC 307 – blockchain and distributed ledger technologies in April 2017 [6]. ITU-T setup focus group (FG) on distributed ledger technology in April 2017 and concluded in August 2019. IEEE Blockchain Initiative was effective from January 2018, which include not only blockchain standards but also education, conferences, events, and publications. Australia published its roadmap for blockchain standards in ISO/TC 307 in March 2017 and identified key components in developing blockchain standards including terminology, privacy, security, identity, risk, governance, and interoperability [7]. NIST published an overview on blockchain with discussion on blockchain's high-level components, limitations, and suggested organizations consider operational and governance issues before incorporating it into businesses [8]. European Union established the EU Blockchain Observatory and Forum in 2018 to accelerate blockchain innovation and the development of blockchain ecosystem within EU. EU provided over EUR 200 million in prizes and grants through Horizon 2020 from 2016 to 2020 [9]. The Ministry of Industry

and Information Technology of China published "Blockchain Technologies and Applications in China White Paper" in 2016 and announced the development of a series of national blockchain standards in 2018.

A recent report published by the World Economic Forum [10] conducted research on over 30 standards-setting entities, 185 jurisdictions, and nearly 400 industry consortia, among which are global SDOs with long standard developing history (ISO, ITU-T, IEEE, etc.), technical communities (W3C, IETF, etc.), open source communities (e.g. the Linux Foundation), newly formed industrial consortium dedicated to blockchain (e.g. R3 Consortium, BiTA). The authors analyzed the current landscape and assessed key issues and findings from a legal and regulation perspective. Some key findings include inconsistency in terminology, gaps and overlapping in scope of standards, various levels of geographic expertise and consumer representation. Authors in [11] also pointed out that definitions, building blocks, and architecture of blockchain were not in consensus among stakeholders in 2018. Further, they suggested developing interoperability-related blockchain standards for between blockchains and blockchain-like technologies as well as between blockchains and traditional technologies.

11.3 Landscape of Blockchain Standards

This section introduces the landscape of blockchain standard projects in international and national SDOs as well as some industrial alliances. There were 23 published blockchain standards in total at the end of 2020 from ITU-T, ISO, and IEEE. Adoption of these standards depends on the investment and R&D of blockchain products and services at national and industrial level.

Blockchain standards may be broadly categorized into two groups: one is the application-oriented standards describing how to use blockchain for a given application that mostly fall into verticals, the other is technology-oriented standards defining various technical aspects of blockchain components, protocols, and interoperability.

Figure 11.2 shows a classification of blockchain standard projects in its applicational and technical categories. The application-oriented projects are placed above the technology-oriented projects to illustrate that applications standards are enabled and supported by technological standards. Note that the projects are under development or published from international and national SDOs at the end of 2020. They will be discussed in the following paragraphs.

Figure 11.3 presents the number of published blockchain standards in different categories by ISO, ITU, and IEEE. It is observed that the published standards fall into terminology, architecture, security and privacy, requirements, data management, cryptocurrency, and testing. A majority of 12 standards are in the

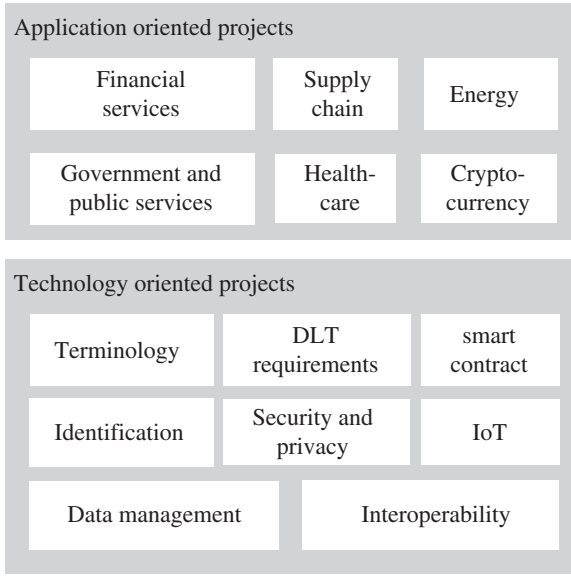


Figure 11.2 Blockchain standard projects classification.

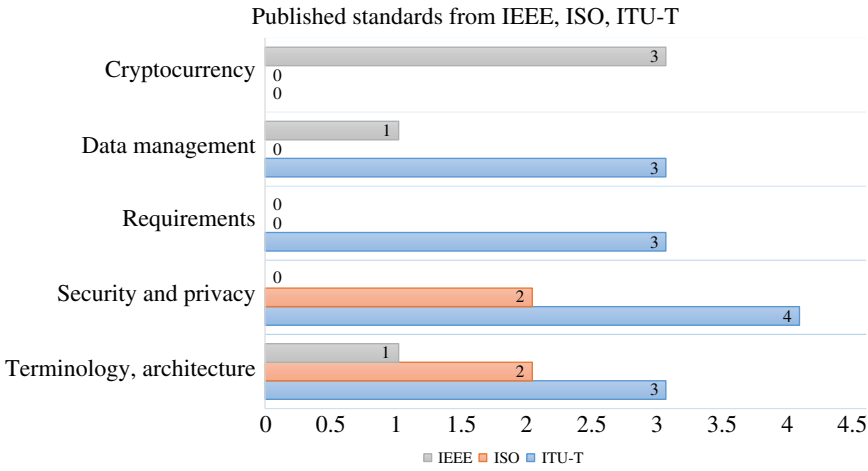


Figure 11.3 Published international blockchain standards.

categories of terminology, architecture, security, and privacy and led by ITU-T and ISO. With more than 50 active standard projects in IEEE covering both application category and technology category, three out of the four published standards are cryptocurrency-related.

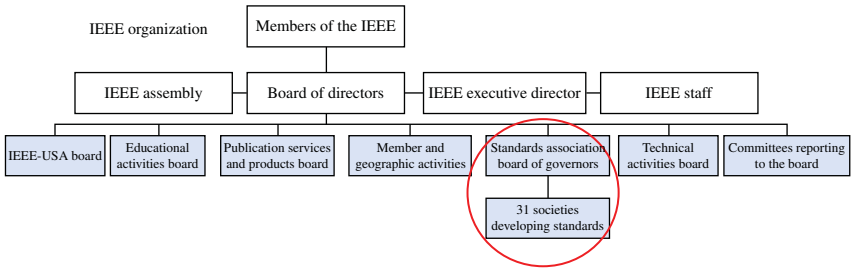


Figure 11.4 IEEE organization.

11.3.1 Blockchain Standards in IEEE

IEEE has 39 technical Societies and seven technical councils representing the wide range of IEEE technical interests. It has over 1200 active standards and more than 650 standards under development [12]. IEEE recognizes the vital role standards will play in the development and adoption of blockchain technologies. Figure 11.4 is adapted from IEEE organization to reflect the position of IEEE-SA in IEEE [13]. It shows that there are 31 Societies developing standards under IEEE-SA.

The Standards Committee (SC) is the organization that assumes responsibility for a particular standards idea within IEEE. The Standards Committee provides technical oversight for the standard. It also determines the scope and nature of the technical content. Figure 11.5 illustrates the organizational structure of IEEE SA. The seven dark gray boxes under Standard Board (SASB) are committees that coordinate the lifecycle of standard projects development, while the WG incubated by SC that is sponsored by societies develop and maintain the standards. There are

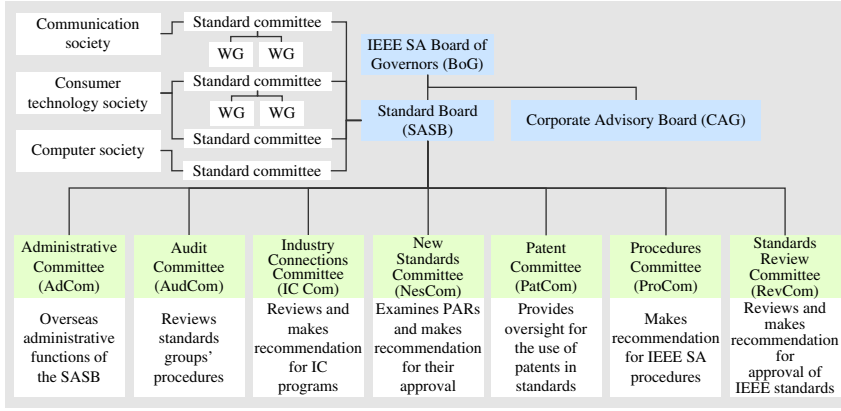


Figure 11.5 IEEE SA organizational structure.

two types of members in IEEE-SA, the corporate member and the individual member. IEEE standards are initiated by IEEE-SA member in a Project Authorization Request (PAR) document, once approved by SASB, a WG is formed. Upon completion of standard draft, it goes through one or more rounds of ballot. After the ballot is finished, the draft is submitted to SASB for final approval and publication. A published IEEE standard is valid for 10 years and can be updated by its standard committee if necessary.

