



Data Fabric and Data Mesh Approaches with AI

A Guide to AI-based Data Cataloging, Governance, Integration, Orchestration, and Consumption

Eberhard Hechler
Maryela Weihrauch
Yan (Catherine) Wu

Apress®

Data Fabric and Data Mesh Approaches with AI

**A Guide to AI-based Data
Cataloging, Governance,
Integration, Orchestration,
and Consumption**

**Eberhard Hechler
Maryela Weihrauch
Yan (Catherine) Wu**

Apress®

Data Fabric and Data Mesh Approaches with AI: A Guide to AI-based Data Cataloging, Governance, Integration, Orchestration, and Consumption

Eberhard Hechler
Sindelfingen, Germany

Maryela Weihrauch
Charlotte, NC, USA

Yan (Catherine) Wu
San Jose, CA, USA

ISBN-13 (pbk): 978-1-4842-9252-5
<https://doi.org/10.1007/978-1-4842-9253-2>

ISBN-13 (electronic): 978-1-4842-9253-2

Copyright © 2023 by Eberhard Hechler, Maryela Weihrauch,
Yan (Catherine) Wu

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

Trademarked names, logos, and images may appear in this book. Rather than use a trademark symbol with every occurrence of a trademarked name, logo, or image we use the names, logos, and images only in an editorial fashion and to the benefit of the trademark owner, with no intention of infringement of the trademark.

The use in this publication of trade names, trademarks, service marks, and similar terms, even if they are not identified as such, is not to be taken as an expression of opinion as to whether or not they are subject to proprietary rights.

While the advice and information in this book are believed to be true and accurate at the date of publication, neither the authors nor the editors nor the publisher can accept any legal responsibility for any errors or omissions that may be made. The publisher makes no warranty, express or implied, with respect to the material contained herein.

Managing Director, Apress Media LLC: Welmoed Spahr
Acquisitions Editor: Jonathan Gennick
Development Editor: Laura Berendson
Editorial Assistant: Gryffin Winkler

Cover designed by eStudioCalamar

Cover image by starline on FreePik (www.freepik.com)

Distributed to the book trade worldwide by Apress Media, LLC, 1 New York Plaza, New York, NY 10004, U.S.A. Phone 1-800-SPRINGER, fax (201) 348-4505, e-mail orders-ny@springer-sbm.com, or visit www.springeronline.com. Apress Media, LLC is a California LLC and the sole member (owner) is Springer Science + Business Media Finance Inc (SSBM Finance Inc). SSBM Finance Inc is a **Delaware** corporation.

For information on translations, please e-mail booktranslations@springernature.com; for reprint, paperback, or audio rights, please e-mail bookpermissions@springernature.com.

Apress titles may be purchased in bulk for academic, corporate, or promotional use. eBook versions and licenses are also available for most titles. For more information, reference our Print and eBook Bulk Sales web page at <http://www.apress.com/bulk-sales>.

Any source code or other supplementary material referenced by the author in this book is available to readers on GitHub (<https://github.com/Apress>). For more detailed information, please visit <http://www.apress.com/source-code>.

Printed on acid-free paper

*To my wife, Irina, and our two sons, Lars and Alex,
for their continuing support and understanding in
writing this book on long evenings and weekends
instead of spending time with them.*

Eberhard Hechler

*To my husband, Frank, for his continuing
motivation, support, and understanding to
prioritize this book above other projects.*

Maryela Weihrauch

*To my husband, Ming, for his encouragement
and tremendous efforts in caring for our three
children while I devoted my time to the book.*

Yan (Catherine) Wu

Table of Contents

About the Authors	xiii
About the Technical Reviewer	xvii
Acknowledgments	xix
Introduction	xxi
Foreword	xxxii
Part I: Data Fabric and Data Mesh Foundation	1
Chapter 1: Evolution of Data Architecture	3
Introduction.....	3
Data Architectures: Values and Challenges	5
Enterprise Data Warehouse (EDW)	6
Big Data, Data Lake, and Data Lakehouse	9
Key Takeaways.....	13
References	15
Chapter 2: Terminology: Data Fabric and Data Mesh	17
Introduction.....	17
Data Fabric Concept.....	18
Data Fabric Framework	20
AI-Infused Data Fabric.....	23
Data Mesh Concept.....	26
Relationship: Data Fabric and Data Mesh	29

TABLE OF CONTENTS

Data Product 31

Key Takeaways..... 39

References..... 40

Chapter 3: Data Fabric and Data Mesh Use Case Scenarios..... 43

Introduction..... 43

Automated and Consistent Governance 50

 Include IBM zSystems Data in AI Governance 52

Unified View of Data Across a Hybrid Cloud..... 55

Provide a Comprehensive View of Customers, Vendors, and Other Parties 60

Unlock the Trustworthy AI Concept 64

Key Takeaways..... 67

References..... 69

Chapter 4: Data Fabric and Data Mesh Business Benefits..... 71

Introduction..... 71

Business Requirements and Pain Points for Data Management and Consumption 72

Benefits of a Data Fabric and Data Mesh for Technical Teams Managing Data..... 76

Benefits of a Data Fabric and Data Mesh for Business Teams Consuming Data..... 79

Key Takeaways..... 83

References 84

Part II: Key Data Fabric and Data Mesh Capabilities and Concepts 87

Chapter 5: Key Data Fabric and Data Mesh Capabilities 89

Introduction..... 90

Knowledge Catalog 90

Active Metadata.....	91
Data Curation.....	95
Semantic Knowledge Graphs	96
Self-Service Capabilities	101
Trustworthy AI	103
Introduction	104
Model Fairness	105
Drift Detection	108
Model Explainability	111
Model Quality Metrics.....	113
Intelligent Information Integration	115
Key Takeaways.....	118
References.....	119
Chapter 6: Relevant ML and DL Concepts.....	123
Introduction to AI, ML, and DL	124
ML and DL Industry Use Cases	128
Data Exploration and Preparation	131
Model Selection, Training, and Evaluation	133
Model Deployment.....	136
Natural Language Processing (NLP)	139
Key Takeaways.....	142
References.....	144
Chapter 7: AI and ML for a Data Fabric and Data Mesh.....	147
Introduction.....	147
General Overview	148
Cataloging	154

TABLE OF CONTENTS

AI-Infused Understanding of Assets 158

- Asset Discovery 159
- Asset Profiling 160
- Automatic Asset Quality Assessment 161
- Asset Access 162

AI/ML for Entity Matching 164

AI/ML to Activate the Digital Exhaust 168

AI/ML for Semantic Enrichment 171

Key Takeaways..... 173

References 174

Chapter 8: AI for Entity Resolution 177

- Introduction..... 178
- Introducing Entity Matching 179
- Traditional Entity Resolution Approaches..... 181
- Use of AI to Resolve Entity Challenges..... 185
- The Benefits and Cost of an AI-Based Solution..... 188
- Considerations for MDM Solutions..... 189
- Key Takeaways..... 192
- References 193

Chapter 9: Data Fabric and Data Mesh for the AI Lifecycle 195

- Introduction to the AI Lifecycle..... 196
- Key Aspects: DataOps, ModelOps, MLOps..... 200
- Case Study 1: Consolidating Fragmented Data in a Hybrid Cloud Environment..... 205
- Case Study 2: Operationalizing AI 211
- Accelerate MLOps with AutoAI 215

Deployment Patterns for AI Engineering	219
Key Takeaways.....	224
References.....	226
Part III: Deploying Data Fabric and Data Mesh in Context.....	229
Chapter 10: Data Fabric Architecture Patterns.....	231
Introduction.....	232
Data Fabric and Data Mesh Evolution.....	232
Data Consumption Patterns	236
Data Fabric for a Data Mesh Solution	243
Data Mesh Self-Service Capabilities	243
Data Mesh Architecture Overview Diagram.....	248
Intelligent Information Integration Styles.....	249
Key Takeaways.....	252
References.....	254
Chapter 11: Data Fabric Within an Enterprise Architecture	257
Introduction.....	258
What Is Enterprise Architecture?	259
What Is Application Architecture?	262
Data Fabric as a Data Architecture	266
Sample of a Data Fabric Within an Enterprise Architecture.....	271
Key Takeaways.....	273
References.....	274
Chapter 12: Data Fabric and Data Mesh in a Hybrid Cloud Landscape	277
Introduction.....	278
What Is Hybrid Cloud?.....	278

TABLE OF CONTENTS

- Key Challenges for Data Architecture 281
- Data Fabric and Data Mesh in Hybrid Cloud 281
 - Data Fabric Architecture in Hybrid Cloud..... 282
 - Data Mesh Solution in Hybrid Cloud 284
- Benefits of Data Fabric and Data Mesh for Hybrid Cloud..... 288
- Key Takeaways..... 289
- References 290

- Chapter 13: Intelligent Cataloging and Metadata Management293**
 - Introduction to Metadata Management..... 293
 - Key Aspects of Intelligent Cataloging..... 297
 - Build an Intelligent Catalog by Automating Data Discovery and Enrichment.... 299
 - Find Data Assets with Semantic Search and Recommendation 302
 - Provide Data Insight and Provenance as Data Flows Across the Enterprise.... 306
 - Key Takeaways..... 308
 - References 309

- Chapter 14: Automated Data Fabric and Data Mesh Aspects311**
 - Introduction..... 312
 - Intelligent Automation of Metadata..... 313
 - Automated Analysis and Profiling of Data 316
 - Automated Tagging, Annotation, and Labeling..... 321
 - Automated Data Quality Assessment..... 324
 - Key Takeaways..... 328
 - References 330

- Chapter 15: Data Governance in the Context of Data Fabric and Data Mesh.....333**
 - Introduction..... 334
 - Importance of Data and AI Governance..... 335

Key Aspects of Data and AI Governance 337

Establishing a Data Governance Foundation with a Data Fabric
Architecture 341

Establishing Automated Regulation with a Data Fabric Architecture..... 342

Automatic Enforcement of Data Regulations in Data Fabric 344

Automate Quality Analysis with Data Fabric 346

Key Takeaways..... 350

References 351

Part IV: Current Offerings and Future Aspects..... 353

Chapter 16: Sample Vendor Offerings355

Introduction..... 356

IBM Cloud Pak for Data 358

Amazon Web Services..... 361

Microsoft Azure 363

Denodo..... 366

Informatica..... 368

Key Takeaways..... 370

References 372

Chapter 17: Data Fabric and Data Mesh Research Areas375

Introduction..... 376

AI-Based Augmented Insight..... 376

AI-Infused Automated AI Governance..... 380

Hyper-automated Data and AI Fabric 386

Key Takeaways..... 389

References 390

TABLE OF CONTENTS

Chapter 18: In Summary and Onward	393
Data Fabric and Data Mesh Summarized.....	394
Where to Go from Here.....	397
Key Takeaways.....	398
Abbreviations	403
Index.....	409

About the Authors



Eberhard Hechler is an executive architect at the IBM Germany R&D Lab. He is a member of the Data and AI development organization and addresses the broader analytics scope, including machine learning (ML). After more than two years at the IBM Kingston Lab in New York, he worked in software development, performance optimization, IT/solution architecture and design, Hadoop and Spark integration, and mobile device management (MDM).

Eberhard worked with Db2 on the MVS platform, focusing on testing and performance measurements. He has worked worldwide with IBM clients from various industries on a vast number of topics such as data and AI, information architectures, and industry solutions. From 2011 to 2014, he was at IBM Singapore, working as the lead big data architect in the Communications Sector of IBM's Software Group throughout the Asia-Pacific region.

Eberhard has studied in Germany and France and holds a master's degree (Dipl.-Math.) in Pure Mathematics and a bachelor's degree (Dipl.-Ing. (FH)) in Electrical Engineering. He is a member of the IBM Academy of Technology and has coauthored the following books:

- *Enterprise Master Data Management*, Pearson, 2008, ISBN: 0132366258
- *The Art of Enterprise Information Architecture*, Pearson, 2010, ISBN: 0137035713

ABOUT THE AUTHORS

- *Beyond Big Data*, Pearson, 2014, ISBN: 013350980X
- *Deploying AI in the Enterprise*, Apress, 2020, ISBN: 1484262050



Maryela Weihrauch is an IBM Distinguished Engineer in the Data and AI development group for IBM Z Technical Sales and is a customer success leader. She has extensive experience with relational databases in terms of systems, application, and database design. She is engaged with enterprises across the world and helps them adopt new data and analytics technologies. Her former roles in Db2 for z/OS development have involved determining a Db2 for z/OS strategy for HTAP (Hybrid Transaction and Analytics

Processing), including the Db2 Analytics Accelerator strategy and implementation as well as Db2's application enablement strategy.

Maryela consults with enterprises around the globe on many data modernization initiatives and leads an effort to develop a methodology to determine the best data architecture for a given application based on data architecture decision criteria.

Maryela holds two master's degrees in Computer Science from Technical University Chemnitz, Germany, and California State University, Chico, California, USA. She holds more than 30 patents and is a member of the IBM Academy of Technology. She frequently shares her experiences at conferences around the world.



Yan (Catherine) Wu is the program director at the IBM Silicon Valley Lab. She is an engineering leader with deep expertise in data governance, artificial intelligence (AI), machine learning (ML), enterprise design thinking, and pragmatic product marketing. She has extensive experience working with large clients to discover use cases for data governance and AI, explore how the latest technologies can be applied to resolve real-world business challenges, and deploy these

technologies to accelerate enterprise digital transformation. She has a proven track record in translating customer needs into software solutions while working collaboratively with globally distributed development, design, and offering management teams.

Prior to her current position at IBM United States, Catherine was the lab director of the Data and AI development lab at IBM China. In these roles, Catherine demonstrated her ability to think horizontally and strategically to bring teams together to create innovative solutions for complex problems.

Catherine was an ambassador for the Women in Data Science (WiDS) organization (www.widsconference.org/). She is passionate about inspiring and educating data scientists worldwide, particularly women in this field. She organized WiDS regional events over the past three years.

Catherine holds a master's degree in Computer Science from the National University of Singapore and a bachelor's degree in Computer Technology from Tsinghua University.

About the Technical Reviewer



Akshay R. Kulkarni is an AI and machine learning evangelist and a thought leader. He has consulted several Fortune 500 and global enterprises to drive AI and data science-led strategic transformations. He is a Google Developer Expert, author, and regular speaker at major AI and data science conferences (including Strata, O'Reilly AI Conf., and GIDS). He is a visiting faculty member for some of the top graduate institutes in India. In 2019, he has been also featured as one of the top 40 under 40 Data Scientists in India. In his spare time, he enjoys reading, writing, coding, and building AI products.

Acknowledgments

We are eternally grateful to the many colleagues in several IBM development labs and client-facing organizations we have worked with around the globe, colleagues who have challenged and inspired us with their points of view and critical questions. Collaborating with universities provided us with an additional invaluable and product-agnostic view regarding Data Fabric and Data Mesh topics.

The best way to connect with someone is not by talking, but by listening. We could not have undertaken this endeavor without the numerous enterprises and organizations that we have had an opportunity to listen to and work with in recent years that have provided us with an insight into elaborating on some of the key Data Fabric and Data Mesh challenges and – most importantly – how to deploy these concepts in an existing IT and business landscape. We are grateful to the many clients that we have worked with; their opinions and views have without a doubt increased our insight and genuine interest in this topic.

We are grateful to *Akshay Kulkarni*, who has conducted a fantastic technical review, which has greatly improved the readability of this book.

Last but not least, thanks to everyone on the Apress Media team who helped us so much: Coordinating Editor *Jill Balzano* and Development Editor *Laura Berendson*. Special thanks to *Mark Powers*, ever-patient Editorial Operations Manager our amazing Editorial Assistant *Gryffin Winkler*, our production editor *Krishnan Sathyamurthy*, and the greatest project manager *Dulcy Nirmala*.

Introduction

Even with more data engineers, the demand for data and data products consistently outgrows the ability to deliver business value with existing data architecture capabilities. This book provides methods and guidance to successfully deploy Data Fabric architectures and Data Mesh solutions, which can lower the needed skill level to discover, access, prepare, and consume data through easy access to metadata in a business context and to intelligently automate the main tasks of a data engineer. It presents new AI-based Data Fabric and Data Mesh capabilities, such as self-service data discovery and data enrichment, automated data quality assessments, and data matching techniques, which helps breaking down the silos of responsibilities of a data source owner, data engineer, and data consumer as well as removing the dependency on the data engineer for data consumption, allowing business priorities to be implemented without the data engineer support capacity as a bottleneck.

The book describes the role of AI in the context of intelligent information integration methods, data knowledge graphs based on enriched metadata, semantic insight for intelligent data consumption, and intelligent cataloging with automated metadata management. It presents Data Fabric and Data Mesh as essentially a metadata- and AI-driven approach of connecting a disparate collection of data and AI assets; it includes data and AI in hybrid cloud environments, Data Mesh with its particular focus on enabling a data marketplace with data products, data lake implementations, and traditional DWH/analytical systems.

This book is for a reader who is looking for guidance and recommendations on how to successfully deploy a state-of-the-art Data Fabric architecture underpinning Data Mesh solutions and is

INTRODUCTION

furthermore eagerly interested in getting a comprehensive overview on how a Data Fabric architecture uses AI and ML for automated metadata management and self-service data discovery and consumption. The reader is furthermore interested to learn how Data Fabric and Data Mesh relate to other concepts, such as DataOps, ModelOps, and AIOps – to name a few. The anticipated reader is looking for examples on how to modernize the consumption of data and AI assets to enable a shopping-for-data (data-as-a-product) experience.

The chapters of this book are organized into four main parts.

Part 1, “Data Fabric and Data Mesh Foundation,” sets the scene for the book in terms of providing an introduction to these concepts, how some of the terms, for instance, Data Fabric architecture, Data Mesh solution, and data product, relate to each other, presenting some of the most relevant use case scenarios, and describing the key business benefits.

It consists of the following four chapters:

- **Chapter 1, “Evolution of Data Architecture,”** introduces the motivation for looking into data architectures. It shares an overview about data architecture evolution coming from traditional data warehouses to big data, data lakes, and data lakehouses and their main characteristics, value, and challenges. It outlines industry requirements in a data-driven world that led to the Data Fabric and Data Mesh concepts.
- **Chapter 2, “Terminology: Data Fabric and Data Mesh,”** explains the key terms that will be used throughout the book, particularly the terms *Data Fabric*, *Data Mesh*, *data marketplace*, and *data product*, and how these terms relate to each other. We introduce the term *data-as-a-product* in the context of a Data Mesh and highlight the relationship of Data Fabric and Data Mesh concepts to DataOps, ModelOps, and

AIOps, among others. It furthermore addresses the specific needs for knowledge catalog orchestration and metadata exchange in distributed organizationally structured Data Mesh deployments.

- **Chapter 3, “Data Fabric and Data Mesh Use Case Scenarios,”** walks through several use cases for implementing a Data Fabric and Data Mesh that also would be valid entry points. Data and AI governance and privacy initiatives are ongoing in almost every organization, enabling access to enterprise data and AI across platforms to the people who have a business need. Other use cases are driven by hybrid cloud data integration; the need for a comprehensive view on customers, vendors, and other parties for better business outcome; and development and integration of trustworthy AI into business processes.
- **Chapter 4, “Data Fabric and Data Mesh Business Benefits,”** dives into business needs and pain points that we hear in our conversations with enterprises. We will discuss business benefits of creating a Data Fabric and Data Mesh from the perspective of the technical team as well as the business teams consuming data and AI assets.

Part 2, “Key Data Fabric and Data Mesh Capabilities and Concepts,” presents key Data Fabric and Data Mesh capabilities and focuses on AI and ML methods applied to those Data Fabric capabilities. It introduces the reader to the most relevant AI and ML concepts that are required to implement a Data Fabric architecture and Data Mesh solution. It furthermore discusses the AI usage for entity matching purposes and how a Data Fabric implementation is leveraged for the entire AI lifecycle.

It consists of the following five chapters:

- **Chapter 5, “Key Data Fabric and Data Mesh Capabilities,”** introduces the key Data Fabric and Data Mesh capabilities, such as self-service, AI and ML, trustworthy AI, intelligent information integration, and active metadata – among other topics. It also discusses semantic knowledge graphs (semantic networks) as underpinning of a Data Fabric and Data Mesh. A section on AI and ML to enable the “digital exhaust” elaborates on pattern recognition and correlation discovery from the “digital exhaust” to augment and operationalize this insight especially into the Data Fabric. This chapter discusses the concepts to deliver trustworthy and explainable AI and provides some examples, for example, to discover drift and bias in AI models. Finally, intelligent (smart) information integration capabilities are described.
- **Chapter 6, “Relevant ML and DL Concepts,”** explains the key ML and DL concepts used for building Data Fabric and Data Mesh capabilities. It starts with an introduction to AI, ML, and DL, their connections and differences, and how these technologies are being used to accelerate enterprise digital transformation initiatives. It also introduces key techniques in the AI lifecycle. Starting from data, it explains the methods of understanding data and the techniques to transform data into a usable shape suitable for AI model training. It furthermore discusses how to choose, train, and evaluate AI models, as well as how to deploy AI models to infuse AI/ML into the Data Fabric and Data

Mesh. Lastly, it covers Natural Language Processing (NLP) and explains why it's important to make use of unstructured data (especially text) in the context of a Data Fabric and Data Mesh for an enterprise.

- **Chapter 7, “AI and ML for a Data Fabric and Data Mesh,”** provides a deep dive into the exploitation of AI and ML for various topics and tasks, such as data discovery, profiling, and data access, to enable a “digital exhaust,” ML-based entity matching, automated data quality assessments, and semantic enrichment. This is an essential chapter, which highlights novel ideas to augment Data Fabric and Data Mesh concepts with AI and ML. The exploitation of AI for some of these topics is further explored in subsequent chapters.
- **Chapter 8, “AI for Entity Resolution,”** explains what entity resolution, also known as entity matching, is, why the problem has arisen, and why it's important to the business. Next, the reader will learn more about what are the traditional approaches to solving this problem and how AI can reveal new possibilities for solving it, what will be the benefits and potential problems of using AI solutions, and how to choose a fit-for-purpose solution.
- **Chapter 9, “Data Fabric and Data Mesh for the AI Lifecycle,”** explains Data Fabric and Data Mesh capabilities during the entire AI lifecycle. First, it introduces the core ideas and concepts of AI engineering and how AI engineering relates to DataOps and ModelOps. To help readers better understand the essence of the Data Fabric and Data Mesh for the

INTRODUCTION

AI lifecycle, this chapter includes two case studies – the first one shows how Data Fabric can help in integrating data from various data sources in a hybrid cloud enterprise environment, and the second case study introduces operationalizing AI and key benefits Data Fabric and Data Mesh could bring to the production system such as security, explainability, governance, and scalability. It specifically highlights accelerating the implementation of MLOps with AutoAI. It further describes the best practices for operationalizing AI and common deployment patterns for AI engineering.

Part 3, “Deploying Data Fabric and Data Mesh in Context,”

introduces Data Fabric architecture patterns for different usage purposes, for instance, intelligent data integration styles and data consumption patterns. It discusses the meaning of automated Data Fabric and Data Mesh, intelligent cataloging, and augmented metadata management and describes the Data Fabric and Data Mesh concepts in the context of hybrid cloud landscapes, an enterprise data architecture, and data governance initiatives.

It consists of the following six chapters:

- **Chapter 10, “Data Fabric Architecture Patterns,”** provides a high-level overview of the Data Fabric architecture evolution and elaborates on key Data Fabric architecture patterns, such as a Data Fabric architecture serving as the underpinning for a Data Mesh solution, intelligent information integration styles, and data consumption patterns. This chapter describes the Data Fabric and Data Mesh evolution, discusses data consumption patterns, and provides a high-level Data Mesh architecture overview diagram.

- **Chapter 11, “Data Fabric Within an Enterprise Architecture,”** is a continuation of Chapter 10; it describes how a data architecture needs to be looked at in conjunction with the implemented enterprise and application architecture in an enterprise. Many organizations are in the process to modernize their application and data landscape. Applications have different requirements with respect to data characteristics that may recommend one data architecture implementation over another, for example, data access through virtualization or data replication and transformation.
- **Chapter 12, “Data Fabric and Data Mesh in a Hybrid Cloud Landscape,”** introduces hybrid cloud, integrating IT on-premises and public cloud, and describes how to deploy a Data Fabric architecture and Data Mesh solution in hybrid cloud environments. This creates new challenges for accessing and integrating data and AI across organizations and boundaries of public cloud providers. It addresses the challenges and benefits of a Data Fabric and Data Mesh in a hybrid cloud landscape.
- **Chapter 13, “Intelligent Cataloging and Metadata Management,”** introduces metadata management, followed by the key aspects of intelligent cataloging. It provides a deep dive into each key aspect and elaborates how Data Fabric and Data Mesh capabilities realize automated data discovery, classification of data assets, assignment of data assets with business terms, and creation of enterprise knowledge graphs by building connections between data and AI assets.

It also highlights why data lineage and provenance are needed and how to implement them with a Data Fabric.

- **Chapter 14, “Automated Data Fabric and Data Mesh Aspects,”** focuses on intelligent automation aspects. The desire to create an enterprise-wide description of data and AI is not a new concept. It was considered a failure about two decades ago. This chapter describes the usage of intelligent automation to collect metadata from different data sources and catalog them, automatically checking data quality and augmenting the data as well as automating data and AI governance services as a foundation of a Data Fabric and Data Mesh.
- **Chapter 15, “Data Governance in the Context of Data Fabric and Data Mesh,”** explains why data and AI governance and privacy are critical to a data-driven strategy for enterprises. It introduces the key aspects of data governance – people, process, data regulations, data rules, data protection methods, etc. It further explores how Data Fabric and Data Mesh capabilities establish a data and AI governance foundation for an enterprise, make data trustworthy by automated quality analysis, and protect data by automatic enforcement of data protection regulations.

Part 4, “Current Offerings and Future Aspects,” discusses a few sample vendor offerings and current Data Fabric and Data Mesh research areas. This part finishes with a short summary and key takeaways.

It consists of the following three chapters:

- **Chapter 16, “Sample Vendor Offerings,”** introduces how different vendors implement a Data Fabric architecture and Data Mesh solution with commercial software or SaaS (Software as a Service). It further delves into each vendor’s offering and explains how it works and what are the strengths.
- **Chapter 17, “Data Fabric and Data Mesh Research Areas,”** discusses Data Fabric and Data Mesh challenges as they are addressed by current research and academia initiatives, such as hyper-automation and knowledge-based consumption. It also describes the confluence of AI and Data Fabric and Data Mesh and outlines the road toward an AI-driven Data Fabric and Data Mesh.
- **Chapter 18, “In Summary and Onward,”** summarizes the main messages of this book and provides suggestions on how to get started on a Data Fabric and Data Mesh journey.

Foreword

This book in one sentence: “Data Mesh is being referred to as a solution, which is built on top of Data Fabric as an architecture, and the key message brought across in this book is that infused AI is the magic sauce that is going to help you being successful in implementing these concepts in your organization.”

The structure of this book is very well organized taking you on a cohesive journey from a to z and covering traditional data management topics while linking it to buzzwords like Data Fabric, Data Mesh, AI, ML, DL, automation, etc.

For instance, the important topic of identity matching will briefly take you through the traditional way of doing it and identifying shortcomings and then expand on how AI, ML, and in particular DL, can help solve the challenge much more efficiently and accurately. In essence, this is applied AI to automate data management tasks, and who don't like tedious task to become automated.

It's filled with nice charts and diagrams to support the written text, and at the end of each chapter you have key take aways that serves as a summary as well a quick refresher at a later state.

Another key aspect is illuminating the importance of not only managing data as an asset but equally important AI related artefacts. This becomes even more significant with the emerging data and AI convergence happening with real time inference in a hybrid processing scenario.

My favorite part of this book is the stressing of the ed how noteworthy it is to have a single view across the enterprise when it comes to metadata or what they refer to as the Knowledge Catalogue. This is not to say that there

FOREWORD

will be one single catalog reflecting all about everything, but a virtually connected catalog with support of Egeria Open Cohort for metadata exchange, which becomes the glue that can hold it all together.

As they rightfully point out this becomes even more essential in a hybrid cloud scenario with a mix of public/private cloud as well an on-premises platform like the good old Mainframe still being the backbone of many large enterprises. A key concern of mine has always been to ensure that we don't lose control of the data governance when Db2 for z/OS are being unlocked for cloud-initiated modernization initiatives.

Many organizations have already invested an immense amount of time and effort in creating metadata to meet the requirements of GDPR, but from a business user perspective this is often too detailed to support them on their Data Mesh journey to create data products. Hence, building on what's already there, but enriching it to bring it higher level of abstraction seems like an appealing idea to pursue.

Adapting the guidelines and recommendations in this book is what would make me sleep well at night during such transition phase going from predominantly on-premises based to a hybrid cloud setup.

—Bjarne Nelson, IBM Champion

PART I

Data Fabric and Data Mesh Foundation

CHAPTER 1

Evolution of Data Architecture

This chapter introduces the motivation for looking into data architectures. It shares an overview about data architecture evolution transitioning from traditional data warehouses to big data and data lakes and their main characteristics, values, and challenges. It outlines industry requirements in a data-driven world that ultimately led to the concept of a Data Fabric.

Introduction

When you look back in time, data architectures were developed in response to pain points with existing IT solutions and new business needs, typically in periods of newly emerging technologies. Understanding this evolution helps position new data architecture trends such as Data Fabric and Data Mesh in context of the existing data landscape in an organization.¹ To demonstrate the value of a new data architecture, it is critical to identify use cases that would benefit from the new data architecture trend and define projects well scoped to deliver value to the business in reasonable time.

¹See Reference [1] for more information on the evolution of data architectures.

The need to structure the data of an organization in a way that the data can be analyzed to answer business-driven questions has existed since the invention of databases, especially Relational Database Management Systems (RDBMS) with a standardized application programming interface (API), Structured Query Language (SQL), in the 1970s.² In the mid-1990s, the data architecture of a data warehouse was primarily developed for reporting and data analysis of structured historical data, potentially combining different data stores.

In the 2010s, with wider adoption of artificial intelligence, Internet of Things (IoT), and edge computing, the demand for processing massive amounts of structured, semi-structured, and unstructured data emerged. It led to the data architecture of a data lake with a very compelling promise, storing data from disparate sources in a single place in its raw format, usually as files for any data consumption needed. As a further evolution, data lakehouse³ architectures emerged, a merge of data lake and data warehouse concepts, which enabled machine learning (ML) and business analytics on data lakes.

Recently, readily available processing power and AI exploitation for managing the metadata of a data architecture itself led to Data Fabric and Data Mesh architecture and solution concepts that facilitate end-to-end integration of various data pipelines and cloud environments using intelligent and automated systems.

This high-level architecture evolution is depicted in Figure 1-1.

² See Reference [2] for more information on RDBMS and SQL.

³ See Reference [3] for more information on the data lakehouse architecture.

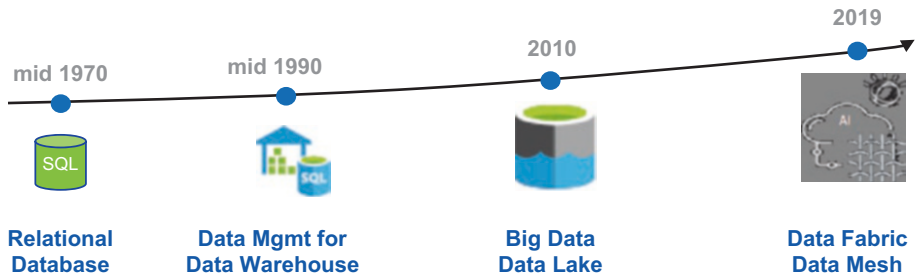


Figure 1-1. *Data Architecture Evolution Leading to Data Fabric and Data Mesh*

Many data architecture capabilities such as data integration, data replication, Extract-Transform-Load (ETL) pipelines, data governance, etc. have existed for a long time. In the past, a specific set of those capabilities were dictated for all use cases, even if they did not satisfy the use case requirements. A Data Fabric architecture brings the promise to use the right capabilities for the right use cases guided by intelligent metadata.

Data Architectures: Values and Challenges

RDBMS introduced several functions that became the foundation of later data architectures. The first function is the separation of a logical data model – the relational model – from the physical storage of the data. The separation created a need to introduce a metadata catalog describing the logical layout of the data in tables and its mapping to physical storage, maintaining statistics such as data size, data distribution, and other characteristics.

RDBMS provide functionality to maintain the physical storage of the data by reorganizing the data into the defined order when too many data changes made the physical layout and therefore data access suboptimal. It also ensures the availability of the data by providing backup and recovery capabilities exploiting available system capabilities.

The second function is a standardized Structured Query Language (SQL) to access the data. Today, support of SQL exists in all generations of programming languages such as Assembler, COBOL, Java, and Python. SQL is continuously expanded to include data types such as XML and JSON and new functionality, most recently for interference of ML and deep learning (DL) models. Both key functionalities of a RDBMS became the foundation for establishment of enterprise data warehouse (EDW) architectures, which we explore in the following section.

Enterprise Data Warehouse (EDW)

Interestingly, RDBMS were initially used to store data for reporting purposes using SQL and SQL-generating tools as a fast way to get results. In the 1970s, performance of data access in a RDBMS was not sufficient for transactional applications compared with storing data in datasets or hierarchical databases. As available compute power increased, the data of most transactional systems were migrated to relational databases. It increased productivity of application programmers significantly.

The following two main business needs led to the development of an EDW architecture:

- **Separation of databases for transactional and reporting applications:** Transactional systems such as core banking or airline ticketing applications are business critical and have different availability and response time characteristics compared with reporting applications. The business requested the separation of databases used for transactional and reporting applications to not impact performance or availability of the transactional applications.

- **Integrate data from multiple sources and restructure for better performance and data consumability:** Data from transactional applications was not stored optimal for reporting and analytical applications. The need became apparent to copy the data from the source(s) and transform it into a format optimal for reporting and analytics.

The preceding needs led to an EDW⁴ architecture, where the data is uploaded from the transactional application(s) to a staging area, followed by sometimes complex data cleansing and data transformation steps before the derived data is stored in the EDW. Typically, data is stored several times in intermittent staging areas.

This EDW architecture is depicted in Figure 1-2.

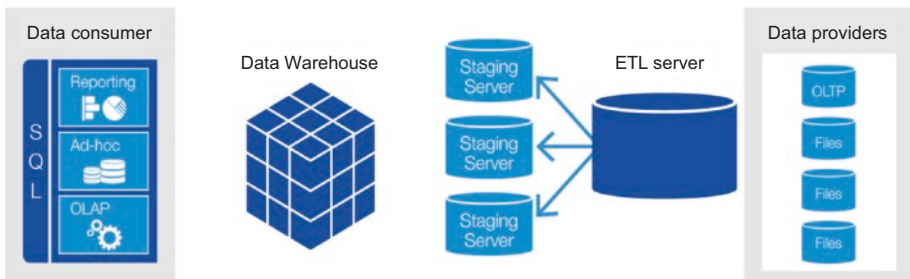


Figure 1-2. Enterprise Data Warehouse Architecture

The EDW architecture delivered a solution to the business needs with the following characteristics:

- Merging and cleansing (make consistent) data from multiple transactional sources.
- Separation of data used for transactional and reporting applications.

⁴See Reference [4] for more information on a data warehouse.

- Maintaining data history even when the source data did not.
- Transforming data into a data model optimal for reporting performance (using star or snowflake schemas). The transformation pipeline feeds dimension tables (e.g., time, sales regions, products, etc.), and the fact table(s) in the middle keeps the relational records between the dimensions, for example, sales of a product in a region in a time period.
- Making the data consumable for business users.

The EDW architecture was hugely successful, and almost every organization implemented it. EDWs are still used today for many reporting needs in many if not all organizations.

Nevertheless, the EDW architecture created *new challenges* as the business wanted to have insight into more, also semi-structured and unstructured data to make better, data-driven business decisions:

- The ETL process also created new data quality issues in the EDW. The transactional applications continuously added data elements to their databases (schema changes). Those new or changed data elements needed to be reflected in the ETL pipeline. It is not uncommon that EDW implementations do not adequately reflect the data of the transactional applications because the ETL processes were not updated correctly or those changes were missed altogether.
- The data latency for data in an EDW is typically longer than 1 day, which is not acceptable for many analytics applications any longer. Some reporting applications operate on data that is older than a week.

- New business requests for data analysis based on the data stored in the EDW are typically declined if the data is not already available in the EDW or it takes many months to implement the ETL processes to add new data elements.
- Even though many transactional applications today are required by law to keep historical data or could implement history data with little overhead using SQL constructs such as bitemporal functionality, ETL processes still spend most of the processing to calculate historical data, which increases the data latency and data quality challenges.
- Finally, an EDW is great for structured data. Integrating semi-structured and unstructured data is a challenge.

Those requirements in the context of newly available technologies delivered with the Apache Hadoop open source project using a network of many computers and massive inexpensive storage led to the data lake architecture.

Big Data, Data Lake, and Data Lakehouse

Business users require access to their organizations' data to explore the content, select and annotate, and access information using their terminology with an underpinning of data protection and governance, for example, a marketer seeking data for new campaigns, which requires ad hoc access to a wide variety of data sources and support for decision-making.

Storing and processing big data required a new data architecture. The massive number of new data sources generating structured, semi-structured (e.g., XML, JSON), and unstructured (e.g., text, audio, pictures) data could not be processed in an EDW.

The Apache Hadoop open source project⁵ was born out of the need to process this massive amount of data. It delivered key technologies that became the foundation of many data lake implementations:

- **Hadoop Distributed File System (HDFS)**, a distributed file system that provides very scalable and well-performing access to data stored on the file system.
- **MapReduce**, a massively parallel, batch-oriented programming model for large-scale data processing. Apache Spark and Apache Hive exploit in-memory technology to deliver even faster processing of large amounts of data as needed for ML algorithms.
- **YARN**, a resource management platform responsible for managing compute resources in clusters and Hadoop common libraries and utilities.

With the wider acceptance of public cloud solutions, today data lakes could be implemented using cloud storage services besides on-prem Hadoop implementations.

A data lake architecture usually has a single repository of data, typically stored in its raw format. It does not require schema definitions, rigorously maintained in an EDW. Instead, users build custom schemas into their queries that makes integration of data sources much more flexible.

Figure 1-3 shows a data lake architecture. On the left are all possible data sources that are copied into the data lake using different technologies such as streaming, data replication, and data integration. It has multiple data zones for different types of data processing. The landing zone stores the raw format of all incoming data. The analytical zone and archive stores transformed data used for reporting, advanced analytics, and ML. Data

⁵See Reference [5] for more information on the Apache Hadoop project with further links to HDFS, MapReduce, and YARN.

values may be replicated in multiple data zones in the data lake. On the right side are data-consuming applications such as decision management systems and predictive analytics tools. The concept of information governance was introduced to manage access authority to the copy of the data that would have been strictly protected in the original data source.

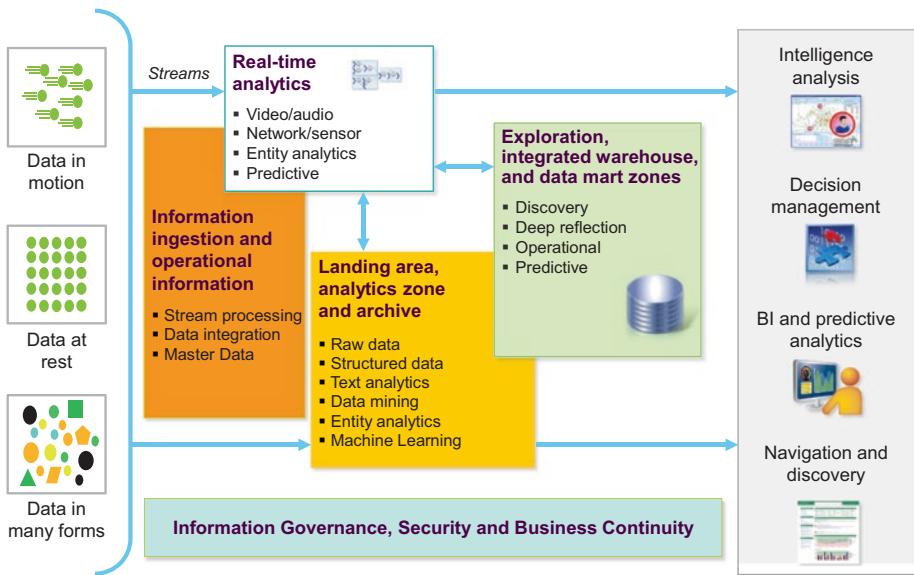


Figure 1-3. Big Data Platform – Data Lake Architecture

Many organizations that undertook data lake projects realized that the *store everything in one place* model creates not just opportunities but also new challenges. *The biggest challenge is not creating a data lake but taking advantage of the opportunity it presents.*⁶

⁶ See Reference [6] for more information on data lake challenges.

If not carefully managed, copying a massive amount of data can lead to the following challenges:

- Introduces the risk of using stale data for analytics due to the update latency into the data lake.
- Complexity to understand and resolve the version of truth among a growing number of data sources.
- Increasing storage and integration cost and complexity as data and the number of sources increase.
- Exposes the business to a lot of regulatory risks. Data governance is extremely difficult and costly as metadata of the original data sources changes constantly, resulting in adaptations in the data lake.

As a result, many organizations start to question the derived business value of a data lake compared with the original promise. The data lake becomes a data swamp. The challenges of the EDW and data lake created the requirements for new data management concepts, the data lakehouse, which is an open data management architecture that combines the flexibility, cost efficiency, and scale of data lakes with the data management and ACID (Atomicity, Consistency, Isolation, Durability) transactions of an EDW, enabling traditional Business Intelligence (BI) and ML on a variety of data structures. Thus, a data lakehouse combines characteristics and capabilities of a data lake with an EDW. This is often based on open source components, such as a delta lake, which is an optimized storage layer that provides the foundation for storing data and tables in a data lakehouse platform. A delta lake is open source software that extends Parquet data files with a file-based transaction log for ACID transactions and scalable metadata handling.⁷

⁷See Reference [7] for more information on data lakehouse and delta lake concepts.

Implementing a data lakehouse as a new data and AI platform requires a new data architecture, the Data Fabric. In Chapter 10, we further elaborate on the evolution of the Data Fabric architecture and Data Mesh solution. Today's enterprises need an agile data architecture – a Data Fabric architecture – that hides the ever-increasing complexity of the data source landscape and enables easy data consumption by business users. Taking an intelligent and agile Data Fabric architecture as its underpinning, the Data Mesh solution approach with its promise to establish a data marketplace with data-as-a-product consumption patterns is the way to go.

In the next chapter, we discuss the relationship and coexistence of a Data Fabric architecture and Data Mesh solution.

Key Takeaways

We conclude this chapter with a few key takeaways as summarized in Table 1-1.