IEEE Standard Association (SA), a globally recognized standards-setting body within IEEE, has been actively pursuing blockchain standardization efforts through various activities in multiple industry sectors, including healthcare, energy, agriculture, and finance. As of the end of December 2020, there are three published IEEE blockchain standards and 56 active standards projects under a number of standard committees [14], including Blockchain Standard Committee (BSC) and Digital Finance and Economy Standard Committee (DFESC) sponsored by Consumer Technology Society (CTS), Blockchain and Distributed Ledgers Standard Committee sponsored by Computer Society, and some others fall into the umbrella of vertical sector societies such as Power and Energy Society, Medicine and Biology Society. The development status of these standard projects varies as they're industrial efforts pushed forward by the WG Chair and its members while being over-sighted by its sponsoring society and IEEE-SA. The driving forces lay in the matureness of Blockchain technologies, marketing effectiveness, consumer acceptance, regulation recommendations, etc. Figure 11.6 shows the trend of blockchain standards development in IEEE-SA from 2017 to 2020.

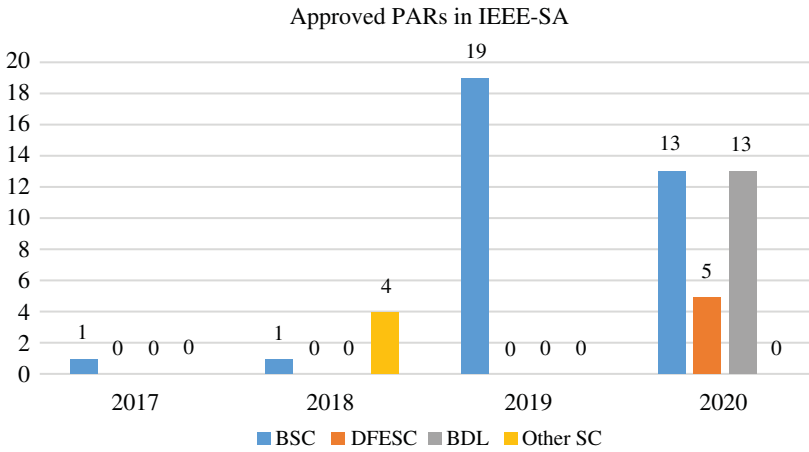


Figure 11.6 IEEE-SA approved PARs from 2017 to 2020.

It is observed that the earliest project is P2418.1 [15], which was approved on 15 June 2017, and it is followed by 10 other projects in the 2418 family. Most of the 2418.x projects explore blockchain applications in verticals except P2418.2 [16] (moved to C/BDL). These applications expand from agriculture, energy, vehicles, healthcare and social science, governmental applications, financial applications (supply chain finance, digital asset management and security tokens), and Internet of Things (IoT).

The three IEEE standards on blockchain all fall into the category of cryptocurrencies, reflecting the vitality of the multi-billion-dollar-trading-volume cryptocurrency market and the relative early stage of the applications of blockchain in other verticals. Based on technological and operational experiences of providing cryptocurrency-related services, the WGs of P2143 and P2140 published their standards. A summary of them is listed below.

1. **P2143.1-2020** [17] defines the general process of cryptocurrency payment between consumers and merchants.
2. **P2140.1-2020** [18] defines the general requirements for cryptocurrency exchanges and describes the exchanges' business logic, operational procedures, transaction specifications, user authentication programs, and fair voting system.
3. **P2140.5-2020** [19] defines a standard framework of a custodian service for cryptocurrency and digital assets.

As blockchain application trials and pilots emerge around the world, two SCs, Blockchain and Distributed Ledgers SC in Computer Society and Digital Finance and Economy SC in CTS, are approved by IEEE-SA and they're incubating blockchain standard projects within their scopes.

CTS/DFESC was officially approved on 5 March 2020 with the scope as follows [20].

The scope of the Standards Committee is to develop and maintain standards, recommended practices and guides for digital finance and economy (including but not limited to the digital forms, digital representations, digital embodiments, or digital equivalents of currencies, issuance, custody, payment, insurance, funds, shares, stocks, equities, securities, bonds, treasury bills, options, derivatives, futures, forward commitment, contingency claims, hedge funds, portfolio management, public benefit, mutual assistance, sharing economy, credit score, credit ratings, valuation, risk analysis, etc., and related hardware, software, systems, services, and applications in various scenarios), using an open and accredited process, and to advocate them on a global basis. Its technical scope is intended to be flexible and is ultimately determined by the sum of its approved PARs.

C/BDL was founded on 5 September 2019 by SASB with the scope as follows. The IEEE Computer Society Blockchain and Distributed Ledger Standards Committee manages the development of standards within the area of blockchains and distributed ledgers, including standards for relevant data formats, the development and implementation of blockchains and distributed ledger systems (DLS), and for applications of blockchain and distributed ledgers to specific sectors, industries, and processes. Until the end of 2020, BDL housed 17 active projects. These projects cover blockchain access control, interoperability, digital asset, identity system, consensus framework, governance, testing and evaluation, etc. Refer to their website for more information [21].

Readers are suggested to refer [22] for a list of active blockchain standard projects in IEEE-SA.

Key takeaways: The active blockchain standard projects in IEEE-SA cover a wide range of topics mainly because of its governance model, standard setting procedures, and member compositions. With different levels of developing progress, the impact of these standards will be largely dependent on the marketing of products employing the technologies defined in the standards. It is an internationally recognized platform for individuals, enterprises, and institutions that work in blockchain area.

11.3.2 Blockchain Standards in ITU-T

The ITU-T stands for International Telecommunication Union – Telecommunication Standardization Sector. ITU started in 1865 with the first International Telegraph Convention in Paris signed by 20 founding members, and the International Telegraph Union was established to supervise subsequent amendments to the agreement. It became a specialized agency of the United Nations in 1947 and was renamed ITU-T in 1993. Now ITU-T assemble experts from around the world to develop international standards known as ITU-T Recommendations which act as defining elements in the global infrastructure of ICTs. ITU-T includes both governments as member states and organizations such as operators, vendors, financing institutions, research institutions, and regional telecommunication organizations as sector members. There are nearly 200 member states and more than 900 members, including companies, universities, and international organizations, as well as regional organizations.

ITU-T has two approval processes for standardization, namely AAP (alternative approval process) and TAP (traditional approval process). The commonly used one is the AAP that is a fast track procedure to bring the demanded standards to market in time. Figure 11.7 shows the AAP procedure. The final decision at the Study Group (SG) meeting must not be opposed by more than one Member State present at the meeting.

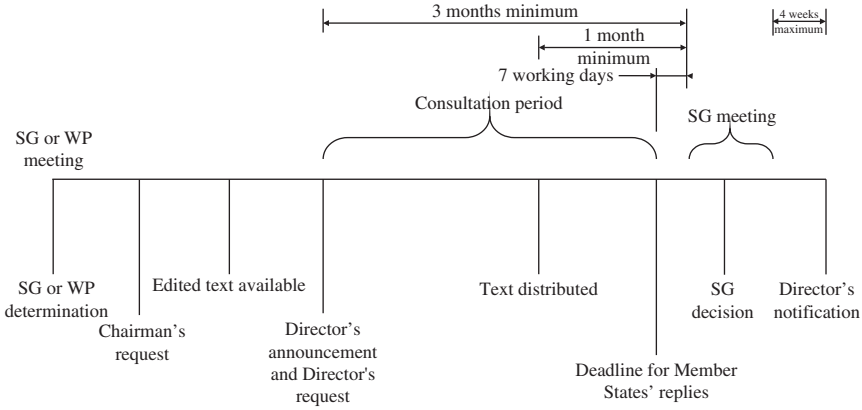


Figure 11.7 The alternative approval process of ITU-T.

The TAP is used for Recommendations that may have regulatory or policy implications. It needs a more strict and longer process, and the final decision must be unopposed.

ITU-T has 11 SGs, and now the six of them are developing the Recommendations-related blockchain. As of December 2020, ITU-T released 14 blockchain/distributed ledger technology-related standards. Four of the published standards are security-related, and three of them are related to data management and IoT, which shows that ITU attaches great importance to security issues and IoT and data issues. Figure 11.8 shows the SGs currently working on the blockchain standards.