Table 1-1. Key Takeaways

#	Key Takeaway	High-Level Description
1	Data architectures are an evolution.	Data architectures are typically developed based on business requirements exploiting state-of-the-art technologies.
2	Relational databases and data access via SQL.	Relational databases introduced the separation of data storage and logical data organization as well as standardized data access via SQL.
3	An enterprise data warehouse (EDW) integrates data from disparate sources.	An enterprise data warehouse is a central repository of integrated and transformed, structured data from disparate sources and used for reporting and data analysis.
4	EDWs are populated by complex ETL or ELT processes.	EDWs are populated by complex ETL- or ELT-based processes that create data latency, data quality, and flexibility challenges.
5	A data lake stores raw structured and unstructured data.	A data lake is a system of repositories storing structured and unstructured data in its raw format in a central place.
6	Data lakes can become data swamps.	Poorly managed data lakes have been called data swamps and deliver little value to the business.
7	Challenges of EDWs and data lakes lead to Data Fabric.	Challenges implementing an EDW and/or data lakes created requirements for a new data architecture – a Data Fabric.
8	Data Fabric is an integrated layer (fabric) of data and connection processes.	Data Fabric is an integrated layer of data sources and connection processes based on active metadata.

References

- [1] IBM, Sarkar, S., *An evolutionary history of enterprise data architectures*, 2021, www.ibm.com/blogs/services/2021/08/03/an-evolutionary-history-of-enterprise-data-architectures/ (accessed August 19, 2022).
- [2] Codd, E. F., *A relational model of data for large shared data banks*, 1970, <https://dl.acm.org/doi/10.1145/362384.362685> (accessed August 19, 2022).
- [3] IBM, *What is a data lake?*, www.ibm.com/topics/data-lake (accessed August 19, 2022).
- [4] Kimball, R., Ross, M., *The Data Warehouse Toolkit: The Definitive Guide to Dimensional Modeling*, 3rd Edition, Wiley, 2013, ISBN-13: 978-1118530801.
- [5] Apache Hadoop, *Apache Hadoop*, <https://hadoop.apache.org/> (accessed August 19, 2022).
- [6] PricewaterhouseCoopers, Stein, B., Morrison, A., *The enterprise data lake: Better integration and deeper analytics*, 2014, www.pwc.com/us/en/technology-forecast/2014/cloud-computing/assets/pdf/pwc-technology-forecast-data-lakes.pdf (accessed August 19, 2022).
- [7] L'Esteve, R., *The Azure Data Lakehouse Toolkit*, Apress, 2022, ISBN-13: 978-1-4842-8232-8.

CHAPTER 2

Terminology: Data Fabric and Data Mesh

This chapter explains the key terms that will be used throughout this book, the terms *Data Fabric* and *Data Mesh*, and how these two terms relate to each other. We introduce the term *data-as-a-product* or *shopping-for-data* and provide a high-level introduction into AI-infused Data Fabric capabilities. The chapter concludes with a description of a data product as a key concept of a Data Mesh.

Introduction

The terms *Data Fabric* and *Data Mesh* are often viewed as different, conflicting, or at the best overlapping data architectures or frameworks, data management concepts, or approaches to discover, explore, govern, and consume data. However, these concepts are related to each other, where each concept emphasizes specific imperatives or objectives.

A Data Fabric¹ has its focus more on the architectural underpinning, technical capabilities, and intelligent analysis to produce active metadata supporting a smarter, AI-infused system² to orchestrate various data

¹ See Reference [1] for a high-level introduction into a Data Fabric.

² See Chapter 7 for more details.

integration styles, enabling trusted and reusable data in a hybrid cloud landscape to be consumed by humans, applications, or other downstream systems. Data cataloging to generate and leverage active metadata is seen as a vital component of any Data Fabric.

A Data Mesh³ views data primarily as organized around domain owners who create business-focused *data products*, which can be aggregated and consumed across distributed consumers, organizations, and Line of Business (LoBs) in a *self-service* and *shopping-for-data* fashion. Transforming data from disparate data sources to be consumed as *data-as-a-product* is an essential paradigm of any Data Mesh. The key phrases *self-service*, *data marketplace*, *data-as-a-product*, and *shopping-for-data* are described further down in this chapter.

In this chapter, we convey a perspective of a Data Fabric to serve as an architectural underpinning to build a Data Mesh solution, where a Data Fabric with corresponding technical capabilities can additionally serve as an architectural underpinning for other solutions as well, for example, a traditional DWH solution. By using the terms *Data Fabric* and *Data Mesh*, we furthermore refer to structured and unstructured data, ETL stages, as well as other AI-related artefacts, for instance, ML/DL models, pipelines, etc.

Data Fabric Concept

The term *Data Fabric* was coined by NetApp⁴ in a white paper from 2016 as a “vision for data management... that seamlessly connects different clouds, whether they are private, public, or hybrid environments.” The NetApp description of a Data Fabric is focusing on the hybrid cloud landscape, highlighting key aspects like data security, governance, integration, the

³ See Reference [2] for an introduction into a Data Mesh.

⁴ See Reference [3] for the NetApp Data Fabric Architecture Fundamentals.

applications and services layer, etc. However, it neglects essential Data Fabric capabilities, for instance, the active metadata-driven approach, AI/ML-infused tasks, and self-service delivery.

Gartner⁵ defines Data Fabric as a “design concept that serves as an integrated layer (fabric) of data and connecting processes. A data fabric utilizes continuous analytics over existing, discoverable, and inferred metadata assets to support the design, deployment, and utilization of integrated and reusable data across all environments, including hybrid and multi-cloud platforms.” In their Data Fabric architecture, Gartner emphasizes key characteristics, such as embedded AI/ML, semantic knowledge graphs, automating repetitive tasks, active metadata (technical, business, operational, and social), dynamic data integration, data cataloging, and automated data orchestration. With these characteristics, Gartner’s Data Fabric architecture description is rather exhaustive and state of the art.

Forrester⁶ defines a Data Fabric to deliver a “unified, integrated, and intelligent end-to-end data platform to support new and emerging use cases. It automates all data management functions – including ingestion, transformation, orchestration, governance, security, preparation, quality, and curation – enabling insights and analytics to accelerate use cases quickly.” In their document, Forrester highlights built-in automation and intelligence across all data management functions, governance and compliance requirements, data cataloging, integration, AI/ML, knowledge graphs, and integration with Master Data Management (MDM), which is a rather comprehensive and complete set of Data Fabric capabilities.

⁵ See Reference [4] for the Gartner Data Fabric article.

⁶ See Reference [5] for The Forrester Wave: Enterprise Data Fabric, Q2 2022.

Data Fabric Framework

As you have seen in Chapter 1, the Data Fabric is the result from the data architecture evolution. However, many of its capabilities were in existence already long before the term *Data Fabric* was coined. For instance, data integration and data governance have been on the agenda of the IT industry for decades. So what are the new imperatives and capabilities that stand out?

It is foremost the AI/ML-based augmentation and automation of all Data Fabric areas, including AI/ML-based smart integration,⁷ generating active metadata and knowledge graphs to be captured in the knowledge catalog. Furthermore, ensuring trustworthy AI and broadening the scope of traditional data governance toward unified AI governance is a relatively new area that goes far beyond just focusing on data by including AI models and corresponding AI artefacts into the governance realm. Finally, automated generation, discovery, and enforcement of data rules and policies is increasingly underpinned with AI/ML.

These new imperatives and characteristics are depicted in Figure 2-1, which is a high-level illustration of a state-of-the-art Data Fabric framework.

⁷The terms *smart integration* and *intelligent information integration* are further described in Chapters 5 and 10.

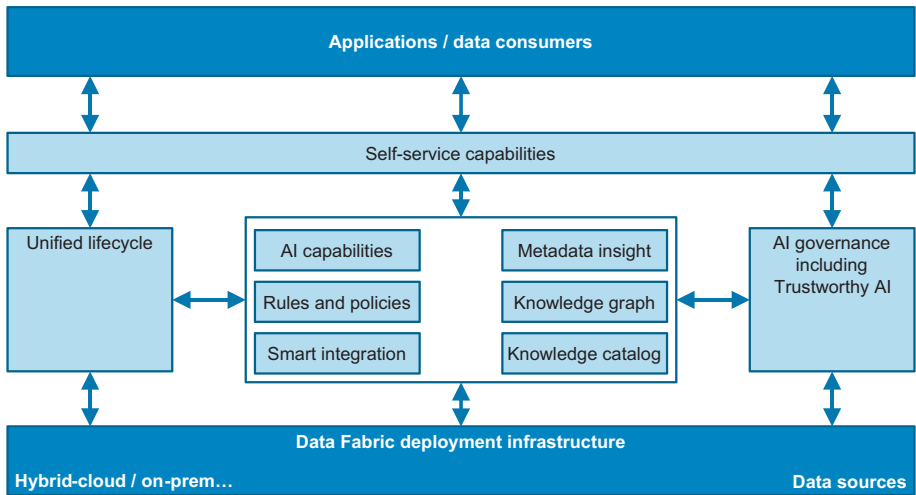


Figure 2-1. *Data Fabric Framework*

The following is a short description of the essential Data Fabric tasks and capabilities, which can be used to implement a Data Mesh or other solutions. These tasks and capabilities⁸ can easily be mapped to the Data Fabric framework as depicted in Figure 2-1:

- **Catalog all your data:** Including business glossary and design-time and runtime metadata (business, technical, and operational metadata)
- **Enable self-service capabilities:** Addressing data discovery, profiling, exploration, quality assessment, consumption of data-as-a-product, etc.

⁸We explore these tasks and capabilities further in Chapter 5.

- **Provide a knowledge graph:** Visualizing how data, people, processes, systems, etc. are interconnected, deriving additional actionable insight
- **Provide intelligent (smart) information integration:** Supporting IT staff and business users alike in their data integration and transformation, data virtualization, and federation tasks
- **Derive insight from metadata:** Orchestrating and automating tasks and jobs for data integration, data engineering, and data governance end to end
- **Ensure trustworthy AI:** Ensuring explainability, transparency, and trust of AI methods and outcomes, taking appropriate actions in case bias, drift, decreasing accuracy and precision, etc.
- **Enforce local and global data rules/policies:** Including AI/ML-based automated generation, adjustments, and enforcement of rules and policies
- **Manage an end-to-end unified lifecycle:** Implementing a coherent and consistent lifecycle end to end of all Data Fabric tasks across various platforms, personas, and organizations
- **Enforce data and AI governance:** Broadening the scope of traditional data governance to include AI artefacts, for example, AI models, pipelines, etc., and taking into consideration new or emerging data governance regulations⁹ and privacy laws

⁹See Reference [6] for more information on the draft EU regulation on AI.

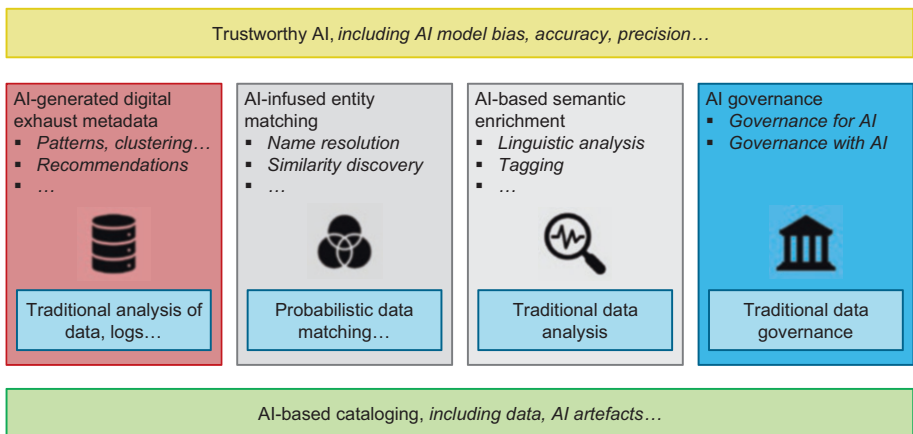
Since we have referenced AI/ML-based augmentation and automation of all Data Fabric areas as a new key imperative, this is the topic we elaborate on next.

AI-Infused Data Fabric

AI with its siblings ML and DL is the key to understanding what differentiates a Data Fabric concept or architecture from conventional data architectures that we have been dealing with in the past.

Gaining more insight into data, simplifying data access, enabling shopping-for-data, augmenting traditional data governance, generating active metadata, and accelerating development of products and services are enabled by infusing AI into the Data Fabric architecture. An AI-infused Data Fabric is not only leveraging AI but also likewise an architecture to manage and deal with AI artefacts, including AI models, pipelines, etc.

Figure 2-2 depicts the essential components that an AI-infused Data Fabric is composed of.



2

Figure 2-2. AI-Infused Data Fabric Concept

The following is a high-level description of these components¹⁰:

1. **AI governance:** The same statement that we made previously for an AI-infused Data Fabric applies to AI governance¹¹ as well: traditional data governance is obviously improved by leveraging AI methods (governance with AI), for instance, by applying ML methods to analyze data governance regulations and privacy laws or to implement governance-related data rules and policies. However, it also needs to broaden its scope by including governance for AI artefacts (governance for AI), for instance, by managing and governing AI models and other AI artefacts.
2. **AI-based cataloging:** The AI-based knowledge catalog constitutes another essential connecting component of an AI-infused Data Fabric by storing automatically generated metadata, leveraging AI methods to generate augmented knowledge, generating and storing knowledge graphs, storing the Data Fabric-based digital exhaust metadata, and enabling semantic enrichment.
3. **Trustworthy AI:** Deploying and operationalizing AI and consuming and interpreting AI outcomes require permanent transparency and explainability. The aim of trustworthy AI is to detect bias, drift, and decreasing accuracy and precision of AI models and to measure other AI-related KPIs during the entire lifecycle. It furthermore needs to propose corrective actions to realign AI artefacts to business goals.

¹⁰We discuss these components in every detail in Chapter 7.

¹¹ See Reference [7] for more information on AI governance.

4. **AI-generated digital exhaust metadata:** All Data Fabric tasks generate additional data, for instance, LOGs, quality measures of data, AI artefacts, data access paths, etc. We call this data digital exhaust, which can be transformed into digital exhaust metadata that can be further analyzed to gain additional insight to improve corresponding Data Fabric tasks.
5. **AI-based semantic enrichment:** Applying AI provides automated enrichment to contextualize data with semantic knowledge, for instance, based on knowledge graphs and other existing metadata in the catalog. Semantic enrichment thrives for simplification and optimization of data consumption by applications and business users by leveraging this semantic insight. It can further simplify and optimize some of the key Data Fabric tasks, such as searching and discovering assets in a particular business context.
6. **AI-infused entity matching:** Heterogeneous and dispersed data landscapes, systems, and applications have prevented business users from matching core information (master data), such as persona data (e.g., customers, business partners, employees, citizens, etc.), products, and services. AI-infused entity matching complements existing deterministic and probabilistic matching techniques, yielding much more accurate matching results.

Let us now address the key aspects of the Data Mesh concept.

Data Mesh Concept

Similar to the “Data Fabric Concept” section, we begin this section with a brief discussion about Gartner’s and Forrester’s views.

According to Gartner,¹² Data Mesh is a “solution architecture for the specific goal of building business-focused data products; it is a solution architecture that can guide design within a technology-agnostic framework.” In their article, Gartner highlights the essential imperatives of a Data Mesh, such as data-as-a-product, distributed data governance and authority, and presenting Data Mesh as a solution. In addition, Gartner also specifically describes the relationship of a Data Mesh and a Data Fabric, which we address in the following section.

According to Forrester,¹³ Data Mesh “is a sociotechnical approach to share, access, and manage analytical data in complex and large-scale environments – within or across organizations.” In their article, Forrester addresses the essential *sociotechnical principles* of a Data Mesh, such as “domain-oriented, data-as-a-product, self-service, federated, computational data governance. It puts the soft side of data and business outcomes first.”

In essence, a Data Mesh solution organizes data around business domain owners and transforms relevant data assets (data sources) to data products that can be consumed by distributed business users from various business domains or functions. These data products are created, governed, and used in an autonomous, decentralized, and self-service manner. Self-service capabilities, which we have already referenced as a Data Fabric capability, enable business organizations to entertain a data marketplace with shopping-for-data characteristics.

¹² See Reference [8] for Gartner’s view of a Data Mesh.

¹³ See Reference [9] for Forrester’s definition of a Data Mesh.

Figure 2-3 is a conceptual depiction of a Data Mesh solution, referencing its key characteristics or solution design imperatives. The heterogeneous and dispersed source data landscape contains the source data that is relevant to build a Data Mesh solution. To explain Figure 2-3, we need to point out one specific aspect in implementing a Data Mesh: as you can see in Figure 2-3, it can also include core information, depicted as `data_source_2` (master data) that is managed by an MDM system. This master data may be relevant for several data products and therefore must be discoverable, accessible, and consumable by different product owners. Therefore, a business user (for instance, from the `business_domain_B`), who is typically accessing data products from its `business_domain_B`, may also require access to this master data, which is owned by `data_domain_owner_A`. This need is illustrated in Figure 2-3 by the arrow from the middle data consumer accessing the master data as a data product, which is owned by `data_domain_owner_A`.

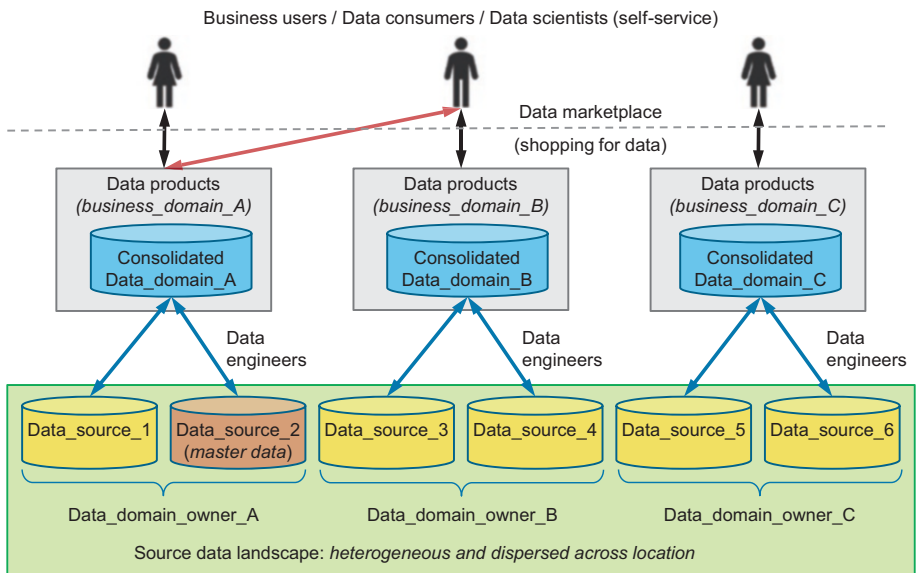


Figure 2-3. *Data Mesh Solution Design Imperatives*

Let us detail out the key design imperatives to build a Data Mesh solution as depicted in Figure 2-3. Again, when we refer to data, we mean data in a traditional sense, but also including AI artefacts:

- **Data sources:** The relevant data that is used to eventually build a data product can be stored in transactional systems, DWH systems, or a data lake or data lakehouse or managed by applications and other systems. It serves as a base for creating data products.
- **Data domain owners:** Source data is in custody of specific data domain owners, who perform data governance tasks and have responsibility for the consumability of the underlying source data. Data domain ownership is distributed with clear responsibility associated to those corresponding data assets (data sources).
- **Data domains:** Since a Data Mesh solution is inherently distributed across business domains, the underlying consolidated data domains are strictly within the realm of a particular business domain (line of business).
- **Data products:** Within a particular business domain, the corresponding data products are created. Again, this is a distributed undertaking, where specific data products are owned, managed, and consumed by data consumers that are associated with that business domain.
- **Data marketplace** (shopping-for-data): Data products need to be made available in a data marketplace environment for corresponding business users. This includes search, discovery, and consumption of *data-as-a-product*.

The value of a Data Mesh solution is that it assigns the creation of data products to data engineers and subject matter experts upstream who are most familiar with the business domains and corresponding needs.

As it was illustrated earlier, business users from other business domains may have an interest in accessing data products from another business domain. This cross-business domain data product consumption needs to be enabled by a Data Mesh solution as well.

In the previous sections, the relationship between a Data Fabric architecture and a Data Mesh solution seems to be already inherently perceived, although we have not quite clearly described this yet. It is this relationship that we elaborate on in the next section.

Relationship: Data Fabric and Data Mesh

When discussing the Data Mesh solution, we have highlighted key solution imperatives, such as the business-oriented data products, the distributed data domain ownership, self-service capabilities, etc. This was done without explicitly spelling out the technical capabilities required to implement the Data Mesh solution. Our Data Mesh solution design imperatives and key characteristics can be seen as a set of guidelines for implementing a solution. However, it is the Data Fabric architecture that enables the Data Mesh. In other words, the Data Fabric is the architectural underpinning to implement a Data Mesh solution.

As you have seen earlier in this chapter, Data Fabric is composed of technical capabilities, such as data profiling, automated metadata enrichment, business glossary, AI governance, etc., to deliver data-as-a-product. The *Data Mesh sociotechnical approach* and concept – to use Forrester’s term – is implemented via these Data Fabric technical capabilities. According to Gartner,¹⁴ a Data Fabric “can be built without

¹⁴ See Reference [8] for Garther’s comparison of Data Fabric and Data Mesh.

following data mesh practices. A data mesh must utilize the discovery and analysis principles that are intrinsic to a data fabric.”

In working with IBM customers, we have often been confronted with the perception that Data Fabric and Data Mesh approaches are alternative ones, where a conscious “either-or” decision must be made. We rather tend to agree with Gartner’s view, where the Data Fabric is an underlying architecture to build a Data Mesh solution. In addition, a Data Fabric is not limited as an architectural underpinning of a Data Mesh; it can also be purposely adjusted and designed to serve other solutions, for example, a state-of-the-art DWH system.

This relationship between these two terms is depicted in Figure 2-4. Based on the Data Mesh solution design imperatives, a set of Data Fabric technical capabilities are identified. The Data Mesh solution design imperatives, which have been deduced from the Data Mesh business requirements, are specifications for the Data Mesh solution.

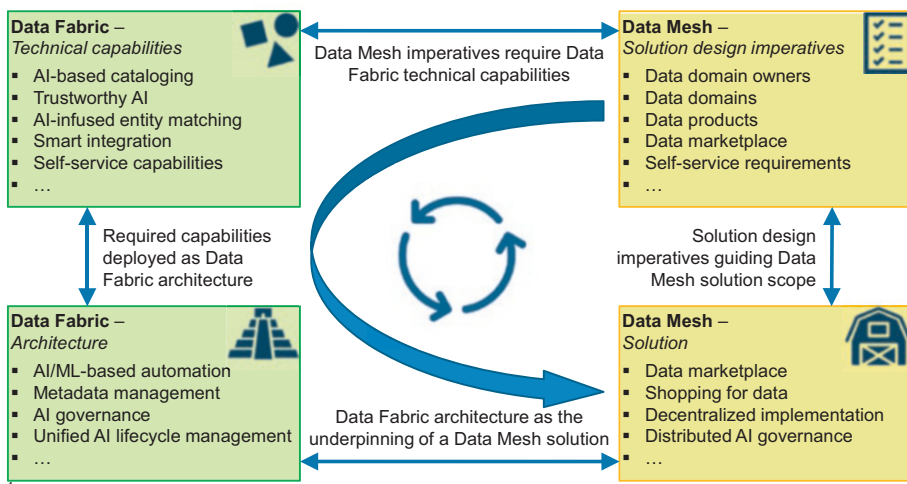


Figure 2-4. Data Fabric and Data Mesh Relationship

Since there are several distinct Data Fabric use case scenarios¹⁵ or entry points, not all available technical capabilities are required – at least not initially. Once a set of technical capabilities have been identified, a suitable initial Data Fabric architecture can be derived – that is, the identified Data Fabric capabilities are deployed within a corresponding Data Fabric architecture, which eventually serves as the architectural underpinning for the required initial Data Mesh solution.

As additional Data Mesh business requirements will be identified over time, the corresponding Data Fabric technical capabilities can be concluded, which results in adjustments and extensions of the Data Fabric architecture and subsequently enhancements of the Data Mesh solution. Thus, there is an iterative adjustment taking place over time to accommodate additional Data Mesh business requirements that may surface over time.

Data-as-a-product and *data product* are one of the most essential terms that are consistently used when describing a Data Mesh solution. It is these terms that we need to shed more light on, describing them in more detail.

Data Product

When does data become a data product? What needs to happen to raw data to be transformed into a consumable data product? Transforming raw data and building data products begins with clearly defined data ownership of raw data and ends with easy consumption of corresponding data products by data consumers, for example, business users.

Building a data product is *enabled* by the data domain owner; however, building a data product itself is primarily driven by the data product owner, which can be a marketing or a customer care organization, an after-sales

¹⁵ See Chapter 3 for more details.

team, or even an individual business user. The data product owner is collaborating with data engineers, data scientists, and other subject matter experts throughout the entire data product build process.

Figure 2-5 is a high-level depiction of what it takes to build a data product.¹⁶

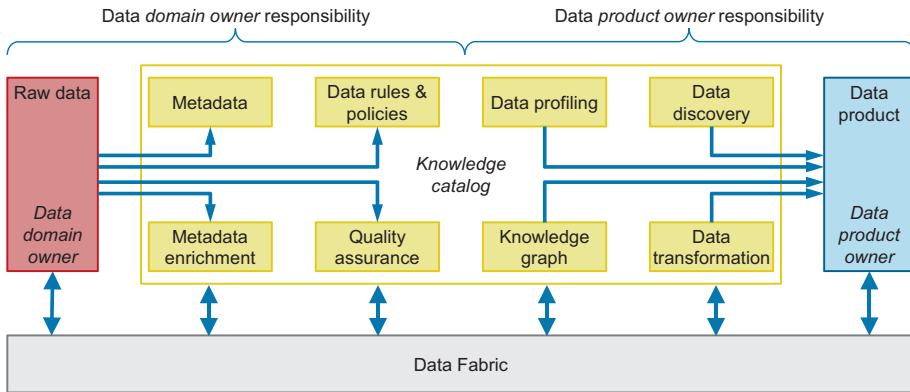


Figure 2-5. *Building a Data Product*

Aside from the Data Fabric and the knowledge catalog, Figure 2-5 depicts the two major areas of responsibilities, which are related to the data domain owner (left side) and the data product owner (right side).

The following are the data domain owner and data product owner responsibilities:

- Data domain owner responsibilities:** Data must be available and registered in the knowledge catalog, and the corresponding metadata should be enriched, which we detail out in Chapter 5. The availability, quality, consistency, completeness, and currency of data must be guaranteed via SLAs (service-level agreements).

¹⁶All characteristics and concepts depicted in Figure 2-6 are further detailed out in subsequent chapters of this book.

Data governance – including data quality – may be guaranteed via enforced data rules and policies. Most importantly, data domain ownership is not fading away simply because of data consumers accessing and possibly even copying the data to other data stores; it persists regardless of where data resides and who is consuming the data products for whatever purpose.

- **Data product owner responsibilities:** Building a data product requires easy discoverability of data; performing data profiling; gaining additional insight, for instance, via semantic knowledge graphs; and executing necessary data transformation jobs. These tasks are facilitated via a knowledge catalog and need to be performed in self-service fashion. A data product is a consumable outcome, meaning the result of these processes, where raw data is transformed into a meaningful business context. Data product ownership comes with guaranteed SLAs related to the availability, business relevance, easy consumability, completeness, and accuracy of the data product. In a way, we may consider the term *shopping-for-data* as a description for building a data product.

The underlying Data Fabric architecture enables DataOps, ModelOps, and AIOps to orchestrate required business and technical adjustments of data products throughout their entire lifecycle.

A data product¹⁷ is based on raw data that is transformed into a meaningful business context and easily consumable by business users. It is enabled via a knowledge catalog and can be characterized by persistent

¹⁷ See Reference [10] for more information on data products and their relationship to a Data Fabric.

raw data domain and data product ownership, which comes along with SLAs as described previously. Producing data products requires Data Fabric capabilities and GUI-based applications or tools with self-service capabilities, facilitating the steps to produce data products and to register them in the knowledge catalog for business consumption.

The following are the key characteristics of a data product. We can view these characteristics as data product specifications that are stored in the knowledge catalog itself; the data product itself is stored and made available for consumption in the data marketplace:

1. **Data product owner:** There is a clearly defined data product owner, who specifies their business objectives, features and characteristics, and road map with possible enhancements to be implemented over time.
2. **Data product description:** The data product needs to be well described, similar to any other product, that is, a software program or a software product. This may include the purpose or business intent of the data product, its version, legal aspects and usage limitations, currency of the underlying raw data, cost of using the data product, etc.
3. **Semantically related data and AI assets:** A data product consists of a set of semantically related data and AI assets, which can be a set of relational tables, but also a set of pipelines, a chain of ETL stages, a Jupyter notebook, etc. This semantic relationship is defined by the business context and usage purpose. The process of identifying semantically related data and AI assets is enabled via metadata enrichment capabilities of the knowledge catalog.

4. **Access methods:** The data product must include access methods, for example, REST APIs, SQL, NoSQL, etc., and where to find the data asset. This could, for instance, be a URI endpoint to be accessed by a defined REST API that is associated to the data product.
5. **Policies and rules:** This includes a description of who is allowed to consume the data product for what purpose. It furthermore addresses access control, authorizations, governance aspects, etc., all managed by data product policies and rules defined in the knowledge catalog.
6. **SLAs:** These are SLAs defined between the data product owner and the data product consumers and may include agreements regarding the data product availability, performance characteristics, functions, cost of data product usage, required explainability of AI models, trust and quality scores of the underlying data and AI assets, etc.
7. **Defined format:** A data product needs to be described using a defined format, which can, for instance, be achieved via a JSON or XML file (or any other data exchange format) that includes or points to the relevant assets (e.g., data product description, data assets, Jupyter notebook, AI model), the SLA, access methods, etc. The data product may, for instance, be stored in the knowledge catalog as a JSON or XML file.

8. **Cataloged:** All data products need to be registered in the knowledge catalog, where the JSON or XML documents need to be stored as well. These data products need to be searchable and discoverable by potential data product consumers and business users.
9. **Consumption-ready:** the data products need to be ready for consumption, meaning that for instance the underlying raw data and AI assets, data pipelines, ETL transformation jobs, etc. are available and accessible.

Creating a data product may also depend on raw data from multiple data domain owners. Furthermore, other data products may as well serve as additional input to build yet another data product. For instance, a data scientist may develop an AI model by taking either raw data from multiple data domain owners or several available data products as input, where the resulting AI model serves as input for a marketing organization to build a targeted marketing campaign offering as a set of data products.¹⁸ We may therefore establish a hierarchy of data products with corresponding dependencies of data products from other data products and data and AI assets.

We need to reemphasize that the data products themselves are not stored in the knowledge catalog; they are rather registered in the knowledge catalog via the data product specifications, which are considered metadata as well. Thus, the data product relates to the product specification like any other data or AI asset to its corresponding metadata.

¹⁸Please, refer to Chapter 12 where we provide a more detailed description of these two scenarios.

Making the data product specification available in the knowledge catalog as an XML or JSON file (or any other data exchange format) makes the data product searchable, discoverable, and consumable by business users and other data product developers.

It is somewhat indicative for a Data Mesh solution to focus on data-as-a-product in the context of organizational needs, thus adhering to a distributed and federated implementation approach. This, however, moves knowledge catalog orchestration and metadata exchange between multiple knowledge catalogs center stage in any Data Mesh implementation.

This knowledge catalog orchestration and metadata exchange can, for instance, be achieved via the Open Data Platform initiative (ODPi) Egeria, an open source Linux Foundation project,¹⁹ which is based on Open Metadata and Governance (OMAG) and Open Metadata Repository Services (OMRS). ODPi is a nonprofit organization accelerating and standardizing the open ecosystem of big data technologies for the enterprise; OMAG is a project to create a set of open APIs, types, and interchange protocols to allow all metadata repositories to share and exchange metadata; and OMRS are services to enable metadata repositories to exchange metadata.

This metadata exchange is enabled via an Egeria open metadata repository cohort, which is a collection of servers sharing metadata using a peer-to-peer exchange, as depicted in Figure 2-6. Any Egeria-compliant knowledge catalog can join the Egeria open cohort to exchange and share its metadata using open metadata and governance formats and interfaces.

¹⁹ See Reference [11] for more information on ODPi Egeria.

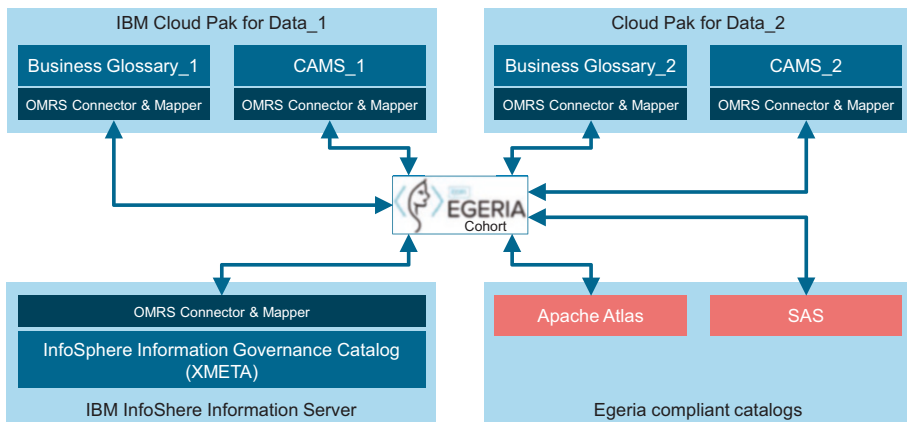


Figure 2-6. Knowledge Catalog Orchestration and Metadata Exchange

For instance, metadata exchange between IBM Watson Knowledge Catalog (WKC) and other Egeria-compliant repositories can be implemented over an Egeria cohort. IBM Watson Knowledge Catalog manages data assets (e.g., datasets, tables, columns, views, connections, and files as well as their relationships to glossary artefacts) in Common Assets Managed Services (CAMS) repositories. Business glossary artefacts (e.g., business terms, categories, data classes, policies, and governance rules) are managed in the business glossary repository, as depicted in Figure 2-6. CAMS has an embedded Egeria engine, which allows catalogs and the business glossary managed by IBM Watson Knowledge Catalog to be added to the OMRS cohort.

Any asset added to the knowledge catalog triggers Kafka OMRS messages, which are sent to the Egeria cohort. This requires the Kafka server to be accessible from IBM Watson Knowledge Catalog.

As we have pointed out earlier in this chapter, the Data Fabric is the architectural underpinning to implement a Data Mesh solution. In the subsequent chapters, we are therefore using the term *Data Fabric* when we are specifically referring to the architectural aspects, whereas the term *Data Mesh* is used when we are specifically emphasizing the data product (data-as-a-product) and data marketplace aspects.

Key Takeaways

We conclude this chapter with a few key takeaways as summarized in Table 2-1.

Table 2-1. *Key Takeaways*

#	Key Takeaway	High-Level Description
1	Realize the new aspects of Data Fabric.	AI/ML-based augmentation and automation of all Data Fabric areas, including AI/ML-based smart integration, generating active metadata and knowledge graphs to be captured in the knowledge catalog, are the novel aspects of a Data Fabric.
2	Enable trustworthy AI.	A Data Fabric can be used to enable trustworthy AI to measure correct bias, drift, and accuracy and precision of AI models.
3	Leverage the digital exhaust.	AI-generated digital exhaust metadata can be analyzed to further optimize and simplify the Data Fabric architecture.
4	Augment entity matching with AI.	AI-infused entity matching extends traditional entity matching solutions beyond just probabilistic and deterministic matching.
5	Data Mesh is a solution.	A Data Mesh should be seen as a solution, which is underpinned by a Data Fabric architecture.
6	Enable semantic enrichment.	AI-based semantic enrichment should be an essential component of the Data Fabric approach.
7	Data Mesh is about delivering data-as-a-product.	The aim of a Data Mesh solution is to establish a data marketplace where data can be searched for, discovered, and consumed as a product.

(continued)

Table 2-1. (continued)

#	Key Takeaway	High-Level Description
8	Relationship of Data Fabric and Data Mesh.	Data Fabric is an underlying architecture to build a Data Mesh solution, where Data Mesh is specified through a number of solution design imperatives.
9	Data Fabric enables DataOps, MLOps, and AIOps.	Realizing and enabling DataOps, MLOps, and AIOps to implement adjustments of data products requires some of the described Data Fabric capabilities.
10	A data product is raw data transformed into a business context.	A data product is based on semantically related raw data that is transformed into a meaningful business context and easily discoverable and consumable by business users.
11	Data products are registered in the knowledge catalog.	Data product specifications (XML, JSON, etc.) are the means by which data products are registered in the knowledge catalog; data products themselves are not stored in the knowledge catalog.
12	An Egeria cohort enables metadata exchange.	Knowledge catalog orchestration and metadata exchange can, for instance, be achieved via the Open Data Platform initiative (ODPi) Egeria, an open source Linux Foundation project.

References

- [1] LaPlante, A., *Data Fabric as Modern Data Architecture*, O'Reilly, 2021, ISBN: 9781098105945.
- [2] Dehghani, Z., *Data Mesh: Delivering Data-Driven Value at Scale*, O'Reilly, 2022, ISBN-13: 978-1492092391.

- [3] NetApp, CaraDonna, J., Lent, A., *NetApp Data Fabric Architecture Fundamentals*, Version 3 | WP-7218, 2016, https://cdn2.hubspot.net/hubfs/525875/Data-Fabric/Data_Fabric_Architecture_Fundamentals.pdf (accessed July 5, 2022).
- [4] Gartner, Gupta, A., *Data Fabric Architecture Is Key to Modernizing Data Management and Integration*, 2021, www.gartner.com/smarterwithgartner/data-fabric-architecture-is-key-to-modernizing-data-management-and-integration (accessed July 6, 2022).
- [5] Forrester, Yuhanna, N., *The Forrester Wave™: Enterprise Data Fabric, Q2 2022*, 2022, <https://reprints2.forrester.com/#/assets/2/73/RES176390/report> (accessed July 6, 2022).
- [6] European Commission, *Proposal for a Regulation laying down harmonised rules on artificial intelligence*, 2021, <https://digital-strategy.ec.europa.eu/en/library/proposal-regulation-laying-down-harmonised-rules-artificial-intelligence> (accessed July 8, 2022).
- [7] Hechler, E., Oberhofer, M., Schaeck, T., *Deploying AI in the Enterprise*, Apress, 2020, ISBN-13: 978-1484262054.
- [8] Gartner, Beyer, M., Zaidi, E., Thanaraj, R., De Simoni, G., *Quick Answer: Are Data Fabric and Data Mesh the Same or Different?*, 2021, www.gartner.com/doc/reprints?id=1-292DG4LD&ct=220209&st=sb (accessed July 10, 2022).

- [9] Forrester, Goetz, M., *Exposing the Data Mesh Blind Side*, 2022, www.forrester.com/blogs/exposing-the-data-mesh-blind-side/ (accessed July 10, 2022).
- [10] Starburst, Abadi, D., *Can the data fabric automate the creation of data products?*, 2022, www.starburst.io/blog/can-the-data-fabric-automate-the-creation-of-data-products/ (accessed September 15, 2022).
- [11] The Linux Foundation Projects, *Egeria*, <https://egeria-project.org/> (accessed October 23, 2022).

CHAPTER 3

Data Fabric and Data Mesh Use Case Scenarios

This chapter walks through several use cases for implementing a Data Fabric and Data Mesh that also represent business-relevant entry points. Data governance and privacy initiatives are ongoing in almost every organization, enabling access to enterprise data and AI artefacts across platforms to the people who have a business need. Other use cases are driven by hybrid cloud data integration; the need for a comprehensive view on customers, vendors, and other parties for better business outcome; and development and integration of trustworthy AI into business processes.

Introduction

Many organizations realized the importance of implementing a data-driven business to stay competitive. Therefore, a chief data officer (CDO) of an organization is challenged to create and implement a data strategy as a foundation for a data-driven business.

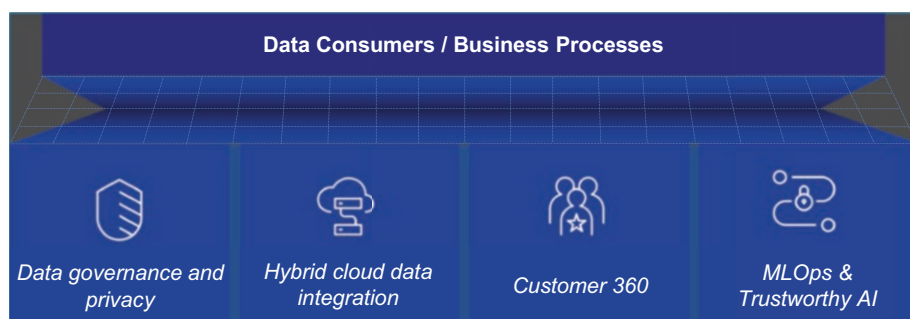


Figure 3-1. Common Data Fabric/Data Mesh Use Case Scenarios

The essential components of a Data Fabric architecture and Data Mesh solution as described in Chapter 2 align with typical data strategy focus areas. Therefore, use cases centered around components of these two concepts create an opportunity to demonstrate the value and meaningful entry points for your Data Fabric and Data Mesh journey.

Figure 3-1 shows common use cases as meaningful and business-relevant entry points for starting to implement a Data Fabric/Data Mesh and progressing on a journey over time. A more in-depth treatment of these four entry points are provided in later chapters of this book:

- **Data governance and privacy:** Establish an environment for automated and consistent AI governance, which addresses data in a traditional sense, but also embraces AI artefacts, for example, ML models, which need to be governed as well.
- **Hybrid cloud data integration:** Create a unified view of the data across the hybrid cloud, which includes public and private cloud deployments, as well as traditional on-premises systems.

- **Customer 360-degree view:** Provide a comprehensive view of customers, vendors, and other parties by infusing AI into the matching process to guarantee increased accuracy and confidence in this core information.
- **MLOps and trustworthy AI:** Implement trustworthy AI and MLOps to detect and address bias, drift, and worsening quality metrics for AI models that are deployed and in production and, furthermore, to address AI model explainability.

The priority in the data strategy defines the use case or entry point that is most applicable and relevant for an organization. In starting your Data Fabric and Data Mesh journey in the context of these entry points, you could select either one or even several use cases, especially since there is some relationship between some of these scenarios. For instance, trustworthy and explainable AI is conceptually related to AI governance.

To ensure success of introducing a Data Fabric as a new data architecture, it is critical to scope the use case well and define an outcome that delivers value to the business as defined in your data and AI strategy. It should show the differentiation of the outcome in a Data Fabric architecture compared with the existing data landscape in the organization, for example, a data lake or traditional EDW.