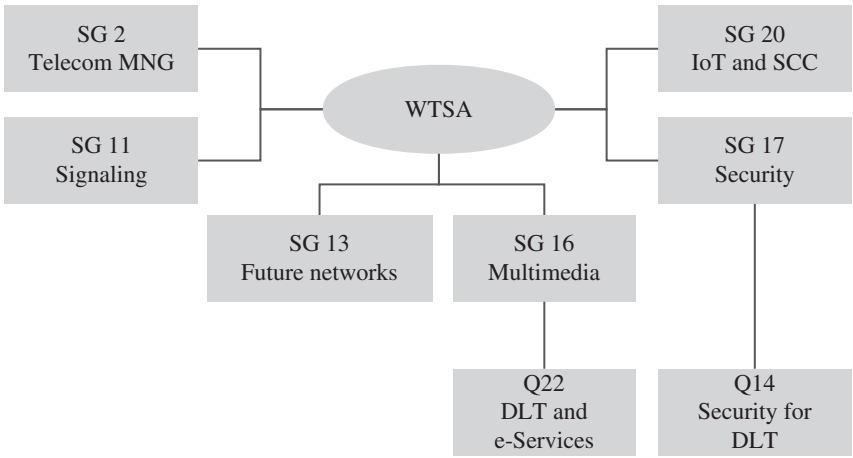


Figure 11.8 The ITU-T Study Groups currently developing blockchain-related standards.

Table 11.1 Q14/17 published items on blockchain.

Rec.#r	Title	Year
X.1400	Terms and definitions for distributed ledger technology	2020
X.1401	Security threats of distributed ledger technology	2019
X.1402	Security framework for distributed ledger technology	2020
X.1403	Security considerations for using DLT data in Identity Management	2020
X.1404	Security assurance for distributed ledger technology	2020

ITU-T SG 17, titled security, coordinates security-related work across all ITU-T SGs. It often works in cooperation with other SDOs and various ICT industry consortia focusing on security aspects. Question 14 in SG 17 (Q14/17) was established in 2017 to develop Recommendations on security aspects for distributed ledger technologies (DLT). It has published five Recommendations (Table 11.1). Table 11.2 lists the items under development.

There are other Questions in SG 17 that are developing DLT-related Recommendations. Q8/17 is the Question that develops cloud computing and Big data infrastructure security standards and is working on X.BaaS-sec, Guideline on blockchain as a service (BaaS) security.

SG 16 Multimedia has established Q22, distributed ledger technologies, and e-services in 2019. Q22/16 has published three Recommendations (Table 11.3) and two technical papers based on the FG-DLT deliverables (Table 11.4).

SG 16 is also developing many items related to DLT in Q22 and Q24, Human factors-related issues for improvement of the quality of life through international telecommunications (Table 11.5).

At the September meeting in 2017, SG 17 proposed establishing a FG under TSAG (Telecommunication Standardization Advisory Group) to research the need for DLT standards development comprehensively in the scope of the entire ITU-T and over. During the TSAG meeting, the need for study on digital currency was recognized. As a result, two focus groups of FG-DLT, "focus group on applications of DLT" and FG-DFC, "focus group on digital currency, including digital fiat currency," were established. The groups had worked for three years, and the results were sent to corresponding SGs, for example, SG 16 and SG 17, to be considered for appropriate standards. Typical examples are D.1.1, which are the basis of X.1400

Table 11.2 Q14/17 blockchain-related standard items underdevelopment.

Item acronym	Title	Year
X.str-dlt	The security threats and requirements for digital payment services based on distributed ledger technology	2021
X.ss-dlt	Security Services based on Distributed Ledger Technology	2021
X.stov	Security threats to online voting using distributed ledger technology	2021
X.das-mgt	Security framework for the data access and sharing management system based on the distributed ledger technology	2021
X.tf-spd-dlt	Technical framework for secure software programme distribution mechanism based on distributed ledger technology	2021
X.srip-dlt	Security requirements for intellectual property management based on distributed ledger technology	2021
X.sc-dlt	Security controls for distributed ledger technology	2022
X.srscm-dlt	Security requirements for smart contract management based on DLT	2022
X.sa-dsm	Security architecture of data sharing management based on DLT	2023
TR.qs-dlt	Guidelines for quantum-safe DLT system	2023

Table 11.3 Q22/16 published blockchain standards.

Rec.#r	Title	Year
F.751.0	Requirements for Distributed Ledger Systems	2020
F.751.1	Assessment criteria for DLT	2020
F.751.2	Reference framework for distributed ledger technology	2020

Table 11.4 Q22/16 published FG-DLT technical papers.

Item acronym	Title	Year
HSTP.DLT-RF	Distributed ledger technology: regulatory framework	2019
HSTP.DLT-UC	Distributed ledger technologies: use cases	2019

Table 11.5 SG 16 blockchain items.

Item acronym	Title	Year
H.DLT-DE	Digital evidence services based on DLT	2021
F.BVSSI	Scenarios and requirements for blockchain in visual surveillance system interworking	2021
F.DLT.HC	Requirements of distributed ledger technologies (DLT) for human-care services	2021
F.DLT.PHR	Service models of distributed ledger technologies (DLT) for personal health records (PHRs)	2021
F.HFS-BC	Requirements and framework for blockchain-based human factor service models	2021
F.Supp-OCAIB	Overview of convergence of artificial intelligence and blockchain	2021
F.DLT-FIN	Financial distributed ledger technology application guideline	2021
HSTP.DLT-GTI	DLT governance and technical interoperability framework	2021
HSTP.DLT-INV	General framework of DLT-based invoices	2021
HSTP.DLT-TFR	Technical framework for DLT regulation	2021
HSTP.DLT-VERI	Formal verification framework for smart contract	2021
HSTP.DLT-Risk	DLT-based application development risks and their mitigations	2021
F.DLIM-AHFS	Requirements of the distributed ledger incentive model for agricultural human factor services	2021
F.DLS-SHFS	Requirements of distributed ledger systems (DLS) for secure human factor services	2021
F.Med-VHN	Framework of telemedicine service based on distributed virtual healthcare network	2021

of SG 17, and D3.1 and D3.3, which are the basis of F751.1 and F.751.2 of SG 16, respectively. Table 11.6 lists the deliverables from FG-DLT.

FG-DFC deliverables are not directly focused on DLT, but they provide general requirements to consider when using digital currency, and this also applies to the DLT-based digital currency. FG-DFC closed in 2019, but the work continues now in Digital Currency Global Initiative, a collaboration between the ITU and Stanford University's Future of Digital Currency Program.

SG 13, titled as "Future networks, with focus on IMT-2020, cloud computing and trusted network infrastructure," has led work on next-generation networks

Table 11.6 FG-DLT deliverables.

WG #	Title	Deliverables
1	State of Art: Ecosystem, Terms, Definitions, Concepts	D1.1 DLT terms and definitions D1.2 DLT overview, concepts, ecosystem D1.3 DLT standardization landscape
2	Applications and Services	D2.1 DLT use cases
3	Technology Reference Framework	D3.1 DLT reference architecture D3.3 Assessment criteria for DLT platforms
4	Policy Reference Framework	D4.1 DLT Regulatory framework
5	Standardization Roadmap	D5.1 Outlook on DLTs

and aspects of mobile telecommunications. This group is interested in the use of blockchain in the next-generation network and is working on standardization around the concept of BaaS. SG 13 has published two recommendations and is developing two more.

- **Y.3550**: Cloud computing – functional requirements for BaaS.
- **Y.2342**: Scenarios and capability requirements of blockchain in next-generation network evolution.
- **Y.SCid-fr**: Requirements and Converged Framework of Self-Controlled Identity based on Blockchain.
- **Y.NRS-DLT-reqts**: Scenarios and requirements of network resource sharing based on distributed ledger technology.

SG 20 develops the Recommendations on the IoT and smart cities and communities (SC&C). It started standardizations with the concept of blockchain of things (BoT) in 2018 and now has published 4 Recommendations (Table 11.7) and been working on five items related to blockchain (Table 11.8).

SG 20 also agreed on the supplement of Y.Supple.62, titled "Overview of blockchain for supporting IoT and smart cities and communities in data processing and management aspects" in 2020.

SG 11, titled "Signalling requirements, protocols and test specifications," started draft Recommendation Q.BaaS-iop-reqts, "Interoperability testing requirements of blockchain as a service."

SG 2, titled "Operational aspects of service provision and telecommunications management," is responsible for the maintenance of ITU's International Numbering Resource (INR) database and also for standards on the management of telecom services, networks, and equipment. It now develops Recommendations

Table 11.7 SG 20 published blockchain IoT items.

Rec. #	Title	Year
Y. 4464	Framework of blockchain of things as decentralized service platform	2019
Y.4560	Blockchain-based data exchange and sharing for supporting Internet of things and smart cities and communities	2020
Y.4561	Blockchain-based Data Management for supporting Internet of things and smart cities and communities	2020
Y.4907	Reference architecture of blockchain-based unified KPI data management for smart sustainable cities	2020

Table 11.8 SG 20 blockchain IoT items under development.

Item acronym	Title	Year
Y.dec-IoT-arch	Decentralized IoT communication architecture based on information centric networking and blockchain	2021
Y.IoT-BC-reqts-cap	IoT requirements and capabilities for support of blockchain	2021
Y.IoT-rf-dlt	OID – based Resolution framework for transaction of distributed ledger assigned to IoT resources	2021
Y.BC-SON	Framework of blockchain-based self-organization networking in IoT environments	2021
Y.blockchain-terms	Vocabulary for blockchain for supporting Internet of things and smart cities and communities in data processing and management aspects	2021
Y.IoT-BoT-peer	Capability and functional architecture of peer of blockchain of things	2022

related to blockchain management. Two Recommendations are under development in SG 2.

- **M.rmbs**: Requirements for management of blockchain system.
- **M.immbs**: Information model for management of blockchain system.

ITU-T Study Group 5 (SG5) is responsible for studies on methodologies for evaluating ICT effects on climate change and publishing guidelines for using ICTs in an eco-friendly way. SG 5 hasn't developed any Recommendations related to DLT yet. However, it established a new Focus Group on Environmental Efficiency for Artificial Intelligence and other Emerging Technologies (FG-AI4EE), which

includes DLT as a part of emerging technologies, that should be considered in view of environmental efficiency.

One of the most important initiatives is the Financial Inclusion Global Initiative (FIGI) Symposium. It will be held on an annual basis from 2017 to 2020, (the last FIGI Symposium has been postponed to 2021) in order to provide a forum for dialogue regulators from telecom and financial services between ITU, World Bank group, Bank for international settlements, and Bill and Melinda Gates foundation. FIGI is composed of four workstreams, which are security, DLT for financial inclusion, quality of service, and trust. It published a Security aspects of DLT report.

Key takeaways: Six out of 11 SGs are developing blockchain-related standard items in ITU-T. As an official SDO in ICT, ITU-T does not stop exploring innovative technologies such as blockchain that has impact on possibly many aspects in ICT. Given member states enforcement power in regional and national level, ITU-T Recommendations are likely to be adopted in the participants' states and have impact on the products that are sold in those states.

11.3.3 Blockchain Standards in ISO

ISO is an independent, non-governmental international organization with a membership of 165 national standards bodies and 23 664 standards touch almost all aspects of daily life, and work for businesses large and small [23]. Its famous standards include the ISO 9000 family on quality management, ISO 14000 family on environmental management, and many more.

In 2016, the International Standards Organization established a new technical committee, ISO/TC 307, with the scope of "standardization of blockchain technologies and distributed ledger technologies." During the last four years with 46 participating members, 13 observing members, and 12 organizations under liaison categories A and B, ISO/TC 307 completed an active working plan within 7 WGs which have delivered 4 published standards and 11 standards under development.

As shown in Figure 11.9, the internal WGs are as follows, with AHG2 for "Guidance for Auditing DLT Systems."

- **WG1**: Foundations.
- **WG2**: Security, Privacy and Identity.
- **WG3**: Smart Contracts and their applications.
- **JWG4**: Joint ISO/TC307 - ISO/IEC JTC1/SC27 WG: Blockchain and distributed ledger technologies and IT Security techniques.
- **WG5**: Governance.
- **WG6**: Use Cases.
- **SG7**: Interoperability of blockchain and distributed ledger technologies.

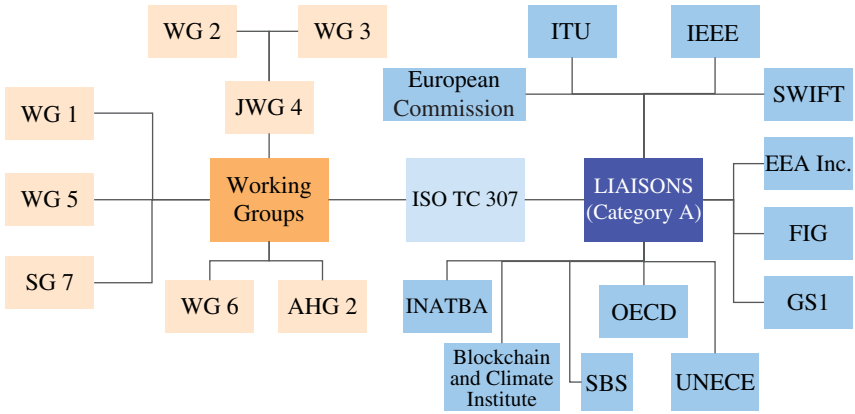


Figure 11.9 ISO/TC 307 WGs and external liaison relationships.

ISO/TC 307 established both internal and external liaison relationships which allow harmonization in transversal areas between different Technical Committees of ISO and other SDOs and consortia.

Figure 11.10 illustrates the internal liaison relationships ISO/TC 307 established with other ISO and ISO/IEC groups. Double-end arrow represents both groups can access each other's documents, while single-end arrow represents one-way document access. It appears that ISO/IEC experts reach consensus that blockchain technologies may affect basic information technology, software and system engineering, data management, IT services and governance, security, and it may also be applicable to biometrics, health informatics, financial services, artificial intelligence, aircraft and space vehicles, business and risk management, etc.

The right part of Figure 11.9 shows the organizations that TC 307 established external liaison relationships with. It is observed that ISO/TC 307 seeks standardization in broad areas of blockchain technologies and blockchain applications in different verticals including economics, finance, and environments.

Two lists below show the published standard items and those under development. The topics cover blockchain terminology, architecture, privacy, smart contract, identification, governance, etc.

Published Standards.

- **ISO 22739:2020**: Blockchain and Distributed ledger technologies – Vocabulary
- **ISO/TR 23244:2020**: Blockchain and distributed ledger technologies – Privacy and personally identifiable information protection considerations.

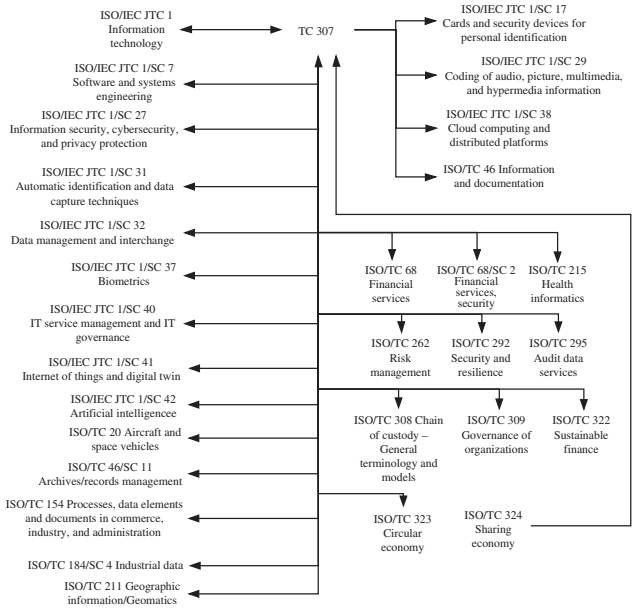


Figure 11.10 ISO/TC 307 internal liaison relationships.

- **ISO/TR 23455:2019**: Blockchain and Distributed ledger technologies – Overview of and interactions between Smart Contracts in blockchain and Distributed ledger technology systems.
- **ISO/TR 23576:2020**: Blockchain and distributed ledger technologies – Security management of digital asset custodians.

Standards Under Development.

- **ISO/DTR 3242**: Blockchain and Distributed ledger technologies – Use cases.
- **ISO/WD TR 6039**: Blockchain and Distributed ledger technologies – Identifiers of subjects and objects for the design of blockchain systems.
- **ISO/AWI TR 6277**: Blockchain and Distributed ledger technologies – Data flow model for Blockchain and DLT use cases.
- **ISO/AWI 22739**: Blockchain and Distributed ledger technologies – Vocabulary.
- **ISO/DTR 23249**: Blockchain and Distributed ledger technologies – Overview of existing DLT system for identity management.
- **ISO/DIS 23257**: Blockchain and Distributed ledger technologies – Reference Architecture.
- **ISO/DTS 23258**: Blockchain and Distributed ledger technologies – Taxonomy and Ontology.
- **ISO/AWI TS 23259**: Blockchain and Distributed ledger technologies – Legally binding Smart contracts.
- **ISO/DTS 23635**: Blockchain and Distributed ledger technologies – Guidelines for governance.
- **ISO/AWI TR 23642**: Blockchain and Distributed ledger technologies – Overview of Smart contract security good practice and issues.
- **ISO/WD TR 23644**: Blockchain and Distributed ledger technologies – Overview of trust anchors for DLT-based identity management (TADIM).

Key takeaways: Since ISO standards cover many aspects of daily life and business, it has an advantage from the interoperability perspective in blockchain standard suite. ISO/TC 307 starts from the basics of blockchain technology, and with internal liaisons with other SCs and TCs in ISO/IEC, TC 307 may potentially have higher leverage on expertise and application scenarios than other SDOs.