This may sound obvious, but many data lake projects were declared a failure because structured data of an EDW with a well-defined schema was moved into a data lake and stored as files in a Hadoop environment. Adopting such an approach lost the benefits of optimized processing of structured data in a relational database and the well-defined schema information, that is, as it is defined in a star or snowflake schema. The data consumer had to re-detect the schema information that was previously readily available in the EDW. It is hard to sell the benefits of a data lake

to the business if the processing of the same analysis takes much longer with the same processing capacity and the data consumer must carry the burden of data structure detection.

The discussed use cases provide a unique opportunity to deliver value to the business by utilizing a Data Fabric architecture and Data Mesh solution.

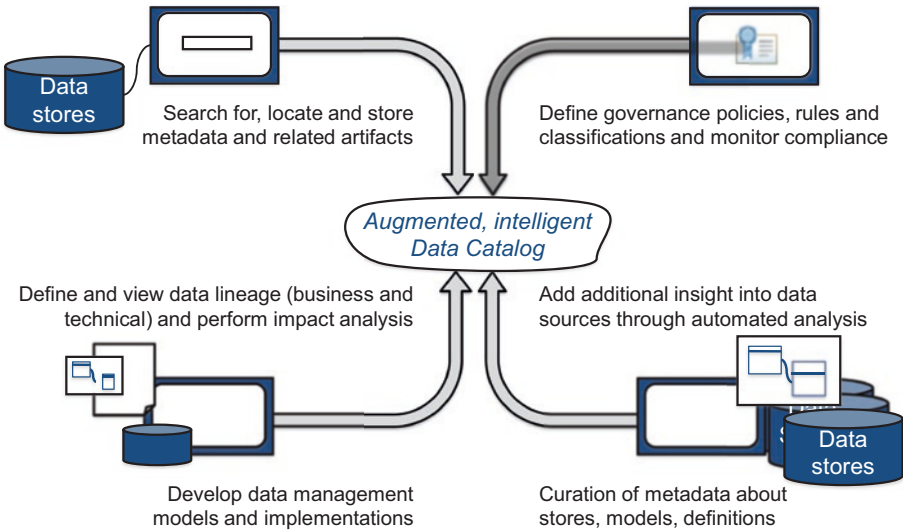


Figure 3-2. Activities to Build an Augmented, Intelligent Data Catalog

Even when the goal of a specific use case varies, all use case implementations will first lead to the activities for building an augmented, intelligent data or knowledge catalog¹ as the core foundation to implement these concepts as shown in Figure 3-2.

¹ We prefer to use the term *knowledge catalog*, because it does justice to the broad AI-related artefacts that go far beyond just traditional data.

The following are the key activities related to the knowledge catalog:

- **Store metadata:** Search for, locate, and store metadata and related artefacts (business, technical, and operational metadata) in the knowledge catalog. Even if it is desired to eventually have all organizational data in the data catalog, it needs to be built up incrementally and is focused on data for the selected use case in a defined business domain or related to a specific organization. An example in the healthcare industry could be patient information. The knowledge catalog should also store metadata related to AI artefacts, such as AI models, Jupyter notebooks, pipelines, etc.
- **Define data lineage:** Define and view data lineage (business and technical) and perform impact analysis. Defined by data architecture implementations such as an EDW and a data lake, data in an organization is frequently copied for consumption by different applications. It is important to make the data pipelines visible. First, regulations such as the General Data Protection Regulation (GDPR)² require an organization to remove personal data in all copies on request. To act on this regulation, it needs to be known which data pipelines processed that data and what copies were created or updated by the data pipeline execution. Second, it also provides an indication about data quality and data latency. It is not a technical challenge to copy data from location A to location B. It is a real operational challenge to keep copies from locations A and B consistent over a longer period of time. The

²See Reference [1] for more information on GDPR.

problem becomes infinitely complex with increasing number of copies and volume of data copied. Third, it makes data integration optimization and alternative approaches visible for data engineers. For example, is it necessary and advisable to copy the original data in a data pipeline, instead of complementing or extending the original data with derived data, which could possibly be a smarter approach?

- **Define data and AI governance:** Definition of data and AI governance policies, rules, and classifications is critical to break down data silos, allow for a uniform data consumption, and prevent misuse of data. It includes monitoring of compliance and enforcement of data and AI rules and policies on an ongoing basis, as well as ensuring compliance with regulations and laws.
- **Add additional insight, enriching data:** Add additional insight into data sources through automated analysis. Augmenting metadata with additional information such as data distribution, data quality, and business terms is critical to provide data as a service (DaaS) to the data and data product consumers. For example, the data analysis of the attribute *telephone number* in a customer record can show the telephone number is not consistently maintained within various data stores and therefore has limited value for a consuming application or user.

Establishing a knowledge catalog by executing the preceding activities is intuitive for a small number of data sources in a static environment. In the real world, the number of data sources and existing copies can be large, and metadata is constantly changing too.

The functionality of products, designed to support the implementation of a Data Fabric and Data Mesh, delivers the differentiating value as listed in the following:

- **Task automation:** Intelligently automate discovery, classification, and cataloging to keep metadata current. For example, new data attributes are added to a customer record in support of new functionality in the transactional application. This metadata change needs to be propagated to the data catalog and consuming applications alerted about the change.
- **Limiting data movement:** Leave original data where it lives and implement alternative data integration techniques using data virtualization to reduce data duplication significantly. It improves usability, latency, and quality of the data. Many decisions to copy data were made at a given time with product and hardware capabilities that are potentially outdated today. It is already a challenge to keep a relatively small number of metadata changes current and consistent over an extended period of time. Keeping many terabytes of data copies consistent over a longer period is realistically impossible.
- **Smart deployment of AI models:** Deploy AI models to intelligently deliver and orchestrate data for specific use cases. For example, a marketing campaign service would better access original data sources through data virtualization to determine the target group for the campaign, whereas data preparation services as input for ML model development could execute data transformation pipelines and store the prepared data

as copies in an analytics zone. Intelligent information integration, meaning AI-infused information integration, supports data engineers in simplifying their tasks.

Automated and Consistent Governance

Establishing an environment for automated and consistent AI governance is a data strategy priority in many organizations and the first use case as an entry point for implementing a Data Fabric architecture and Data Mesh solution as shown in Figure 3-3.

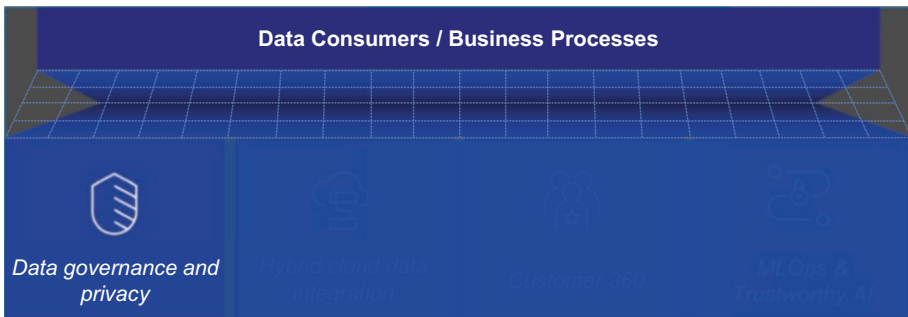


Figure 3-3. *Data Fabric/Data Mesh Use Case – Data Governance and Privacy*

The term *data governance*³ is used for the processes and responsibilities that define, manage, and enforce access, privacy, availability, and security of the organization’s data. It typically includes a set of policies, rules, and data classifications and functionality to monitor and enforce compliance. As stated earlier, we use the term *AI governance* in a broader sense, also including AI artefacts.

³See Reference [2] for a comprehensive treatment on data governance.

Data governance is not a new concept. The need to implement centralized data and AI governance exists to detect data inconsistencies across different data sources. For example, a data attribute such as account number is named the same but has different meanings across data sources and cannot be used to combine related data. Alternatively, the same attribute is stored under different names across data sources, but it is not visible to the data consumer without additional metadata.

Every data platform provides functionality to define and manage data access rules and policies. This siloed implementation is a pain point for data consumption across data sources. For example, when a user needs access to data stored on different data platforms, it typically triggers specific processes for each platform to approve and define the requested data access and takes a long time to complete.

Monitoring compliance with ever-increasing data privacy and protection laws such as GDPR in the European Union (EU) across the different data sources and simply answering the question who has obtained what data access authority in an organization are problems in themselves. Therefore, many organizations started to establish a data and AI governance structure⁴ focusing on the following activities:

1. **Data and AI identification:** Identify data and AI assets and create an inventory for an organization or the entire enterprise with related metadata – a knowledge catalog.
2. **Data management policies:** Define the rules and policies for governing the creation, acquisition, privacy, integrity, security, classification, and quality of the data.

⁴See Reference [3] for an example on the impact of implementing data governance.

3. **Data assessment:** Develop processes to measure the quality, usage, and impact of data and AI artefacts as well as monitor and optimize related actions.
4. **Data communication:** Establish a culture of sourcing data through the knowledge catalog and maintain currency of the metadata stored in the knowledge catalog.

It becomes very visible that a data and AI governance structure⁵ can be established by implementing an intelligent, augmented data catalog as the foundation for a Data Fabric architecture to build a Data Mesh solution.

Any project execution would be very difficult without implementation and usage of the right product capabilities. The selected products should support the data sources and platforms in your organization and provide AI-augmented functionality to ingest and automatically enrich metadata, allowing business users to easily understand, collaborate, enrich, and access the right data, to quickly establish an environment for highly automated and consistent governance and automatically secure data across the organization.

Include IBM zSystems Data in AI Governance

As an example, we would like to introduce you to IBM zSystems,⁶ also known as the mainframe, which runs most business-critical applications for many large enterprises. Seventy-one percent of the Global Fortune 500 companies run their business and store data on IBM zSystems, either in the relational database Db2 for z/OS, in the structured file system VSAM, or in the hierarchical database IMS. Many data engineers and data

⁵ Please, refer to chapter 15 on *Data Governance in the Context of Data Fabric and Data Mesh* for more details.

⁶ See Reference [4] for an introduction into IBM zSystems.

consumers are not familiar with IBM zSystems and see the understanding of and ability to access the operational data stored on IBM zSystems as a major pain point.

An organization that stored data on IBM zSystems and on other platforms expressed their requirements for improved data consumption:

- Make the data attributes available to data consumers on demand.
- Support the coexistence of new and legacy data platforms.
- Provide regulatory compliance for interoperability and security.
- Apply consistent governance to improve data quality for consumers.
- Enable near-real-time access to data.

Figure 3-4 shows an architecture overview diagram (AOD) on how this organization can respond to those requirements by implementing data governance services across data stored in Db2 for z/OS, VSAM, as well as other data sources.

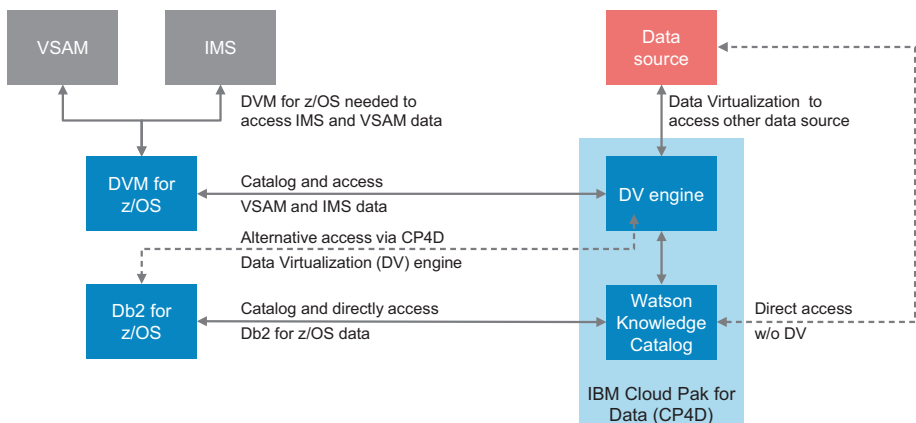


Figure 3-4. Architecture Overview Diagram – Data Governance Scenarios

In this example, IBM Watson Knowledge Catalog on IBM Cloud Pak for Data provides a secure enterprise management platform that is supported by a data governance framework. The data governance framework ensures that data access and data quality are compliant with the defined business rules and processes:

1. **Knowledge catalog:** IBM zSystems data needs to integrate with IBM Watson Knowledge Catalog. Data stored in Db2 for z/OS can be directly integrated into the knowledge catalog, whereas data stored in IMS or VSAM needs to be mapped first to virtual views in IBM Data Virtualization Manager for z/OS (DVM for z/OS). Metadata from DVM for z/OS can also be integrated into the knowledge catalog.
2. **Governance artefacts:** Data stewards and data quality analysts can assign AI governance artefacts, for example, data and AI policies and rules, and analyze and improve the quality of data and AI assets as well as define data masking rules to protect sensitive data.
3. **Rules and policies:** Data governance rules and policies are consistently applied when IBM zSystems data is accessed, for example, inserted, deleted, or updated.

This integration approach demonstrates how data stored on IBM zSystems can be made easily available for smarter consumption by stakeholders without any *mainframe* knowledge while maintaining rigorous data quality and regulatory compliance standards.

Unified View of Data Across a Hybrid Cloud

A hybrid cloud⁷ integrates public cloud services, private cloud services, and on-premises infrastructure and provides orchestration, management, and application portability across all three. As shown in Figure 3-5, its adoption creates the requirement of a unified view of data across a hybrid cloud leading to the next Data Fabric or Data Mesh use case.

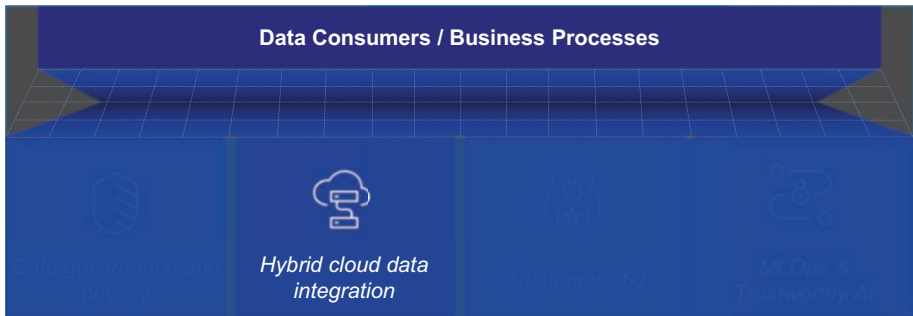


Figure 3-5. *Data Fabric/Data Mesh Use Case – Hybrid Cloud Data Integration*

Traditionally, enterprises managed their own on-premises IT infrastructure and computing centers and installed and maintained hardware and software (operating systems, middleware, and applications). Depending on how organizations kept up with the adoption of modern IT capabilities and the complexity of internal change processes, new business solutions were often too cumbersome, could take a long time, and were too costly to design and implement.

The inability to respond to new business requirements frustrates business units and creates an opportunity for public cloud providers. Public cloud providers focus on automation of the provisioning and metering of service resources and can offer a variety of services. Cloud providers offer infrastructure as a service to quickly provision compute

⁷ See Reference [5] for more information on a hybrid cloud.

resources (e.g., compute power, storage, memory) for the deployment of the needed software stack as well as development and testing of a new application to make it available to the business unit. Other cloud providers offer the subscription to a solution as a service such as salesforce.com for customer relationship management.

The cloud-style provisioning with its benefits of elasticity, scalability, and ease of service delivery led to the development of virtualization, software management, and automation technologies that organizations adopted in their own computing centers, implementing a private cloud.

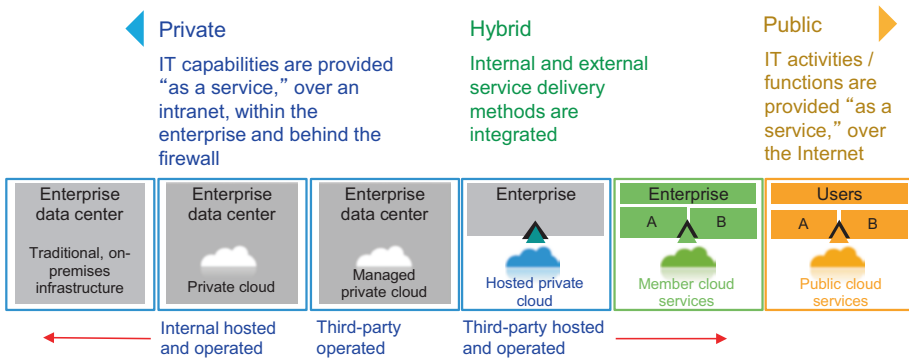


Figure 3-6. *Deployment Options for Cloud Computing*

Figure 3-6 shows the different deployment options for cloud computing. Many organizations choose an application modernization strategy in support of their business needs. For example, an organization could choose to sunset their own custom CRM application and subscribe to a cloud-native service provider. They could decide to move their custom finance application or applications related to gaining analytical insights⁸ to a public cloud provider.

⁸ See Reference [6] for more information on gaining analytical insights using the cloud.

Many IT organizations also chose the approach to implement the technology for deployment and metering automation in their IT data centers creating a private cloud. Another common choice is application modernization, adoption of a modern application architecture such as the microservices architecture in the on-premises infrastructure, which could be driven by regulatory compliance requirements for core business applications.

Figure 3-7 shows the architectural overview of the preceding example, where the data previously stored on-prem (left side of Figure 3-7) in few data sources are now spread across multiple public clouds and multiple on-premises platforms in all possible formats (right side of Figure 3-7). Creating a unified view of the organization's data across the hybrid cloud is now even more complex than on-premises alone.

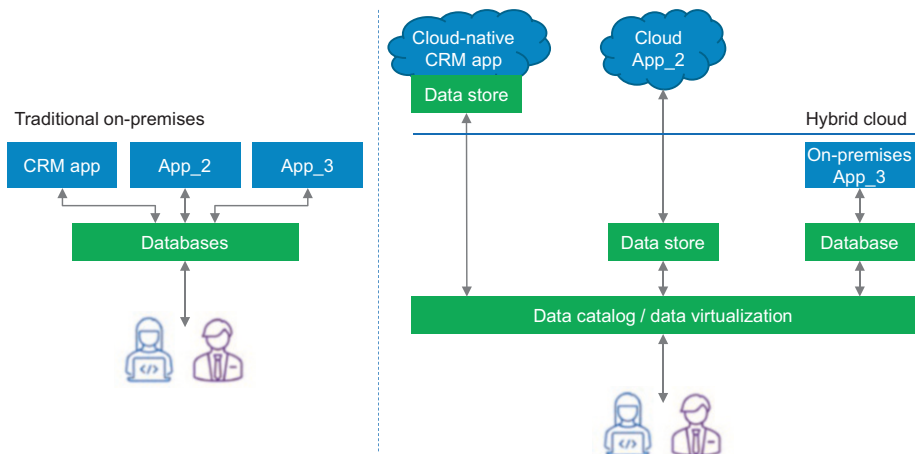


Figure 3-7. Architectural Overview Diagram – Hybrid Cloud Data Integration

The data lake architecture may not be a good answer to this data access requirement as copying massive amounts of data across different cloud providers into a single data store daily may encounter

operational difficulties (e.g., network latency and bandwidth issues, problem determination across many components) and cost may become prohibitive.

Implementing a Data Fabric⁹ across the different data sources with focus on augmented data integration flows making the best use of ETL, data virtualization, and real-time capture for optimized access to those many diverse data sources seems to be a better architecture addressing today's requirements for increased agility and flexibility.

Such a project would be approached with the previously outlined steps, again in a well-defined scope, such as a data domain:

1. **Data identification:** Identify domain-related data assets across hybrid cloud data sources, where it is critical for the success regarding implementation across a hybrid cloud environment with all data source owners carrying responsibility for data as a service to the data consumer and therefore having a strong interest to integrate the metadata of the data source in the domain knowledge catalog. Additionally, they need to provide an applicable business glossary.
2. **Data management policies:** The data source owners need to define the rules and processes for governing the creation, acquisition, privacy, integrity, security, classification, quality, and consumption of that domain data and AI artefacts.
3. **Data assessment:** Develop processes to measure the quality, usage, and impact of data and AI assets, as well as monitor and optimize related actions.

⁹We elaborate further on the role of a Data Fabric in a hybrid cloud landscape in Chapter 12.

4. **Data as a service (DaaS):** It becomes imperative for the organization to provide a marketplace for the data consumer and source the data through the knowledge catalog. The automation of data integration pipelines and recommendation for data and AI asset integration technologies such as data virtualization needs to be part of this use.

Adopting a Data Fabric for hybrid cloud data integration allows to create a view across data stores enabling consistency between operational applications. It is a great opportunity to consolidate and simplify IT infrastructure to run anywhere (on-premises or in any cloud) and automate data operations. The key differentiator is the implementation of intelligent (smart) information integration with augmented data integration flows making the best use of various data provisioning methods, such as ETL, data virtualization, replication, and streaming technologies to optimize access to many diverse data sources.

The Data Fabric provides capabilities to build quality analysis and remediation natively into data pipelines without costly downstream processing. Building a Data Mesh solution within a hybrid cloud environment relies on capabilities of such a Data Fabric architecture.

It is critical for data consumption to support a unified, end-to-end AIDataOps lifecycle¹⁰ through common governance rules, policies, and procedures.

¹⁰ See Chapter 9 for more details.

Provide a Comprehensive View of Customers, Vendors, and Other Parties

There is a common agreement in the business world that understanding your customer is critical for all aspects of the business. Figure 3-8 illustrates Customer 360, the comprehensive view of customer, vendor or other party data as the third use case for a Data Fabric or Data Mesh project.



Figure 3-8. *Data Fabric/Data Mesh Use Case – Customer 360*

It is an essential input to the business strategy, which products or services to invest in for driving innovation and to grow sales successfully as well as provide outstanding customer service. According to a McKinsey survey¹¹ in 2019, only about a fourth of the survey participants described their companies using customer insight consistently within their organizations.

This use case shows how a Data Fabric architecture provides the foundation to tackle the problem of inconsistencies in customer or other party profiles, establishing a single, trusted, 360-degree view of a customer, thus enabling a Data Mesh solution approach that requires a 360-degree view of customer data for building corresponding data products and making them available via a trusted data marketplace.

¹¹ See Reference [7] for the McKinsey survey.

Historically, every application in an organization maintained their own customer, vendor, or other party relationship. In the insurance industry, for example, it was common to have different applications to manage life, car, and home insurances. If a customer owned a car, life, and home insurance policy with that insurance company and changed the telephone number, the telephone number would need to be changed three times.

With the trend of online customer engagement in omni-channel environments, organizations focused on MDM, centralizing the customer data and other core information, integrating it with existing applications. Still, it is difficult to answer questions such as the following:

- Which of my customers are also my vendors to negotiate better contract terms, and what is the likelihood for those customers to accept another contract?
- Is there a relationship between a customer and a service provider that could expose a potential fraud pattern, and what is the likelihood of fraud?
- If multiple entries in the customer database are similar, are they representing the same person or party to be included in a marketing campaign?

Centralizing the customer data into an MDM solution and integrating existing applications may have worked well if all applications run on-premises, preferably on a single platform. Nevertheless, integrating a new data source into an existing MDM solution is a rather complex endeavor and increases time and cost of adoption of new business solutions.

It becomes even more complex and demanding when the organization adopts a hybrid cloud strategy.

Figure 3-9 shows how a Data Fabric architecture¹² can help improve customer insight.

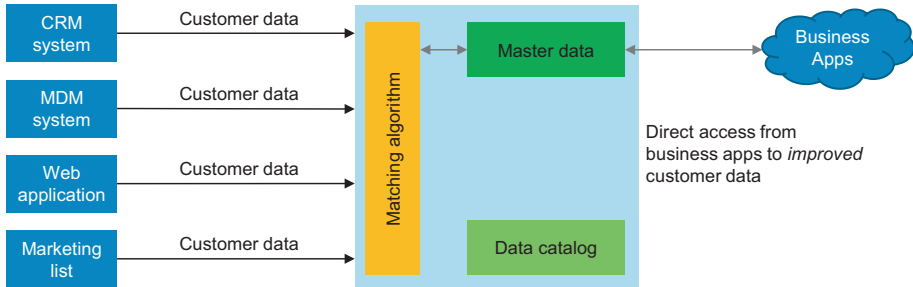


Figure 3-9. Architecture Overview Diagram – Increase Trust in Master Data for Business Consumers

The customer data is chosen as the domain, and the project would include the actions for implementing a Data Fabric with focus on the following:

- **Data identification:** Customer, vendor, or other party data is discovered and identified across data sources, which may be distributed within a hybrid cloud landscape.
- **AI-infused entity matching:** Create high-quality customer, vendor, or other party profiles by applying AI-infused entity matching. AI-infused entity matching complements existing deterministic and probabilistic matching techniques, yielding much more accurate matching results.
- **Enriched customer profile:** Augment customer metadata with additional information such as data distribution, data quality, business terms as well as semi- and unstructured data from customer engagement records, etc.

¹²We elaborate more on this topic in Chapter 8.

- **Data management policies and rules:** The data source owners need to define the rules, policies, and processes for governing the creation, acquisition, privacy, integrity, security, classification, and quality of the customer data. The enforcement of regulations needs to be automated.
- **Data as a service (DaaS):** The data latency and data quality are especially important for provisioning customer and other party data. AI algorithms should be used to intelligently provide recommendations for data delivery and orchestration for specific use cases. The usage of data virtualization to access the operational data source may be the preferred data integration technique for systems of engagements. It avoids data inconsistencies for the data consumers and simplifies integration of new data sources.

The Data Fabric and Data Mesh capabilities centered around an intelligent augmented knowledge catalog support the creation of a comprehensive customer, vendor, or other party profile and therefore increase trust in master data for data consumers. Better customer profiles are created when entities across different data sources are resolved by applying AI-infused entity matching¹³ vs. spending time on hunting quality data that is well understood.

¹³ In Chapter 8, we provide some examples.

Unlock the Trustworthy AI Concept

AI continues to penetrate all aspects of business operations and enables new business use cases such as improving the profitability of the business, augmenting human capabilities and experience. Figure 3-10 illustrates the forth and final use case for a Data Fabric or Data Mesh project.

According to an IBM Institute for Business Value (IBV) survey of C-level executives,¹⁴ 85% of advanced adopters are reducing operating costs with AI, and 99% of the participants report a reduction in cost per contact from using virtual agent technology. While AI has a lot of promise, many people question the trustworthiness¹⁵ of applying AI. Three out of four executives view AI ethics as a source of competitive differentiation. Indeed, many organizations included AI ethics into their business guidelines; they see challenges in applying AI ethics practices and actions in their business and IT processes.

Requirements for trustworthy and explainable AI suggest the implementation of MLOps, bridging the development and operationalization domains, meaning that the Data Fabric and Data Mesh needs to orchestrate the development of AI models on one platform, for example, on an x86 cluster on-premises or in the public cloud, and the deployment and operationalization for scoring and inferencing on another platform, for example, on IBM zSystems.

¹⁴ See Reference [8] for the IBM IBV survey.

¹⁵ See Reference [9] for more information on trustworthy AI.



Figure 3-10. *Data Fabric/Data Mesh Use Case – MLOps and trustworthy AI*

Specifically, for MLOps and trustworthy AI, this includes data access and data integration within the transactional landscape as well.

Before we explain trustworthy AI, let us provide a simple example of MLOps.¹⁶ The meaning of MLOps is to focus on the automation of tasks specific to ML, such as feature engineering, training and validation, version control, and finally deployment and operationalization.

Figure 3-11 is a high-level architecture overview diagram that depicts an AI development environment on the right side, which is based on IBM Cloud Pak for Data with IBM Watson Studio, running on an x86 cluster on-premises or in a public cloud, and an operational environment on the left side, which is based on IBM Watson Machine Learning for z/OS with CICS as a transactional manager, running on IBM zSystems. AI models that have been developed with IBM Watson Studio need to be exported from the distributed side and imported and put into production on the IBM zSystems side, where the scoring and inferencing takes place for each CICS transaction.

Once AI models have been deployed and moved into production, outcomes of the AI models need to be monitored and measured on a regular basis, which could mean daily, weekly, or at longer intervals.

¹⁶ See more on MLOps in Chapter 9.

This is to gain insight and confidence regarding the continuing business relevance of the AI models.

For instance, data scientists and business users have an interest – and are even motivated by upcoming regulations on trustworthy AI – to detect bias and drift related to drop in data accuracy and precision; they need to furthermore understand shift in fairness or whether quality metrics are deteriorating during the operationalization of the AI models. In addition, business users increasingly require explainability and insight into how AI model outcomes are derived, for instance, in terms of influencing features; they need to gain trust and confidence. Addressing these challenges is summed up with the term *trustworthy AI*.¹⁷

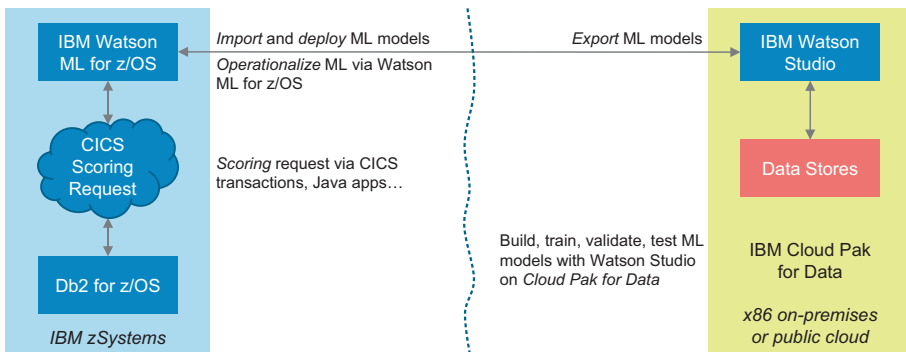


Figure 3-11. Architecture Overview Diagram – ML Operationalization

The following is a short list of trustworthy AI capabilities:

1. **Bias:** Detecting and correcting the fairness (bias) of AI models throughout the entire lifecycle
2. **Drift:** Detecting and correcting drift (drop in accuracy and data consistency)

¹⁷Please, see the section on trustworthy AI in Chapter 5 for more details.

3. **Explainability:** Explaining AI models (transparency of AI model outcome)
4. **Quality metrics:** Measuring AI model quality metrics, including when AI models are operationalized

Implementing trustworthy and explainable AI¹⁸ use cases (or entry points) on a Data Fabric and Data Mesh journey requires distinct functions and features in vendor products. More than that, for instance, if bias or drift has been detected or certain AI model quality metrics are deteriorating while the AI model is in production, corrective actions need to be launched autonomously – at least as far as possible – to address these concerns. This may include triggering a retraining, revalidation, and retesting process for the AI model, including collecting a newer version of the data and performing corresponding data preparation tasks.

Key Takeaways

We conclude this chapter with a few key takeaways as summarized in Table 3-1.

¹⁸See Reference [10] for more information on trustworthy AI and ethics in AI.

Table 3-1. Key Takeaways

# Key Takeaway	High-Level Description
1 Create and implement a data strategy.	Creation and implementation of a data strategy for the organization guides the adoption of the Data Fabric architecture.
2 Building a data catalog is the foundation for every use case.	All use case implementations will first lead to the activities for building an augmented, intelligent data catalog as the foundation of a Data Fabric.
3 Data Fabric architecture for data governance.	Data Fabric architecture can be used to implement an environment for automated and consistent governance.
4 Data Fabric for hybrid cloud strategy.	The requirement of a unified view of the data stored across the hybrid cloud (on-prem and in the cloud) can be solved through a Data Fabric architecture.
5 Data Fabric for 360-degree customer view.	The requirement of a comprehensive view of customers and vendors can be solved through a Data Fabric architecture.
6 Data Fabric for trustworthy AI.	Data Fabric architecture provides the permanent transparency and explainability of data required for trustworthy AI.
7 Trustworthy AI is driven by regulations.	Trustworthy and explainable AI and ethics in AI are driven by forthcoming regulations in the United States and the EU.
8 Define a prioritized use case.	Define a well-scoped use case that delivers value to the business as defined in the data strategy. The discussed use cases are commonly prioritized in organizations.

References

- [1] DDPREU, *Complete guide to GDPR compliance*, <https://gdpr.eu/> (accessed October 16, 2022).
- [2] Eryurek, E., Gilad, U., Lakshmanan, V., Kibunguchy, A., Ashdown, J., *Data Governance: The Definitive Guide*, O'Reilly, 2021, ISBN-13: 978-1492063490.
- [3] IBM, Journey to AI Blog, Kupec, K., *Danske Bank brings teams together with data governance and privacy*, 2021, www.ibm.com/blogs/journey-to-ai/2021/01/danske-bank-brings-teams-together-with-data-governance-and-privacy/ (accessed September 2, 2022).
- [4] IBM, IBM Developer, Mertens, J., *IBM zSystems fundamentals: An introductory Q&A*, 2022, <https://developer.ibm.com/articles/what-is-ibm-z/> (accessed September 2, 2022).
- [5] IBM, Vennan, S., *Hybrid Cloud*, 2021, www.ibm.com/cloud/learn/hybrid-cloud (accessed September 2, 2022).
- [6] Forbes, Davenport, T., *Is The Cloud Slower For Analytical Insights?*, 2022, www.forbes.com/sites/tomdavenport/2022/01/21/is-the-cloud-slower-for-analytical-insights/?sh=56df52d86dca (accessed September 2, 2022).

- [7] McKinsey, Kressmann, F., Ehrlich, O., Lehmann, S., *The customer insights function is ripe for a boost*, 2019, www.mckinsey.com/business-functions/growth-marketing-and-sales/solutions/periscope/our-insights/articles/the-customer-insights-function-is-ripe-for-a-boost (accessed September 2, 2022).
- [8] IBM, IBM Institute for Business Value, *The business value of AI*, www.ibm.com/thought-leadership/institute-business-value/report/ai-value-pandemic (accessed September 2, 2022).
- [9] Springer Link, Pery, A., Rafiei, M., Simon, M., van der Aalst, W. M. P., *Trustworthy Artificial Intelligence and Process Mining: Challenges and Opportunities*, 2022, https://link.springer.com/chapter/10.1007/978-3-030-98581-3_29 (accessed September 2, 2022).
- [10] Ammanath, B., *Trustworthy AI: A Business Guide for Navigating Trust and Ethics in AI*, Wiley, 2022, ISBN-13: 978-1119867920.

CHAPTER 4

Data Fabric and Data Mesh Business Benefits

This chapter dives into business drivers and pain points that we hear in our conversations with enterprises. The business benefits of creating a Data Fabric architecture and also implementing a Data Mesh solution are discussed from the perspective of the technical, primarily data engineering team as well as the business teams consuming the data. We discuss a needed cultural shift related to managing data in an organization that looks at holistic data ownership of data source owners, data engineers, and data consumers.

Introduction

After defining a Data Fabric, also as an underlying data architecture for a Data Mesh implementation, in Chapter 2, let us revisit the key business drivers that justify significant investment into implementing such a new data architecture in the organization.

The overarching motivation or driver is simple: get more value out of the organization's data to keep up with the industry or, better, achieve a competitive advantage.

One example is the worldwide, disruptive nature of the pandemic¹ that created never seen business challenges in all areas. Identifying and solving supply chain problems quickly and providing all business functions online, without need for personal interaction, are just few resulting requirements that caused organizations to revisit their current IT landscape and define new IT priorities and accelerate related IT modernization projects.

Another example is the globally acknowledged climate change² that causes more frequent, natural disasters. Insurance companies need the ability for agile and complex analysis of enterprise and publicly available data to estimate a policy risk accurately to ensure the company can fulfil its obligations and still create profit. Competitively priced insurance policies have a direct effect on the market share of an insurance company.

Such business needs drive application landscape changes, such as applications may move to a cloud service or applications adopt a new application architecture that drives a new data architecture; it leads to new requirements for data management and data consumption in an organization, which has profound ramifications for a sustainable and disruption-proof implementation of both concepts.

Business Requirements and Pain Points for Data Management and Consumption

The business drivers need to be mapped to key drivers for a data architecture. To better understand the technical and business value of a Data Fabric architecture and a Data Mesh solution, let us again travel back in time and summarize the key drivers for implementing data architectures.

¹ See References [1] and [2] for more information on the business impact of COVID-19.

² See Reference [3] for more information on the impact of climate change on insurers.

Figure 4-1 lists the commonly voiced pain points in an organization for a given data architecture.





mid 1970 Relational DB 	mid 1990 Data Mgmt for DWH 	2010 Big Data / Data Lake 	2019+ Data Fabric / Data Mesh 
Need to understand structure of the data	Need data of multiple data sources in one place	Need to keep up with big data demands	Need to locate relevant data faster
Need to apply logic to the data via standard API	Need to run analytics without impacting the operational system	Need to support semi- and un-structured data	Need consumable data
	Need to accelerate business reporting	Need to easily include additional data	Need consistent data and AI governance

Figure 4-1. Key Drivers for Data Architecture Evolution

The key drivers for both concepts are also expressed in statements and questions that may sound familiar:

- **Multiple locations:** The same data is stored in multiple locations, often in a different format and accessible via different methods, for example, SQL, NoSQL, or REST API.
- **Data and AI asset owner:** I have a question about some data and AI assets, but I don't know who owns these assets and whom to reach out to. Examples are: How do I get access to the data and AI assets in the context of a concrete business question? To whom should I forward requests regarding authentication and authorization?
- **Knowledge catalog:** I need a comprehensive overview of data and AI assets, which are relevant and available to me, which should also include an overview of available data products including their specifications.

- **Data quality:** Can I be confident regarding data quality – the effort to remediate data issues is high in terms of manpower, time, and money?
- **Currency and consistency:** I need more current and consistent data and AI assets: Who is responsible for this? Is there an owner for specific data and AI assets?
- **Impact of changes:** What is the downstream impact to my data if changes occur? Are there tools that I can rely on providing accurate data quality measures?

The statements highlight challenges in knowledge about impact of data-producing applications, insight into data pipelines or data lineage, and knowledge about data for data consumers as well as data availability and related automation and governance processes.

Those challenges³ directly impact the ability to deliver data at the speed requested by the business with the available skill. Even with more data engineers, the demand for data and data products consistently outgrows the ability to deliver with existing data architecture capabilities.

The benefits of both concepts can be directly derived from solving the key challenges that exist today after organizations implemented the previous data architectures, especially traditional EDW and even data lakehouse architectures. It is important to point out that a Data Fabric architecture and Data Mesh solution cannot deliver on the full value without considering cultural and organizational changes as well; a more holistic approach is necessary.

Figure 4-2 categorizes Figure 1-3 of Chapter 1 by responsible roles. On the left are the owners of applications that produce data. In the middle are the data engineers who make this data available to the data consumer in the format best suitable to the consuming use case. Until now, there is a

³See Reference [4] for more information on data-related issues and opportunities.

clear division in responsibility and limited collaboration across roles that does not help streamline the data pipeline integration.

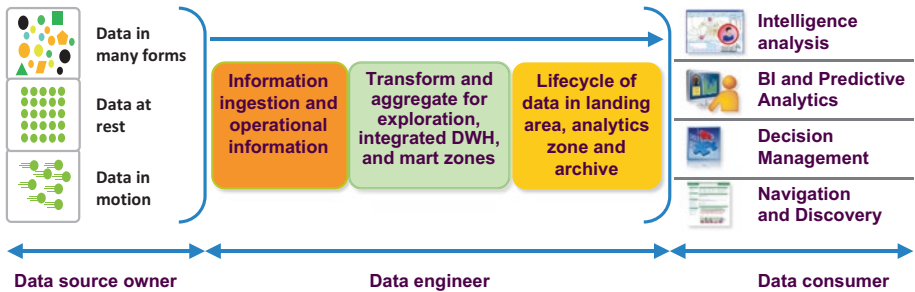


Figure 4-2. *Changing Scope of a Data Pipeline*

Let's look at an example to illustrate. Core applications constantly change data. In the past those transactional applications were so optimized for response time and scalability that they did not keep history of the data changes. With today's processing power, regulatory requirements, and database capabilities, it is much more efficient to just keep history when data is changed in the core application. Still, most data changes are recreated as part of the acquire-transform-manage data pipeline, making data pipelines more complex and typically causing bigger than 50% of the pipeline processing.

The goal of both concepts alike is to lower the needed skill level to discover, access, prepare, and consume data through easy access to metadata in a business context and intelligently automate the main tasks of a data engineer. By breaking down the silos of responsibilities of a data source owner, data engineer, and data consumer as well as removing the dependency on the data engineer for data consumption, business priorities can be implemented without the data engineer support capacity as a bottleneck.

Benefits of a Data Fabric and Data Mesh for Technical Teams Managing Data

Chapter 3 described activities to implement the foundation and entry points for a Data Fabric architecture and Data Mesh solution. The following are several benefits for technical teams, for example, data engineers, resulting from implementation and changes in responsibilities, responding to the key drivers⁴:

- **Faster and simpler data delivery process:** Previously, the data engineer needed to investigate applicable data sources manually; developed data pipelines, for example, coded ETL jobs; and informed the data consumer how to consume the derived (transformed and aggregated) data. Depending on the business priority, this effort took weeks to months. By creating a knowledge catalog and using annotation and analytics capabilities, applicable data sources can be identified easier, and tools can be used to automate data integration for faster data delivery. Breaking down silos between the data source owner and data engineer, the data engineer moved from being the perceived *owner* of the data for the data consumer to being the enabler for the data source owner to get the data faster to the consumer. This way, the data source owner is motivated to look beyond the functional and non-functional requirements of the transactional application and include requirements for better integration of the data source into the Data Fabric and Data Mesh such as providing the metadata and business glossary to the knowledge catalog.

⁴See References [5] and [6] for benefits and value propositions of a Data Fabric approach.

- **Reduction in efforts for data access management:**
Introduction of global data governance rules, policies, and processes across data and AI assets that are managed via the knowledge catalog and automation by global policy enforcement significantly reduce the effort and time to give access to the data consumer. They also allow the LoB to become part of data product design, implementation, and ownership.
- **Decreased effort to maintain data quality standards:**
The first milestone of implementing a Data Fabric and Data Mesh is cataloging the existing data sources and existing data pipelines. A key capability of a knowledge catalog is the quality assessment of the data source. The data quality is expressed by data completeness and data correctness and currency of data. For example, a telephone number is part of a customer record. If the telephone number is only maintained in 60% of the records, an analysis by area code would not correctly represent the true customer distribution. Another example is that data in a database needed to be expanded and therefore was moved into another column. The ETL process was not correctly updated and still read data from the old column that might not have been removed. Such a problem could make the derived data unusable. Data consumers may not have been aware of such a data quality problem.⁵ After assessment of the data quality of the original data source, including data lineage makes the data

⁵ Please, refer to Chapter 8, where we elaborate on applying AI methods to address these and related issues.

flow consumable by tools. What is otherwise buried in thousands of transformation jobs becomes visible to the data engineer and creates an opportunity to reduce the number of copies. Without question, technology is available today to copy large amounts of data from a source to a target. In practice, it is a huge operational challenge to keep copies of data consistent over a longer period. Therefore, reducing copies of data always increases the quality and therefore the trust in data.