11.3.4 Regional, National, and Industrial Blockchain Standards

Besides international SDOs discussed above, there are a number of regional, national, and industrial blockchain standards are published or under development. They are connected with ISO, ITU-T, and/or IEEE through membership or liaison relationships.

11.3.4.1 ETSI

European Telecommunication Standards Institute (ETSI) is an independent, not-for-profit, standardization organization in the field of telecommunications, broadcasting, and other electronic communications networks and services. ETSI is officially recognized by European Union as one of the three ESO (European Standards Organization). There are over 900 member organizations drawn from 65 countries and five continents including large and small private companies, research entities, academia, government, and public organizations [24].

ETSI launched a new industry specification group (ISG) on permissioned distributed ledger (PDL) in December 2018 with founding members Ericsson, Huawei, Intel, Telefónica, Vodafone. The target adopters of PDL are industries and governmental institutions. According to ETSI press release, the new group will work on the operation of PDLs, business use cases, functional architecture and solutions for the operation of PDLs, including interfaces/APIs/protocols and information/data models as well as other topics [25]. PDL released the first batch of group reports in February 2021. There are two other reports released earlier in 2020 on PDL landscape review [26] and PoC framework [27], respectively.

- **ETSI GR PDL 002**: "Applicability and compliance to data processing requirements" was published in November 2020 [28]. This report focused on the scenario where data sources (sensors, gateways, etc.) interacted with PDL platforms. Some topics including sensors/devices attributes, data privacy, certifications are discussed in this report.
- **ETSI GR PDL 003**: "Application Scenarios" was published in December 2020 [29]. It introduced a high-level PDL reference framework, PDL's integration with ICT infrastructure, applications, and PDL governance.
- **ETSI GR PDL 004**: "Smart Contracts: System Architecture and Functional Specification" was published in February 2021 [30]. It presented a reference architecture of the smart contracts, its lifecycle phases, applications, and a discussion on threats and limitations of smart contracts.

The ISG PDL plans to develop normative specifications based on published reports and feedbacks from operational experience with emphasis on new application environments enabled by the emergence of next-generation networking infrastructures [31]. Leveraging ETSI's partner with the international Third-Generation Partnership Project (3GPP), PDL may have advantages incorporating blockchain technologies with 4G and 5G mobile communications, and machine-to-machine communications.

11.3.4.2 DIN in Germany

The DIN is the Standards Association of the German Industry. It was founded in 1917 and became an ISO member in 1951 as the sole national standards

organization for Germany. In 1961 DIN became a member of the European Committee for Standardization (CEN). DIN publishes standards and specifications (DIN SPECs). DIN SPECs were introduced in 2009 to allow faster development without requiring full consensus.

DIN has published several DIN SPECs regarding Blockchain and DLT:

- **DIN SPEC 3103**: Blockchain and distributed ledger technologies in application scenarios for Industrie 4.0 (June 2019)
- **DIN SPEC 3104**: Blockchain-based validation of data (April 2019)
- **DIN SPEC 4996**: Blockchain-based approach to the transfer of software licenses (April 2020)
- **DIN SPEC 4997**: Privacy by Blockchain Design: A standardized model for processing personal data using blockchain technology (April 2020).

DIN SPEC 4997 – Privacy by Blockchain Design [32] defines a standardized model for processing personal data using blockchain technology. This DIN SPEC classifies different approaches of handling personal data and exemplifies them with architectural blueprints. The latter describes technical design patterns in order to mitigate the risk and therefore raise the data protection level in an IT system. It is guided by the European General Data Protection Regulation (GDPR) and its "Privacy by Design" principle. The DIN SPEC 4997 committee claims to provide:

- A description of the functional requirements for DLT systems, in particular blockchain, to achieve compliance with GDPR requirements;
- A guide to the handling of personal data using blockchain technology;
- Architectural blueprints to illustrate the uses of blockchains to improve privacy;
- Procedures for business processes for the iterative maintenance and quality assurance of data protection.

First, the chapter "GDPR awareness" discusses how to apply GDPR to DLT systems. Then the chapter "Principles of data protection and their risks from the perspective of Privacy by Design" introduces the principles of the GDPR focused on the risk-based approach. It introduces "Privacy by Design" as a way to mitigate these risks by technical measures. It links the legal principles to technical measures like pruning, off-chain storage and hashing, K-anonymity, salting, ring signatures, fully homomorphic encryption, secure multiparty computation, trusted execution environments, and zero-knowledge proofs. Finally it provides an architectural blueprint for an IT system processing personal data utilizing a blockchain-based tamper-proof access log.

11.3.4.3 UNE CTN 71/SC307 in Spain

UNE is the Spanish Standardization Association within the Kingdom of Spain for ISO, the National Body of Spain, committed since the beginning with the

establishment of ISO/TC 307 and in particular worked on identity management. Recently the Spanish Bulletin published the UNE 71307-1 Part 1: Reference Framework: Decentralized Identity Management Model on Blockchain and other Distributed Ledger Technologies [33] which is the world's first standard on decentralized digital identity for blockchain and distributed ledger technologies.

UNE Standard 71307-1 defines a generic reference framework for the decentralized issuance, administration and use of attributes that facilitate the characterization (identification) of individuals or organizations; allowing individuals to create and control their own digital identity in a self-managed manner, without the need for centralized authorities.

Adoption of standardized models for decentralized management of identity information is the optimal method that will ensure the security of organizations' processes and individuals, protect their privacy while maintaining full control over their individual identity and its use as opposed to centralized traditional models where individuals have very little or no control over the use of their identity. Among other advantages, it also helps preventing possible identity theft on the Internet.

It considers several basic concepts and processes of decentralized identity management to facilitate that the technological systems supporting them may be compliant with the relevant business, contractual and regulatory requirements. This standard has been developed in UNE's CTN 71/SC 307 technical standardization committee, with the participation and consensus of all its members. The committee CTN 71 on digital enabling technologies was established as part of the initiative of the Secretary of State for Digitization and Artificial Intelligence.

The publication of the standard is a result of the culmination of intensive work that began in July 2019 with the establishment of the CTN 71/SC 307 GT1 WG. It will serve as the basis for the future development of other standards globally in the field of decentralized management of identity information; therefore, the Spanish standardization body has proposed that it become the European standard for the European standardization bodies CEN and CENELEC.

11.3.4.4 LACChain Alliance in Latin America and the Caribbean

LACChain is the Global Alliance for the Development of Blockchain Ecosystem in Latin America and the Caribbean led by the innovation lab of Inter-American Development Bank (IDB Lab). LACChain has developed a techno-legal framework based on a set of standards, protocols, technologies, rules, policies, and agreements that enable the building of a multi-purpose blockchain network of networks that is robust, reliable, sustainable, compliant with regulation, and scalable (the LACChain infrastructure). The LACChain infrastructure is provided as a public good to Latin America and the Caribbean [34].

One of the main characteristics of the LACChain Infrastructure is the capacity to resolve disputes based on an underlying orchestration entity that is neutral and agnostic to technology as well as accountable with privacy and data protection. By the end of 2020, the LACChain Blockchain Network was being used for more than 40 entities including governments, financial institutions, start-ups, multilaterals, and universities. They are leveraging it for a diverse spectrum of use cases, including trade, digital diplomas, time-stamping, digital identity, procurement processes, supply chain, and traceability of food products, among others.

The Self-Sovereign Identity (SSI) standards (e.g. decentralized identifiers and verifiable credentials) are used across the LACChain Infrastructure and different platforms and applications built on top of it. The use of SSI in LACChain is aligned with the industry needs, respectful with the role of government in the identification [35].

LACChain is also confronting the global pandemic by setup partnerships with sustainable developing goals (SDG) 17 whereby it is deploying a COVID-Net for universal use cases such as traceability of vaccines and digital vaccination certificates.

11.3.4.5 ISO, ITU Participation, and National Blockchain Standards for Financial Asset Management in Russia

In parallel with the works of ISO and ITU in the field of international standardization of blockchain and DLTs, at the national level in Russia, similar works are carried out by national technical committees on standardization, TC 26 "cryptography and security mechanisms" [36] and TC 159 "distributed ledger technology and blockchain software and hardware" [37].

National experts on standardization developed terms and definitions for blockchain and DLTs [38], which were used to assess the compliance with national legislation in the field of information security of the first information system designed using blockchain and DLTs – "Masterchain" in Russia.

Terms and definitions for blockchain and DLTs were published by the national standardization committee TC26. This document was applied to provide legitimacy to the first certified blockchain platform in Russia – "Masterchain."

The technological architecture of the "Masterchain" was used in the development of the reference architecture of the ITU-T Focus Group on Application of Distributed Ledger Technology [39]. A description of the technology was also presented in the ISO/DTR 23249 Blockchain and distributed ledger technologies – Overview of existing DLT systems for identity management [40].

Technical Committee 159 works on the development of the guideline "Messaging protocol for message exchange between DLT-platforms." The guideline aims to describe the principles of creating a consortium-based blockchain network with an arbitrary number of DLT platforms as participants of consortium networks.