- **Reduced infrastructure and storage cost:** The reduction in data copies and consolidation of data management tools and the shift in responsibility of primarily the data engineer directly lead to reduced infrastructure and storage cost, which is nevertheless an essential goal for the CDO and CIO and their corresponding organizations.

Figure 4-3 depicts the four business outcomes for technical teams as it relates to specific value drivers and corresponding measurements or KPIs. This list of value drivers and KPIs is certainly incomplete and serves more as an illustration. Nevertheless, it represents an entry point or a basic template in measuring IT-related business outcomes derived from technical teams implementing a Data Fabric architecture or a Data Mesh solution.

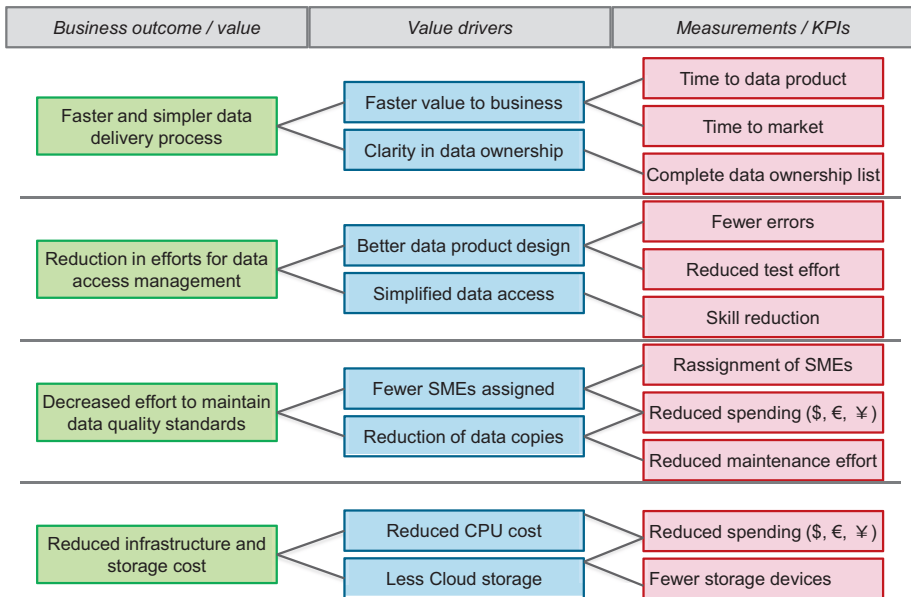


Figure 4-3. Value Proposition for Technical Teams

Of course, this method needs to be further refined and completed. We encourage you to use this template for your initiatives, as it enables you to develop a comprehensive and measurable value proposition, which is essential to receive required funding.

Benefits of a Data Fabric and Data Mesh for Business Teams Consuming Data

The technical benefits of a Data Fabric and Data Mesh as listed previously can be directly mapped into benefits for business teams.⁶ These business outcomes relate to implementing both concepts:

⁶ See Reference [7] for business benefits of a Data Fabric.

- **Self-service data shopping:** Self-service is implemented in many areas. Hotel reservations are done through portals instead of calling the hotel. Product checkout in supermarkets is done through self-service pay stations. Therefore, the expectation of a business team member to shop for the organization's data and AI assets in a company marketplace portal does not sound so far-fetched. A Data Fabric with its capabilities provides the plumbing for implementing such a data and AI asset marketplace, which can provide a frustration-free full self-service data shopping experience for business users. Automatic enforcement of data and AI governance rules, policies, and processes ensures that business owners only have access to data (clear or masked) that they are entitled to see and process.
- **Compliance and security:** Automatic enforcement of data governance rules and processes is not just the prerequisite for a data marketplace. First, data governance rules, policies, and processes need to be consistently defined within an organization to automatically enforce them. This is a huge improvement compared with today where compliance definitions and tools vary by business domain, applications, or even responsible people and every platform and system has its proprietary way to define and ensure data security. So despite making the data and AI available for full business analytics utilization, the data compliance and security is actually increased.
- **Faster and more accurate insight:** By lowering the skill level for data business consumers and simplifying data access through a self-service marketplace, the

data consumer gains faster and more business-relevant insight. Additionally, data quality, completeness, and timeliness are significantly improved by applying intelligent and automated data management as well as documented lineage for data and AI assets, leading also to a more accurate and business-relevant insight.

- **Spent time on analyzing data:** Data consumers, especially data scientists today, spend most of their time finding and preparing data instead of focusing on analyzing the data or building AI models. This ratio of data exploration and preparation tasks to the actual time needed to develop, for instance, AI models (including training, validation, test, etc.) is typically in the range of 80/20. This faster, self-service access to relevant data and AI assets allows data consumers, especially data scientists, to focus their time on analyzing the data, also contributing to faster insight into business questions.

In summary, a Data Mesh⁷ data marketplace based on a Data Fabric architecture allows the data consumer to discover and access high-quality data faster in self-service mode (enabling a data marketplace with data-as-a-product), removing potentially long delays waiting on the data engineer⁸ to prepare data and to build ETL stages or pipelines for business consumption.

As a result, relevant and accurate insight into business challenges is gained faster and can be acted on with increased agility.

Figure 4-4 depicts the four business outcomes for business teams as it relates to specific value drivers and corresponding measurements or KPIs. Again, like the list of value drivers and KPIs for the technical teams

⁷ See Reference [8] for more benefits of a Data Mesh.

⁸ See Reference [9] for more information regarding challenges addressed by a Data Mesh.

managing data and AI assets, this list of value drivers and KPIs for the business teams consuming data and AI assets is incomplete as well.

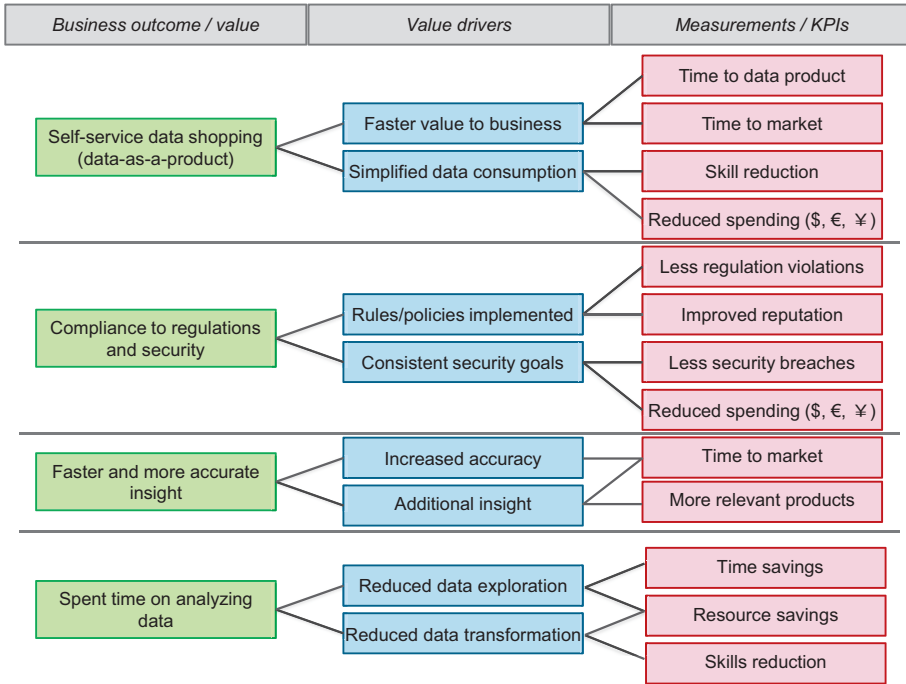


Figure 4-4. Value Proposition for Business Teams

Nevertheless, this approach also represents a meaningful entry point in measuring the value proposition derived from business teams implementing any of the two concepts.

The two lists of value propositions⁹ derived from technical and business teams can and should be refined and extended according to chosen entry points and specific use case scenarios. For instance, the KPIs could be related and further detailed out to initiatives and actions, features, dependencies, and acceptance criteria based on your specific project requirements.

⁹ See Reference [10] for more information on building a business value proposition.

Key Takeaways

We conclude this chapter with a few key takeaways as summarized in Table 4-1.

Table 4-1. *Key Takeaways*

#	Key Takeaway	High-Level Description
1	Motivation to get more value out of the organization's data.	The overarching motivation or driver is simple: get more value out of the organization's data to keep up with the industry or even better achieve a competitive advantage.
2	Key drivers for a Data Fabric architecture and Data Mesh solution.	The key drivers for a Data Fabric or Data Mesh are the need to locate data faster, to simplify access and consumption of data, and to apply consistent data and AI governance.
3	Need for more data grows faster than data delivery.	The demand to gain actionable insight from data in a more agile way and to produce data products consistently outgrows the ability to deliver with existing data architecture capabilities.
4	Responsibility of the data engineer changes.	The data engineer moves from being the perceived <i>owner</i> of the data to being the enabler for the data source owners.
5	Data Fabric and Data Mesh capabilities correspond to key drivers.	The Data Fabric and Data Mesh capabilities deliver noticeable value by directly addressing the key business and IT drivers, specifically addressing tasks to be performed with increased automation and in a self-service fashion.
6	Faster and more accurate insight.	Data Fabric and Data Mesh capabilities enable a business analyst to gain accurate insight into business challenges faster.
7	A business outcome should be refined on a project-specific level.	Business outcomes that are derived from technical and business teams should be depicted in terms of value drivers and measurements (KPIs).

References

- [1] McKinsey & Company, *COVID-19: Implications for business in 2021*, 2021, www.mckinsey.com/business-functions/risk-and-resilience/our-insights/covid-19-implications-for-business-2021 (accessed August 25, 2022).
- [2] Akkaya, B., Jermittiparsert, K., Abid Malik, M., Kocyigit, Y., *Emerging Trends in and Strategies for Industry 4.0 During and Beyond Covid-19*, Sciendo, 2021, ISBN-13 : 978-8366675384.
- [3] Reuters, *Climate change is hurting insurers, report says*, 2022, www.reuters.com/business/finance/climate-change-is-hurting-insurers-report-2022-05-17/ (accessed August 25, 2022).
- [4] Accenture, *Data is the new capital*, www.accenture.com/_acnmedia/PDF-157/Accenture-Data-Is-The-New-Capital-POV2.pdf (accessed August 25, 2022).
- [5] eWeek, *IBM's Daniel Hernandez on AI and Data Fabric*, 2021, www.eweek.com/big-data-and-analytics/ibm-daniel-hernandez-ai-data-fabric/ (accessed August 25, 2022).
- [6] Teradata, Dave, P., *Seven Benefits of a Powerful Data Fabric*, 2022, www.teradata.de/Blogs/Seven-Benefits-of-a-Powerful-Data-Fabric (accessed August 25, 2022).
- [7] IBM, *Data fabric architecture delivers instant benefits*, www.ibm.com/downloads/cas/V4QYOAPR (accessed August 25, 2022).

- [8] Radiant Digital, *Benefits of Data Mesh*, <https://radiant.digital/benefits-of-data-mesh/> (accessed August 25, 2022).
- [9] IBM, IBM Developer, Das, S., Sharma, N., *An introduction to data mesh*, 2022, <https://developer.ibm.com/articles/data-mesh-a-peek-into-this-new-paradigm/> (accessed August 25, 2022).
- [10] Harvard Business School, Business Insights, *How To Create An Effective Value Proposition*, <https://online.hbs.edu/blog/post/creating-a-value-proposition> (accessed August 28, 2022).

PART II

Key Data Fabric and Data Mesh Capabilities and Concepts

CHAPTER 5

Key Data Fabric and Data Mesh Capabilities

A state-of-the-art Data Fabric architecture and Data Mesh solution is unquestionably linked to the knowledge catalog¹ as one of its prime components. What differentiates a modern knowledge catalog from traditional ones are AI-infused capabilities to automate tasks and to provide self-service capabilities. AI is without dispute an inevitable domain that characterizes a modern Data Fabric and Data Mesh. Infusing AI generates additional added value specifically for business users, such as delivering trustworthy AI.

Dealing with diverse and heterogeneous source data, which is typically distributed across various organizations and systems, requires intelligent information integration concepts that go far above and beyond traditional data federation or virtualization techniques.

¹ See References [1] and [2] for more information on knowledge catalogs and metadata management solutions available by different vendors.

Intelligent information integration needs to include AI artefacts (for instance, AI models, Jupyter notebooks, ETL stages and pipelines, dashboards, physical and logical data models), automating the integration of these artefacts for timely insights and business decisions.

Introduction

This chapter introduces key capabilities for both concepts, such as a knowledge catalog with active metadata, data curation, rules and policy management, semantic knowledge graphs (semantic networks), and self-service capabilities. A subsection elaborates on trustworthy AI, which has been described in Chapter 3 as one of the four key use case scenarios or entry points. It also includes activation of the digital exhaust, where we elaborate on pattern recognition and correlation discovery from the digital exhaust to augment and operationalize this insight into the Data Fabric and Data Mesh.

Finally, we elaborate very briefly on intelligent information integration concepts as an integral part for both concepts. Although we address intelligent information integration approaches, the focus of this chapter is nevertheless on the knowledge catalog and trustworthy AI topics.

Knowledge Catalog

As we have already seen in Chapter 2, the knowledge catalog is one of the most essential components for a Data Fabric architecture and Data Mesh solution and is inevitably associated with enabling execution of relevant tasks, such as cataloging all your assets, generating active metadata, performing data curation tasks, implementing self-service capabilities, providing a knowledge graph (semantic networks), enforcing

local and global data rules and policies, and enforcing unified data and AI governance. To investigate additional facets, we feature the knowledge catalog in several additional chapters, for example, Chapters 13, 14, and 15.

Let's start with a description of what we mean by metadata and by generating active metadata.

Active Metadata

The treatment of this topic requires us to briefly revisit what we mean by metadata. Metadata is simply data that describes data; it is typically categorized into business metadata, technical metadata, and operational metadata:

1. **Business metadata:** Provides business-relevant context to data and AI artefacts. It describes business aspects, provides meaning to the data and AI assets, and is primarily added and consumed by business users. Business metadata enables LoB users to use an organization- or enterprise-wide consistent language, for instance, by maintaining industry- or domain-specific business glossaries. Business metadata helps bridge the chasm between IT and business teams. Examples of business metadata are annotations to business reports or AI model outcomes, business glossaries, and information about federal and international laws or regulations relevant to business operations and how to manage data and AI assets, ownership of AI assets, security and privacy levels, etc.

2. **Technical metadata:** Is primarily concerned about the organization of data and AI artefacts; it describes its structure and physical attributes, location of AI assets, quality metrics, connections available, access methods, etc. Examples of technical metadata are database table and column names, indexes, data types, names and attributes of AI models (i.e., ML/DL algorithms used), Jupyter notebooks (i.e., languages used), logical and physical data models, XML schema definitions, stored procedure definitions, BI artefacts, etc. Technical metadata enables data professionals to give adequate attention to how to manage, access, transform, and consume data and AI assets.
3. **Operational metadata:** Often called process metadata, is concerned about the creation and transformation of AI assets, including when they were updated or deleted by whom and why; it describes events and processes that affect AI assets. Examples of operational metadata are ETL stages, Insert/Update/Delete (I/U/D) logs, pipelines, job execution logs including runtime parameters, SQL query execution logs (i.e., access path information), AI asset usage logs, etc. In combination with business and technical metadata, process metadata can, for instance, be used to visualize data lineage or data provenance.

Before moving ahead toward active metadata, we provide another angle to metadata, which became popular in recent years – social metadata. This is often mistaken for user-generated content. However, social metadata is data about social data and its originator, for instance, annotations, descriptions, trust scores, relevance, completeness, etc.

Let us now set ourselves the task to explain what we mean by active metadata and why it matters. According to Gartner,² “active metadata management is an emerging set of capabilities across multiple data management markets resulting from continuous metadata management innovation.” In our view, active metadata is AI/ML-augmented metadata, meaning it is generated by applying AI/ML techniques to metadata to gain additional actionable insight, which can be used to further automate Data Fabric or Data Mesh tasks. In other words, active metadata is presumably derived from traditional metadata (business, technical, and operational) via AI/ML capabilities to enable automated consumption by Data Mesh-relevant applications and systems. In addition, users may be allowed to intercept the execution of automated tasks to validate, adjust, or even reject those automated tasks; recommendations and suggestions may have to be evaluated prior to implementing them. AI/ML techniques are particularly well suited to further augment automation and simplification of essential tasks.

² See Reference [3] for Gartner’s market guide for active metadata management.

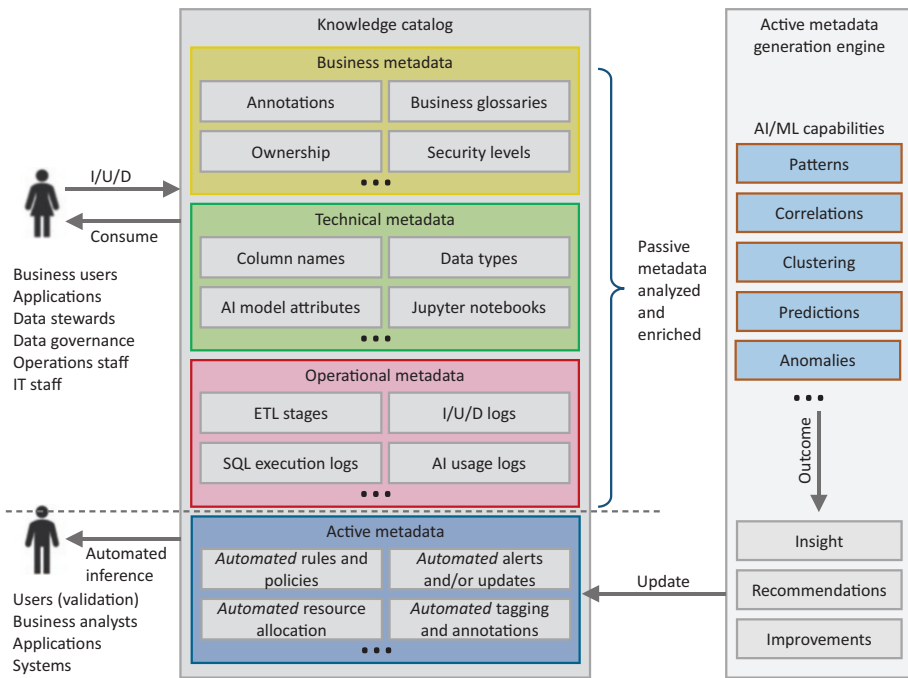


Figure 5-1. Active Metadata

Figure 5-1 depicts the key components to generate and consume active metadata; it features prominently the knowledge catalog with its business metadata, technical metadata, and operational metadata in the middle, which often is referred to as passive metadata, and the active metadata generation engine with its AI/ML capabilities on the right side.

The active metadata generation engine applies the AI/ML capabilities, such as pattern and correlation discovery, clustering and prediction, and anomaly detection, holistically to the entirety of passive metadata (business, technical, and operational) to derive to actionable insight and recommendations for improvements or simplifications of relevant tasks. The primary objective of active metadata is geared toward automated inferencing into Data Mesh applications, systems, or components. Nevertheless, business analysts, data stewards, data governance personnel, and data scientists may benefit from active metadata as well by gaining

additional insight, for instance, by receiving alerts related to shifting business outcomes, violations to regulatory compliance, degrading data quality metrics, or recommendations to expedite data exploration tasks.

Semantic enrichment and activating the digital exhaust from Data Fabric or Data Mesh processes, which we further explore in Chapter 7, are additional examples related to generating active metadata.

Data Curation

We refer to data curation as a set of processes to add data and AI assets to a knowledge catalog, enriching them by assigning classifications, data classes, and business terms, automatically generating and assigning asset rules and policies, providing a self-service way to discover and share enterprise assets, and analyzing, ensuring, and improving quality of the assets. Data stewards and data engineers curate data and AI assets by intelligent cataloging and automated metadata management, including importing and enriching metadata, preparing the assets, enriching the assets by assigning governance artefacts, and publishing the assets into a knowledge catalog.

Intelligent cataloging and automated metadata management are further examined in Chapters 13 and 14. In this section, we provide a high-level overview of the key data curation tasks.

The following is a brief description of key data curation tasks:

1. **Create metadata:** Automatically generate metadata of relevant data and AI assets, for example, source data tables, and capture the metadata in the knowledge catalog.
2. **Assign business terms:** Discover business terms based on available business glossaries or propose new business terms, for instance, based on available taxonomies, and assign these terms to corresponding assets.

3. **Analyze asset quality:** Perform profiling of data and AI assets, conduct quality assessments, and propose recommendations for quality improvements to be accepted or rejected by the data steward.
4. **Refine assets:** Prepare, transform, and improve quality of data and AI assets, for instance, based on data rules and policies, including automatically generated recommendations to be accepted or rejected by the data steward.
5. **Enriching metadata:** Assign data classes, perform semantic enrichment (e.g., generate semantic knowledge graphs) and data classification, add tags and annotations, etc.
6. **Assigning governance artefacts:** For instance, automatically generate rules and policies according to governance and regulation imperatives.

Most of these data curation tasks, for example, semantic enrichment and automatically generating and assigning asset rules and policies, may very well require applying AI/ML techniques. We refer to AI/ML-infused data curation as *advanced or intelligent data curation*.³

Let us now proceed to the exciting topic of semantic knowledge graphs.

Semantic Knowledge Graphs

A knowledge graph – often referred to as a semantic network – is a directed labeled graph, which comprises three main components: nodes, edges, and labels.

³See Reference [4] for more information on data curation.

Figure 5-2 is a simple knowledge graph depicting the relationship of a customer *A* to a vendor *B*: *A* is a customer of *B*. Depending on the application area, *A* is referred to as *subject*, *B* is referred to as *object*, and the label is referred to as *predicate*. A directed graph where the nodes are classes and subclasses of objects of a particular domain (e.g., car, engine, cylinder, camshaft, etc.), and the edges describe the subclass relationship, is called a taxonomy. The conceptual compatibility of knowledge graphs to active metadata seems to be obvious: knowledge graphs represent enriched metadata that is derived by analyzing passive metadata and by applying AI/ML capabilities to gain additional non-obvious insight. For instance, knowledge graphs should be created by visualizing *ownership-asset* relationships and enriching the knowledge graph with detected anomalies in those relationships or non-obvious correlations between some data and AI assets.

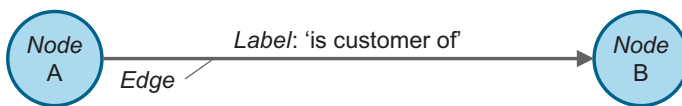


Figure 5-2. Knowledge Graph

We may call those AI/ML-enriched knowledge graphs as *semantic knowledge graphs*.⁴ They are stored in graph databases (GDBs) as a logical component of a knowledge catalog. Examples of GDBs are Amazon Neptune,⁵ Neo4j,⁶ and IBM Graph,⁷ among others. Figure 5-3 is an example of a semantic knowledge graph that visualizes an AI model in

⁴ See Reference [5] for more information on semantic knowledge graphs in the context of a Data Fabric.

⁵ See Reference [6] for more information on Amazon Neptune.

⁶ See Reference [7] for more information on Neo4j.

⁷ See Reference [8] for more information on IBM Graph.

context. It characterizes the AI model in terms of its type and subtype, for example, the AI model can be of type *machine learning* with an associated subtype of a *binary classification model*, as it is very common for fraud prevention models.

As you can easily imagine, the knowledge graph could also visualize additional characteristics of the AI model, such as when the model was trained and deployed, when and how often it was retrained, and related AI artefacts, for example, Jupyter notebooks, pipelines, required features, etc.

The knowledge graph also depicts the personas, which have developed the AI model and its current ownership, including their corresponding roles and organizations they belong to. In our example *John*, who is a data scientist in the development organization, has developed the model, and *Joan*, who is a business owner in the fraud department, has business responsibility for the AI model. Additional personas with their corresponding roles and organizations could be included as well, for example, linking the model to the AI governance and IT operational organizations.

A knowledge graph needs to give adequate attention to key quality metrics that need to be regularly measured during the operationalization phase of the AI model. This is exemplarily depicted on the left side of Figure 5-3, where you see the area under the ROC (Receiver Operating Characteristic) and PR (Precision-Recall) curves⁸ and F1 measure, among others.

⁸We further examine these and other quality metrics, for example, the F1 measure, in the section on trustworthy AI later in this chapter.

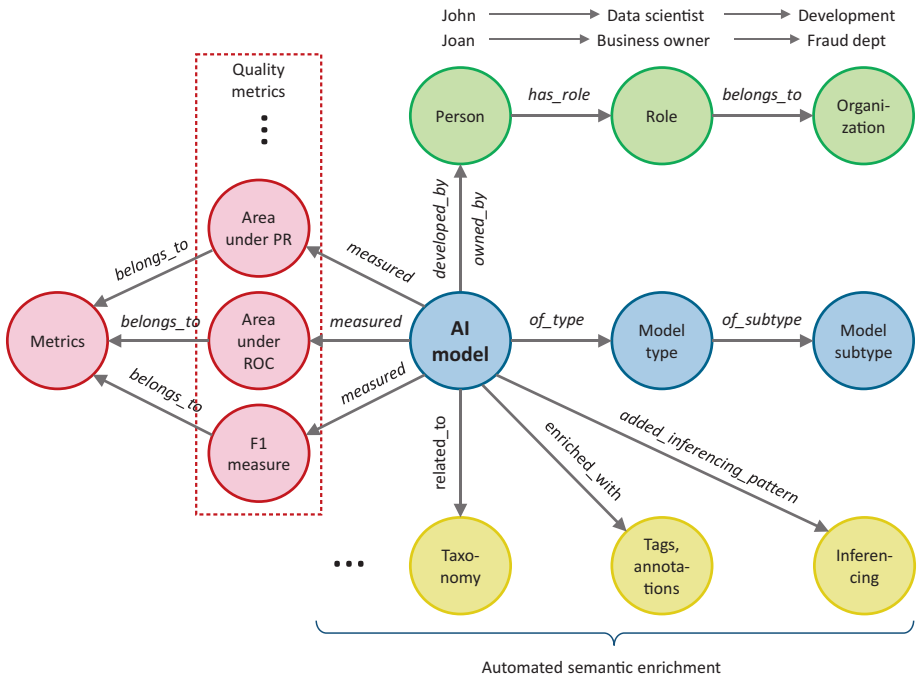


Figure 5-3. *Semantic Knowledge Graph Example*

Additional metrics and insight into the model could be visualized by the knowledge graph, such as bias and drift, including the corresponding times when those metrics were measured. Additional insight could be related to the explainability of the AI model, for example, based on automated discovery of features influencing scoring. Analyzing passive metadata available in the knowledge catalog – in combination with AI/ML techniques – enables enrichment of a knowledge graph of AI models.

Metadata of potentially unprecedented diversity in terms of pervasive business glossaries, structured and unstructured data, and particularly operational metadata, entangled with external data, such as industry- or domain-specific taxonomies and ontologies, makes AI/ML particularly well suited to enrich the knowledge graph with semantic knowledge. In our semantic knowledge graph as depicted in Figure 5-3, we have indicated just a few examples of automated semantic enrichments. The AI

model could, for instance, be related to a specific taxonomy or ontology, including suggestions to adjust and refine the feature set by embracing specific terms.

Furthermore, the AI model could be tagged; annotations could be added to make it more explainable to business users. Finally, insight into the scoring and inferencing patterns could shed light on the business relevance of the AI model, indicating, for instance, the number of prevented fraudulent transactions (true positive rate, TPR) and names of the calling applications or components, including corresponding statistics that may even rank the AI model regarding its business relevance to other AI models. As we have described in the “Data Product” section of Chapter 2, a data product consists of a set of semantically related data and AI assets. Therefore, this automated semantic enrichment capability is essential to create a semantic knowledge graph, which can be used to build data products.

In Chapter 7, we further examine the semantic enrichment process that provides *intelligent* and *automated* enrichment to contextualize assets with semantic knowledge mainly by using external data sources, for instance, external knowledge graphs. Semantic enrichment in tandem with knowledge graphs and other input, for example, domain- or industry-specific taxonomies or ontologies, constitutes a key capability of an intelligent approach.

Although taxonomies, ontologies, and knowledge graphs are often used synonymously and these are related concepts, there are differences: Taxonomies can be viewed as a subset of knowledge graphs, providing structure to objects that belong to a particular domain, where the nodes are classes and subclasses of these objects. Ontologies are concerned about formal naming conventions, including definition of types, properties, and correlation of entities, where the entities are depicted as nodes and the relationships as edges.

Let us move on to discuss self-service capabilities.

Self-Service Capabilities

As we have seen in Chapter 2, a state-of-the-art Data Fabric architecture and Data Mesh solution is inevitably affected by its self-service capabilities. However, we need to shed more light on what is really meant by *self-service*. Self-service is the digitalization of processes and tasks that would otherwise require intervention and support by personas with other, non-business-related roles and responsibilities. In simple words, self-service capabilities of a Data Mesh solution enable a business user to perform business tasks, for example, generating a report, discovering and accessing data, understanding the quality and trustworthiness of AI assets, etc. without support and dependencies from IT or other business units. Since today's dependency of business users from IT organizations represents a challenge in producing data products in a more agile fashion, self-service capabilities are vital for a Data Mesh solution.

The following Table 5-1 describes the most essential self-service capabilities for both concepts.

Table 5-1. *Essential Self-Service Capabilities of a Data Fabric and Data Mesh*

#	Capabilities	High-Level Description
1	Access to AI assets without additional credentials	The knowledge catalog includes metadata of AI assets, which are accessible by users (based on their role and responsibility), for example, business users, data scientists, AI governance personnel, data engineers, etc., without the need to request additional credentials.
2	Information regarding access methods	All metadata in the knowledge catalog should include information regarding the access methods of the corresponding AI assets, for example, SQL, NoSQL, REST APIs, Java SDK, .NET SDK, etc.

(continued)

Table 5-1. *(continued)*

#	Capabilities	High-Level Description
3	Information about related AI assets	The knowledge catalog should include information of related AI assets, providing a holistic and business-relevant picture of particular AI assets, including information of governance artefacts, Jupyter notebooks, etc.
4	Generating active metadata	AI/ML-augmented metadata, for instance, triggering automated resource allocations, tagging and annotations, and automated allocation of resources for peak AI asset consumption intervals.
5	Semantic search capabilities	AI/ML-based search to discover, for instance, non-obvious correlation among AI assets or recommendations concerning additional AI assets that are relevant for a particular business purpose.
6	GUI-based data exploration and preparation	Easy-to-use, GUI-based data exploration and preparation tools that can be used by non-IT personas, such as business users, business analysts, data governance officers, etc., in a self-explanatory way.
7	Performing data curation tasks	Some of the data curation tasks, which we have listed earlier in this chapter, need to be performed by business users without intervention or assistance by subject matter experts or IT personnel, for example, data quality assessments and data refinement tasks.

Some of these capabilities go far above and beyond what a knowledge catalog can provide. For instance, a GUI-based data exploration and preparation tool may leverage the data in the knowledge catalog. Nevertheless, it requires a separate set of tools that are part of the Data Fabric architecture.

As you can easily see from this table, self-service capabilities enable business organizations to entertain a data marketplace with *data-as-a-product* and *shopping-for-data* experience.

Trustworthy AI

This section examines one of the four key pillars or use cases of a Data Fabric or Data Mesh, which we have introduced in Chapter 3, namely, trustworthy AI. Once AI models are prepared and deployed for scoring, outcomes of the AI models need to be regularly monitored and measured to gain insight and confidence regarding the continuing business relevance. Data scientists and business users alike need to understand shifts in fairness or bias, drift concerning drop in accuracy or data consistency, and deteriorating quality metrics during the operationalization of the AI models. In addition, business users increasingly require explainability and insight into how AI model outcomes are derived, for instance, in terms of influencing features; they need to gain trust and confidence. Tackling these challenges is subsumed under the term *trustworthy AI*.

Trustworthy AI, however, is not only a clever acronym: in recent years, it became a prominent theme embraced by regulatory bodies and government agencies,⁹ also entangling trust in AI with ethics, transparency, and explainability of AI. Given these regulatory guidelines and imperatives, enterprises must increasingly contend with these challenges. In this section, however, we provide concrete examples regarding the technical aspects of trustworthy AI, leaving aside the ethical, lawful, and societal aspects.¹⁰

⁹ See References [9] and [10] for details on guidelines of the European Commission and the US Department of State regarding trustworthy AI.

¹⁰ See Reference [11] for more information on trustworthy AI, including social aspects.

Introduction

A thorough treatment of trustworthy AI could certainly fill an entire book by itself; we therefore must consciously sample out the vital issues that are most instrumental for both concepts.

The following are the key issues – or rather challenges – that we consider as essential in the context of a Data Fabric and Data Mesh:

1. **Model fairness**¹¹ (bias): Deals with measuring and managing desirable or undesirable preferences for certain values of chosen features determining the outcome of an AI model. This is especially important if these features are gender, age, religion, race, nationality, etc.
2. **Drift detection**: Measures two kinds of drift, (a) the ML model drift, which measures the drop in accuracy and drop in data consistency by comparing accuracy during runtime with the accuracy during training, and (b) data drift, which is comparing key characteristics of the dataset, for example, value distributions of key features for the ML model used for training with the dataset during runtime.
3. **Explainability**: Provides insight and transparency of AI model outcomes to business users with no or limited data science skills and allowing for *what-if* scenarios.

¹¹ We are using the terms *fairness* and *bias* interchangeably. However, fairness has a more social connotation, whereas bias is used more in a mathematical context.

4. **Model quality metrics:** Metrics such as precision, areas under the ROC and PR curves, true positive rate (TPR), true negative rate (TNR), etc. need to be collected. Measuring these model quality metrics during the entire lifecycle and allowing corrective actions to be taken is an essential feature for both concepts.

We examine these four challenges of trustworthy AI in the subsequent subsections. Let us begin with model fairness (bias).

Model Fairness

As stated previously, bias is concerned about preferences regarding favorable AI model outcomes for specific feature values. For instance, a binary classification model that predicts acceptance for a marketing campaign may include gender (e.g., female and male) as a feature input. In preparation for deployment and operationalization of this ML model, initial bias needs to be measured to understand, for instance, favorable outcomes regarding females in relationship to males.

Once the ML model is in production, changes in bias need to be monitored at regular intervals to enable data scientists to act once bias falls below a defined threshold.

In this subsection, we examine an example of how bias can be measured and monitored,¹² by using the above-mentioned ML model with gender (female and male) as a feature, where we intend to monitor bias for females.

¹² See References [12] for more details on IBM Watson OpenScale, which this example is based on.

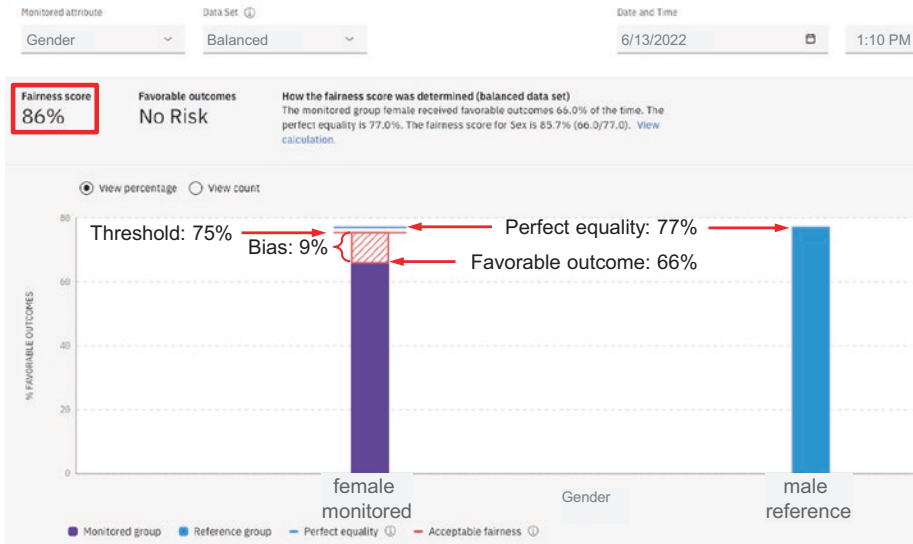


Figure 5-4. Fairness/Bias Score for Gender

Initial bias is described by calculating three metrics – perfect equality, favorable outcome, and fairness score:

- Perfect equality:** An initial, balanced dataset is used, where records with *females* as favorable outcomes (female records) are added to the male records by just changing the gender value from *female* to *male*. This new perturbed dataset is used to measure the percentage of favorable outcomes. This metric is called perfect equality and can be seen in Figure 5-4 as 77%.
- Favorable outcome:** Adding records with *males* as favorable outcomes to the female records by just changing the gender value from *male* to *female* gives a new perturbed dataset, which is used to measure the percentage of favorable outcomes. This metric is called a favorable outcome and can be seen in Figure 5-4 as 66%.

- **Fairness score:** Dividing the favorable outcome by the perfect equality yields the fairness score for gender, which is 86% (66/77).

The creation of the perturbed datasets as described previously to calculate the favorable outcome for the male records, which are used as references to calculate perfect equality, and the monitored female records, which are used to calculate favorable outcome, entangles female and male records in both directions and subsequently generates more balanced perturbed datasets.



Figure 5-5. Evaluations of Fairness/Bias for Gender

The hatched area is indicating the initial bias for females, which is measured against a defined threshold; it is the difference between the threshold – in our example set at 75% – and the favorable outcome (66%), which is 9%. This initial bias should not necessarily be seen in a negative way: depending on the underlying business model, bias regarding a certain feature value – *females* in our example – could very well represent truthful behavior. Data scientists need to carefully examine and possibly adjust the ML model till the outcome is satisfactory regarding the underlying business model.

Once the initial fairness score and bias have been measured based on the balanced and perturbed datasets, changes of those metrics should be monitored once the ML model has been deployed into production. This is depicted in an exemplary way in Figure 5-5, where the favorable outcome for females is measured on an hourly basis. Monitoring results can be shown as dashboards and by generating alerts if the bias is greater than a defined value.

Evaluations of fairness should be enhanced by considering correlations of the monitored feature value (e.g., females in gender) with other features, such as salary levels, age, etc. Furthermore, AI-infused measurements of fairness should predict when a creeping deterioration of fairness may reach a certain level, alerting business users and data scientists well in advance. Furthermore, actionable recommendations should be provided to data scientists on possible adjustments, for example, changing weights of features, choosing a different ML algorithm, or adjusting hyperparameters. Finally, in an automated AI-infused Data Fabric or Data Mesh, *self-correction* capabilities of a trustworthy AI module should rebuild a di-biased model automatically, either relieving data scientists from taking actions themselves or enabling them to validate and approve these automated corrections.

Let us now elaborate on drift detection.

Drift Detection

An AI model needs to accurately reflect the underlying business reality (e.g., customer behavior, fraud scenarios, etc.) and the data available for scoring and inferencing (e.g., customer profile data, transactional banking data, insurance claims data, etc.). But this can change over time, resulting in a drop of some or all AI model quality metrics – as depicted in Figure 5-6.

Both changes of the business reality and data consistency issues may lead to decreasing AI model quality metrics. Figure 5-6 depicts the declining area under the ROC curve, as an example¹³ of a particular quality metric, and the drop in data consistency.

Using a few examples, we briefly examine what we really mean by changes of business reality and drop in data consistency:

- **Changes of business reality:** Behavior or preferences of customers purchasing goods, signing up for loans and insurance contracts, leveraging services or delivery channels, using the Internet or social media, etc. may change over time and subsequently do not represent the customer behavior or preferences anymore during the AI model training time.
- **Drop in data consistency:** Production data available for scoring and inferencing may deviate from pre-production data used for AI model training, validation, and test. An increase in records in the production data that are like those that the AI model did not evaluate correctly in the pre-production data, changes in frequency distributions or different ranges of numeric values of some features, data patterns, and rare combinations of feature values are some examples.

¹³ We elaborate on additional AI model quality metrics further in this chapter in the following.

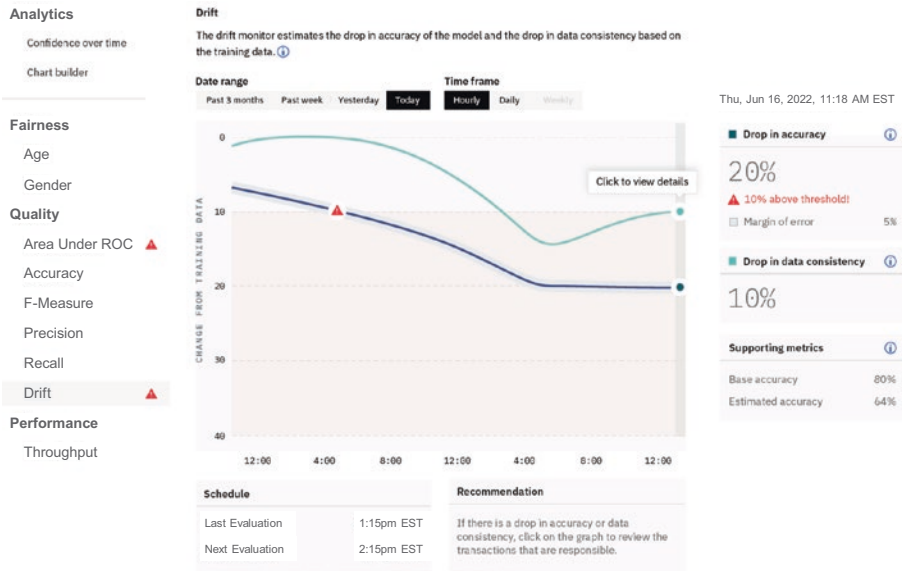


Figure 5-6. Drift Detection

As mentioned previously, these changes in business reality and the drop in data consistency can lead to a drop in AI model quality metrics. The declining area under the ROC curve and the drop in data consistency over time are depicted in Figure 5-6. Other quality metrics could be displayed as well, for example, accuracy, precision, area under the PR curve, etc.

A state-of-the-art Data Fabric or Data Mesh needs to complement drift detection with alerts and predictive and automated self-corrective capabilities – like what we have described under the preceding model fairness.

Let us move on to model explainability.

Model Explainability

As we mentioned previously, model explainability provides insight and transparency of AI model outcomes to business users with no or limited data science skills and allows for *what-if* scenarios. This is rated very highly in importance when it comes to trust, transparency, and interpretability of AI models. Depending on the type of an AI model, evaluation results can include different types of analysis, which are typically based on popular open source frameworks, such as LIME (Local Interpretable Model-Agnostic Explanations) and SHAP, or the ability to test *what-if scenarios* on the data. LIME is a Python library to analyze the input and output values of an AI model to create human-understandable interpretations of the model. LIME reveals which model features are most important for a set of specific data points. There are typically several thousand perturbations done for a particular set of model features that are relatively close to the data points of the model features. In an ideal setting, the features with high importance in LIME are the model features that are most important for those specific data points.

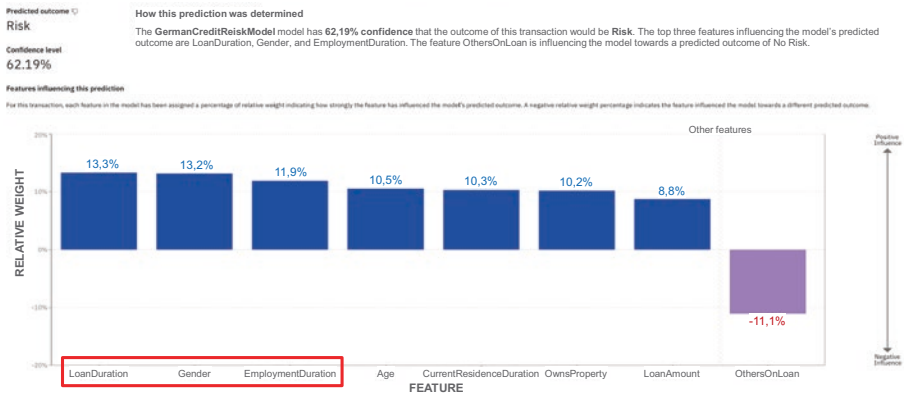


Figure 5-7. Explainability for a Specific Transaction

SHapley Additive exPlanations (SHAP) is a Python library with a method that explains the output of any ML model. It connects optimal credit allocation with local explanations by using Shapley values and their related extensions. SHAP assigns each feature of an ML model an importance value for a particular prediction, which is called a Shapley value. The Shapley value is the average marginal contribution of a feature value across all possible groups of features. The SHAP values of the input features are the sums of the difference between baseline or expected ML model output and the current ML model output for the prediction that is being explained. The baseline ML model output can be based on the summary of the training data or any subset of data that explanations must be generated for.¹⁴

We illustrate explainability with a simple example by using a credit risk model, as depicted in Figure 5-7. The outcome of this ML model produces a *risk score* for a loan approval process, which uses several features as input to the model, such as *LoanDuration*, *Gender*, *EmploymentDuration*, *Age*, *CurrentResidenceDuration*, *OwnsProperty*, *LoanAmount*, and *OthersOnLoan*, which you can see on the horizontal feature axis.

As you can see in the upper-left corner of Figure 5-7, the ML model predicts risk for a loan approval for a particular customer or transaction with a confidence level of 62.19%. The graphic shows the most important features influencing the risk score. The top three features influencing the model's predicted outcome are *LoanDuration* (13.3%), *Gender* (13.2%), and *EmploymentDuration* (11.9%). The feature *OthersOnLoan* is influencing the model toward a predicted outcome of *No Risk*.

Explainability should also allow for GUI-based *what-if* scenarios to enable business users or data scientists to better understand the impact of adjusted feature values on the predicted outcome. This includes scenarios

¹⁴ Refer to Reference [13] for a comparison of the LIME and SHAP methods.

where existing features are neglected or additional features are added. Any adjustment done in such a *what-if* scenario should indeed generate the same information as included in Figure 5-7: confidence level (risk score) and ranking of most influential features. Given the variety of models (e.g., regression, classification, clustering, DL models), this is easier said than done. Our preceding example is based on a relatively simple binary classification model.

Since we have already referenced AI model quality metrics, we briefly describe some of the key ones in the forthcoming subsection.

Model Quality Metrics

As we have seen in this section, measuring quality metrics plays a fundamental role to enable trustworthy AI, specifically for drift detection and explainability; they should be measurable on a regular basis once the AI model has been deployed into production. Relevant quality metrics depend on the type of the AI model. The following is a list of quality metrics for binary classification problems:

- **True positive rate** (TPR), also called sensitivity, recall (R), or hit rate: $TPR = TP / (TP + FN)$
- **False positive rate** (FPR), also called fall-out: $FPR = FP / (FP + TN)$
- **True negative rate** (TNR), also called specificity or selectivity: $TNR = TN / (TN + FP)$
- **False negative rate** (FNR), also called miss rate: $FNR = FN / (FN + TP)$
- **Accuracy** (ACC): $ACC = (TP + TN) / (TP + TN + FP + FN)$
- **Precision** (P): $P = TP / (TP + FP)$

- **Area under the ROC curve:** Is created by plotting the true positive rate (TPR) vs. the false positive rate (FPR) at various threshold settings.
- **Area under the PR curve:** Is created by plotting the precision (P) vs. the recall (R) or true positive rate (TPR), where the balance of the curve is determined by business decisions. The ROC and PR curves are interrelated and should be evaluated in combination based on business imperatives.
- **F1 measure**, also called harmonic mean of precision and sensitivity: $F1\ measure = 2TP / (2TP + FP + FN)$
- **Logarithmic loss** measures the performance of a classification model whose output is a probability value between 0 and 1, where the logarithmic loss increases as the predicted probability diverges from the actual classification:

$$\text{Logarithmic loss} = - (y \log(p) + (1 - y) \log(1 - p)),$$

where y is the predicted probability of the true label p .

Most of these metrics can easily be comprehended from the confusion matrix. A concrete example of a confusion matrix is depicted in Figure 5-8, where you can see the predicted vs. the actual outcomes. Figure 5-8 displays the values TN, FN, FP, and TP, which are taken as input to calculate the quality metrics, which you can see in the top part of the figure, for example, accuracy, recall, precision, etc. You can also see the areas under the ROC and PR curves, which are derived by measuring the TPR and FPR at different threshold settings.¹⁵

¹⁵ See References [14] and [15] for good introductions of ML, where the concepts of the confusion matrix, areas under the ROC and PR curves, etc. are explained.

Area under ROC	Area under PR	Accuracy	TPR	FPR	Recall	Precision	F1 measure	Logarithmic loss
0.76	0.69	0.81	0.62	0.09	0.62	0.78	0.69	0.43

		Prediction		Total
		No Risk	Risk	
Actual	No Risk	120	12	132
	Risk	26	42	68
Total		146	54	200

True negatives (TN) points to 120
False negatives (FN) points to 26
False positives (FP) points to 12
True positives (TP) points to 42

Figure 5-8. Confusion Matrix for Binary Classification Problems

There are additional quality measures for regression problems and quality metrics for multiclass classification problems. The following are some quality metrics for regression problems: R squared, proportion explained variance, root mean squared error (RMSE), mean absolute error, and mean squared error.

Finally, we are listing some quality metrics for multiclass classification problems: accuracy, weighted true positive rate (wTPR), weighted false positive rate (wFPR), weighted recall, weighted precision, weighted F1 measure, and logarithmic loss.

Intelligent Information Integration

For decades, information integration has been a well-known IT domain, which includes but is not limited to ubiquitous challenges that stem from the diversity as well as the disparity of information and data sources. What has changed in recent years, however, is an ever-increasing need to intelligently accommodate different integration methods in an automated fashion, for example, traditional ETL and replication, real-time streaming of data, messaging, data virtualization and federation, microservices, etc., in a wide variety of architectural settings, including hybrid cloud environments.

Furthermore, information integration must incorporate a rich variety of artefacts, other than just structured and unstructured data, for example, AI models, pipelines, ETL stages, and even application logic. Data Mesh solutions, with their key imperatives to establish a data marketplace for business users with self-service shopping-for-data (data-as-a-product), are only adding further demand for automated and intelligent information integration techniques.

A modern Data Fabric architecture or Data Mesh solution needs to tackle these challenges. AI is particularly well suited for this purpose. We use the term *intelligent information integration* as an AI-infused information integration layer, which constitutes a vital capability for both concepts to automate information integration tasks as far as possible.

Table 5-2 lists essential intelligent information integration capabilities that are needed for any modern approach.

Table 5-2. *Intelligent Information Integration Capabilities*

#	Capabilities	High-Level Description
1	Automated workload distribution	AI-enabled automated workload distribution should be implemented, considering underlying system availability and capabilities, resource consumption, SLAs, and performance requirements.
2	Self-service information integration	Business users and data engineers need to perform information integration tasks (including data exploration, data preparation and transformation in a self-service manner) with transparency from the complexity and diversity of source systems and corresponding artefacts.

(continued)

Table 5-2. *(continued)*

#	Capabilities	High-Level Description
3	Active metadata exploitation	Business users should transparently leverage active metadata to gain pervasive and relevant insight regarding the underlying assets, search and discovery, access methods, policies, etc.
4	Semantic knowledge graph exploitation	Business users need to exploit semantic knowledge graphs to understand relevance of taxonomies and ontologies, get further insights from tags and annotations, and get relevant recommendations for asset consumption and inferencing into applications.
5	Learnable information integration	Intelligent information integration needs to be learnable, meaning that AI methods should be applied to adjust, improve, simplify, and optimize integration flows and tasks over time.
6	Automated corrections of integration flows	AI should be applied to faulty or bad-performing information integration flows, jobs, and tasks, generating recommendations for IT personnel and automatically implementing meaningful corrections.
7	Leveraging the digital exhaust ¹⁶	Information integration should be improved over time by automatically activating the digital exhaust, for instance, to optimize resource allocation for end-of-month or end-of quarter integration tasks.

¹⁶ See Chapter 7 for more information on activating the digital exhaust.

Key Takeaways

We conclude this chapter with a few key takeaways as summarized in Table 5-3.

Table 5-3. Key Takeaways

#	Key Takeaway	High-Level Description
1	Active metadata is a key characteristic of Data Fabric and Data Mesh.	Active metadata is AI/ML-augmented metadata, generated by applying AI/ML techniques to metadata to gain additional actionable insight from metadata, which can be used to further automate Data Fabric and Data Mesh tasks.
2	AI/ML techniques need to be applied to data curation tasks.	AI/ML-infused data curation is referred to as advanced or intelligent data curation; it enables automated data curation with intelligent cataloging.
3	Semantic knowledge graphs should be an integral part of a Data Fabric.	AI/ML-enriched knowledge graphs are called semantic knowledge graphs or semantic networks, depicting non-obvious relationships of objects and assets, providing further actionable insight to data consumers and business users.
4	Self-service is a vital Data Fabric and Data Mesh characteristic.	Self-service capabilities of a Data Mesh solution enable a business user to perform business tasks without support and dependencies from IT or other business units.
5	Trustworthy AI means to measure and ensure AI model fairness.	Fairness or bias deals with measuring and managing desirable or undesirable preferences for certain values of chosen features determining the outcome of an AI model.

(continued)

Table 5-3. (continued)

#	Key Takeaway	High-Level Description
6	Trustworthy AI means to detect drift.	Drift measures the drop in accuracy and drop in data consistency by comparing accuracy during runtime with the accuracy during training and by comparing key characteristics of the dataset used for training with the dataset during runtime.
7	Trustworthy AI means to provide explainability.	Explainability provides insight and transparency of AI model outcomes to business users with no or limited data science skills and allows for <i>what-if</i> scenarios.
8	Model quality metrics serve as input to enable trustworthy AI.	There are several AI model quality metrics that are automatically calculated; they can also be visualized to provide further insight to data scientists to optimize model outcomes.
9	AI/ML introduces automation and intelligence to information integration tasks.	The term <i>intelligent information integration</i> relates to an AI-infused information integration layer, which constitutes a vital capability of a modern Data Fabric architecture and Data Mesh solution.

References

- [1] Gartner, Peer Insights, *Metadata Management (EEM) Solutions Reviews and Ratings*, www.gartner.com/reviews/market/metadata-management-solutions (accessed July 22, 2022).

- [2] Gartner, Gartner Research, *Gartner Magic Quadrant for Metadata Management Solutions*, www.gartner.com/en/documents/3993025 (accessed July 22, 2022).
- [3] Gartner, Gartner Research, *Market Guide for Active Metadata Management*, www.gartner.com/en/documents/4004082 (accessed July 23, 2022).
- [4] IBM, IBM Cloud Pak for Data, *Data curation*, <https://dataplatform.cloud.ibm.com/docs/content/wsj/governance/curation.html> (accessed July 24, 2022).
- [5] Data Science Central, Aasman, J., *The Foundation of Data Fabrics and AI: Semantic Knowledge Graphs*, May 19, 2022, www.datasciencecentral.com/the-foundation-of-data-fabrics-and-ai-semantic-knowledge-graphs/ (accessed July 24, 2022).
- [6] AWS, *Amazon Neptune Documentation*, <https://docs.aws.amazon.com/neptune/index.html> (accessed July 23, 2022).
- [7] Neo4j, *Neo4j Graph Data Platform*, <https://neo4j.com/> (accessed July 23, 2022).
- [8] IBM, *IBM Graph, An enterprise-grade property graph as a service, built on open source database technologies*, www.ibm.com/analytics/ca/en/technology/cloud-data-services/graph/ (accessed July 23, 2022).

- [9] European Commission, *Ethics Guidelines for Trustworthy AI*, <https://ec.europa.eu/futurium/en/ai-alliance-consultation.1.html> (accessed July 26, 2022).
- [10] US Department of State, *Artificial Intelligence (AI)*, www.state.gov/artificial-intelligence/ (accessed July 26, 2022).
- [11] Heintz, F., Milano, M., O'Sullivan, B. (Eds.), *Lecture Notes in Artificial Intelligence, Trustworthy AI – Integrating Learning, Optimization and Reasoning*, Springer, 2021, ISBN: 978-3-030-73958-4.
- [12] IBM, IBM Cloud Pak for Data, *Watson OpenScale on Cloud Pak for Data*, www.ibm.com/docs/en/cloud-paks/cp-data/4.5.x?topic=services-watson-openscale (accessed July 27, 2022).
- [13] Poduska, J., *SHAP and LIME Python Libraries: Part 1 – Great Explainers, with Pros and Cons to Both*, 2018, www.dominodatalab.com/blog/shap-lime-python-libraries-part-1-great-explainers-pros-cons (accessed December 18, 2022).
- [14] James, G., Witten, D., Hastie, T., Tibshirani, R., *An Introduction into Statistical Learning with Applications in R*, Springer, 2015, ISBN: 978-1-4614-7137-0.
- [15] Flach, P., *Machine Learning*, Cambridge University Press, 2012, ISBN: 978-1-107-42222-3.

CHAPTER 6

Relevant ML and DL Concepts

At the heart of a Data Fabric and Data Mesh is the use of *artificial intelligence* (AI) and *machine learning* (ML) technologies to automate complex data tasks to the greatest extent possible. Therefore, understanding the concepts of AI and ML is the foundation for implementing both concepts in an enterprise. If you are already an AI/ML practitioner, you might skip this chapter. If you are not sure, please take a quick assessment by answering the following questions, and please continue with this chapter should you need more clarity:

- Do you know the correlation of AI, ML, and DL?
- Do you know what types of problems AI and ML can solve?
- Do you know relevant business scenarios for AI?
- Do you know the lifecycle of AI models, and do you know the specific tasks in each stage of the AI lifecycle?

Introduction to AI, ML, and DL

Broadly speaking, AI is a system that can simulate human perception, learning, reasoning, and interaction. It covers a wide range of fields, including autonomous driving, robot control, computer vision, Natural Language Understanding (NLU), and more. The current AI technology for commercial use is still relatively narrow, where AI models require large amounts of data to be trained for a single task and a single domain and AI models for new domains require new data to be acquired. In the meantime, a great deal of research is being done to broaden the scope of AI. For example, meta-learning and N-shots learning¹ focus on training models using small amounts of samples. Transfer learning² reuses AI models from one domain in other domains to handle multi-tasks and multi-domains. Federated learning³ enables multiple parties to collaboratively learn a shared model without sharing the data.

ML is one way to implement AI systems. ML uses algorithms to parse data, learn from it, and then make decisions or make predictions about real-world events. Unlike traditional software programs that are hard-coded to solve a specific task, ML uses large amounts of data to train algorithms to learn how to accomplish a task from data. Rule-based systems are an example of a non-ML system. In rule-based systems, it is up to humans to clearly define the parameters of branching conditions, such as the if-statements present in the implementation code. ML has a strong assumption that the history of what has happened has an inherent pattern and thereby that the same results under similar conditions will occur in the future. The process of training is to find a function $f(x)$ that optimally describes the pattern of existing samples and then use this function to make predictions for new samples.

¹ See Reference [1] for more details on N-shots learning.

² See Reference [2] for more details on transfer learning.

³ See Reference [3] for more details on federated learning.

There are three types of ML: supervised learning, unsupervised learning, and reinforcement learning.

Supervised Learning derives the prediction function from the labeled training data. It's like taking the exam (inference) after studying past exams' questions (data) and answers (labels).



Van Gogh



Monet



Picasso

Figure 6-1. *Training Datasets for Supervised Learning*

For example, given several previous artworks by Van Gogh, Picasso, and Monet (as depicted in Figure 6-1), who has painted the one in Figure 6-2?



Figure 6-2. Testing Data for Supervised Learning

Unsupervised Learning infers conclusions from unlabeled training data. Although the training data is not labeled, the machine learns to find common features, a structure or pattern in the input data, or if there are some associations between the features. The most typical unsupervised learning is clustering, which groups similar data from many data samples. For example, a system for grouping customers based on purchase behaviors is unsupervised learning. By grouping customers according to the characteristics of their purchase history, different marketing campaigns can be implemented for each group.

For example, although no labels are defined for the pictures in Figure 6-3, ML models can distinguish one category from the other category.



Figure 6-3. *An Illustration of Unsupervised Learning*

In the real world, there are many situations where only some data is labeled because of the high cost of manual labeling. One way to resolve this problem is to use a small number of labeled data to train a model and use this model to label yet unlabeled data and then create a new dataset to improve the model. This type of learning is known as semi-supervised learning⁴ and is now a popular area of research.

Reinforcement learning is concerned with how a software agent can take actions in an environment to maximize some cumulative return. Unlike supervised learning and unsupervised learning, reinforcement learning doesn't require a large amount of input data. It has a beginning state and enters a new state when an action is made. The system will give signals – positive for desired results and negative for failure. The whole

⁴ See Reference [4] for more details on semi-supervised learning.

training process is constantly exploring the possible actions through many trials and errors in a changing environment until then finding the best path to get maximum outcome based on predefined criteria. Common applications include chess play, robotics control, and autonomous driving.

Deep learning (DL) is “a subset of ML, which is essentially a neural network with three or more layers.”⁵ The strength of DL is automatic feature extraction. Instead of hand-picking features from datasets, DL can discover and learn good feature representation. For example, an image can be represented in a variety of ways, such as a vector of intensity values per pixel or more abstractly as a series of edges, regions of a particular shape at a different layer of DL architecture. DL has a wide range of applications, such as image recognition, video analytics, object detection, machine translation, etc. However, DL requires large amounts of labeled data and extremely powerful computation resources, like high-performance Graphical Processing Units (GPUs), to find the right architecture and best set of parameters for the entire architecture. It usually takes millions of images and thousands of hours of video for training.

ML and DL Industry Use Cases

To discover AI use cases for each industry, understanding what types of problems ML and DL are good at solving is a must. The following five types of problems are often seen resolved by ML and DL:

1. **Classification problem:** The first type is the classification problem, whether the sample is type A or type B. For example, *is this patient highly likely to have diabetes?* Or *is this client at risk of churn?* These are typically binary classification problems. Supervised learning algorithms are widely used for classification problems.

⁵ See Reference [5] for IBM’s definition of DL.

2. **Regression problem:** The second type is the regression problem, for example, predicting house prices, stock prices, and commodity prices based on historical data and relevant external events.
3. **Anomaly detection:** The third type of problem is anomaly detection. A very common anomaly detection scenario in IT operations is whether the current CPU, memory, and storage consumption are as expected based on workload patterns. For instance, *will this cause any performance degradation or even outage?*
4. **Clustering problem:** The fourth type is the clustering problem, where unsupervised learning algorithms excel. A typical case is that a company looks to improve business performance by using customer segmentation for precision marketing.
5. **The next best action problem:** Finally, the next best action problem. Reinforcement learning algorithms are used to solve this type of problem. In addition to robot control and autonomous driving, transactions in the financial service industry, treatments in healthcare, and online recommendations in ecommerce are popular reinforcement learning use cases.

After understanding the problems that AI can solve, let's look at industry-related AI use cases and AI applications:

Manufacturing: Predictive maintenance is a method of preventing the failure of expensive manufacturing equipment by analyzing data throughout the production process. A variety of data includes vibrations, temperature, ultrasonics, and acoustics through sensors built into the

equipment. These data help create ML models to identify abnormal behaviors in advance to ensure that necessary maintenance actions are taken to minimize production downtime. Inspecting goods and products can be quite a cumbersome task for any large manufacturing company. Computer vision can provide the analysis of real-time information obtained from captured images to perform complex inspection tasks. It can help verify the correct number of items in the warehouse, monitor the staff's operations for compliance with safety regulations, and check for the presence of defective products.

Financial service: AI has a wide range of application scenarios in the financial industry, including anti-money laundering, capital position forecasting, smart loan approval, credit card fraud detection, intelligent investment, etc. The financial industry adopted information technology many years ago, so it has accumulated a large amount of data, laying a good foundation for ML. However, the financial industry is also a highly regulated industry, and if the models are not well interpreted, they cannot be used in production systems due to compliance. Thus, the scenarios of DL are limited in this respect.

Telecommunications: Telecommunications is another industry with a wealth of accumulated data. AI is used for intelligent planning of mobile sites, fast turnup of base station services, intelligent path planning, and automatic deployment of optical transmission networks. It can forecast the network based on network history data, manage network resources dynamically and adaptively, and adjust parameters. As operations in the telecom industry are supported by large network equipment, predictive maintenance of equipment is a very important AI scenario. Models are built for network health analysis and prediction and network self-healing, reducing the workload of operation and maintenance personnel, improving the efficiency of operation and maintenance fault handling, and continuously promoting fundamental changes in network operation and maintenance modes.

Automobile: There is no doubt that the most critical application of AI in the automotive industry is autonomous driving. In addition to this, there are many other AI application scenarios in the automotive industry. For example, supply chain management in the automotive manufacturing industry is complex, and it is very valuable to use AI to optimize the supply chain. At the same time, quality inspection of the automobile production line, abnormality detection during driving, and interactive experience in the car are all perfect scenarios for AI applications. In addition, the repair procedures for cars are complex. Using chatbots can help staff find the information they need most to complete their work in a timely manner.

After understanding what business problems AI can solve, let's move on to some concepts in various stages of AI.

Data Exploration and Preparation

The first stage in ML is data exploration and data preparation.⁶ At this stage, an initial exploratory analysis of the data needs to be done to understand the distribution of the data and the state of the data quality. Descriptive statistics⁷ is a technique to describe the characteristics of a dataset. It provides measures to summarize the central value of a dataset (e.g., mean, median, and mode measures) and the dispersion of data within the dataset (e.g., min, max, and variance measures). The histogram is another powerful exploratory tool for data understanding. It shows the distribution of continuous data, allowing easy detection of outliers.

In addition to understanding the distribution of data, the quality of data needs to be inspected too. The reality is that data can be subject to a variety of errors depending on the source and the way it is generated.

⁶ See Reference [6] for more information on data preparation.

⁷ See Reference [7] for more information on descriptive statistics.

For example, when entered manually, there may be missing values, duplicate values, incomplete values, and outliers. Or there may be inconsistencies in the format and caliber of the data when it is extracted from various application systems. These data quality issues can lead to irrelevant or even incorrect analysis results. Therefore, the next important data preparation task is *data cleansing*, which is a process of detecting and correcting erroneous data in a dataset.

Data quality can be improved by using *data transformation*. The following transformation methods are commonly used:

- **Missing values:** Fill missing values with default values, for example, using the average of the data values in this column or with the values of adjacent data.
- **Various data formats:** Turn the data with various date formats into a uniform format, for example, extending all the phone numbers with country codes.
- **Categorical values:** Encode categorical values to numerical values, for example, by determining the numerical range of all features and adjusting them to a uniform scale (normalization).
- **Outliers and duplicate values:** Remove outliers and duplicates from further inclusion.

Once you have clean and tidy data, the next step is *feature engineering*, which aims to get better training data for optimal models. Feature engineering includes sub-problems: feature selection, feature construction, and feature extraction.

Feature selection is the technique of selecting the subset of input features that are most relevant to the target variable and discarding the features that are less relevant. For example, if the correlation between

features is strong, it means that these features are redundant. Selecting redundant features or irrelevant features only increases the training time and does not improve the performance of models.⁸

Feature construction is the process of constructing new features from original features. Creating new features requires data scientists' deep insight and analytics skills to identify meaningful transformation from existing features, for example, combining attributes such as date-time-location, decomposing or slicing the original features such as from one feature that has three values [*green, yellow, unknown*] to three Boolean features – *is_green, is_yellow, and is_unknown* – or adding arithmetic operations such as $x^3 + y$ as a new feature.

Feature extraction is the process of reducing the dimensionality of the dataset by applying various dimensionality reduction algorithms, such as Principal Component Analysis (PCA), Independent Component Analysis (ICA), and Linear Discriminant Analysis (LDA). The purpose of feature extraction is to minimize the dimensionality of the data while ensuring that the important information of the target is retained.

Model Selection, Training, and Evaluation

Choosing the right AI/ML algorithms to train your model is a creative process of mapping your business problem to an AI/ML toolset. Depending on the nature of the data you collect and prepare, you can determine which algorithms can solve the problem. Model selection, training, and evaluation are iterative processes in the AI lifecycle, in which data scientists do various experiments by exploring different models using different features.

The following Table 6-1 summarizes key questions that should be considered in AI model selection, training, and evaluation regarding business, data, and environment.⁹

⁸ See Reference [8] for more information on feature selection.

⁹ See Reference [9] for a comprehensive review on choosing the right ML algorithm.

Table 6-1. Key questions to be answered when choosing and evaluating an AI model

	Model Selection/Training	Model Evaluation
Business	<ul style="list-style-type: none"> – Is it an exploration or prediction? – Do you need to detect anomalies or outliers? – Do you want a recommendation? 	<ul style="list-style-type: none"> – Are there any domain-specific metrics to justify the model? – What is the expected output of the business problem? – Are there any quality and acceptance measures?
Data	<ul style="list-style-type: none"> – Is there a target to predict? – Is the prediction numeric or categorical? – Does the data need to be grouped or scaled? – What is the range of values for features and target in the data? – Do you need dimensionality reduction or manifold learning? – How to address missing data values? – Do you have labeled data? – Do you need to process unstructured data (i.e., text)? – What are potential hyperparameters? 	<ul style="list-style-type: none"> – What is the type of model (classification, regression, clustering, etc.)? – What’s expected estimation error and approximation error? – What’s expected confusion matrix and precision, recall, and F1 measure? – What’s expected accuracy and error rate? – What’s expected ROC and AUC? – What’s expected cost curve? – What’s expected Shapley Additive exPlanations (SHAP) values’ – What’s expected performance on the test/holdout dataset? – When conducting HPO, and how do you design hypothesis tests?

(continued)

Table 6-1. (continued)

	Model Selection/Training	Model Evaluation
Environment	<ul style="list-style-type: none"> – What ML frameworks do you have access to (proprietary or open source)? – What computing resources do you have? – Where is the data? How large is it? – Which platform will the model be deployed to? – Frequency of model retraining. – Is this a distributed training or deployment pattern? 	<ul style="list-style-type: none"> – Do you need to explain the model? – Do you need to explain the output of the model? – Is it online or batch prediction (different way of collecting evidence for evaluation)? – What is the size of the model? – Is it possible to repeat the training process and gain the same model?

In the model selection and model training phases, the questions to be thought about are similar, and therefore they have been combined into one column in the preceding table. However, the focus of the same question at these two stages is different. For example, for the question *Is the prediction numeric or categorical?*, model selection aims to find the characteristics of different data types, while model training is to identify applicable algorithms in conjunction with the data types.

The evaluation method depends on the nature of the problems. Classification models favor accuracy, precision, recall, F1 measure, Receiver Operating Characteristic (ROC), and area under the ROC curve, whereas regression models prefer mean absolute error (MAE), mean squared error (MSE), root mean squared error (RMSE), etc. Model evaluation is critical because AI models have the potential to make incorrect predictions.

A model that has a zero error rate on the training dataset doesn't mean it will perform well for unforeseen data in the future. A model can be tuned to master every detail of the training dataset, but it may fail to generalize prediction of new samples. This is called as *overfitting*. Model evaluation (a.k.a. offline testing) aims to overcome overfitting in the training process.

As a rule of thumb for practitioners, follow these three steps to set up the model selection and evaluation process:

- Design a sampling strategy to separate data into mutually exclusive datasets for training and testing.
- Choose best-fit evaluation metrics to gauge performance of models.
- Conduct tests using different datasets to make comparisons against trained models and make the selection.

Model Deployment

Model deployment is the final stage of delivering the model. It is the process of infusing the model into the production system (either in real time or through a batch interface). The previous chapter explains that a Data Fabric and Data Mesh is designed to address the challenges of accessing data in a distributed hybrid environment. Model scoring needs real-world data to make a prediction, and thereby model deployment also faces the challenge of accessing distributed data. Therefore, the model deployment pattern does not solely rely on how to integrate with applications, but the residency and the size of input data required by models, latency requirements, and so on. This section introduces some key decision factors for deploying AI/ML models.

One aspect to consider is the AI/ML model format. There are many language-agnostic and vendor-specific serialization formats that exist in the industry. AI deployments¹⁰ can be characterized by using various programming languages and AI model standards. There are quite a few popular AI-related programming languages, such as Python (the most

¹⁰ See Reference [10] for more information on deploying AI, especially in regard to developing scorecards and performing comprehensive self-assessments.

prominent one), Scala, R, Julia, and additional languages for notebooks, such as Ruby, Perl, F#, and C#. The choice of different programming languages allows for the implementation of different AI/ML frameworks and the corresponding model formats. In the open source world, several AI model-related standards¹¹ exist, where the Predictive Model Markup Language (PMML), Open Neural Network eXchange (ONNX), and Portable Format for Analytics (PFA) are prominent examples. Although Spark is not a standard, it is still popular among data scientists and provides a de facto standardization, especially for Spark ML model exchange and portability.

The following Table 6-2 highlights features of popular language-agnostic formats for the AI/ML model, namely, PMML, PFA, and ONNX.

Table 6-2. *The language-agnostic formats of AI Model*

Format	Human-Readable	Runtime Support
PMML	Yes (XML)	Python, R, C++, Java, or Scala
PFA	Yes (JSON with the Avro schema for data types)	PFA-enabled runtime
ONNX	No	TensorFlow, PyTorch, CNTK, CoreML, PaddlePaddle, ONNX-enabled runtimes

The choice of AI languages and standards depends on individual data scientists' use cases, skills, and preferences. Some AI models can be converted or integrated using specific standards. Python Scikit-learn models, for example, can be put into production using PMML or ONNX. However, in other cases, the owner of the AI model does not want to expose the model and therefore needs to consider how to package the model. For maximum compatibility, the model should be exported to a language-independent format.

¹¹ See Reference [11] for more information on PMML, ONNX, and PFA.

To protect the model (i.e., hide the implementation details of the model), it can be deployed in a proprietary format with a scoring engine that can be used to parse the model and perform inference on it.

The next decision is about serving the model. There are a few patterns to choose from.¹² To determine a suitable pattern, you'll need to consider the following factors: the size and the location of data, the size of the model, the latency of request, the cost of serving, etc.

As shown in Figure 6-4, a RESTful API is suitable for real-time requests on a small amount of data, usually one record per request. The cost of serving a RESTful API is high since you need dedicated computing resources to make the service highly available. Streaming API should be used when a huge amount of data needs to be processed and the result is expected in near real time. Streaming API is asynchronous by design. Consider a messaging system with a queue to handle such requests.

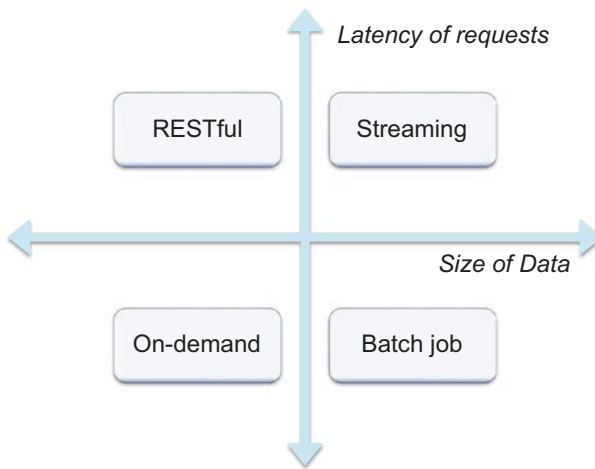


Figure 6-4. *Model Deployment Patterns*

¹²See Reference [10] for the IBM Data Science Best Practices on deployment architecture.

If you need to process a large amount of data but do not need the result right away, deploy the model into a batch job, which can be scheduled to run on predefined triggers (a specific time or an event) and yield output to external storage, for example, a database table. If the request is not sensitive to either latency or size of data, you can deploy the model on demand. This is sometimes referred to as serverless, which is more cost-effective since the computing resources are only required when there are incoming workloads.

Natural Language Processing (NLP)

Natural languages such as Chinese, English, and Japanese are flexible and versatile, but they cannot be well understood by computers. Natural Language Processing (NLP) was born to achieve communication between humans and computers using natural language. NLP is a field that integrates linguistics, computer science, mathematics, and other disciplines to study not only linguistics but also how to make computers process these languages. It is divided into two main directions: Natural Language Understanding (NLU), which is listening and reading, and Natural Language Generation (NLG), which is speaking and writing.

By the 1980s and 1990s, statistical ML algorithms were introduced into NLP, and rule-based approaches were gradually replaced by statistical-based approaches. During this phase, NLP made substantial breakthroughs and moved toward practical stages. From around 2008, as DL neural networks achieved remarkable results in image processing and speech recognition, DL was also applied to NLP: from the very first word embedding and word2vec to neural network models such as RNN, GRU, LSTM, and, more recently, attention mechanism, pre-trained language models, etc. With the addition of DL, there is significant progress made in NLP.

In natural language, words are the most basic units. For a computer to understand and process natural language, we first must encode words. Although it is easy to construct word vectors using one-hot encoding,¹³ it does not represent the semantics of the words very well. Word2vec represents words as a fixed-length vector and learns the semantic information of words by context. It contains two models – one is predicting the context by a central word, and the other one is predicting a central word by the context.

At the same time as word vectors were proposed, the DL RNN framework was applied to NLP with great success in combination with word vectors. RNN is not perfect. It has problems like difficulty to parallelize and establish long-distance and hierarchical dependencies. The recent attention mechanism can help resolve these issues. The core idea is to filter out a small amount of important information from a large amount of information and focus on this information. It focuses less on other external data and instead focuses only on the input data itself and is better at capturing the relevance of the data internally.

While the application of DL gave NLP its first leap, the advent of pre-training models has given NLP its second leap. Pre-training learns a powerful language model from large-scale corpus data by self-supervised learning (without annotation), which is then migrated to specific tasks by fine-tuning to eventually achieve significant results.

After introducing the popular research advances in the field of NLP, let's look at the business value of NLP, where still 80% of organizations are waking up to the fact that 80% of their content is unstructured.¹⁴ The unstructured data includes surveillance video and images, customer support recordings, various reports, social media posts, historical documentation, operating manuals, and more. NLP can help companies uncover knowledge from text-related unstructured data.

¹³ See Reference [12] for more details on one-hot encoding.

¹⁴ See Reference [13] for more information on Woodside's story.

For example, Woodside Energy¹⁵ harnesses the power of NLP technology powered by IBM Watson to extract meaningful insights from 30 years of complex engineering data and help workers quickly find information and synthesize it into informed business decisions.¹⁶

There are many other use cases for NLP:

- **Virtual agents and chatbots:** A question expressed in natural language is analyzed to some extent (e.g., entity links, relational formulas, forming logical expressions, etc.). After the analysis is completed, possible candidate answers are found in the knowledge base, and the best answer is found by sorting. For example, auto-response customer service is widely used to improve and optimize customer relationship in the ecommerce industry to filter out a number of repetitive questions by replying to many basic and repetitive questions, thus enabling human customer service to serve customers better.
- **Machine translation:** The most well-known NLP scenario is obtaining text in a source language and automatically translating the input source language text into a target language text. Today, this scenario already works very well.
- **Audio transcript:** Convert an audio file into a text file that can support the audience gaining a better understanding of the content in the audio and make the audio content searchable for future reference.

¹⁵ See Reference [14] for more information on Woodside Energy leveraging IBM Watson.

¹⁶ See Reference [15] for more information on NLP use cases.

- **Sentiment analysis:** Analyzing the sentiment of massive customer support calls and social media comments to achieve a timely response to public opinion.
- **Text summarization:** Extract key information from a huge amount of text to save people time and effort reading the entire document, finding relevant information easily and quickly.
- **Grammar check and correction:** Automatically correct grammar or spelling mistakes to improve the quality and correctness of the writing.

Key Takeaways

We conclude this chapter with a few key takeaways as summarized in Table 6-3.

Table 6-3. *Key Takeaways*

#	Key Takeaway	High-Level Description
1	There are three types of ML.	Supervised learning derives the prediction function from the labeled training data, while unsupervised learning finds hidden patterns with unlabeled training data. Reinforcement learning doesn't require a large amount of training data but does trial and error to get maximum outcome based on predefined criteria.
2	There are some types of problems that ML and DL are good at solving.	ML and DL can solve the problems like classification (A or B), regression (how much), anomaly detection, clustering, and best next action.

(continued)

Table 6-3. *(continued)*

#	Key Takeaway	High-Level Description
3	ML/AI addresses a broad set of industry use cases.	Predictive maintenance in manufacture, fraud detection in finance, network planning in telecom, in-car interactive experience in automobile, etc.
4	Feature engineering includes several subfields.	Feature engineering includes sub-problems, which are feature selection, feature construction, and feature extraction.
5	The model evaluation method depends on the nature of the business problems.	Classification models favor accuracy, precision, recall, F1 measure, ROC, and area under the ROC curve, whereas regression models prefer mean absolute error (MAE), mean squared error (MSE), and root mean squared error (RMSE).
6	It's a multi-factor decision to choose a model deployment pattern.	Model deployment patterns do not solely rely on the integration with applications but the residency and size of input data required by models, latency requirements, etc.
7	NLP is divided into two main directions.	Natural Language Understanding (NLU), which is listening and reading, and Natural Language Generation (NLG), which is speaking and writing.
8	There are many other use cases for NLP.	Virtual agents and chatbots, machine translation, audio transcript, sentiment analysis, text summarization, grammar check and correction, etc.

References

- [1] Chatterjee, P., *How do zero-shot, one-shot and few-shot learning differ?*, <https://analyticsindiamag.com/how-do-zero-shot-one-shot-and-few-shot-learning-differ> (accessed August 4, 2022).
- [2] Donges, N., *What Is Transfer Learning? Exploring the Popular Deep Learning Approach*, <https://builtin.com/data-science/transfer-learning> (accessed August 4, 2022).
- [3] McMahan, B., Ramage, D., *Federated Learning: Collaborative Machine Learning Without Centralized Training Data*, <https://ai.googleblog.com/2017/04/federated-learning-collaborative.html> (accessed August 4, 2022).
- [4] Altexsoft, *Semi-Supervised Learning, Explained with Examples*, www.altexsoft.com/blog/semi-supervised-learning/ (accessed August 4, 2022).
- [5] IBM, *Deep Learning*, www.ibm.com/cloud/learn/deep-learning (accessed August 4, 2022).
- [6] Patel, A., *Part-3 Data Science Methodology from Understanding to Preparation*, <https://medium.com/ml-research-lab/part-3-data-science-methodology-from-understanding-to-preparation-a666a8203179> (accessed August 4, 2022).

- [7] Hayes, A., *Descriptive Statistics*, www.investopedia.com/terms/d/descriptive_statistics.asp (accessed August 4, 2022).
- [8] Brownlee, J., *How to Choose a Feature Selection Method for Machine Learning*, <https://machinelearningmastery.com/feature-selection-with-real-and-categorical-data/> (accessed August 4, 2022).
- [9] IBM, *Evaluate and select a machine learning algorithm*, www.ibm.com/garage/method/practices/reason/evaluate-and-select-machine-learning-algorithm/ (accessed August 4, 2022).
- [10] IBM Data Science Best Practices, *Deployment Architecture*, <https://ibm.github.io/data-science-best-practices/architecture.html#deployment-architecture---pipelines> (accessed August 4, 2022).
- [11] Open Standards for Machine Learning Model Deployment <https://community.ibm.com/community/user/datascience/viewdocument/open-standards-for-machine-learning>.
- [12] Brownlee, J., *Ordinal and One-Hot Encodings for Categorical Data*, <https://machinelearningmastery.com/one-hot-encoding-for-categorical-data/> (accessed August 4, 2022).
- [13] IBM, *Woodside Energy*, www.ibm.com/case-studies/woodside-energy-watson-cognitive (accessed August 4, 2022).

- [14] IBM, *Preserving institutional wisdom*, www.ibm.com/watson/stories/woodside (accessed October 9, 2022).
- [15] IBM, *Natural Language Processing (NLP)*, www.ibm.com/cloud/learn/natural-language-processing (accessed August 4, 2022).

CHAPTER 7

AI and ML for a Data Fabric and Data Mesh

This chapter provides a deep dive into the exploitation of AI and ML for various Data Fabric and Data Mash topics and tasks, such as data discovery, data access, and data profiling, analyzing the “digital exhaust” of Data Fabric and Data Mesh process steps, ML-based entity matching, automated data quality assessments, and semantic enrichment of the underlying data.

This is an essential chapter, which highlights some novel ideas to augment both concepts with AI and ML.

Introduction

Today’s data-rich enterprises are frequently confronted with the challenge to manage a heterogenous and highly distributed data landscape, where business data resides in diverse data stores, leading to the segmentation of data across various organizations. Discovering, understanding, and gaining trust in data and accessing relevant business data for downstream consuming purposes constitutes a huge challenge for business organizations. The limitation of human cognitive capabilities to understand data in context and gain semantic knowledge has not only created a bottleneck for data exploration tasks but has also led to

ever-increasing difficulties to gain relevant and actionable business insight. Thus, a human-centric manual Data Fabric or Data Mesh approach is lacking the means to democratize access to data in a self-service fashion and to enable *data-as-a-product* (*shopping-for-data*) as a key Data Mesh principle.