At the level of national regulation in Russia, some laws were introduced to adopt blockchain assets: "digital financial assets" and "digital currencies." Special were defined for a new type of non-credit financial institution and roles – "operator of the information system for digital financial assets" [41].

The Bank of Russia produces research on the central bank's digital currency (CBDC) – "A Digital Ruble" [42]. At the moment, the Central Bank of Russia started a public consultation. This research mentions blockchain and DLTs for the purposes of CBDC, including various mechanisms for the technical implementations of a CBDC, information security specifics of decentralized and hybrid technologies.

To summarize, all this work aimed at formalizing the properties of blockchain/DLT systems in order to make them interoperable with existing systems, and there have been some successes along the way, but much remains to be done, especially in matters of confidentiality and interoperability.

11.3.4.6 Blockchain Standards in China and Financial Sector Application

In order to standardize the development of applications based on DLT in the financial sector, the People's Bank of China has successively published two industry standards, JRT 0184-2020 "Financial Distributed Ledger Technology Security Specification" and JRT 0193-2020 "Blockchain Technology Financial Application Evaluation Rules" [43, 44].

In February 2020, JRT 0184-2020 "Financial Distributed Ledger Technology Security Specification" was released. It specifies the security system of financial distributed ledger technology, including basic hardware, basic software, cryptographic algorithms, node communications, ledger data, consensus protocols, smart contracts, identity management, privacy protection, regulatory support, operation and maintenance requirements, and governance mechanisms. It is applicable to institutions engaged in the construction of DLS or service operations in the financial field.

In July 2020, JRT 0193-2020 "Blockchain Technology Finance Application Evaluation Rules" was released. This standard is based on JRT 0184-2020, and it stipulates the implementation requirements, evaluation methods, and criteria for the application of blockchain technology in the financial field. The standard defined three levels of evaluation of blockchain financial applications including basic requirements, performance, and security, so as to assess whether the system can guarantee the safe and stable operations of financial facilities and applications. It mainly targeted financial institutions to carry out product design, software development, and system evaluation to financial applications of blockchain technologies.

The releases of the two standards provide standardized guidelines for financial institutions to build blockchain systems and services. It helps financial

institutions to implement system deployment and maintenance in accordance with appropriate security requirements. It provides business assurance capabilities and information risk-resistant capabilities for large-scale applications of blockchain. It may be referenced for future blockchain applications in the financial sector in China.

The Standardization Administration of China (SAC), a representative of China in ISO, approved the first national blockchain-related project in December 2017. The national blockchain and DLT standard committee was approved with 22 national standards plan that ranges from the basic blockchain and DLT technologies, services and applications, processes and methods, trust and interoperability, and information security.

DCEP pilots in China: Electronic payment has spread to all aspects of life in China, but all electronic payments require linking to a bank account registered in China, which is the real-name registration account. The digital currency DCEP (Digital Currency Electronic Payment) issued by the central bank is a "digital payment tool with value characteristics." It is essentially a substitute for banknotes in its digital form. Its functions and properties are exactly the same as banknotes (M0), and has features such as liquidity, storability, offline transaction, controllable anonymity, non-forgeability, non-repeatable transaction, and non-repudiation. DCEP is able to complete the payment without a user's bank account. It aims to reduce the degree of account reliance on transactions, which is conducive to the circulation and internationalization of CNY. At the same time, DCEP can realize real-time collection of data such as currency creation, accounting, and flow, which provides a useful reference for currency placement and monetary policy formulation and implementation.

DCEP's design and issuance adopts a two-tier operating system of "People's Bank-Commercial Bank-Users" for R&D and exchange. The first layer is for the central bank to issue and withdraw digital currency to commercial banks, and the second layer is for commercial banks to encrypt the digital currency amount and ownership authentication information are sent to the terminal for acceptance, and the wallet data change is completed. The two-tier model can effectively guarantee the purity of the creditor's rights and debt relationships in the existing monetary system. The centralized management model safeguards the central bank's authority in currency circulation, provides undifferentiated credit guarantees for digital currencies, and provides basic support for the circulation of digital currency.

Several pilot projects of the DCEP application were rolled out [45]. In January 2020, PBOC completed the top-level design, standard formulation, function research and development, joint debugging and testing of DCEP. In March 2020, experts from the National Development and Reform Commission recognized the

importance of digital economy and accelerated the launch of DCEP. The first batch of pilots are performed in developed cities of China.

In May 2020, DCEP was issued to Suzhou, Jiangsu Province, as a form of transportation subsidy to employees of district-level agencies, institutions, and directly affiliated enterprises. In October 2020, DCEP passed digital red envelope testing in Shenzhen, which is in wider scope and more scenarios than the Suzhou pilot. The retail merchants participant covers large commercial supermarkets, food catering, retail supermarkets, and everyday life services. The Shenzhen pilot demonstrated the advantages of digital currency in consumer retail. In December 2020, another Jiangsu pilot launched DCEP red envelopes that were worth 100 000 digital RMB. This is the first time in the DCEP tests that payment were completed without a payment network. As more pilots rolled out, through the continuous development of DCEP and the improvement of various functions, DCEP may be promoted and used on a larger scale in the future.

11.3.4.7 Blockchain Standards in Communication Networks

Blockchain technology and governance stacks support services and applications such as trusted data exchanges, distributed financial applications for enterprises and governments as the "trust infrastructure." In fact, the building blocks of blockchain exploit the communication networks that provides communication, networking, storage, and computing capabilities. Standing at the OSI Network Model's point of view, blockchain services and applications are built on top of the underlying communication networks that may be managed under a distributed, consensus-based, incentive-supported governance model depending on the specific requirements of the scenario.

As discussed in Section 11.3.4.1, ETSI ISG PDL identified Internet service provider (ISP) application on Internet resources management. IETF launched a Distributed Internet Infrastructure research group (DINRG) in September 2017 to investigate open research issues in decentralizing infrastructure services such as trust management, identity management, name resolution, resource/asset ownership management, and resource discovery [46] and there are no documents available yet. According to the TMF report [47], communication service providers (CSP) including Telefonica, Globe Telecom, Deutsche Telekom, Vodafone, AT&T, Colt Technology Services, SoftBank conducted research and completed a number of PoCs on settlement, roaming, payments, identity management, and authentication in cross-carrier and/or cross-border scenarios.

There are potential benefits of blockchain for a number of collaborative network control and management tasks including domain networking, network slicing, spectrum sharing, and identity management. In 5GPPP key achievements v 3.0, blockchain is listed under "Security, Privacy and Resilience"

category as authentication and authorization mechanism [48]. The deliverable presented 5G use case in smart energy vertical that integrated the NRG-5 networking VNFs (Virtualized Network Function) with blockchain-based identity management [49].

Key takeaways: Collaboration among International SDOs, regional, national, and industrial SDOs and alliances are established to work on blockchain standards, promotion, and pilot projects. The terminology and basic components of blockchain will reach consensus within the trading boundaries, if not at global scale but at least by industrial sector. The standardization of interoperation between blockchain and existing infrastructure may be driven by CSPs and ISPs. Regarding blockchain applications in industrial verticals, the degree of standardization will vary by industrial characteristics such as power of customer, power of provider, regulation, and legality. Examples above show that blockchain applications in IoT, finance, and banking would have higher level of matureness compared to other sectors.

11.4 From Blockchain Standards to Industrial Adoption

As introduced and discussed in Section 11.3, the participants in blockchain standards are mainly from industry and government regulation bodies. This sends a message that while blockchain technologies are moving fast into pilots and trials in various business scenarios, it was and will continue to be regulated due to its embedded innovation from community design to financial applications.

- Blockchain standards before regulations
  As much as the capital market and government are trying to regulate the space, the prerequisite of jurisdictional regulation is international standardization. Standards help to level out the playing field of the technology used, be transparent to the mechanisms used and coordinate the various types of regulations moving forward.
  Taking web technologies as an example, Web 2.0 is mainly centralized, rent-seeking, and requires plenty of human intermediaries as inputs. Web 3.0 seeks to elevate the foundational technology stack with decentralization, revenue-generating economics and machines as intermediaries.
  Thus, this is the reason that blockchain standards are so important. There are different blockchain models and architectural structures. The different underlying foundational technology stack highlights the different types of economics that will exist on the application layers of these technology stacks.
  Standardization also aids in regulation, to better understand this pandora box, and to constrain or mitigate its risks while allowing for innovation to flourish.