The implications of AI and ML (AI/ML) to simplify and optimize a Data Mesh solution cannot be overestimated. Especially ML as the prevalent AI technology facilitates simplified discovery, access, and profiling of data that is stored in a variety of data stores, for example, RDBMS, NoSQL databases, or key value stores that are distributed enterprise-wide. The exploitation of AI/ML in Data Fabric¹ scenarios has incentivized exploration and consumption of data for business organizations that have otherwise struggled even finding relevant data for their business purpose.

Apart from exploring the usage of AI/ML to improve and simplify a Data Fabric architecture or Data Mesh solution, most data-centric enterprises are looking for current capabilities used for their existing AI initiatives. In this chapter, however, we elaborate on AI/ML to be used for both concepts.

General Overview

The following is a short, introductory description of some key Data Fabric and Data Mesh areas that can derive benefits from ML/DL. These areas are further discussed in the following corresponding sections. Since our focus is exclusively on the ML/DL aspects, we are assuming the reader to have a basic knowledge about these concepts in general. For instance, we don't elaborate in-depth about data governance or data quality:

¹ See Reference [1] for an introduction into an AI Data Fabric.

1. **Cataloging:** One of the most essential components for concepts is a knowledge or governance catalog. It contains business, technical, and operational metadata. Although governance catalogs and cataloging in general have existed already for quite some time, the ever-increasing diversity and complexity of the IT and business landscape and emerging regulations induce the need for AI-infused cataloging.² For instance, AI is leveraged to automatically extract and propose new business terms for new regulations as candidates into a knowledge catalog. Registering and cataloging new data sources is based on automated metadata discovery, AI-based data rule discovery, automated detection of sensitive data, AI-based classification, and automated assignment of business terms.
2. **Data discovery:** The process of searching, understanding, and evaluating data is usually encapsulated as data discovery, which is enabled via automated data cataloging. Apart from data discovery, which should also include the discovery of data transformation stages or AI artefacts, such as ML/DL models, data engineering pipelines, etc., AI/ML itself should be used for data discovery tasks. ML-augmented data discovery and visualization includes semantic search, data affinity and non-obvious relationship discovery, identification of relevant data objects and AI artefacts for corresponding business tasks, and automated discovery to monitor data quality.

² See Reference [2] for more details on AI-infused data governance.

3. **Data profiling:** The process of examining, analyzing, and checking the content of all data attributes of relevant data sources to get a first path in understanding data quality is referred to as data profiling. The outcome of data profiling tasks is metadata that is captured in a knowledge catalog. This data relates to quality, structure, content, and relationship of the underlying data sources. For structured and relational data, profiling includes determination of data types, column and domain analysis, primary key analysis of corresponding tables, and cross-domain and foreign key analysis. Especially for non-structured and non-relational data, profiling is overlapping with activity or user behavior, customer classification, etc. ML can be used to accelerate and automate data profiling tasks; even DL techniques can be deployed to calculate quality measures, for instance, of text data.
4. **Data access:** There are several AI-infused data access and data integration areas that further simplify and optimize data access with minimal data movement and high automation, such as self-service data access, ML-based query optimization, and ML-infused data abstraction layers where, for instance, data federation and virtualization is augmented with intelligent caching to allow for centralized access to data stored in disparate data sources including multiple clouds, semantic SQL, and confidence-based SQL statements.

5. **Automated data quality assessment:** Data can only be used by business organizations if its quality within a business context and the source it originates from are trusted and its content and structure is well understood. AI-infused data quality assessments³ enable automation, including calculations of data quality scores, detection of data anomalies and data drift, and to auto-analyze data quality issues with the goal to suggest remediation strategies for improved business consumption. Delivering reliable and trusted data in a timely fashion for business consumption is a continuous process.
6. **Entity matching**⁴: Establishing a single, trusted version of the truth (360-degree view) of core business entities, in particular persona data, for example, customers, citizens, employees, and business partners, is required to deliver business insight. Since these persona records are typically distributed across multiple systems and applications across multiple clouds with different identifiers, ML-infused entity matching techniques can help resolve these entities, complementing traditional MDM solutions.
7. **Digital exhaust:** Any Data Fabric architecture or Data Mesh solution that is not only comprised of data discovery, profiling, and access but also of data exploration and transformation tasks, development and operationalization of ML/DL models, and orchestration and management of data and AI

³See Reference [3] for a high-level overview on data quality and the connection to ML.

⁴We further elaborate on AI-infused entity matching or resolution in Chapter 8.

artefacts generates a digital exhaust, which can be leveraged to create additional metadata, which we call digital exhaust metadata. This is related to data lineage and data provenance events, data transformation stages, data knowledge graphs, data quality assessments, and shifts in key data quality measures. AI/ML can be applied to this metadata to gain additional relevant and actionable insight and to further optimize corresponding scenarios. For instance, shifts in data quality measures can be correlated to drift and degrading accuracy and precision of ML models during their entire lifetime.

8. **Semantic enrichment:** Whereas the digital exhaust is geared toward enhancing and further optimizing relevant scenarios themselves, semantic enrichment thrives for simplification and optimization of data consumption by applications and business users. Applying AI provides automated enrichment to contextualize data with semantic knowledge, for instance, based on knowledge graphs and other existing metadata in the catalog. Enriching the Data Fabric or Data Mesh with semantic knowledge can shield data and business users from the complexity of IT, for example, the heterogeneous source data landscape, different data access methods (SQL vs. NoSQL), and various data formats (relational, XML, JSON, text, etc.).
9. **Governing:** Data and information governance has existed for decades. Since this book does not provide a review of data governance⁵ in general,

⁵Please, see Chapter 15 for more details.

we treat the AI-related aspects of governance in the sections on cataloging, data quality assessment, entity matching, etc.

Figure 7-1 is a high-level illustration of the relationship or coherence of these key Data Fabric and Data Mesh areas, which we have elaborated on earlier.

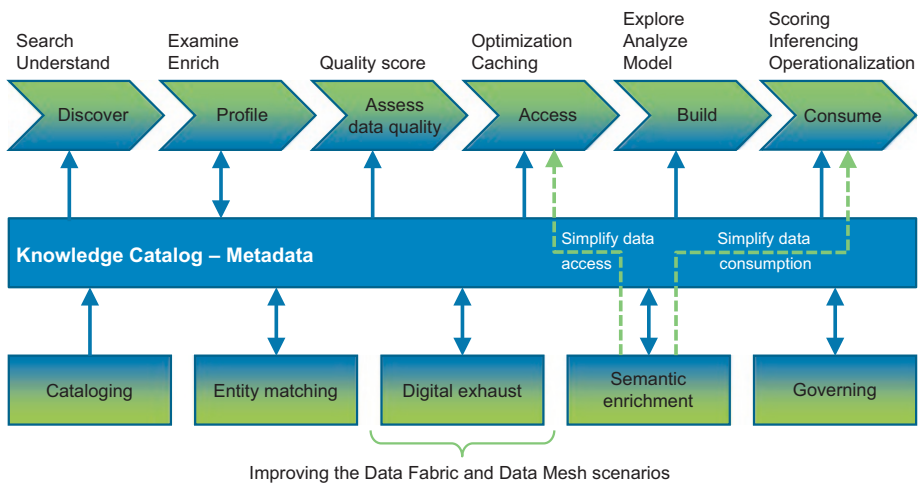


Figure 7-1. Relationship of Key Data Fabric or Data Mesh Areas

The knowledge or governance catalog, depicted as the middle layer, contains the required business, technical, and operational metadata. The layer above it represents the data consumer or business user, who discovers and profiles data, assesses data quality, and finally accesses the data to build business-related artefacts that can then be used to consume data in a business-relevant context. The layer below it represents key Data Fabric or Data Mesh tasks that are performed by data stewards, data or information governance professionals, or data engineers. These consist of registering or cataloging new data sources or other entities into the knowledge catalog, performing entity matching tasks (in addition to

a broader set of data quality measures), activating the digital exhaust, performing semantic enrichments, and conducting a broad range of data governance tasks.

Some of these tasks may be performed by different personas, depending on the usage scenario, for example, data consumption vs. data source registration. For instance, assessing data quality may be done by data consumers, business users, data stewards, or data governance professionals alike.

The outcome of the digital exhaust further improves scenarios for both concepts, whereas the outcome of the semantic enrichment can further simplify and optimize data consumption and data access.

The next section provides more details about these AI-infused topics.⁶ While doing so, our focus is specifically on the AI/ML aspects.

Let us begin with cataloging.

Cataloging

The knowledge or governance catalog is an essential component in a Data Fabric architecture and Data Mesh solution. By using the term *cataloging*, we refer to data-centric tasks, which all generate business, technical, or operational metadata that is stored in the knowledge catalog. These tasks are not limited to data itself, but refer to IT- or AI-related artefacts, such as ETL⁷ stages and process flows, ML/DL models, data pipelines, etc.

The following are a few examples:

- Registering a new data source, ETL stage, or AI artefact, for example, an ML/DL model
- Proposing and classifying new business terms for new regulations or laws

⁶ Additional AI-infused areas are discussed in Chapter 17.

⁷ ETL stands for Extract-Transform-Load.

- Discovering and creating new asset rules or policies, for example, related to data, ML/DL models, etc.
- Detecting sensitive data
- Profiling data
- Assessing data quality
- Assessing ML/DL model accuracy, precision, and other quality-related KPIs
- Generating and assigning new business terms

In addition to the preceding tasks, the operational environment may generate additional metadata, for instance, by continuously measuring data quality KPIs, for example, a data quality score, understanding drift and bias in ML/DL models, capturing data lineage or data provenance events, etc. Activating the digital exhaust and performing semantic enrichment⁸ generate metadata as well. Some of these tasks are performed in a certain sequence within the context of a cataloging process. For instance, registering a new data source may very well include the tasks to detect sensitive data and to generate and assign new business terms.

Figure 7-2 is an illustration of the *registering a new asset* process. We describe this process regarding a data source, IT asset (ETL stage), and AI artefact (ML model), whereby the focus is on the AI-infused augmentation. Let us discuss Figure 7-2 in more detail. The metadata of a new data source needs to be discovered automatically. This includes ownership, access methods (SQL, NoSQL, REST API, etc.), structured or unstructured data, read and/or write access, authentication requirements, and so forth.

⁸We elaborate on the digital exhaust and semantic enrichment in dedicated sections further in the following.

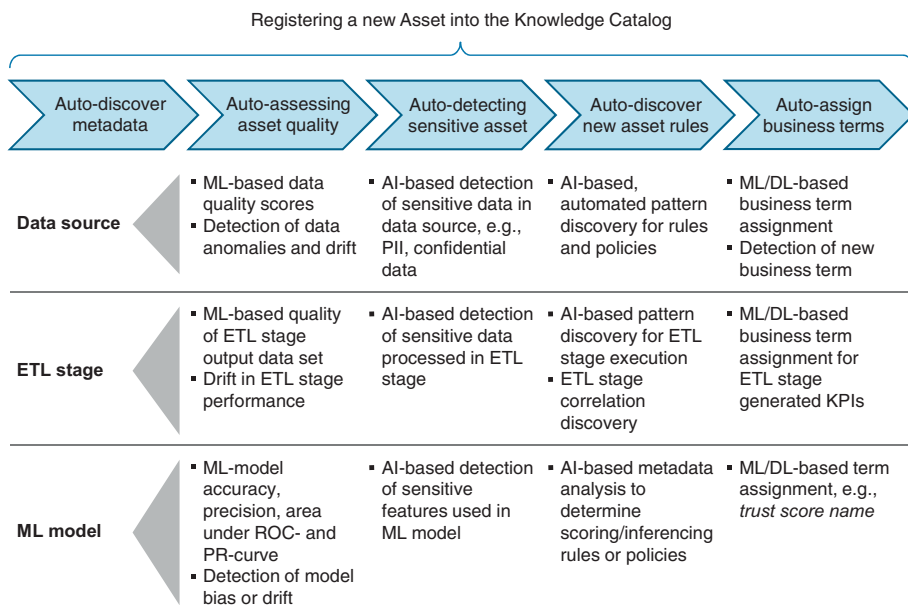


Figure 7-2. Registering a New Asset

The metadata of a new ETL stage includes ownership, execution pattern, source data required, downstream consuming systems, applications, languages used (e.g., SQL, Java, C++, etc.) or additional ETL stages that depend on the outcome of the ETL stage, and so forth. The metadata of a new ML model includes ownership, status of the ML model (e.g., trained, validated, tested, deployed, etc.), business purpose, required data sources, features required from those data sources, data pipeline required, and so forth.

A data steward needs to validate the cataloging steps and may even have to manually complete missing metadata.

Auto-assessment of the new data source quality includes AI-based generation of a quality score and detection of data anomalies, missing attributes, or data drift that is especially important for AI model development and adjustments. Detecting data drift may be a repetitive process.

The auto-assessment of the ETL stage includes ML-based quality evaluation of the ETL stage output dataset in terms of completeness and trustworthiness and drift detection in the ETL performance, the latter one being a repetitive process.⁹ The auto-assessment of an ML model quality is itself based on ML methods to measure, for instance, the ML model accuracy, precision, and areas under the ROC and PR curves¹⁰ and detection of model fairness (bias) and drift, the latter one being a repetitive process as well.

Auto-detecting sensitive data is based on AI-infused methods to discover, for instance, PII¹¹ or confidential data. A data steward may perform a final validation to either confirm or decline the recommendation from the AI model. For an ETL stage, AI-based detection of sensitive data that is processed in the ETL stage has to be performed. Likewise, for a new ML model, AI-based methods need to be applied to auto-detect sensitive features used in the ML model.

Auto-detecting new asset rules is an essential step in the registration process. For a new data source, this includes AI-based, automated pattern discovery for new data rules and policies or assignment of existing data rules or policies. For an ETL stage, this includes AI-based pattern discovery for the ETL stage execution or correlation discovery of a particular ETL stage with additional ETL stages, which results in corresponding rules or policies. A new ML model needs to be understood in terms of its scoring and inferencing needs. AI-based metadata analysis to determine a scoring or inferencing pattern generates either new or assigns existing ML model scoring- and inferencing-related rules or policies.

Auto-assignment of business terms related to a new data source includes ML/DL-based assignment of existing business terms that are already included in the knowledge catalog; it may also generate new

⁹ See Reference [4] for more information on measuring quality of ETL processes.

¹⁰ See Chapter 6 for more details on the ROC and PR curves.

¹¹ PII stands for Personally Identifiable Information.

business terms that are, for instance, derived from new regulations or laws. A new ETL stage may generate new KPIs, which aren't reflected yet in the knowledge catalog. ML/DL-based business term assignment for such a new ETL stage relates to those new KPIs.

Finally, an ML model generates outcomes, for example, a trust or confidence score and so on, which may be captured with a new business term or assigned to an existing one.

Let us move on to examine what is meant by AI-infused understanding of AI assets.

AI-Infused Understanding of Assets

In the previous section, we have broadened the scope above and beyond just dealing with data, incorporating ETL stages and ML models into our discussion.

Moving on, we continue to focus on data and ML models. This lives up to the expectations that today's enterprises are dealing with an ever-increasing number of diverse assets. When we use the term *asset* in this section, we refer to data and AI models, where we limit our scope to AI-infused understanding.

Figure 7-3 depicts the set of AI capabilities that are applied to the four process steps: *discovering*, *profiling*, *assessing quality*, and *accessing assets*. Some of the AI capabilities may be applied to several process steps, such as the quality assessment that is part of the profiling and quality assessment processes. The profiling process is limited to a quality score, whereas the quality assessment process determines quality KPIs at a much more detailed level, including recommendation to improve the quality of assets.

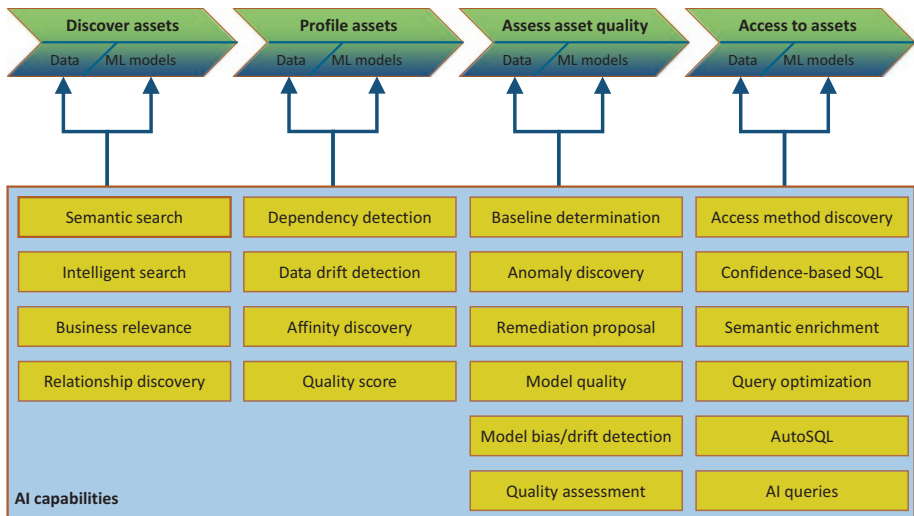


Figure 7-3. Overview of AI-Infused Understanding of Assets

We discuss these AI capabilities in the forthcoming four subsections. Let us begin with asset discovery.

Asset Discovery

Once a new asset – again, we stay focused on data sources and ML models and focus on the AI-infused capabilities – is registered in the knowledge catalog, we can assume the availability of necessary metadata in the catalog. This enables the business user and other data and AI asset consumers to conduct semantic and intelligent searches, to understand the business relevance, and to discover relationships of assets.

Semantic search¹² describes the business user’s or search engine’s intent to generate the most accurate search results possible by considering the search intent, query context, and relationship between relevant assets or entities. These entities can be structured data (e.g., relational data, XML

¹² See Reference [5] for more information on semantic search.

or JSON files, hierarchical data, CSV files) or non-structured data (e.g., textual data, image files, video and audio files). Thus, semantic search needs to address both types of data.

AI-based intelligent search¹³ helps surface information and answers that are specific to a business intent. Even if relevant assets have been identified via the available metadata in the knowledge catalog, semantic and intelligent searches over these assets (e.g., data, ML models, etc.) need to be performed to gain further insight and to validate the relevance for a particular business intent.

Discovering assets and their relationship needs to be performed considering the corresponding business relevance or business objective. This requires search tools to be equipped with understanding human language and the capability to learn and become more targeted via ML/DL. This scope of understanding assets reaches clearly above and beyond the discovery and profiling phases, where structured and unstructured data, ML models, and other assets need to be further explored and understood in a business context.

Let's move on to this profiling phase.

Asset Profiling

Once assets have been discovered, they need to be further understood in terms of their dependencies from other assets. For ML models, this means to understand the required features for scoring and inferencing, including systems and data sources those features are generated with and stored at, respectively. ML/DL techniques can be used to understand the affinity of an asset to other assets that are reflected in the knowledge catalog. Asset profiling - including affinity discovery - is not limited to analyzing the metadata in the knowledge catalog; discovered assets need to be analyzed

¹³ See Reference [6] for more information on intelligent search.

by looking at the assets directly. For instance, data assets need to be scanned to understand their content and structure, data type, primary and foreign key relationship, and also affinity to other assets.

Quality scores may have to be calculated using ML models and statistical samplings on a well-defined subset of the data, predicting the quality of the entire asset. In some situations, profiling on a subset of the data may not be sufficient; it thus may have to be done on the entire dataset. Data drift detection is a repetitive process that is particularly important considering the underlying data for ML models.

Calculating quality scores is the first step of a more pervasive asset quality assessment, which we discuss next.

Automatic Asset Quality Assessment

Understanding anomalies especially in data sources is a key area within the automated asset quality assessment. This requires the determination of a data baseline to really understand deviations from that baseline. AI/ML techniques can be used to determine the baseline of almost all relevant data components. Corresponding products need to come with prepared model templates and GUIs to facilitate these quality assessments and – most importantly – to minimize skill requirements for business users to determine baselines and discover anomalies. Sophisticated tools should also react to discovered anomalies by comparing them to known issues and subsequently proposing remediation strategies. This does not only apply to data anomalies, but to issues with other assets as well.

For instance, discovering degrading ML model quality measures, for example, model bias or drift, reduced ML model accuracy or precision, reduction of the area under the ROC and/or PR curve, etc., should be conveyed with recommendations or autonomously performed correction, such as retraining of an ML model or adjustments of hyperparameters. Quality assessment of any asset is a repetitive, ongoing process, not only executed when trying to understand an asset after its initial discovery.

Once assets are understood, they can finally be accessed or reaccessed. How AI can support this is the topic of our next subsection.

Asset Access

An ML-infused data abstraction layer should support the business user or asset consumer to discover the best access method. For instance, several data sources may have to be accessed considering a specific business purpose. The business intent may be indicated by leveraging the business glossary, specifying specific business terms and KPIs of interest.

Using ML methods under the cover could suggest corresponding methods to access the required data sources via data virtualization with federation and intelligent caching to provide a centralized access, which hides the complexity and diversity of the underlying set of data sources from the asset consumer. This simplifies and optimizes data access for the data consumer and reduces the appearance of data silos by providing an ML-underpinned convergence of data stores. Such an ML-infused data abstraction layer increases self-service data access needs and minimizes data movement.

Improving SQL query performance and optimizing resource consumption is another key dimension of data access. ML-infused query optimization in RDBMS¹⁴ can support an optimizer to learn from past experience in comparing chosen access paths and corresponding elapsed times of a particular SQL query to refine the access path with each execution – resulting in reduced query elapsed times, further optimizing resource consumption, and making critical insight available to the business in a much shorter time.

IBM Db2 AI for z/OS¹⁵ is an example, where AI/ML is used to optimize operational SQL query performance.

¹⁴ RDBMS stands for Relational Database Management Systems.

¹⁵ See Reference [8] for more information about IBM Db2 AI for z/OS.

Let us now discuss a few enhancements that are broadening the scope of standard SQL to enable additional use cases:

1. **AutoSQL:** The diversity of data stores and methods, such as data lakes, data lakehouses, relational data stores, DWH, potentially several knowledge catalogs, and streaming data, requires AI-based SQL capabilities to hide the business user from using multiple query engines and moving or replicating data. AutoSQL¹⁶ is a technology that automates the access, integration, and management of data for AI – regardless of where it resides and how it is stored – without having to move assets. In conjunction with data virtualization and intelligent caching capabilities, AutoSQL empowers business users to easily query data across hybrid cloud and multi-vendor environments.
2. **Confidence-based query matching:** Another means to improve data access is ML-infused confidence-based query matching, which improves query results and accuracy, even when the data resides outside of initial search parameters. This feature dramatically expands the range of possible data tasks that can be done without involving a data scientist.
3. **Semantic SQL:** Enhancing traditional SQL with semantic query capabilities allows easier access to data, without the need for costly ETL or getting data scientists engaged. It also enables additional

¹⁶See Reference [7] for more information on AutoSQL.

use cases that were not possible with the standard SQL. For instance, Db2 for z/OS with SQL Data Insights (SDI)¹⁷ leverages DL methods and extends the standard SQL to enhance the traditional data processing in a relational database. It extrapolates unsupervised learning to train neural network models for discovering, matching, and grouping records with similarities, dissimilarities, and clusters in Db2 for z/OS data. For example, by learning from a large amount of training data, SDI can infer hidden relationships across two different records that are traditionally not considered an exact match.

AI/ML for Entity Matching

One of the key challenges in any Data Fabric architecture and Data Mesh solution is entity resolution and matching for core information (master data). Since data is stored in different systems and applications, core entities, for example, customers, business partners, citizens, employees, products, services, etc., are usually represented quite differently, making it hard to match these entities to derive a trusted, complete, single version of the truth (golden record). This has been a known issue for decades.

AI/ML has found its way to tackle this challenge, complementing existing deterministic and probabilistic matching techniques, yielding much more accurate matching results. Since we discuss entity matching¹⁸ in much more detail in Chapter 8, this section provides just a conceptual view of the AI/ML-infused state-of-the-art entity matching concepts.

¹⁷ See Reference [9] for more information on SQL Data Insights.

¹⁸ We use the terms *entity matching* and *entity resolution* synonymously. The following terms are used as well: *data matching*, *string matching*, *object identification*, etc.

Figure 7-4 depicts the AI/ML-based matching engine in the middle box with its key AI/ML capabilities, such as nickname resolution, pattern discovery, similarity discovery, distance measuring, ML-based prediction, and a standardizer function. Various inconsistent but similar records (depicted on the left side) serve as input to the matching engine, which can be adjusted and tuned by setting various parameters (depicted on the right side), such as matching attributes, threshold sensitivity, match strength setting, and tuning of the underlying ML algorithms.

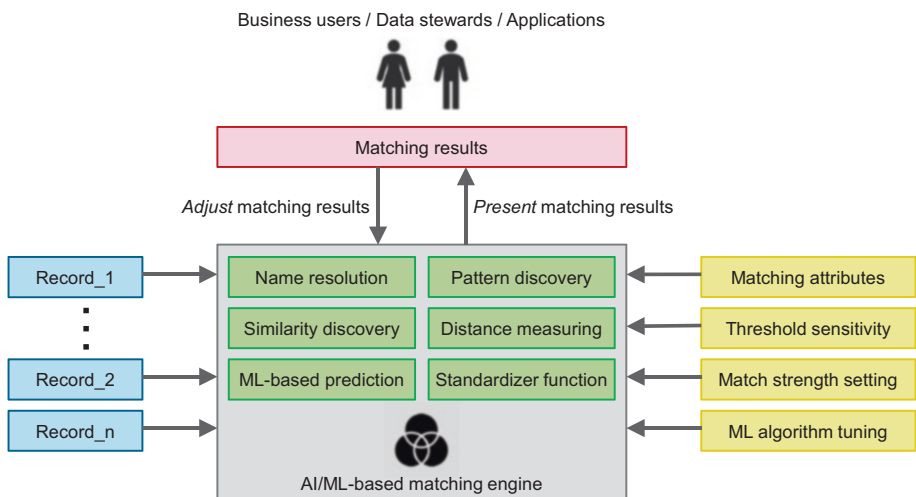


Figure 7-4. *AI-Infused Entity Matching*

The following is a list of the four parameters, which can be adjusted to influence the matching result.

1. **Matching attributes:** Need to be defined based on the record content. These attributes can, for instance, be names, addresses, nicknames, phone numbers, city names, etc.; they are used as input for the comparison to match incoming records to create master data entities.

2. **Threshold sensitivity:** Needs to be defined as lower and upper boundaries for non-matches of records, enabling fine-tuning of the matching engine.
3. **Match strength settings:** Can be used to set the similarity strength, for instance, with values between 0 and 10, and to see an estimate of how changes to attributes and threshold sensitivities affect the matching algorithm.
4. **ML algorithm tuning:** Influences the ML algorithms of the matching engine by setting distance measuring, standardization of entities, etc.

Matching results are presented to the business users, data stewards, or requesting applications. If needed, further adjustments can be triggered by these personas or applications by further tuning the matching engine via the preceding parameters and sending corresponding records back to the matching engine.

The following list describes the key AI/ML-based capabilities of the matching engine:

1. **Name resolution:** Names may be fully spelled out, abbreviated, or provided as nicknames. Name resolution may use graph representations to indicate relationship with other entities, such as persona, city, employer, or attributes, like profession, age, etc. This allows Graph Neural Networks (GNNs) to resolve names or nicknames. GNNs can also be used to discover additional insight that may, for instance, be relevant to prevent fraud.
2. **Pattern discovery:** A large number of records and attributes can either be automatically clustered according to discovered patterns or along a well-

chosen subset of attributes or attribute types. This simplifies and accelerates the discovery of similar entities.

3. **Similarity discovery:** The similarity discovery algorithms should be adjustable by the business user via match strength settings or weights. Different similarity algorithms may be used based on distance calculations, semantic similarity relations, GNNs, etc.
4. **Distance measuring:** Calculating the distance is related to similarity discovery, which can be facilitated via adjustable weights as well.
5. **Standardizer function:** Standardizers are used by the matching algorithm to convert the values of different attributes to a standardized representation that can be processed by the AI/ML-based matching engine. Multiple standardizers should be used by the matching algorithm depending on the specific attribute types found in the records.
6. **ML-based prediction:** Since there may be potentially millions of entities to be resolved, ML algorithms can be used on a much smaller subset of the data, for instance, 5-10%, to predict matching entities on the entire dataset or to support data stewards in further fine-tuning the matching engine parameters. This improves performance and reduces resource consumption.

Depending on the data type, other AI/ML methods and algorithms may be used for matching purposes. For instance, comparing and determining the similarity of two texts or determining whether two texts portray a similar opinion about a certain subject may be done by applying DL algorithms. This topic, however, is beyond the scope of this book.

AI/ML to Activate the Digital Exhaust

As you have seen in Chapters 3 and 5, a Data Fabric architecture or Data Mesh solution is comprised of a vast number of components that can be used to implement a variety of scenarios. Most associated tasks, regardless of whether they are performed by data stewards, business users, data engineers, data scientists, or other IT- and business-related personas, generate additional metadata in the knowledge catalog.

During the operations of systems, applications, and tools, additional data is generated. This digital footprint is related to the execution of ETL jobs and SQL queries, the scoring and inferencing of ML/DL models, changes to quality KPIs of data, ML/DL models, and other assets. Additional useful information can be found in systems and application LOGs, via insights from data lineage or data provenance events, data quality assessments, or data knowledge graphs.

This digital footprint may also be related to registering a new asset in the knowledge catalog, gaining additional insight about the data and AI assets, executing a data quality improvement or data matching process, exploring and transforming assets, and consuming or using an asset for a particular business purpose. The digital exhaust can be leveraged to create additional metadata, where AI/ML can be applied to this newly generated metadata to gain additional insight and to further optimize processes for both concepts.

The following are just a few examples of additional insight derived from the digital exhaust:

- Shifts in data quality KPIs can be correlated to drift and degrading accuracy and precision of some ML models during their lifetime.
- New data sources or other assets may have been registered in the knowledge catalog that could potentially be used to increase the relevance of BI reports or improve the accuracy of ML models.

- Correlation of LOG data to SQL query elapsed time behavior could yield additional insight regarding constraints in consumption of resources, for example, memory, CPU, I/O, etc.

Figure 7-5 depicts the knowledge catalog in the center, which contains the metadata related to registering, cataloging, understanding, etc., which we have already discussed in the previous sections. It also contains new types of metadata, which is derived from the digital exhaust data, which we call *digital exhaust metadata*.

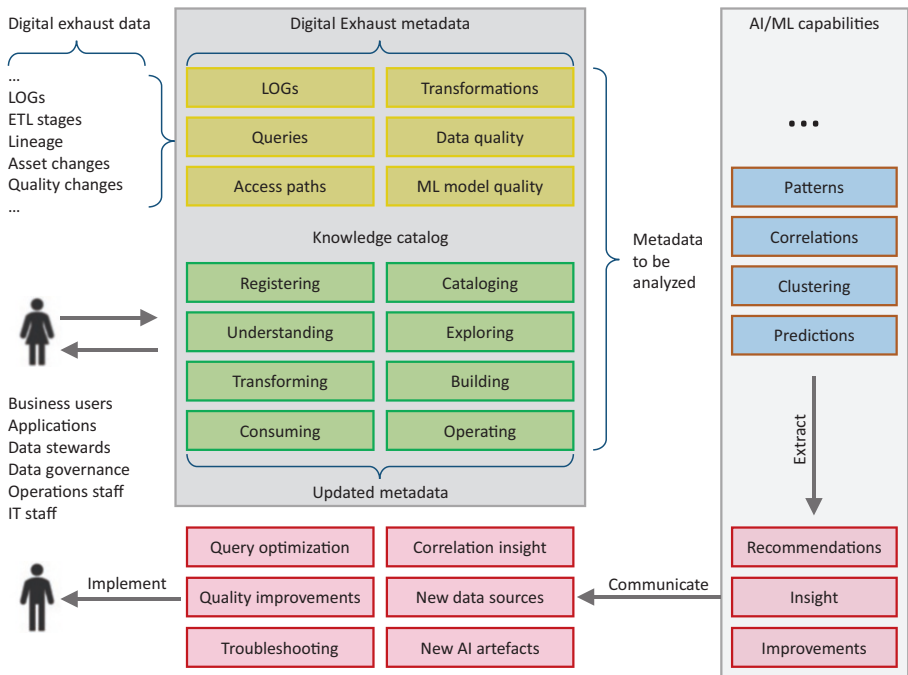


Figure 7-5. *Digital Exhaust Metadata*

This new type of metadata is generated by the digital exhaust data, which is composed of LOGs, executions of ETL stages and SQL queries, etc. This metadata needs to be generated based on the digital exhaust.

This can be done in a straightforward manner by simply storing, for instance, LOG data, data lineage events, SQL query elapsed times and resource consumption, SQL access path information, etc. in the knowledge catalog.

To some degree, existing metadata is updated as well. This combination of digital exhaust metadata and updated metadata needs to be analyzed via AI/ML capabilities to derive additional insight with recommendations to improve, simplify, and optimize corresponding Data Fabric or Data Mesh scenarios. AI/ML techniques are primarily used to discover patterns and non-obvious correlations or perform clustering and predictions.

For instance, ML methods can be used to predict data quality that is likely to degrade below a defined threshold or to predict improved SQL performance based on resource adjustments, for example, adjusting the size of buffer pools. The Data Fabric and Data Mesh solution should autonomously implement adjustments or provide recommendations that should be implemented.

Activating the digital exhaust may also yield additional business insight simply by correlating non-obvious data points. Of course, this depends on the scope and content of the digital exhaust data captured. For instance, if ML model scoring or inferencing events related to marketing campaigns are captured, this data could be correlated with data quality initiatives or availability of new data sources, possibly suggesting or autonomously implementing adjustments that can improve the accuracy of the marketing campaign.

Let us now elaborate on the final topic, which is related to using AI/ML for semantic enrichment.

AI/ML for Semantic Enrichment

Semantic enrichment¹⁹ is the process of adding meaning to data, which is represented as additional metadata in the knowledge catalog. The intent of semantic enrichment is to simplify and optimize some of the key Data Fabric and Data Mesh tasks, such as search and discovery of assets, access, and consumption of assets by applications and business users to build corresponding data products. Again, when we use the term *assets*, we refer to data, ML/DL models, and other AI artefacts. Applying AI to the semantic enrichment process provides *intelligent* and *automated* enrichment to contextualize assets with semantic knowledge mainly by using external data sources, for instance, based on knowledge graphs, domain- or industry-specific taxonomies or ontologies, business and IT glossaries, and other existing metadata in the knowledge catalog.

Enriching the Data Fabric and Data Mesh with semantic knowledge makes it easier for data and business users to search, discover, access, and consume assets and can shield especially business users from the complexity of IT, the heterogeneous source data landscape, different data access methods (SQL vs. NoSQL), and various data formats (relational, XML, JSON, text, etc.).

Semantic enrichment is often used in the context of tagging, indexing, classification, markup, annotation, etc. The challenge of semantic enrichment is to provide automation as much as possible, primarily via infusing AI/ML into the semantic enrichment process.

Figure 7-6 is a high-level depiction of the AI/ML-infused semantic enrichment engine, which is comprised of AI/ML capabilities, such as clustering, pattern discovery, correlation or relationship discovery, concept matching, etc., plus additional semantic enrichment capabilities, such as linguistic analysis, text analysis, tagging, indexing, etc. As it is depicted on the left side of Figure 7-6, the assets to be semantically

¹⁹ See Reference [10] for more information on semantic enrichment.

enriched could be data (structured or unstructured), ML/DL models or other AI-related artefacts, text, XML or JSON documents, data science-related pipelines, ETL stages, digital exhaust metadata, etc.

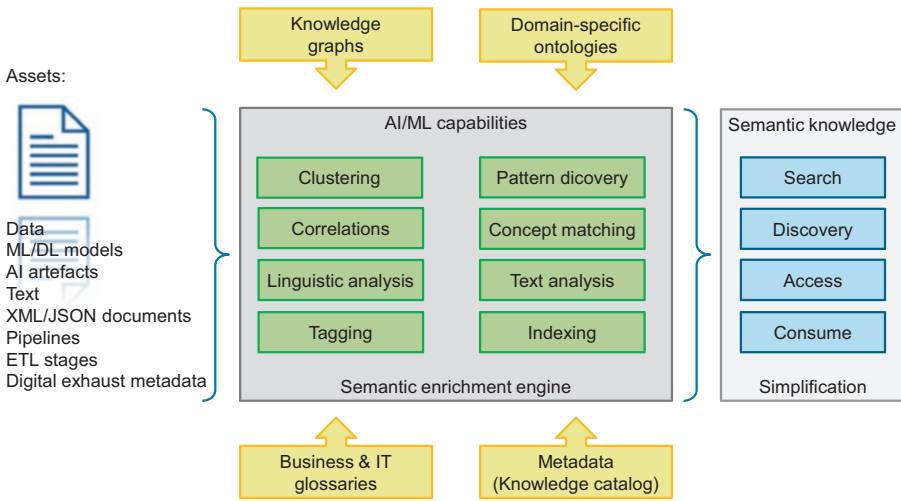


Figure 7-6. *Semantic Enrichment*

The semantic enrichment engine is using additional external data sources, such as knowledge graphs, domain- or industry-specific taxonomies or ontologies, and business and IT glossaries, to generate the semantic metadata or knowledge. Existing metadata that is already stored in the knowledge catalog may be used as additional input as well.

Once the semantic metadata is generated, it can then be used by business users to simplify and optimize Data Fabric and Data Mesh-related tasks, such as search, discovery, access, and consumption of assets, and – most importantly – to build data products in self-service fashion. In addition, the semantic insight should be explainable, actionable, and visualized²⁰ via a GUI.

²⁰ See Reference [11] for more information on visualizing semantic enrichment.

Key Takeaways

We conclude this chapter with a few key takeaways as summarized in Table 7-1.

Table 7-1. Key Takeaways

#	Key Takeaway	High-Level Description
1	AI/ML methods are essential to augment the Data Fabric and Data Mesh.	AI/ML-infused Data Fabric and Data Mesh democratizes access to data in a self-service fashion and enables <i>data-as-a-product (shopping-for-data)</i> as a key Data Mesh principle.
2	Registering a new asset should automatically generate metadata.	Registering a new data or AI asset into the knowledge catalog should be done in correlation with auto-assessing asset quality, auto-detecting sensitive data, auto-discovering new asset rules, and auto-assigning business terms.
3	Enhancing standard SQL with AI.	There are essential enhancements to standard SQL, such as AutoSQL, confidence-based query matching, and semantic SQL.
4	AI/ML should be used for entity matching.	AI/ML has found its way to tackle the entity matching challenge for core information (master data), complementing existing deterministic and probabilistic matching techniques, yielding much more accurate matching results.
5	AI/ML can be used to activate the digital exhaust.	The digital exhaust can be leveraged to create additional metadata (digital exhaust metadata), where AI/ML can be applied to this metadata to gain additional insight and to further optimize Data Fabric and Data Mesh scenarios.
9	The semantic enrichment engine should be enhanced with AI/ML.	The goal of semantic enrichment is to simplify and optimize some of the key Data Fabric and Data Mesh tasks, such as search and discovery of assets, access, and consumption of assets by applications and business users.

References

- [1] Sengupta, S., *The Role of AI and ML in Building a Logical Data Fabric*, 2021, www.rtinsights.com/the-role-of-ai-and-ml-in-building-a-logical-data-fabric/ (accessed June 23, 2022).
- [2] Hechler, E., Oberhofer, M., Schaeck, T., *Deploying AI in the Enterprise*, Apress, 2020, ISBN-13: 978-1484262054.
- [3] Talend, *Data Quality and Machine Learning: What's the Connection?* 2022, www.talend.com/resources/machine-learning-data-quality/ (accessed June 21, 2022).
- [4] Theodorou, V., Abelló, A., Lehmer, W., Thiele, M., *Quality Measures for ETL Processes: From goals to implementation*, 2016, <https://upcommons.upc.edu/bitstream/handle/2117/100626/16.CCPE.pdf> (accessed June 23, 2022).
- [5] IBM, *IBM Watson Explorer*, 2022, www.ibm.com/docs/en/watson-explorer/12.0.x (accessed June 24, 2022).
- [6] IBM, *Intelligent Search*, 2021, www.ibm.com/cloud/learn/intelligent-search (accessed June 24, 2022).
- [7] IBM, *5 Things You Need to Know About IBM's Next Generation Cloud Pak for Data and New Data Fabric*, 2021, <https://newsroom.ibm.com/5-Things-to-Know-about-Cloud-Pak-for-Data-and-New-Data-Fabric> (accessed June 25, 2022).

- [8] IBM, *IBM Db2 AI for z/OS*, 2022, www.ibm.com/products/db2-ai-for-zos (accessed June 25, 2022).
- [9] IBM, *IBM Db2 13 for z/OS and More*, IBM Redbooks, 2022.
- [10] Clarke, M., Harley, P., *How Smart Is Your Content? Using Semantic Enrichment to Improve Your User Experience and Your Bottom Line*, 2014, www.councilscienceeditors.org/wp-content/uploads/v37n2p40-44.pdf (accessed June 29, 2022).
- [11] Schlegel, A., Heese, R., Hinze, A., *Visualisation of Semantic Enrichment*, 2012, <https://cs.emis.de/LNI/Proceedings/Proceedings208/1047.pdf> (accessed June 29, 2022).

CHAPTER 8

AI for Entity Resolution

It is widely accepted that an organization's success is increasingly dependent on its ability to derive value from the data it has. However, many organizations are still stuck on the first step – understanding the data – especially as the volume and complexity of data continue to grow. Think about a simple question: *How many customers does your enterprise have?* For many businesses, providing an accurate answer to this question remains difficult.

The reason is quite simple: large organizations often have multiple departments, product lines, and sales channels that collect a wide variety of customer information. A customer may be registered with a different email address and be identified as a different user in multiple systems. Not to mention the new companies that come in through acquisitions. The format in which customer information is stored varies widely. Therefore, identifying the same customer across multiple systems is key for laying the trust foundation for analytics and AI for the enterprise and for implementing a sustainable Data Fabric architecture and corresponding Data Mesh solution.

This chapter is a deep dive into the third use case scenario (customer 360), which we have elaborated on in Chapter 3. It outlines novel AI-infused capabilities for entity matching, which can significantly improve trust in consuming core information.

Introduction

The data used to describe the core business entities of an enterprise is referred to as master data, managed via MDM systems. Enterprise master data has high business value, can be reused across business units within an enterprise, and is often distributed across multiple heterogeneous application systems.

There are many types of enterprise master data. The most common are suppliers and customers. In addition, industries may have specific types of master data.¹ Ecommerce companies must have a holistic view of channels and distributors for cross-selling and upselling. Healthcare requires a 360-degree view of patients for personalized treatment, healthcare services, and insurance plans. Manufacturers care about materials, parts, and prices, which are critical to their procurement.

For different business departments within an enterprise, their master data is also different. Customer information is undoubtedly critical for sales and marketing departments. Product information including product releases and categories is a core asset for R&D departments. The employee information, organizational structure, as well as departmental hierarchies are of primary concerns to the human resource department.

Having complete and accurate sets of master data helps enterprises streamline the process, improve customer experience, and optimize resources. However, in the real world, the same data can be represented in different formats by various systems because of spelling, abbreviations, languages, format, etc. Let's look at the example in Figure 8-1. In this example, *name*, *address*, and *phone number* are identity attributes. However, in the *Employee* table, *name* and *address* are composite attributes, while in the *Order* table, they are atomic attributes. But it's easy to map the composite attributes in the first table to the second table.

¹ See Reference [1] for more information on industry use cases.

Employee Table

First Name	Middle Name	Last Name	Primary Phone	Address Line 1	Address Line 2	City	Zip code	State	Country
MS. Jennifer	R	Gates	01-55-555-1234	123 1st Street		Oakland	94500	CA	US
Mr. Philip		Jones	+44(0) 1222 525555	2 Kennet Rd		Surrey	CR3 6YA		UK

Order Table

Name	Shipping Address	Country	Mobile Phone	T-Shirt Size
Jennifer Gate	123 1 st Street, Oakland, 94500	USA	555-555-1234	M
Dr Philip Jones	2 Kennet Rd, Surrey, CR3 6YA	United Kingdom	1222 525555	L

Figure 8-1. *Different Records for the Same Person*

The formats of the preceding relevant attributes are slightly different. The mobile phone in the *Order* table doesn't have the country code. The data value in the column *Country* is also presented differently. Nevertheless, Jennifer Gates and Philip Jones in both systems are most likely referring to the same customers, respectively, because of the similarity of corresponding attributes.

Going back to the question at the beginning of this chapter, *how many customers does the business really have?* If duplicate records are not removed from the same system or merged across different systems, the business may have far more customers than it actually has, which can lead to bad decisions or even serious business ramifications.

Let us begin with an introduction into entity matching.

Introducing Entity Matching

Entity matching² is referring to the process of recognizing the same entity (people, products, suppliers, etc.) in the real world across datasets from the same or disparate data sources. Matching within the same data source is mainly for the purpose of identifying duplicate entity records, where the similarity of the same attributes is calculated directly. Matching between

² See Reference [2] for more details on the definition of entity matching.

data sources is done to establish linkages between entities from different data sources and thus creates a comprehensive view of the entities. The calculation of similarity is a bit more complex. The similarity of attributes is first calculated, followed by the similarity computation of the values of similar attributes.

For example, the *Insurance* database as depicted in Figure 8-2 has a *Customer* table with the attribute *Name*. The values corresponding to the attribute *Name* are Bob Lyle and Robert Lyle. They might refer to the same person. The *Retail* database also has a *Customer* table with two attributes, *Last Name* and *First Name*. First, match the *First Name* and *Last Name* and *Name* in both tables. Then compare the record values of these attributes to see if the records are similar.

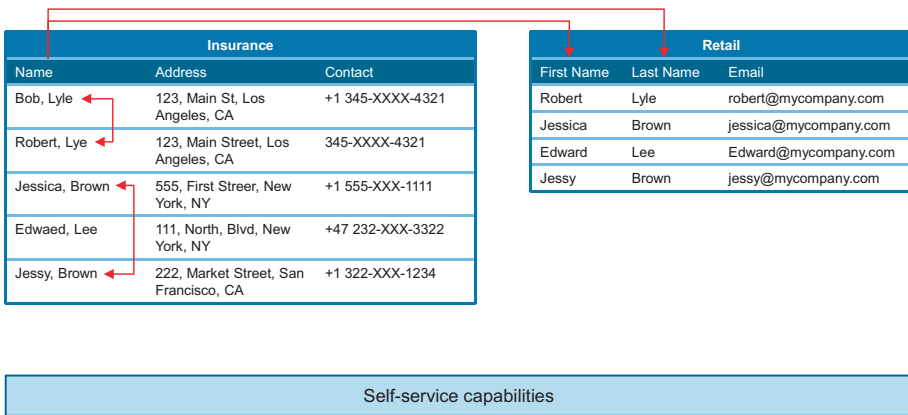


Figure 8-2. *The Illustration of Entity Matching*

What is somewhat ironic, however, is that entity matching itself has many variations of what it is called – entity resolution,³ record linkage, duplication, etc.

³See References [5], [6], and [7] for more on entity resolution.

Why does entity matching matter? With the explosive growth of data volume, enterprises are facing an increasingly severe data silo problem. The causes of data silos are both internal and external. First, when IT applications go live, they do not consider a unified data architecture, and the data models in the applications can vary, for example, the design of the customer table for the credit card system may be different from the deposit and loan systems. In addition, if data are outsourced, such as potential customer data for marketing activities, the definition of the prospect entity is likely to be different from the self-built systems. Lacking an effective way to identify the relationship between these records and to merge the records for the same entity, the value of this data will diminish greatly. The problem not only arises from the business side but also from the end user, where some information may be entered incorrectly or where a change of name, a change of cell phone number, or a move causes changes in personal information, resulting in inconsistent records from period to period.

Entity matching is the foundational technology that provides a single truth. Companies can use entity matching to see the whole picture: on the one hand, to customize the experience for customers and increase sales and, on the other hand, to continuously assess risk and identify suspicious behaviors in a timely manner.

Traditional Entity Resolution Approaches

Entity matching is generally considered to be a process consisting of multiple steps. Although the specific steps vary, they all include the steps depicted in Figure 8-3.

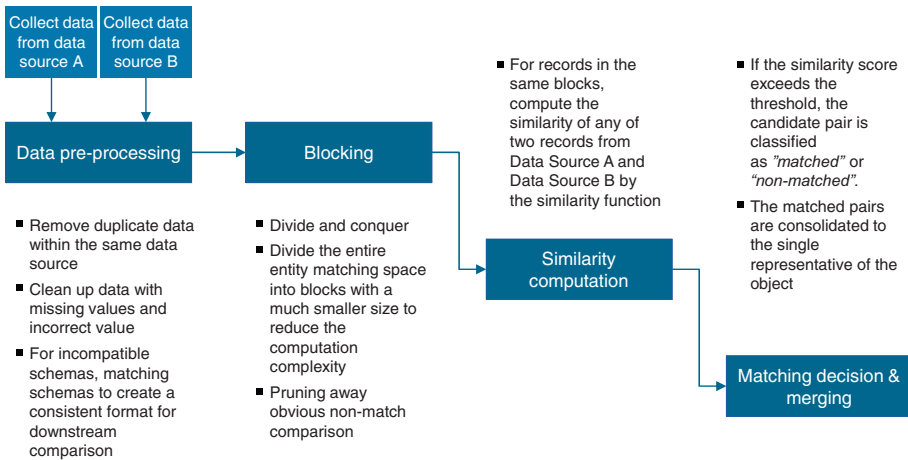


Figure 8-3. *The Reference Model of Traditional Entity Matching*

The first step is data preprocessing. The goal in this stage is to transform data into a consistent structured format with the same caliber, such as units, types, and abbreviations. Data preprocessing also includes schema matching. The aim is to identify attributes between data sources that should be compared with each other, essentially identifying semantically relevant attributes. For example, the *Address* column in one table should be linked to one or many columns (*Address Line 1*, *Address Line 2*, *Street Name*, *City Name*). The traditional way is to build a global dictionary that consists of all possible ways of a single attribute representation. In this case, the entry in the dictionary is (Key: *Address*, Value: *Address Line 1*, *Address Line 2*, *Street Name*, *City Name*).⁴

The second step is blocking,⁵ as shown in Figure 8-4. This phase is particularly important in the context of big data.

⁴ See Reference [3] for more details on schema matching.

⁵ See Reference [4] for more on blocking techniques.

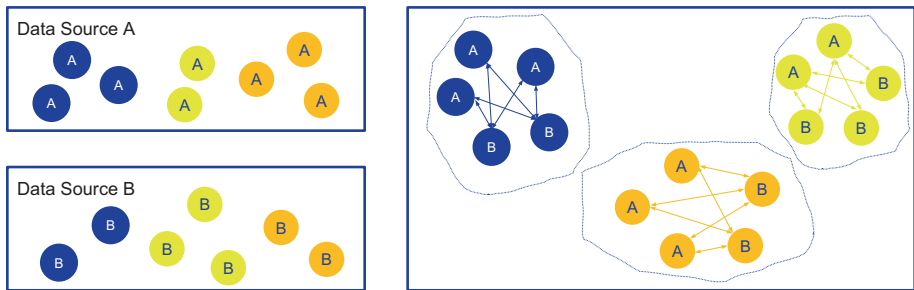


Figure 8-4. *The Illustration of Blocking*

The computational complexity of entity matching is related to the amount of the data sample. For example, if there are 10,000 customer records in a table from data source A and 100,000 customer records in a table from data source B, then a billion pairwise comparisons ($100,000 \times 10,000$) will be performed, which is very time-consuming. The idea of blocking is to reduce the complexity of the computation by dividing the data from disparate data sources into much smaller blocks and comparing only the data records within the blocks. This significantly reduces the complexity of the computation from a quadratic amount of total data to a quadratic number of blocks. The key steps in blocking as illustrated in Figure 8-4 are extracting tokens (combinations of attributes) from records and mapping the records to one or more blocks. There are many blocking algorithms.⁶ The two main categories are hash-based and sort-based. As the name suggests, hash-based blocking is mapped to blocks by tokens, while sort-based is sorting the tokens and taking a fixed-size window as blocks.

The next step is similarity computation. The similarity between two records is calculated by aggregating the similarity value of corresponding attributes on the assumption that attributes are independent. The core elements of this stage are the similarity function and attribute selection.

⁶ See Reference [4] for a blocking algorithms survey.

The classic method is rule-based. For example, *if two records have similar names and similar addresses, they refer to the same entity*. In this example, the selected attributes are *name* and *address*. *Similarity* is a probabilistic matching and is usually defined by a similarity function, such as the Jaccard similarity or index,⁷ which measures the similarity of two datasets by the percentage of the total number of common values and the total number of distinct values in two datasets, as illustrated in Figure 8-5.

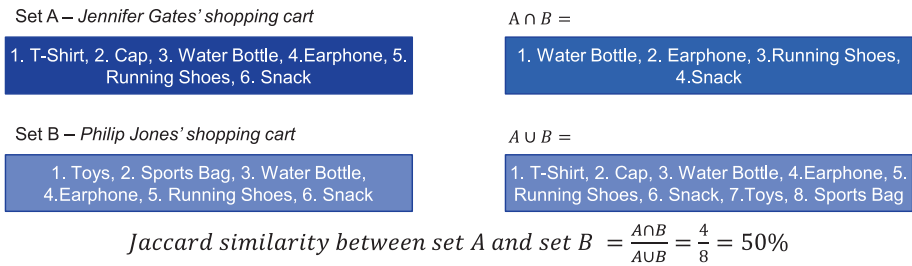


Figure 8-5. An Illustration of the Jaccard Similarity Between Two Sets

The rule-based method is straightforward and easy to implement. However, it usually requires deep domain knowledge to be constructed manually, and it's hard to adjust. In addition, it is not scalable. Once the problem space increases, rule-based systems quickly become unmanageable.

The final stage is matching decisions and merging, as depicted in Figure 8-6. The matching decision is made by comparing the similarity value with the attribute-level threshold, and the record-level threshold determines whether the two records match. Once two records are identified as the same entity, these two records are consolidated into a single representative of the object.

⁷ See Reference [8] for more on Jaccard indexes.

	Birth date	Customer Lifetime Value	Gender	Home telephone	Legal name		Primary residence			
					Given name	Last name	Address line 1	City	Postal code	State/Province value
4	1953-05-00	2685.901421	F	520-267-3352	JULIE	JACKSON	8388 SOUTH CALIFORNIA ST.	TUCSON	85708	AZ
	1953-05-00	8343.412804	F	520-267-3352	JULIE	JACKSON	8388 SOUTH CALIFORNIA ST.	TUCSON	85708	AZ
	1953-05-00	2685.901421	F	520-267-3352	JULIE	JACKSON	8388 SOUTH CALIFORNIA ST.	TUCSON	85708	AZ
	1953-05-00	5749.883799	F	520-267-3352	JULIE	JACKSON	8388 SOUTH CALIFORNIA ST.	TUCSON	85708	AZ

Figure 8-6. An Example of Merging

Use of AI to Resolve Entity Challenges

Many problems in the entity matching process can be solved by ML methods. Chapter 6 explains the difference between non-learning programming and ML. Instead of manually selecting the attributes/tokens and constructing the rules manually for entity resolution, the ML method learns an optimal solution for matching results from training datasets. Let us examine for each entity matching stage which ML method can be applied.

- Data preprocessing:** Schema matching is to find similar attributes between source schema A and target schema B. The problem can be translated to a classical clustering problem given the features of attributes in source A and target B. The features of attributes are extracted by metadata of the schema (column name, the type of data, unique constraints) and the statistics of data in the field (max, min, average for numeric data, etc.). With extracted features for each attribute, K-means and SOM (Self-Organizing Map) put attributes with similar features to clusters.

- **Blocking:** The essence of blocking is to put data records that are likely to refer to the same entity in one data block and discard dissimilar and irrelevant data as much as possible. With the training datasets having labels where matching pairs and non-matching pairs are marked, supervised learning helps select the attributes/tokens and the transformation functions that map matching pairs to the same block and non-matching pairs to disjunctive blocks.
- Even if the labeled data is not available either because no prior knowledge is available or labeling is too expensive, unsupervised learning can come into play, using clustering algorithms to facilitate subsequent supervised learning. There are several clustering algorithms that help to obtain approximate labels.
- **Similarity computation:** The learning goal is finding the best similarity computation method for known matched and unmatched data in the training set. SVM (Support Vector Machine) and decision tree are classification algorithms that are often used in practices to find the best similarity computation method.
- **Matching decision:** Learning the right thresholds for matching decisions is a supervised learning problem. Using different thresholds influences the matching decision.

With the rise of DL, a neural network has a wide range of uses in the entity matching process.⁸ Originally it is merely a classifier for a determination of whether the records are matching or not. Now it is expanding the role to address the challenges of every step in the process. Feature extraction, schema matching, blocking, and similarity measures are especially interesting in this regard since DL methods increasingly reduce the need for tedious handcrafting efforts. The basic DL methods in this space are Recurrent Neural Networks (RNNs).⁹

Active learning is also applicable for entity matching to overcome the challenges of lack of annotated data. The idea behind is that the active learner selects a small set of the most informative records from the unlabeled datasets for the expert to label and then learns an effective classifier from labeled data. The goal is to achieve high classifier performance with relatively low labeling costs.

Figure 8-7 is an illustration of the reference model for an AI-based entity matching process.

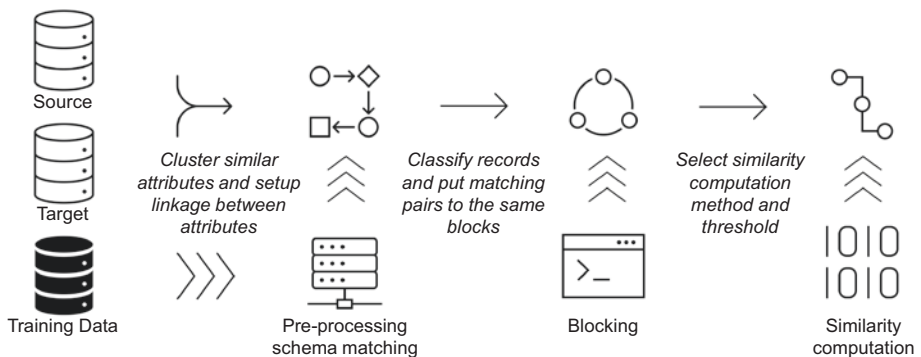


Figure 8-7. Illustration of an AI-Based Entity Matching Process

⁸ See Reference [9] for more on applying neural networks for entity matching.

⁹ See Reference [10] for more information on Recurrent Neural Networks.

The Benefits and Cost of an AI-Based Solution

There are several limitations to the traditional entity matching approach. Firstly, rule-based matching approaches completely rely on the knowledge of domain experts. However, this method is not very accurate and is somewhat subjective. For example, consider the rule: *if two records have a similar address value, then they are matched*.

Imagine the situation because if you are currently renting a place where a previous tenant had credit risks, then your credit will also be impacted assuming you are the same person, which is obviously unfair. Also, the cost of constructing rule-based is high and requires the heavy lifting of coding all the knowledge into rules for enumeration.

Similarly, the traditional approaches require manual efforts to select attributes for schema matching and blocking. This is not just labor-intensive but also error-prone. The selection of the appropriate blocking strategy and the ad hoc adjustments make this task nontrivial. If the manual selection strategy is not optimal, very likely similar records are assigned to different blocks, which leads to a high rate of missing pairing, or too many unrelated records are assigned to the same block, resulting in a waste of computational resources.

AI-based solutions go a long way to solving these challenges. Firstly, AI learns based on historical data, breaking through the limitations of expert experience and also unearthing potentially hidden patterns from the data to achieve optimal solutions. At the same time, the use of ML or DL can automate time-consuming and laborious tasks to the greatest extent possible and will also reduce human error.

However, all advantages come at a cost. In general, the execution time of AI-based solutions is significantly worse than that of non-learning methods and not scalable for large datasets. This is especially true for DL, which requires specific hardware acceleration. Also, the effectiveness of

ML approaches is known to depend on the provision of training data. It takes a lot of time to receive and clean the data, not to mention the need to correctly label the data for a high-quality model. This all adds to the difficulty of building an AI solution.

The distribution of data and the positive and negative instances have a great impact on the accuracy and fairness of models. Last but not the least, the use of DL also faces the problem of interpretability. While there are many benefits to AI-based solutions, there are also costs associated with them that need to be weighed carefully.

Considerations for MDM Solutions

As mentioned earlier, entity resolution is the core of MDM. Having understood how entity resolution works, let's return to the MDM solution. Without a doubt, one of the most important considerations in choosing a master data solution is the accuracy, efficiency, and automation level of entity resolution. Ideally, the solution should provide rich built-in models for attribute mapping and record mapping and automate the entire matching process as much as possible when data is ingested.

See one example in Figure 8-8. When the *Auto Insurance Customers_shaped* table is ingested, the columns are automatically mapped to built-in data models.

Auto Insurance Customers_shaped
 Last mapping update: Apr 26, 2021, 7:13 PM

Auto map Profile

Find an attribute or field Filter by: All columns (17)

CUSTOMER	NAME	COUNTRY	LATITUDE	LONGITUDE
Mapped Mapped to: Customer Key	Mapped Mapped to: Legal name - Full name	Mapped Mapped to: Primary residence - Country value	Mapped Mapped to: Primary residence - Latitude degrees	Mapped Mapped to: Primary residence - Longitude degrees
AA71604	Janine Cockshot	US	33.599728	-111.98813
AB96670	Wynnie Dunnnett	US	39.2781	-120.1203
AC58002	Waylan Trelevan	US	47.76121	-122.3464
AE23906	Brynn Jurgensen	US	34.16851	-118.615569
AG22225	Mariellen Wippermann	US	39.74740931	-104.999675
AH65907	Gretta Rown	US	33.656757	-112.351061

Figure 8-8. Auto-mapping

Moreover, a generic model may not be applicable because of the different domains in which companies operate. This requires an MDM solution to provide capabilities to allow users to bring their own customized models and have the flexibility to map the attribute of the model to data columns, as depicted in Figure 8-9.

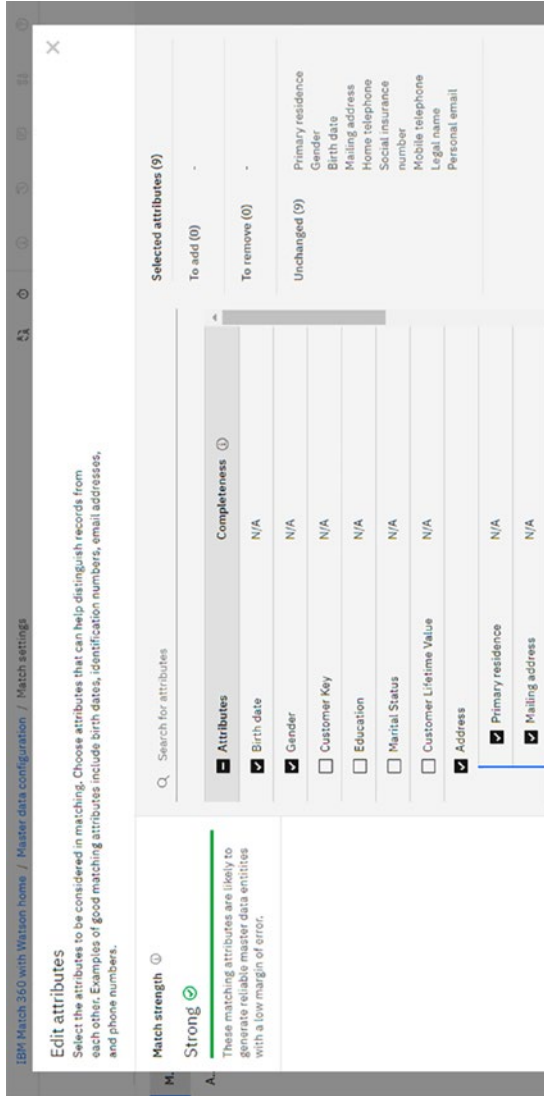


Figure 8-9. Customized Model

In addition to the core functions, the upstream of the MDM system should be tightly integrated with the enterprise data architecture, allowing streamlined access to different types of data sources, to either ETL, virtualize, or replicate.

In the meantime, it needs to be tightly integrated with the enterprise asset catalog. Master data, as an important asset, is published in the enterprise asset catalog for various departments within the enterprise to query and consume.

Key Takeaways

We conclude this chapter with a few key takeaways as summarized in Table 8-1.

Table 8-1. *Key Takeaways*

#	Key Takeaway	High-Level Description
1	There are many types of enterprise master data.	The most common master data are suppliers and customers. There are also industry-specific master data – <i>channels</i> and <i>distributors</i> for ecommerce, <i>patients</i> for healthcare, <i>materials and parts</i> for manufacture.
2	Entity matching is referring to the process of recognizing the same entity.	Matching within the same data source is mainly for the case of duplicate entity records in the data source. Matching between data sources is for better correlating entities between different data sources for a more comprehensive view of the entity.
3	Entity matching itself has many variations.	It is called entity resolution, record linkage, duplication, etc.

(continued)

Table 8-1. (continued)

#	Key Takeaway	High-Level Description
4	Entity matching is a process consisting of multiple steps.	It consists of preprocessing to clean data and match schemas, blocking to reduce the comparison complexity, and similarity computation and merging for outcome.
5	Many problems in the entity matching process can be solved by ML methods.	Essentially entity matching is a classification problem for labeled data and a clustering problem for unlabeled data. SVM, K-means, and decision tree are common algorithms.
6	Entity matching is a key capability for Data Fabric and Data Mesh.	A Data Fabric architecture and Data Mesh solution needs to include an entity matching capability for core information to guarantee trust in core information, that is, customer, partner, supplier, citizen, etc.
7	A MDM solution is more than an entity matching engine.	The capabilities of allowing users to bring their own models and have the flexibility to map the attributes of the model to data columns are also important.

References

- [1] Goyal, S., *An introduction to Entity Resolution – needs and challenges*, <https://towardsdatascience.com/an-introduction-to-entity-resolution-needs-and-challenges-97fba052dde5> (accessed August 22, 2022).
- [2] Talburt, J. R., *Entity Resolution and Information Quality*, Morgan Kaufmann, 2011, ISBN-13: 978-0123819727.

- [3] Sahay, T., Mehta, A., Jadon, S, *Schema Matching using Machine Learning*, 2019, <https://arxiv.org/pdf/1911.11543.pdf> (accessed August 22, 2022).
- [4] Papadakis, G., Skoutas, D., *Blocking and Filtering Techniques for Entity Resolution: A Survey*, <https://dl.acm.org/doi/abs/10.1145/3377455> (accessed August 22, 2022).
- [5] Altwaijry, H., Sharad, M., *Analysis-Aware Approach To Entity Resolution*, <https://escholarship.org/uc/item/0nc1x8nf> (accessed August 22, 2022).
- [6] Chistophides, V., Efthymiou, V., Palpanas, T., Papadakis, G., Stefanidis, K., *An Overview of End-to-End Entity Resolution for Big Data*, <https://dl.acm.org/doi/abs/10.1145/3418896> (accessed August 22, 2022).
- [7] Kopcke, H., Thor, A., Rahm, E., *Evaluation of entity resolution approaches on real-world match problems*, www.researchgate.net/publication/220538608_Evaluation_of_entity_resolution_approaches_on_real-world_match_problems (accessed August 22, 2022).
- [8] Statistics How To, *Jaccard index*, www.statisticshowto.com/jaccard-index/ (accessed August 22, 2022).
- [9] Barlaug, N., Gulla, J. A., *Neural Network for Entity Matching: A Survey*, <https://dl.acm.org/doi/10.1145/3442200> (accessed August 22, 2022).
- [10] Salel, F. M., *Recurrent Neural Networks: From Simple to Gated Architectures*, Springer, 2022, ISBN-13 : 978-3030899288.

CHAPTER 9

Data Fabric and Data Mesh for the AI Lifecycle

With the development of AI in technology and business, AI is no longer an experiment limited to a select few data scientists. It will penetrate all aspects of enterprise business operations and continue to innovate and optimize for new business scenarios. Now the focus shifts from the competition of AI algorithms to how to combine the strength of expert teams and AI technology for the actual needs of the enterprise and industry to generate business value.

To put data and AI into production means enterprises not only need to create AI models but also operationalize AI workloads, which means practicing AI effectively and efficiently and, more importantly, doing it in a way that instills confidence in the outcome. Therefore, it is crucial to establish a verifiable AI full lifecycle management system within the enterprise.

Let us dive into this topic by introducing the AI lifecycle.

Introduction to the AI Lifecycle

AI and software engineering have fundamental differences. Traditional software is rule-driven, where the programmer translates the solution to a problem into clear logical rules, while AI is more data-driven, which is based on training sets of data and a set of selected algorithms. Traditional software decomposes the problem into components, modules, and functions till the lines of code. It always has a deterministic output for each building block, and all building blocks orchestrate to produce deterministic outputs for the systems for the same input.

AI works in a very different way. Without explicit rules defined, the AI model is trained to find an approximation of the optimal solution to the problem. The efficiency of the solution depends on the quality of the data used for training, the effectiveness of selected features, and the sophistication of the algorithm. It implements using a *stepwise approximation plus search* strategy to find a set of parameters (models) to minimize the loss function.

Therefore, AI engineering is a whole new field. According to Gartner technology trend 2022,¹ “AI engineering automates updates to data, models, and applications to streamline AI delivery. Combined with strong AI governance, AI engineering will operationalize the delivery of AI to ensure its ongoing business value.” In essence, AI engineering is a collection of methods, tools, and practices that expedite the entire AI lifecycle and ensures the efficient delivery of AI models that are robust, trustworthy, and interpretable and that continue to create value for the enterprise.

¹ See Reference [1] for more information on the Gartner Top Strategic Technology Trends for 2022.

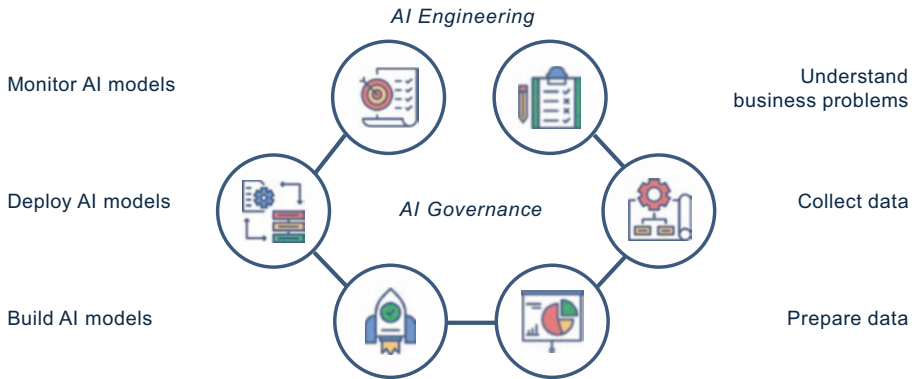


Figure 9-1. *The Stages of the AI Lifecycle*

Although there are multiple reference models² of the AI lifecycle in the industry, most of them comprise the following common stages from conception to maintenance, as shown in Figure 9-1:

1. **Understand the business problem:** Data scientists learn from domain experts to understand business problems, research the necessity and feasibility of AI, and define key metrics of the AI project – not only performance metrics of the model itself but also service-level metrics³:
 - a. **Service latency:** The time it takes to load models and prepare required features
 - b. **Inference latency:** The time the model takes to make a prediction for a given input
2. **Collect data:** Data scientists request data access that is required for AI projects. This stage is often the most time-consuming and labor-intensive

² See References [2] and [3] for more information on reference models of the AI lifecycle.

³ See Reference [4] for more information on service-level requirements.

if the enterprise has not established a trusted enterprise-level data foundation for analytics. Data scientists not knowing what kind of business data is available in what format may struggle to make specific data requests, so much so that the process may be repeated several times until the required data is finally obtained. Therefore, it's not an overstatement that "74% of AI adopters say that data access is a challenge." New industry regulations are making this even more challenging.

3. **Prepare data:** When datasets are available, data scientists start to wrangle, explore, and cleanse datasets. The challenges in this stage for data scientists are the following:
 - a. *How to identify flaws in the data?* Since datasets are from disparate data sources and often produced by different business applications, datasets have various types of quality issues, for example, missing values, duplicated records, inconsistent values, etc.
 - b. *How to visualize large datasets?* Visualization helps data scientists identify outliers in the data, understand the statistics of datasets, and prepare for the feature engineering in the next stage.
4. **Build models:** This stage requires the creativity of data scientists. First, data scientists extract features from datasets, and quite often data scientists derive new features from raw data by aggregation or transformation for better results of prediction. Second, data scientists build models by splitting datasets between training and testing,

training models including HPO (Hyperparameter Optimization), and evaluating models. This is an iterative process.

5. **Deploy models:** This is where ML engineers take control from data scientists. It may involve the reimplementing of models in a scalable fashion because of service-level requirements defined in the first stage. There are several examples:
 - a. Rewrite the Python model into Java/C++ for better performance.
 - b. Rearchitect the model to run in a parallel way.
 - c. Build a feature store for the features less likely to change.
 - d. Deploy the models to the environment close to the data source (data gravity).
6. **Monitor models:** After the model is deployed in a production environment, ML engineers continue to monitor the quality or accuracy of the model and drift, which is the drop in accuracy and in data consistency over time. The degradation of predictive performance triggers pullout of models from the production environment and retraining with recent datasets. In highly regulated industries, the monitoring stage comprises fairness and explainability.
7. **Govern models:** This stage is to capture necessary information and calculate risk scores on an ongoing basis to ensure enterprises govern the creation and adoption of AI throughout the entire lifecycle.

For example, facts and lineage provide metadata tracking of underlying datasets and algorithms. This is very useful to become enterprise-ready for any regulatory requirement that may arise.

AI engineering is the operationalization of AI models to create a sustainable, repeatable, and measurable operationalization process. It involves data engineers, data administrators, data scientists, ML engineers, business analysts, and operations engineers. The goal is to empower every data-related role in the enterprise, regardless of their background and skills, to collaborate closely and smoothly to deliver the full value of AI investments and improve time-to-market.

Key Aspects: DataOps, ModelOps, MLOps

When it comes to operationalizing AI, it is important to understand a few concepts, namely, DataOps, ModelOps, and MLOps. Like DevOps, systematic software development that aims to deliver software from code to production rapidly with high quality and on a continuous basis, DataOps follows the same principles and practices but applies them to data. It is designed to accelerate the collection, processing, and analysis to produce high-quality data for data citizens to fulfill their needs in a compliant way.

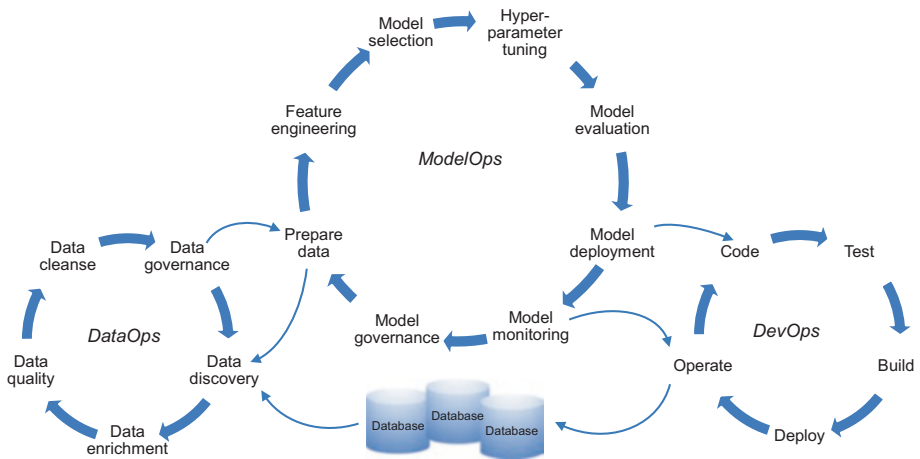


Figure 9-2. *DataOps, ModelOps, and DevOps*

As shown in Figure 9-2, DataOps covers two stages in the AI lifecycle: *Collect Data* and *Prepare Data*. The implementation of DataOps streamlines the processes in these two stages by automating data tasks into data pipelines. A Data Fabric architecture and Data Mesh solution⁴ is an ideal way to implement DataOps practices.⁵

ModelOps was coined at almost the same time as DataOps. It's a combination of AI with analytical models and DevOps, designed to bring some of the proven capabilities of agile software engineering to AI and the analytics space. ModelOps is a DevOps-like framework and set of toolchains and processes that bring together data engineers, data scientists, developers, and operators to accelerate the delivery of models from data preparation to model production through an effective enterprise data and AI strategy while continuously monitoring the models and retraining the models when needed and instilling the trust throughout the entire process. It covers all stages in the AI lifecycle presented in Figure 9-1.

⁴Please review Chapter 5.

⁵See Reference [5] for more information on DataOps.

MLOps is a subfield of ModelOps. The primary goal is to help data scientists with rapid prototyping for ML models and fast delivery of ML models to production systems. MLOps focuses on the automation of tasks specific to ML, as outlined in Figure 9-2: feature engineering, HPO, version control, model evaluation, and finally deployment of inference models at scale. It increasingly employs tools with interpretability, transparency, security, governance, and reproducibility of experiments to incorporate ethics and remove bias throughout the entire ML lifecycle.

ModelOps and MLOps are often used interchangeably. They are very similar in terms of capabilities. But ModelOps has a broader scope including not just ML models but also knowledge graphs, rules, optimization, and natural language techniques and agents, while MLOps focuses on ML model operationalization only, according to Gartner.⁶

As presented in Figure 9-2, the data preparation task in ModelOps triggers DataOps pipelines. The output of DataOps, which is high-quality governed data, goes into the next stage of ModelOps. So where does the output of ModelOps go? The answer is *business application*. Typically, there are several ways to integrate the predictions from ML models with business applications. The first option is offline inference, where the prediction results are stored in a database and the application gets them directly from the database. The second is batch prediction, which makes inference on a set of records on a regular basis, for example, weekly, monthly, or quarterly. This is commonly used for accumulated data and situations that don't need immediate results. The third option is online inference, where the model is called via an API to make predictions. Usually, online inference is expected to have results instantly. Also, the prediction requests can queue up by the message broker and be processed later.

⁶See Reference [6] for more information on the difference between ModelOps and MLOps.

Either way, the model inference is consumed by business applications. Therefore, DevOps also comes into play. Both new deployments of models and anomalies detected during the monitoring phase trigger the DevOps pipelines and actions accordingly. DevOps practices are essential to ensure that AI models can be deployed and injected into the business workflow of the enterprise. The model repositories need to be built for AI model lifecycle management, champion testing, and system testing, and model rollout/rollback mechanisms need to be set up to ensure availability, backed by CI/CD (Continuous Integration/Continuous Deployment).

In summary, AI engineering consists of three pillars: data, model, and code. To achieve best practice in AI engineering, it is recommended to adopt a platform that can provide capabilities for DataOps, ModelOps, and DevOps or integrate seamlessly with external Ops frameworks without supporting all three.

The following are some motivational aspects to implement DataOps, MLOps, and ModelOps in the context of the AI lifecycle:

1. **Complexity of data domains:** Data challenges that enterprises are faced with are not just the explosive growth of data but the complexity of the data domains and the heterogeneity of data sources from hybrid cloud and multi-regions. Data silos are becoming an increasingly serious problem. The advent of Data Fabric and Data Mesh concepts is aiming to resolve this issue by providing smart integration capabilities to help users decide to virtualize, replicate, or transform data depending on various factors, for example, policies, performance, latency, etc. Especially Data Mesh solutions with their organizational and federated approach are geared toward breaking data silos with data source and data ownerships.

2. **AI governance:** One critical aspect of implementing DataOps is to address the need for data and AI governance and privacy. Data Fabric and Data Mesh provide a knowledge-augmented central catalog that contains
 - a. An inventory of data assets with enriched business semantics
 - b. A set of governance artifacts including business glossaries, regulations, data privacy policies, data protection rules, and privacy data classifications
3. **Knowledge catalog:** The Data Fabric architecture with its knowledge catalog capabilities automatically enriches data assets via data discovery and data integration and enforces quality and protection rules throughout all activities in the *data preparation* stage.
4. **Democratization of data and AI:** Furthermore, one key benefit of DataOps is data and AI democratization. The intelligent catalog and semantic search enable everyone in the company to find the data they need to perform their job. In the context of the AI lifecycle, it greatly improves the collaboration and communication among data scientists, data engineers, and IT specialists and reduces the time for *Collect Data*.
5. **Data and AI orchestration:** Last but not the least, the core of DataOps lies in orchestration, which is responsible for moving data and AI between different stages in the pipeline and instantiating the data tools that operate on it. It also monitors

progress and issues alerts for specific data problems. Data Fabric and Data Mesh provide a unified pipeline that composes an end-to-end workflow by reusable data pipelines, which is the core objective of DataOps orchestration.

To illustrate DataOps and ModelOps further, let us dive into a couple of case studies.

Case Study 1: Consolidating Fragmented Data in a Hybrid Cloud Environment

The data needed to build models is scattered across multiple data sources and often across multiple clouds. According to IDC's State of the CDO 2021 study,⁷ data fragmentation and complexity is the number one barrier to digital transformation in 2022. Nearly 80% of organizations surveyed are storing more than half of their data in hybrid cloud infrastructures. Seventy-nine percent of organizations are using more than 100 data sources, and 30% are using more than 1,000 data sources. However, 75% of organizations do not yet have a complete architecture to manage a set of end-to-end data activities, including integration, access, governance, and protection.

One of the ultimate goals of a Data Fabric architecture and Data Mesh solution⁸ is to achieve a single source of truth, which asserts enterprise-wide data coverage across applications that is not limited by any single platform or tool.⁹ This creates both technical and process challenges for groups seeking to access and explore their data. The technical challenges arise from the logistics of extracting data from multiple sources or if

⁷ See Reference [7] for more information on the CDO 2021 study.

⁸ See Chapter 8.

⁹ See Reference [8] for more information on Gartner's vision for data and analytics.

that data is stored in different formats. Making data usable requires a significant amount of time and effort from highly skilled data engineers. More complex are the organizational-level restrictions around who is accessing what data and for what purpose. This is particularly difficult in industries such as healthcare and finance, where sensitive data needs to be handled with care and often with strict regulatory requirements. At the same time, people struggle with analyzing data without replicating it. In the majority of analytics projects, multiple copies of the data are stored in different locations and formats, which creates additional issues such as cost, latency, untrustworthy data, security risks, and more.

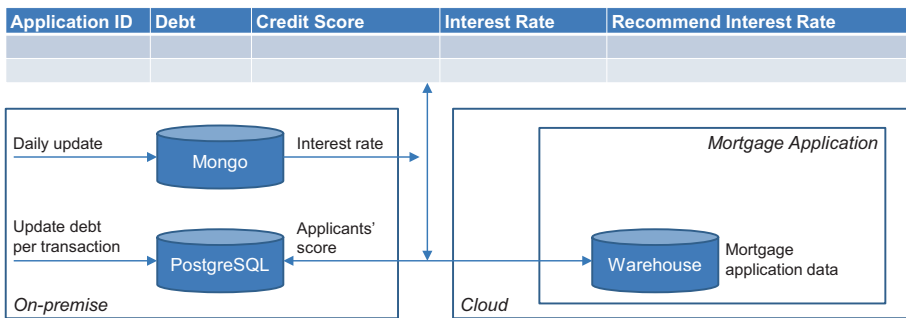


Figure 9-3. *The Data Architecture for a Mortgage Application*

As most organizations struggle with data fragmentation, there is a strong need for a unified enterprise data architecture. This section presents you with a case study to integrate data from multiple sources with the implementation of a Data Fabric architecture. Data and sample projects used in the case study can be found in IBM Cloud Pak for Data Gallery.¹⁰

¹⁰ See Reference [9] for more information on data and sample projects in IBM Cloud Pak for Data Gallery.

In this case study, a bank in the United States wants to offer a smart mortgage service for California residents. The interest rate on the loan is based on a combination of the applicant's personal credit score and the latest interest rate regulations. To implement this service, data engineers in this bank need to collect all key information about the applicant and recommended rates. The key information is spread in different database systems as depicted in Figure 9-3:

- The anonymized mortgage application data and mortgage applicants' PII (Personally Identifiable Information) are stored on a cloud data warehouse.
- The mortgage applicants' credit score data is stored on a PostgreSQL database.
- The interest rate is subject to market change and is scheduled to be refreshed daily, where the latest interest rate data can be retrieved from a MongoDB database.

Data engineers need to find mortgage applicants' credit scores, filter the data by state (to only include records from California), calculate the total debt for each applicant, and then merge credit score ranges into a data asset to be consumed by the data science team.

As explained in the previous section, being able to collect and prepare data with good quality is critical to building a data pipeline. It is important to understand the metadata of the ingested data. Sample data previews can help us understand the values within the data, and statistics and visualizations can help us determine a strategy for connecting multiple datasets.

From the data preview, as depicted in Figure 9-4, it is found that the column ID can be used to connect application data with PII data. It's also found that the column STATE_CODE can be used to filter the data instead of using the state name (which requires an additional step for data

transformation). It's noticeable that the mortgage applicant's credit score data uses a different code in the column ID, which cannot be used as a key for further connections. So the column EMAIL_ADDRESS is chosen instead.