- How blockchain standards (as Layer 1 technology protocols) can help with enterprise applications?
  Blockchain is a technology stack where innovative mechanisms and digital solutions can be built upon. We saw more and more traditional industries (e.g. agriculture, manufacturing) looking into digital transformation and finally accepting that digitization is the basis of competitiveness. For example, International SDOs and different levels of national and industrial organizations identify blockchain as one of the enabling technologies for smart manufacturing transformation.
  Blockchain standards help enterprises be comfortable with investing in this technology. One aspect in information economics is that enterprises invest heavily in a legacy technology, which makes it difficult to switch to another technology. It is both costly in monetary terms and the time and energy required in training. The important decision-making variable is the risks involved in this investment in not fully understanding the technology. Thus, blockchain standards help to reduce these risks by defining and interpretating the technology and its showcases in a broader scale, and signaling to decision makers that these standards and protocols are adopted in similar business scenarios. That means the investment will be put on an innovative yet known technology that will bring more benefits than exposing risks, which justifies the costly investment.
  Standardization also plays an indispensable role in interoperability. Data and digital information have to be able to speak to each other in various technology stacks. Having international standardization helps to limit the interoperability issues, which can be another cost to "translate" from one technology stack to another.
- What is next for blockchain standards (Layer 2 protocols and application protocols, such as CBDC)?
  As the final chapter comes to a conclusion, it's important to look ahead and be prepared for what is next. As fascinating and promising blockchain can be, crossing the chasm to mainstream adoption is yet to come. The key of crossing lies in the matureness of technology, promotion and marketing, and in blockchain's case, the regulation. As mentioned previously, decentralization changes the economics from rent-seeking to revenue-generating ecosystems. Instead of the usual independent demand and supply, this is the design of such new economics, where tokens can act as incentives to affect participants in the system, be it machines (mining or validating) or people (transacting on-chain). Blockchain's layer 1 focuses its economics around computer and communication science, which is primarily driven by machines and smart contracts. In the application layer, where there are more actions by human behaviors, the economics around such ecosystems would be infinitely more complex.

Some examples may include: central bank digital currencies as a country-level application, decentralized finance or decentralized ownership of assets as a citizen-level application.

In conclusion, blockchain standards are a great first step toward the future being built. It is expected that top-down regulations in layer 1 technology stack, and standardizations from bottom-up around the application and ecosystems will be built on the layer 1 tech stack. Blockchain standards define the functions, performance indicators, control, management, and governance mechanisms in a vertical stack, as well as horizontal integration and interoperation with other devices and systems.

List of Acronyms

3GPP	Third Generation Partnership Project
CSP	communications service provider
DCEP	Digital Currency Electronic Payment
DIN	Deutsches Institut für Normung
DLT	distributed ledger technologies
EC	European Commission
EEA	Enterprise Ethereum Alliance
FG	focus group
FIG	International Federation of Surveyors
ICT	Information and Communication Technology
IEEE	Institute of Electrical and Electronics Engineers
IETF	Internet Engineering Task Force
INATBA	International Association for Trusted Blockchain Applications
IoT	Internet of Things
ITU-T	International Telecommunication Unit-Standardization sector
ISO	International Organization for Standardization
ISP	Internet service provider
OECD	Organization for Economic Co-operation and Development
SBS	Small Business Standards
SC	Standards Committee
SDG	sustainable developing goals
SDO	Standard Development Organization
SG	Study Group
SPEC	specification
SWIFT	Society for Worldwide Interbank Financial Telecommunication
TC	Technical Committee
UNECE	United Nations Economic Commission for Europe

W3C	World Wide Web Consortium
WG	Working Group

References

1 Hawkins, R., Blind, K., and Page, R. (2017). *Handbook of Innovation and Standards, Edward Elgar books*. , Edward Elgar Publishing, Incorporated. https://books.google.com.sg/books?id=AXkvDwAAQBAJ.

2 Safety standards for vehicles. https://tc.canada.ca/en/road-transportation/safety-standards-vehicles-tires-child-car-seats/safety-standards-vehicles (accessed 17 May 2023).

3 European Commission. Technical harmonisation. https://ec.europa.eu/growth/sectors/automotive/technical-harmonisation_en (accessed 17 May 2023).

4 Pongratz, S. (2020). Harmonized ITU IMT-2020 Standards of 3GPP 5G Technologies Lay the Foundation for a Successful Global Ecosystem. *Tech. Rep.* https://www.delloro.com/knowledge-center/white-papers/harmonized-itu-imt-2020-standards-of-3gpp-5g-technologies-lay-the-foundation-for-a-successful-global-ecosystem/ (accessed 17 May 2023).

5 The Economist (2015). The promise of the blockchain, the trust machine. https://www.economist.com/leaders/2015/10/31/the-trust-machine (accessed 17 May 2023).

6 BSI ISO (2017). Briefing ISO/TC307 blockchain and distributed ledger technologies. https://isitc-europe.com/wp-content/uploads/2017/01/TC307-DLT1-Briefing-v2.pdf (accessed 17 May 2023).

7 Roadmap for Blockchain Standards. https://www.standards.org.au/getmedia/ad5d74db-8da9-4685-b171-90142ee0a2e1/Roadmap_for_Blockchain_Standards_report.pdf.aspx (accessed 17 May 2023).

8 Yaga, D., Mell, P., Roby, N., and Scarfone, K. (2018). Blockchain technology overview. https://nvlpubs.nist.gov/nistpubs/ir/2018/NIST.IR.8202.pdf (accessed 17 May 2023).

9 Drive Innovation B. U. F.3 (2021). European blockchain strategy – brochure. https://ec.europa.eu/digital-single-market/en/news/european-blockchain-strategy-brochure (accessed 17 May 2023).

10 World Economic Forum (2020). Global Standards Mapping Initiative: An overview of blockchain technical standards. https://www.weforum.org/whitepapers/global-standards-mapping-initiative-an-overview-of-blockchain-technical-standards (accessed 30 May 2023).

11 Hyland-Wood, D. and Khatchadourian, S. (2018). A future history of international blockchain standards. *The Journal of The British Blockchain Association*. https://doi.org/10.31585/jbba-1-1-(11)2018.

12 IEEE (2019). *IEEE quick facts*. https://www.ieee.org/about/at-a-glance.html (accessed 17 May 2023).

13 IEEE (2019). *Summary of the IEEE organization*. https://www.ieee.org/content/dam/ieee-org/ieee/web/org/about/corporate/organization_summary.pdf (accessed 17 May 2023).

14 IEEE Blockchain Initiative (2020). https://blockchain.ieee.org/standards (accessed 17 May 2023).

15 IEEE-SA. P2418.1 – *Standard for the framework of blockchain use in internet of things (IoT)*. https://standards.ieee.org/project/2418_1.html (accessed 17 May 2023).

16 IEEE-SA IEEE Std. 2418.2-2020 (2020). *IEEE approved draft standard data format for blockchain systems*.

17 IEEE-SA IEEE Std. 2143.1-2020 (2020). *IEEE standard for general requirements for cryptocurrency exchanges*.

18 IEEE-SA IEEE Std. 2140.1-2020 (2020). *IEEE standard for a custodian framework of cryptocurrency*.

19 IEEE-SA IEEE Std. 2140.5-2020 (2020). *IEEE standard for a custodian framework of cryptocurrency*.

20 IEEE-SA (2020). *Digital finance and economy standard committee (CTS/DFESC)*. https://sagroups.ieee.org/cts-dfesc (accessed 17 May 2023).

21 IEEE-SA (2020). *Blockchain and distributed ledger standards committee*. https://sagroups.ieee.org/bdlsc (accessed 17 May 2023).

22 IEEE-SA. *Standards under development*. https://blockchain.ieee.org/standards (accessed 17 May 2023).

23 ISO. https://www.iso.org/benefits-of-standards.html (accessed 17 May 2023).

24 ETSI (2020). *About ETSI*. https://www.etsi.org/about (accessed 17 May 2023).

25 Antipolis, S. (2021). ETSI blockchain group releases first reports, targeting industry and governmental bodies. https://www.etsi.org/newsroom/press-releases/1884-2021-02-etsi-blockchain-group-releases-first-reports-targeting-industry-and-governmental-bodies (accessed 17 May 2023).

26 ETSI GS PDL 005 V1.1.1 (2020-03) (2020). Permissioned Distributed Ledger (PDL); Landscape of Standards and Technologies. *Tech. Rep*. https://www.etsi.org/deliver/etsi_gs/PDL/001_099/005/01.01.01_60/gs_PDL005v010101p.pdf (accessed 17 May 2023).

27 ETSI GR PDL 001 V1.1.1 (2020-03) (2020). Permissioned Distributed Ledger (PDL); Proof of Concepts Framework. *Tech. Rep*. https://www.etsi.org/deliver/etsi_gr/PDL/001_099/001/01.01.01_60/gr_PDL001v010101p.pdf (accessed 17 May 2023).

28 ETSI GR PDL 002 V1.1.1 (2020-11) (2020). Permissioned Distributed Ledger (PDL); Applicability and Compliance to Data Processing Requirements. *Tech. Rep*. https://www.etsi.org/deliver/etsi_gr/PDL/001_099/002/01.01.01_60/gr_PDL002v010101p.pdf (accessed 17 May 2023).

29 ETSI GR PDL 003 V1.1.1 (2020-12) (2020). Permissioned Distributed Ledgers (PDL); Application Scenarios. *Tech. Rep*. https://www.etsi.org/deliver/etsi_gr/PDL/001_099/003/01.01.01_60/gr_PDL003v010101p.pdf (accessed 17 May 2023).

30 ETSI GR PDL 004 V1.1.1 (2021-02) (2021). Permissioned Distributed Ledgers (PDL); Smart Contracts; System Architecture and Functional Specification. *Tech. Rep*. https://www.etsi.org/deliver/etsi_gr/PDL/001_099/004/01.01.01_60/gr_PDL004v010101p.pdf (accessed 17 May 2023).

31 PDL. Our role and activities. https://www.etsi.org/technologies/permissioned-distributed-ledgers (accessed 17 May 2023).

32 DIN SPEC 4997:2020-04 (2020). *Privacy by blockchain design*. https://dx.doi.org/10.31030/3150127.

33 Spanish Standardization Association UNE 71307-1:2020 (2020). *Digital enabling technologies. Decentralised identity management model based on blockchain and other distributed ledgers technologies. Part 1: Reference framework*.

34 IDB (2018). Global alliance to promote the use of blockchain in Latin America and the Caribbean. https://www.iadb.org/en/news/global-alliance-promote-use-blockchain-latin-america-and-caribbean (accessed 17 May 2023).

35 Lopez, M.A. (2020). Self-Sovereign Identity; The Future of Identity: Self-Sovereignity, Digital Wallets, and Blockchain. *Tech. Rep*. https://publications.iadb.org/publications/english/document/Self-Sovereign-Identity-The-Future-of-Identity-Self-Sovereignity-Digital-Wallets-and-Blockchain.pdf (accessed 17 May 2023).

36 TC 26. Cryptography and security mechanisms. https://tc26.ru/en/ (accessed 17 May 2023).

37 TC 159. Hardware and software for blockchain and distributed ledger technologies (TC159). http://bccmt.ru/ (accessed 17 May 2023).

38 TC 26 (2018). Mr 26.4.001-2018, terms and definitions in the field of blockchain and distributed ledger technologies. https://tc26.ru/standarts/metodicheskie-rekomendatsii/mr-26-4-001-2018-terminy-i-opredeleniya-v-oblasti-tekhnologiy-tsepnoy-zapisi-dannykh-blokcheyn-i-raspredelennykh-reestrov.html (accessed 17 May 2023).

39 ITU-T Focus Group on Application of Distributed Ledger Technology (FG DLT) (2019). Technical specification FG DLT D3.1 distributed ledger

technology reference architecture. https://www.itu.int/en/ITU-T/focusgroups/dlt/Documents/d31.pdf (accessed 17 May 2023).

40 ISO/DTR 23249 (2020). Blockchain and distributed ledger technologies overview of existing DLT systems for identity management. https://www.iso.org/standard/80805.html (accessed 17 May 2023).

41 Federal Law No of July 31 (2020). On digital financial assets, digital currency and on amendments to certain legislative acts of the Russian federation. https://rg.ru/2020/08/06/tsifra-dok.html

42 The Bank of Russia (2020). A digital ruble. http://www.cbr.ru/eng/analytics/d_ok/dig_ruble (accessed 17 May 2023).

43 China Daily and Jia, C. (2020). Digital currency to be based on blockchain. https://global.chinadaily.com.cn/a/202004/20/WS5e9cfc29a3105d50a3d17569.html (accessed 17 May 2023).

44 CBNEditor (2020). PBOC releases safety standards for use of distributed ledger technology by finance sector. https://www.chinabankingnews.com/2020/02/28/pboc-releases-safety-standards-for-use-of-distributed-ledger-technology-by-finance-sector/ (accessed 17 May 2023).

45 Zhang, Z. (2020). China's digital Yuan: development status and possible impact for businesses. https://www.china-briefing.com/news/chinas-digital-yuan-status-roll-out-impact-businesses/ (accessed 17 May 2023).

46 IETF. https://datatracker.ietf.org/rg/dinrg/about/ (accessed 17 May 2023).

47 Ridgewell, P. (2019). Blockchain: Where's the Value for Telecoms? *Tech. Rep*. https://www.accenture.com/_acnmedia/PDF-101/Accenture-Blockchain-Wheres-the-Value-for-Telecoms.pdf (accessed 17 May 2023).

48 5G-MoNArch. Security, privacy and resilience. https://5g-ppp.eu/key-achievements-v3/#Business (accessed 17 May 2023).

49 Voulkidis, A., De Mori, L., Duke, A. et al. (2019). Enabling smart energy as a service via 5G mobile network advances. http://www.nrg5.eu/wp-content/uploads/2017/07/NRG5_D2.3_v1_compressed.pdf (accessed 17 May 2023).

Index

Blockchains: Empowering Technologies and Industrial Applications, First Edition.
Edited by Anwer Al-Dulaimi, Octavia A. Dobre, and Chih-Lin I.

b

d

m

n

q

r

s

IEEE PRESS SERIES ON DIGITAL AND MOBILE COMMUNICATION

John B. Anderson, *Series Editor*
University of Lund

1. *Future Talk: The Changing Wireless Game*
   Ron Schneiderman
2. *Digital Transmission Engineering*
   John B. Anderson
3. *Fundamentals of Convolutional Coding*
   Rolf Johannesson and Kamil Sh. Zigangirov
4. *Mobile and Personal Communication Services and Systems*
   Raj Pandya
5. *Wireless Video Communications: Second to Third Generation and Beyond*
   Lajos Hanzo, Peter J. Cherriman, and Jürgen Streit
6. *Wireless Communications in the 21st Century*
   Mansoor Shafi, Shigeaki Ogose, and Takeshi Hattori
7. *Introduction to WLLs: Application and Deployment for Fixed and Broadband Services*
   Raj Pandya
8. *Trellis and Turbo Coding*
   Christian B. Schlegel and Lance C. Perez
9. *Theory of Code Division Multiple Access Communication*
   Kamil Sh. Zigangirov
10. *Digital Transmission Engineering,* Second Edition
    John B. Anderson
11. *Wireless LAN Radios: System Definition to Transistor Design*
    Arya Behzad
12. *Wireless Broadband: Conflict and Convergence*
    Vern Fotheringham and Chetan Sharma
13. *Millimeter Wave Communication Systems*
    Kao-Cheng Huang and Zhaocheng Wang
14. *Channel Equalization for Wireless Communications: From Concepts to Detailed Mathematics*
    Gregory E. Bottomley
15. *Handbook of Position Location: Theory, Practice and Advances*
    Seyed A. Reza Zekavat and R. Michael Buehrer
16. *Digital Filters: Principles and Applications with MATLAB*
    Fred J. Taylor

17. *Resource Allocation in Uplink OFDMA Wireless Systems: Optimal Solutions and Practical Implementations*
    Elias Yaacoub and Zaher Dawy
18. *Non-Gaussian Statistical Communication Theory*
    David Middleton
19. *Frequency Stability: Introduction & Applications*
    Venceslav F. Kroupa
20. *Mobile Ad Hoc Networking: Cutting Edge Directions*, Second Edition
    Stefano Basagni, Marco Conti, Silvia Giordano, and Ivan Stojmenovic
21. *Surviving Mobile Data Explosion*
    Dinesh C. Verma and Paridhi Verma
22. *Cellular Communications: A Comprehensive and Practical Guide*
    Nishith Tripathi and Jeffrey H. Reed
23. *Fundamentals of Convolutional Coding*, Second Edition
    Rolf Johannesson and Kamil Sh. Zigangirov
24. *Trellis and Turbo Coding*, Second Edition
    Christian B. Schlegel and Lance C. Perez
25. *Problem-Based Learning in Communication Systems Using MATLAB and Simulink*
    Kwonhue Choi and Huaping Liu
26. *Bandwidth Efficient Coding*
    John B. Anderson
27. *Visible Light Communications: Modulation and Signal Processing*
    Zhaocheng Wang, Qi Wang, Wei Huang, and Zhengyuan Xu
28. *Switch/Router Architectures: Shared-Bus and Shared-Memory Based Systems*
    James Aweya
29. *Handbook of Position Location: Theory, Practice, and Advances*, Second Edition
    Edited by S. A. (Reza) Zekavat and R. Michael Buehrer
30. *Information and Communication Theory*
    Stefan Höst
31. *Blockchains: Empowering Technologies and Industrial Applications*
    Edited by Anwer Al-Dulaimi, Octavia A. Dobre, and Chih-Lin I.

WILEY END USER LICENSE AGREEMENT

Go to www.wiley.com/go/eula to access Wiley’s ebook EULA.