Let's see how to build data pipelines to consolidate fragmented data from disparate data sources, which is a goal of DataOps.

ID	NAME	STREET_ADDRESS	CITY	STATE	STATE_CODE	ZIP_CODE	EMAIL_ADDRESS	CREDIT_SCORE	
1	ET86431	Madeline Augie	1420 Beaumont Avenue	Beaumont	California	CA	92223	mauge1@home.pl	717
2	ET86432	Rosa PAYS	222 North El Dorado Street	Stockton	California	CA	95202	rpaysp@homestead.com	820
3	ET86434	Tiphane Paquet	1002 Divisland Rd	Harlingen	Texas	TX	79152	tpaquet54@mpg.org	569
4	ET86436	Gaylor Haburne	12597 Sunset Hills Rd	Reston	Virginia	VA	20190	ghaburnem@gnv.uk	721
5	ET86438	Adolph Skitch	1001 W 75th Street	Woodridge	Illinois	IL	60557	askitch4@ox.ac.uk	430
6	ET86440	Diamond Dunn	15175 Whittier Blvd.	Whittier	California	CA	90602	odundo@sunnews.com	795
7	ET86442	Augustina Garnell	1454 DS South Foothill Drive	Salt Lake City	Utah	UT	84108	agarnell7@buzfeed.com	794
8	ET86433	Janine Cockshot	12602 N Paradise Village Pkwy	Phoenix	Arizona	AZ	85032	jackshotp@wikimedia.org	464
9	ET86444	Martaine Blackledge	208 LAKEWOOD CENTER MALL	Lakewood	California	CA	90712	mblackledge2@ted.com	708
10	ET86446	Josephine Southern	1021 Third Avenue	New York	New York	NY	10021	jsouthern7y@1and1.de	588
11	ET86448	Gaylor Crosetti	1799 Marlow Road	Santa Rosa	California	CA	95401	gcrosetti2a@1and1.de	672
12	ET86450	Martalen Wippermann	1406 Lanier St	Denver	Colorado	CO	80202	mwippermann72@examiner.com	824
13	ET86452	Myron Bewshaw	2005 North Ocean Blvd	Myrtle Beach	South Carolina	SC	29577	mbewshawck@timesonline.co.uk	434
14	ET86425	Myrryn Morris	15 S. PROSPECT AVE.	Park Ridge	Illinois	IL	60068	mmorrisbm@worldpress.com	430

Figure 9-4. Preview Sample Data

First, adding the amounts of the loan and credit card debt gives the total debt of an applicant. Then query the interest rate table in MongoDB with the credit score to find the corresponding interest rate. Finally, generate recommended interest rate for mortgage applicants. The whole data pipeline looks like Figure 9-5.

However, interest rate data keeps changing. Loan and credit card debt amounts are updated monthly. Credit scores are calculated by other applications and pushed to the PostgreSQL database daily. Interest rates are updated daily. If the tasks of joining of data, the calculation of total debt, and querying of interest rates are separated and run independently,

there's no guarantee the result keeps the expected level of accuracy. Therefore, the pipeline, as depicted in Figure 9-5, needs to be deployed as one job and is scheduled to run regularly. That ensures that the data request of interest rate is fulfilled with up-to-date data.

High-quality ML models require high-quality data. ML pioneer Andrew Ng believes that focusing on the data quality that powers AI systems will help unlock their full power.¹¹ Business leaders like Gartner believe that low-quality data induces high cost and undermines business.¹² Data only delivers business value when it is correlated and can be accessed by any user or application in the organization. Organizations looking to improve data quality typically start with a data and analytics governance program that consolidates fragmented data from hybrid cloud environments. When implemented properly, Data Fabric and Data Mesh help ensure that this value is available throughout the organization in the most efficient and automated manner.

¹¹ See Reference [10] for more information on why Andrew Ng advocates for data-centric AI.

¹² See Reference [11] for more information on the impact of poor-quality data on business from Forbes.

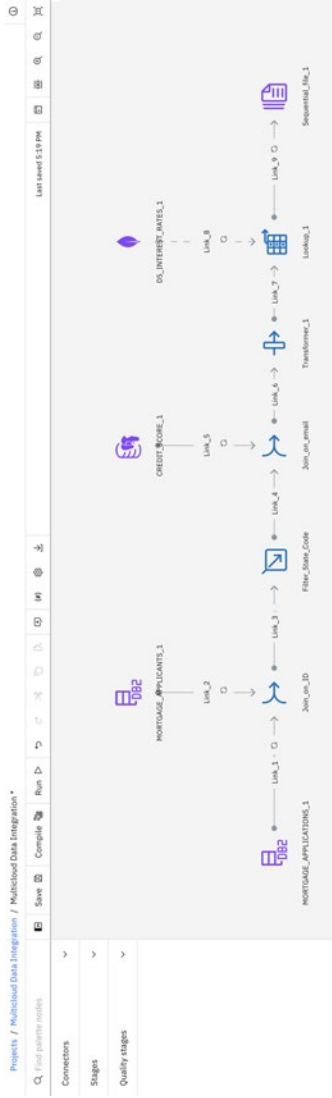


Figure 9-5. Create Data Pipelines

The ideal implementation of both concepts puts together self-service data tools into an intelligent fabric of a heterogeneous data landscape. It provides everyone in the organization with the ability to find, explore, and interrogate all available data, whether on-premises or in a hybrid cloud landscape.

Case Study 2: Operationalizing AI

Operationalizing AI refers to AI lifecycle management, which introduces integration between the data engineering team that builds the data pipeline, the data science team that builds the AI models, and the operation team that deploys and maintains the AI models. Operationalizing AI and AI engineering are often interchangeable. As explained in the previous section, AI engineering involves the core management competencies of ModelOps, DataOps, and DevOps to enable organizations to improve the performance, scalability, interpretability, and reliability of AI models while delivering the full value of AI investments.

In the mortgage business, change is constantly happening based on changing regulations, products, and processes. It is imperative that customers get up-to-date information and timely support to streamline their home-buying experience. The bank in our case study wants to have an expansion of its business by offering low-interest-rate mortgage renewals for online applications. The task for the data scientist team is to train a mortgage approval model to predict which applicants qualify for a mortgage and deploy the model for real-time evaluation based on the applicant's requirements.

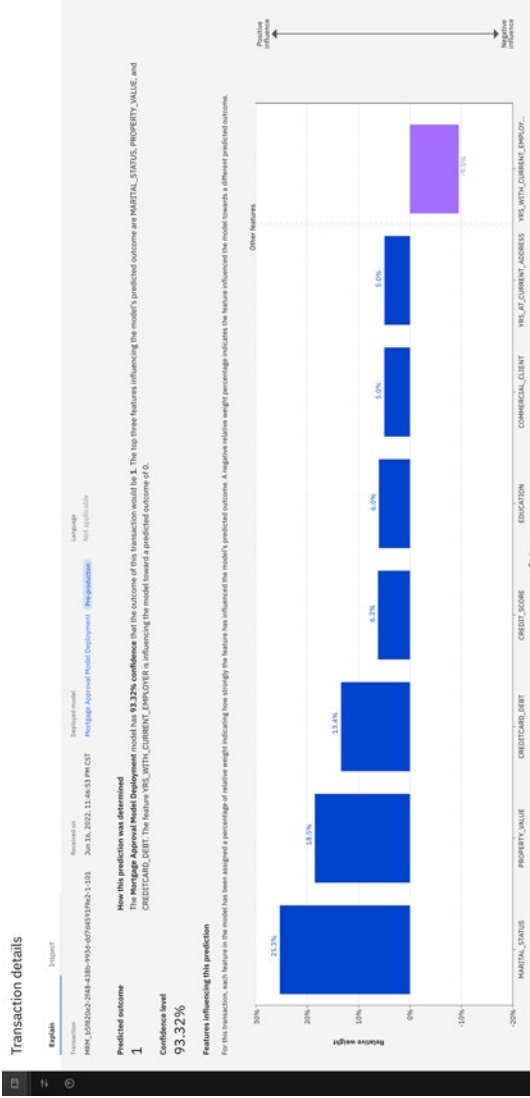


Figure 9-6. Explain How Prediction Is Being Made in Plain English

There are a wealth of algorithms for training the model. Once it's available, the data scientists save it to the project and use a holdout dataset to evaluate the model. Figure 9-6 explains in plain English why the model makes a prediction with a high degree of confidence. When the performance of the model is satisfactory, it can be taken to production.

Moreover, the status and production performance of the models can be monitored at any time from the model inventory, as shown in Figure 9-7.

There are a few challenges when operationalizing AI. The most common one is that deploying AI models into production is expensive and time-consuming. For many organizations, over 80% of the models have never been operationalized. While data science teams build many models, very few are actually deployed into production, which is where the real value comes from. For many organizations, the time it takes them to build, train, and deploy models is 6–12 months.

The issue of AI bias has attracted increasing attention in public. Drift occurs as data patterns change, which leads to a reduction in the accuracy of each model's predictions. When this happens, line-of-business leaders are increasingly losing confidence that their models are producing actionable insights for their business. Fairness is also an area of concern. If the model has produced favorable predicting results for specific groups (gender, age, or nationality), then it can lead to AI ethics discussions and possibly even legal risks.

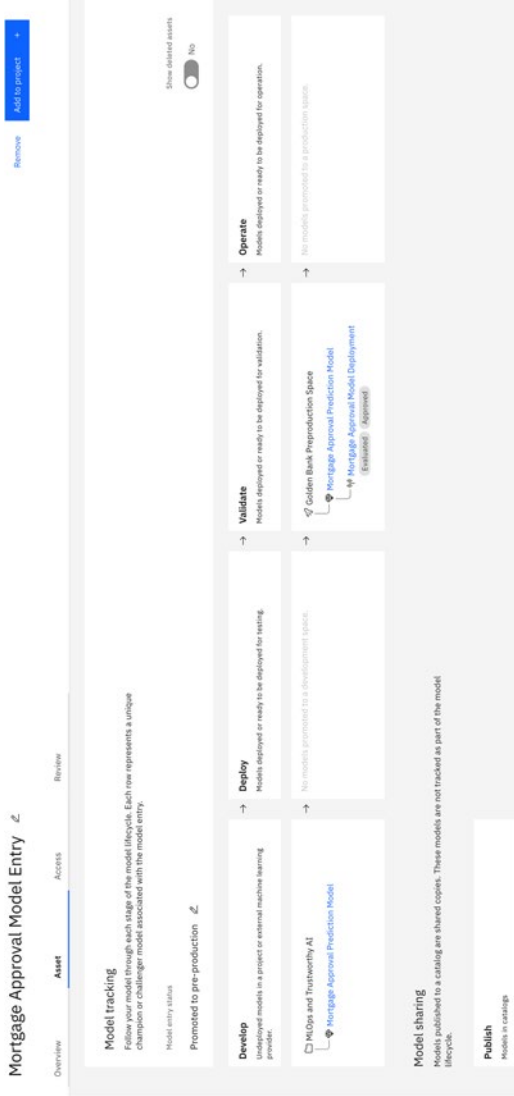


Figure 9-7. The Status of the Model in the Model Inventory

Another AI trust issue that affects model deployment comes from the lack of model lineage analysis. This includes two aspects. One is how the model is built and which features have a decisive role in the final scoring results of the model. This is the area where the interpretability of the model comes into play. The other area is data lineage: where the data used to train the model comes from, whether it is accurate and secure, and whether there is a possibility of tampering.

The fact sheets shown in Figure 9-7 are an example to help business users understand and trust the model.

These challenges need to be considered when an organization chooses a Data Fabric and Data Mesh implementation. The goal of acquiring data is to use it for a particular business purpose. Therefore, the best solution is to have the capabilities to operationalize data and AI implemented within the Data Fabric architecture. It helps organizations reduce the skills required to build and manipulate AI models, speed up delivery time by minimizing mundane tasks and data preparation challenges, and, at the same time, optimize the quality and accuracy of AI models with real-time governance.

Accelerate MLOps with AutoAI

There is a chasm between using Jupyter notebooks to develop models for experimentation and deploying them to production systems. While enterprise investment in AI has been increasing, the percentage of AI models that are delivered to production is still small. Enterprises are recognizing that crossing this chasm has become critical to realizing the value of AI. Due to the success of DevOps, many enterprises are looking to MLOps to solve this problem. As it is explained in the earlier section, MLOps is a set of practices that connect data preparation, model creation, deployment, and monitoring, with a focus on operating ML models effectively.

The success criteria for MLOps are to establish a sustainable set of disciplines for enterprises to roll out experimental models to production smoothly. Listed in the following are challenges that need to be dealt with when implementing MLOps practices.

The first is data preparation. Since ML models are built on data, they are very sensitive to the quality of data, such as the semantics, quantity, and completeness of the data being used to train the model. However, data preprocessing can be very time-consuming, depending on the conditions of the data. As explained in Chapter 6, a sequence of data understanding and data preparation tasks need to be performed. Here are some of the most common examples:

- **Feature selection:** Discard features that are less important to prediction. First, exclude the features that have a constant value throughout the dataset. Second, if the data type is not time or date, ignore the columns that have a unique value, which very likely represents an ID.
- **Missing or incorrect values:** For the missing values or inaccurate values in the datasets, one option is to use a statistical approach to estimate – the mean, median, or average of adjacent records. Another option is to use an ML algorithm to predict the missing or inaccurate value.
- **Feature encoding and scaling:** There are many transformation methods based on data types. For example, you can encode categorical features as ordinal numbers and scale numerical features to see how they affect the performance of the model.

The next phase is model creation. Based on the characteristics of datasets and the nature of business problems, there are a wealth of ML algorithms to use. In reality, the choice of algorithms needs to balance between the accuracy and the time spent for training. Another thing to

consider is the metrics used to evaluate ML models, such as ROC and AUC for binary classification¹³; F1, precision, and recall for multi-class classification; mean squared error (MSE), root mean squared error (RMSE), and R2 for regression; etc.

Feature engineering also plays an important role in model creation. ML algorithms deal with tabular data, and if relationships exist between different features, using data transformation to derive new features and combine them will help reveal the insights hidden in the data.

While these previously described tasks are complex, the approach to when and what method is needed to solve a problem is clear. This lays a good foundation for automation. Many vendors offer AutoAI solutions to automate the preceding tasks.

AutoAI is a no-code/low-code platform that automates several aspects of the MLOps lifecycle. As shown in Figure 9-8, AutoAI provides an easy-to-follow wizard to help you define the model training settings and to choose the target (predicted) feature. AutoAI also provides HPO capabilities that help optimize the hyperparameters of the best-performing pipelines from the previous phases. It uses a model-based, derivative-free global search algorithm, called RBFOpt,¹⁴ which is tailored for the costly ML model training and scoring evaluations.

For each folding and algorithm type, AutoAI creates two pipelines optimizing the algorithm type using HPO: the first optimizes the algorithm type based on the preprocessed (imputed/encoded/scaled) dataset, and the second one optimizes the algorithm type based on optimized feature engineering of the preprocessed (imputed/encoded/scaled) dataset.

The final model can be selected from the set of candidate models. Once the model is built and exported to a Jupyter notebook, the data scientist can further customize the results, which is often required for more complex use cases.

¹³ Please, refer to chapter 6 for an explanation of ROC, AUC, etc.

¹⁴ RBFOpt is an open source library for black-box optimization with costly function evaluations.

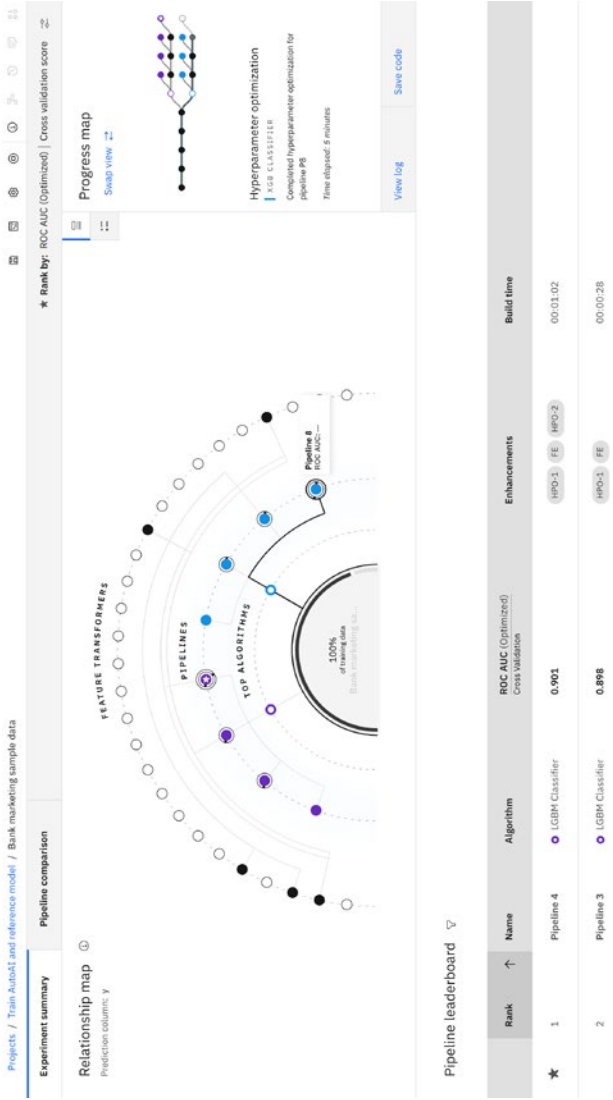


Figure 9-8. AutoAI

AutoAI automates the highly complex process of finding and optimizing the best ML models, features, and model hyperparameters for training.¹⁵ It allows people without deep data science expertise to create models of all types and even those with deep data science expertise to prototype and iterate from them more quickly. AutoAI reduces the effort of building models and increases productivity and accuracy. It provides significant productivity gains for enterprises implementing MLOps.

Deployment Patterns for AI Engineering

When it comes to the training and deployment of AI models, IT often defines the requirements that data science practices must comply with. For example, if training data are stored in a public cloud, model training using that data very likely also must take place in the same cloud to minimize data export (outbound) costs or to comply with governance rules. There are many other factors around model training and deployment, including security, latency, performance, and data residency. Here are a few examples:

- Meet the service-level requirement, for example, response time, latency, and throughput:
 - Co-locating with the data source for easy access to features
 - Co-locating with the applications to reduce network overhead
 - Easy scale-out to accommodate massive inference calls

¹⁵ See Reference [12] for more information about the benefits AutoAI could bring to MLOps and the AI lifecycle.

- Optimize cost for deployment related to hardware, integration, and operation cost:
 - Reuse existing computational resources.
 - Reduce efforts of integrating with existing applications.
 - Reduce operational costs by reusing existing operations infrastructure.

Depending on the importance of these factors, enterprises employ one or more of three common patterns when deciding on ML deployment architecture. The first pattern is to deploy the runtime environment of the ML model to the platform where the data originated.

Business-critical applications in large enterprises are currently deployed in highly reliable and regulated environments. For instance, two of three large financial institutions in the world are running their core banking systems on IBM zSystems. As a result, a large amount of training data for ML and the raw data needed for inference after the ML models go live are still generated and stored on-premises. Moving this data from a secure environment to the public cloud not only introduces latency but also increases the risk of data leakage and data tampering.

The deployment pattern in Figure 9-9 can be an option for organizations that do not want to take legal risks and potential financial losses, but still want to benefit from flexible, scalable, and cost-effective computing resources on the public cloud.

First, the data is masked according to the data protection regulations and migrated to the public cloud platform through intelligent integration technology as the dataset for model training. When the model training is completed, it is deployed to the on-premises system. To obtain better

scalability, it is recommended to use container-based deployment. In addition, multiple versions of the same model can be deployed for subsequent AB testing¹⁶ and gray release.

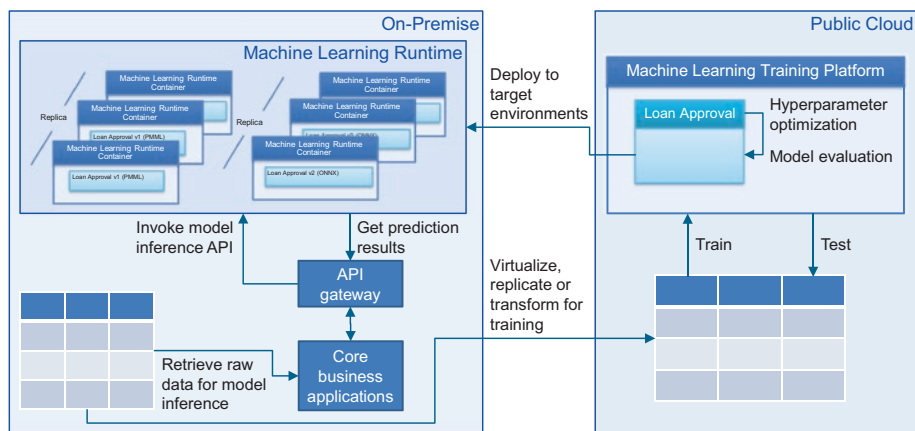


Figure 9-9. Pattern 1: Co-locate ML Runtime with Data for Easy Access of Features

There are several ways in which models and applications can be integrated. To continuously monitor and easily update models, invoking inference APIs of models from applications is the most common way of integration. Considering the scale of the workloads in production systems, which may reach up to 10,000 per second, enterprises usually use an API gateway for load balancing.

In this pattern, the application directly obtains the raw data needed to invoke the model APIs and then sends the request for model inference through the API gateway, which forwards it to the runtime container of the corresponding model version based on the specific information of the request. If too many inference requests are received at the same time, the API gateway may create a new replica of the runtime with the specific model version to maintain the service level for the model.

¹⁶A/B testing is a method of comparing two versions of a web page or app.

The core idea behind this deployment pattern is to train models on public cloud with regulated data from on-premises but deploy models back to on-premises to easily access raw data, thereby improving the performance and quality of model inference. One potential drawback of this pattern is the massive amount of scoring requests that may have an impact on business-critical applications that reside in the same system. Another disadvantage is that skills and toolchains for operationalization are more complex due to cross-platform deployments.

In contrast with data gravity, the second deployment pattern is a cloud-native one. Assuming most data for training and scoring are generated on-premises and modernized applications are running on public cloud, enterprises can consider the deployment pattern presented in Figure 9-10.

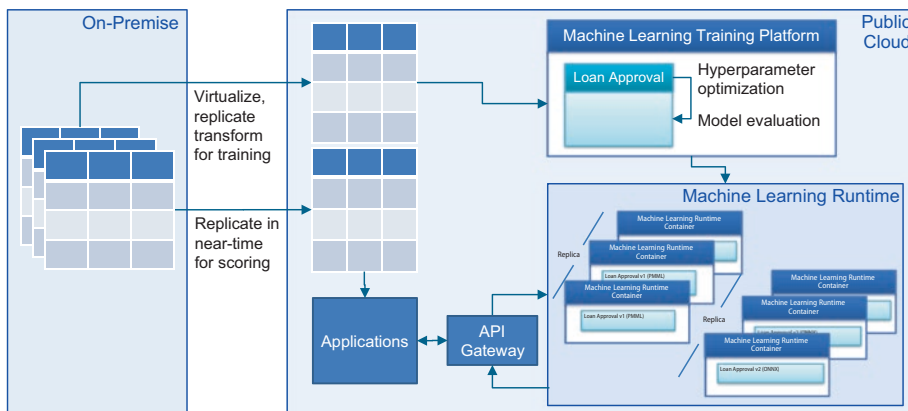


Figure 9-10. Pattern 2: Co-locate ML Runtime with the Application on Cloud

Like the previous pattern, during the training phase, data is moved off from the on-premises system to the public cloud, and after the training is complete, it is deployed directly to the public cloud. Applications running on public cloud retrieve raw data either directly from the data source on-premises or through access to a near-real-time cache of the data source

on-premises. Once applications get the data needed for model inference APIs, they send requests to the API gateway, which dispatches the requests to a specific runtime for models as described in the first pattern.

There are two important consideration factors when deploying this pattern:

1. Whether the latency of copying data or accessing data for model inference is acceptable. Depending on the data source and technology, the latency varies from hours to seconds.
2. Whether the data movement complies with security and privacy regulations. For example, data generated by data centers in the countries in the European Union is not allowed for transit use by applications running on public cloud in the United States.

If the above-mentioned factors are not hindrances, then this deployment pattern has advantages. First, the environment in which the models run can easily scale out through a public cloud infrastructure. Second, the cost of operations and maintenance is relatively low due to the unified operations toolchain. Last but not the least, it has minimal impact on on-premises systems.

The third and final common deployment pattern is an edge deployment one. AI models for image recognition and video analysis are widely used in the manufacturing industry. One typical use case for image recognition is to spot the defects in the parts in the production line to reduce the manual efforts of quality inspection. Another use case is the use of video analytics to monitor whether workers are operating regulation-safe operations.

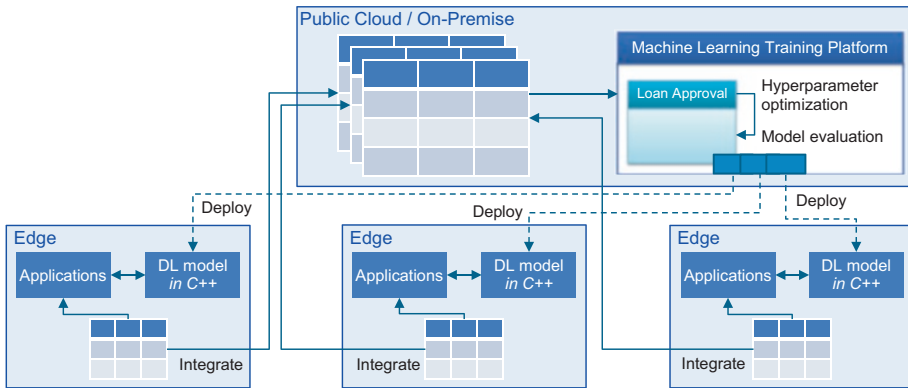


Figure 9-11. Deploy an Inference Service on an Edge Device

In these two examples, very likely the production lines in the shop floor don't have connections to the public cloud. Even if they have connections, the network overhead increases the delay in the return of model inference results, and the performance requirements of the application cannot be met. That's why edge deployment has traction.

In this deployment pattern, as depicted in Figure 9-11, all data captured at edge devices (including images and videos) is sent to public cloud or on-premises for training. Since images and videos are unstructured data, manual annotation is usually required. When the model training is completed, the model is deployed to multiple edge devices. One difficulty with this deployment pattern is that the model may be rewritten in a language like C++ due to the resource constraints of the edge-side devices and the extremely high requirements for performance, which imposes additional difficulties for model upgrades and version management. It often requires an additional component to dispatch the models to edge devices and manage the lifecycle of models at the edge.

Key Takeaways

We conclude this chapter with a few key takeaways as summarized in Table 9-1.

Table 9-1. Key Takeaways

#	Key Takeaway	High-Level Description
1	AI and software engineering have fundamental differences.	Traditional software is rule-driven, where the problem can be coded with clear logical rules and have deterministic output, while AI is more data-driven, which is based on training sets of data and algorithms for approximation of an optimal solution.
2	The AI lifecycle comprises of multiple stages.	The AI lifecycle comprises of business problem understanding, collecting data, preparing data, building the model, deploying the model, monitoring the model, and governing the model.
3	AI operationalization domains include DataOps, ModelOps, and DevOps.	AI operationalization consists of three pillars – data, model, and code. It is recommended to adopt a platform that can provide capabilities for DataOps, ModelOps, and DevOps.
4	Differences between ModelOps and MLOps.	MLOps is a subfield of ModelOps. ModelOps contains the operationalization not just for machine learning but also for knowledge graphs, rules, optimization, Natural Language Processing, agents, etc.
5	The Data Fabric architecture and Data Mesh solution implements DataOps practices.	Data Fabric and Data Mesh provide a unified enterprise data architecture and solution for consolidating dispersed data from a hybrid cloud environment through automated data discovery, smart data integration, and intelligent cataloging.

(continued)

Table 9-1. (continued)

#	Key Takeaway	High-Level Description
6	Majority of models never get into production.	Over 80% of models are never operationalized because the efforts involved in deploying them are enormous and the models are deployed and found to produce drift or fairness issues that outweigh the benefits.
6	AutoAI accelerates MLOps.	AutoAI is a no-code/low-code platform that automates several aspects of the MLOps lifecycle. It allows citizen data scientists to create models of all types and even seasoned data scientists to prototype and iterate from them more quickly.
8	There are multiple deployment architecture patterns.	The choice of deployment architecture is determined by various factors, including but not limited to data locality, performance and latency requirements, and security requirement.

References

- [1] *Gartner Top Strategic Technology Trends for 2022*, www.gartner.com/en/information-technology/insights/top-technology-trends
- [2] Mark Haakman, Luís Cruz, Hennie Huijgens, & Arie van Deursen , *AI lifecycle models need to be revised*, <https://link.springer.com/article/10.1007/s10664-021-09993-1>

- [3] Walch, K. Forbes. *Operationalizing AI*, 2020, www.forbes.com/sites/cognitiveworld/2020/01/26/operationalizing-ai/#49ef691c33df (accessed March 2, 2020).
- [4] **Aparna Dhinakaran**, *Two Essentials for ML Service-Level Performance Monitoring - A guide to optimizing ML service latency and ML inference latency*, <https://towardsdatascience.com/two-essentials-for-ml-service-level-performance-monitoring-2637bdabc0d2>
- [5] *How and Why to DataOps*, www.ibm.com/blogs/academy-of-technology/wp-content/uploads/2022/02/IBMDataOpsHowandWhy_Whitepaper.pdf
- [6] Natasha Sharma, *What Is ModelOps and How Is It Different From MLOps?* <https://neptune.ai/blog/modelops>
- [7] *2021 State of the CDO study*, www.informatica.com/about-us/news/news-releases/2021/12/20211209-informatica-unveils-2021-state-of-the-cdo-study.html
- [8] *Leadership Vision for 2022: Top 3 Strategic Priorities for Data and Analytics Leaders*, www.gartner.com/en/information-technology/insights/leadership-vision-for-data-and-analytics
- [9] *Data and Sample projects in IBM Cloud Pak for Data Gallery*, <https://dataplatfom.cloud.ibm.com/gallery?context=cpdaas&format=project-template&topic=Data-fabric>

- [10] *Why it's time for "data-centric artificial intelligence,"* <https://mitsloan.mit.edu/ideas-made-to-matter/why-its-time-data-centric-artificial-intelligence>
- [11] *Flying Blind: How Bad Data Undermines Business,* www.forbes.com/sites/forbestechcouncil/2021/10/14/flying-blind-how-bad-data-undermines-business/
- [12] *MLOps and Trustworthy AI,* www.ibm.com/products/cloud-pak-for-data/scale-trustworthy-ai
- [13] Meenu Mary John, Helena Holmström Olsson, and Jan Bosch, *Architecting AI Deployment: A Systematic Review of State-of-the-art and State-of-practice Literature,* www.researchgate.net/publication/348655621_Architecting_AI_Deployment_A_Systematic_Review_of_State-of-the-Art_and_State-of-Practice_Literature
- [14] Chaoyu Yang, *Design Considerations for Model Deployment Systems,* <https://towardsdatascience.com/design-considerations-of-model-deployment-system-c16a4472e2be>
- [15] Ryan Dawson, *Navigate ML Deployment,* <https://towardsdatascience.com/navigating-ml-deployment-34e35a18d514>

PART III

Deploying Data Fabric and Data Mesh in Context

CHAPTER 10

Data Fabric Architecture Patterns

A specific Data Fabric architecture is determined by its business and IT context and intent, meaning that not every implementation is identical. A Data Fabric could for instance serve different data consumption patterns, such as real-time transactional inference of AI-based insights, trustworthy AI scenarios, or AI governance purposes. A specific implementation of a Data Fabric also depends on concrete solution requirements, such as the ones associated with a Data Mesh solution (e.g., data-as-a-product) and whether the Data Fabric should serve certain technologies, such as IoT, edge computing, or 5G. Finally, intelligent information integration can be underpinned with different and complementary methods, such as data virtualization, replication, streaming, etc., which has an impact on the underlying Data Fabric architecture. Integration challenges within a hybrid cloud landscape¹ leveraging public cloud services may differ from integration needs within a private cloud and on-premises landscape.

Even AI itself can be characterized by a broad spectrum of application areas prevalent in different industries,² with the result that a Data Fabric architecture implementation may differ significantly case by case, depending on industry imperatives.

¹ See Reference [1] for more information on Data Fabric in a hybrid cloud environment.

² See Chapter 6 for some sample AI application areas in various industries.

Introduction

In this chapter we provide a high-level overview of the Data Fabric and Data Mesh evolution and elaborate on Data Fabric architecture patterns in terms of three related but distinct areas, which we have touched on previously.

First, we present data consumption patterns, where we examine how a Data Fabric that is usually associated with AI and analytics is entangled with transactional processing. We then introduce a Data Fabric architecture that serves as an underpinning for a Data Mesh solution, extending our treatment of this topic from Chapter 2. Finally, we examine some intelligent information integration styles, which are not necessarily mutually exclusive but could complement each other, depending on business and IT requirements and preferences.

Data Fabric and Data Mesh Evolution

As we have seen in Chapters 1 and 2, the Data Fabric and Data Mesh concepts are still relatively nascent occurrences. Nevertheless, there are already evolutionary phases that are worthwhile to give adequate attention to, which is what we aim for in this section.

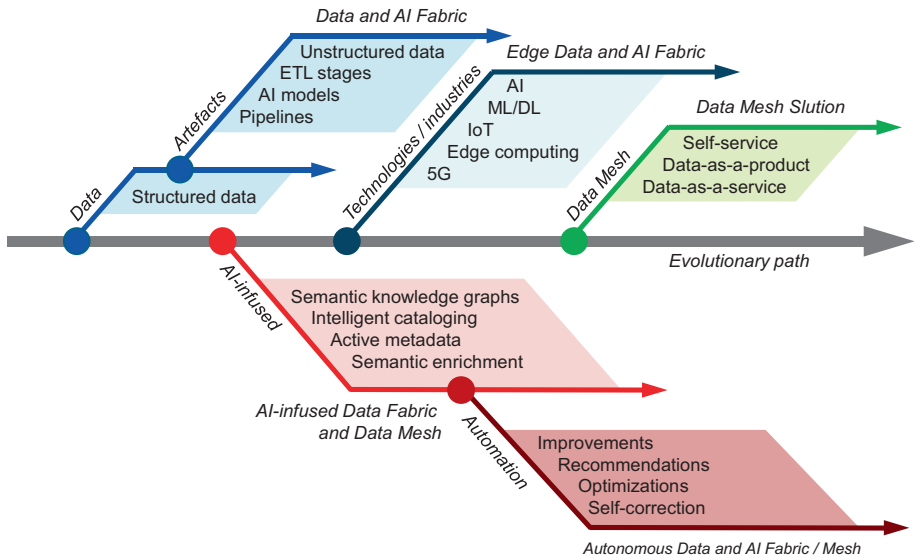


Figure 10-1. *Data Fabric and Data Mesh Evolution*

The original scope of a Data Fabric³ seems to have presumably centered around hybrid multicloud data integration challenges, including corresponding security issues, data consistency needs, data governance, etc. But beyond this, nothing else has rated very highly in importance. For instance, an AI-infused Data Fabric generating active metadata, enabling intelligent cataloging, activating the digital exhaust, semantically enriching data, performing automated Data Fabric tasks, managing artefacts other than data itself, and addressing specific Data Fabric-based solutions, such as a Data Mesh solution, has not been within the original Data Fabric sphere.

Figure 10-1 illustrates these various influencing factors concerning the Data Fabric and Data Mesh evolution, where the starting point is lacking AI, both in terms of AI artefacts to be managed by both concepts and AI capabilities augmenting the Data Fabric and Data Mesh functionality.

³ Refer to Chapter 2, where we have introduced the origin of the *Data Fabric* term.

The Data Fabric evolution is equally influenced by different technologies, such as IoT, edge computing, and 5G. Deploying AI/ML in specific industries requires unique functional capabilities and integration points.

The following list briefly examines the various aspects, which have influenced and still continue to influence the evolution of both concepts. We have consciously avoided quantifying this evolutionary path as depicted in Figure 10-1 with concrete years or timing in scale, since most of these evolutionary aspects are overlapping and occurring mainly in parallel:

1. **From data to AI artefacts:** As we have pointed out throughout this book, a Data Fabric and Data Mesh is not only about data in a traditional sense, specifically structured or relational data – it is about AI artefacts. Instead of using the term *Data Fabric* or *Data Mesh*, it would certainly be more appropriate to use the term *Data and AI Fabric* or *Data and AI Mesh*, to consider the variety of artefacts to be managed. This is illustrated by the left upper branches in Figure 10-1.
2. **AI-infused augmentation:** Infusing AI into both concepts, which we may call an *AI-infused Data Fabric* or *Data Mesh*, constitutes the most essential innovation; it is inevitably the core of what this book is all about. Augmenting capabilities for both concepts by infusing AI enables us to do intelligent cataloging, generate active metadata, build semantic knowledge graphs, etc. and gain necessary and holistic insight to improve and optimize corresponding tasks, for example, building data products.

3. **From insight to automated action:** Gaining insight via AI/ML is a vital first step to provide automation. Moving toward a modern *autonomous Data and AI Fabric or Mesh*, the challenge is to turn insight into automated actions, meaning to, for instance, implement recommendations and proposed optimizations autonomously, limiting intervention by business users or operational staff to review and approval steps. Data Fabric and Data Mesh tasks should be adjusted and corrected in an automated fashion, based on the insight derived from active metadata or by activating the digital exhaust. Furthermore, resource allocations (e.g., compute power, I/O, memory, etc.) regarding data consumption tasks need to be learned as far as possible and autonomously adjustable by the Data Fabric and Data Mesh.
4. **Technology scope and industry needs:** Hybrid multicloud adoption by the Data Fabric is only a first step. The concept of a Data Fabric becomes increasingly interrelated with other deployment options, technologies, and specific industry needs. For instance, a Data Fabric must power industrial IoT, mobile devices, edge computing, and 5G⁴ as well, which means for the Data Fabric to become more dispersed and move geographically closer not only to business users but also to end customers. In this context, data needs to be continuously streamed

⁴See References [2] and [3] for more information on the relationship between the Data Fabric, IoT, and edge computing.

between the core system or applications and their edge devices, a pattern that needs to be embraced by a Data Fabric architecture. Furthermore, AI/ML is increasingly deployed at the intersection between the analytical and transactional domains,⁵ enabling inference of AI model outcomes within transactions at mobile and edge devices. To stress these specific aspects, we may even refer to an *Edge Data and AI Fabric* architecture pattern that can be viewed as an extension to a conventional Data Fabric architecture.

5. **Data Mesh solution:** As we have seen in Chapter 2, the Data Fabric concept is closely interrelated to a Data Mesh solution with its key characteristics, such as enabling departmental ownership and building of data products in self-service fashion, which creates specific imperatives for a Data Fabric architecture.⁶

Now we move on to explore some data consumption patterns.

Data Consumption Patterns

Thinking about today's data consumption patterns leads us unavoidably to move above and beyond just considering *analytical data* as relevant for Data Fabric or Data Mesh scenarios. Instead, we need to embrace the intersection of transactional and analytical data domains as well. For instance, AI models – once they are developed and trained – must be

⁵ We explore this aspect in the following section, “Data Consumption Patterns.”

⁶ We introduce a Data Fabric architecture for a Data Mesh solution later in this chapter.

deployed and operationalized, which involves near-real-time inference and scoring of those AI models within the transactional landscape,⁷ for instance, via REST API calls. In addition, trustworthy AI requirements require us to implement MLOps, bridging the development and operationalization domains,⁸ meaning that the Data Fabric needs to orchestrate data access and data integration within the transactional landscape as well. IoT and edge computing require inferencing and analytics at edge and mobile devices, integrated into the application and transactional landscape. This imposes fundamentally different requirements for a Data Fabric architecture compared with traditional DWH or data lakehouse implementations, which are primarily geared toward analytical purposes, for example, developing AI models, dashboards, or traditional BI reports.

We therefore differentiate between the following two related but nevertheless distinct data consumption patterns:

Pattern A, analytical patterns, which are primarily concerned about analytical data and developing AI models, dashboards, BI reports, etc.

Pattern B, transaction patterns, which are primarily concerned about integration of data and AI artefacts within the transactional landscape

⁷ See Chapter 9.

⁸ See the section “Trustworthy AI” in Chapter 5.

To further explore data consumption patterns, we need to understand the user landscape of a Data Fabric and Data Mesh. Data and AI artefacts can obviously be consumed by different users, addressing different objectives. For simplicity purposes, we differentiate between the following five groups of users:

1. **Business users:** These are primarily executives, managers, and LoB users, who are either responsible for or preparing input for business decisions. These users consume data and artefacts for business purposes, including data-as-a-product as part of a Data Mesh solution.
2. **Developers:** These are primarily application developers, data scientists, and data engineers, who are developing applications, AI models, pipelines, ETL stages, etc., which includes required data exploration and preparation tasks.
3. **End customers:** These are the end customers of an enterprise, who are accessing applications and consuming, for instance, AI artefacts transparently at their edge, mobile and online devices or via other channels, for example, branch offices, ATMs, telephone, etc.
4. **Governance staff:** These are data curators, data stewards, data quality engineers, etc., who are concerned about data and AI governance aspects.
5. **IT staff:** In the context of a Data Fabric or Data Mesh, these are IT operational staff, who are, for instance, concerned about the AI deployment and operationalization aspects, enabling inferencing, scoring, etc.

Both concepts need to address the variety of data consumption needs that are imposed by these users. There exists obviously a much more fine-grained list of users and their corresponding roles and responsibilities. However, we would like to keep it relatively simple and targeted toward our Data Fabric and Data Mesh–related purposes.

Table 10-1 is a list of some key data consumption patterns, which are categorized as either transactional, analytical, or hybrid and furthermore linked to the various user groups.⁹

Table 10-1. *Data Consumption Patterns*

#	Category	Pattern	User	Description
1	Analytical	BI	Business	Traditional BI business reporting, planning scenarios, and real-time dashboarding based on DWH systems
2	Analytical	AI	BusinessDeveloper	AI model development, including training, validation, and testing
3	Analytical	Exploration	Developer	Data exploration and preparation for AI and BI use cases
4	Analytical	Transformation	Developer	Data transformation and ETL for AI and BI use cases

(continued)

⁹ See Reference [4] for more information on data consumption patterns.

Table 10-1. *(continued)*

#	Category	Pattern	User	Description
5	Transactional	Trustworthy AI	BusinessIT staffGovernance	Ensuring trustworthy AI after deployment and during operationalization of AI models into the transactional landscape
6	Transactional	MDM ¹⁰	Business	Enabling a single version of the truth for core information (master data, reference data) and 360-degree customer insight
7	Transactional	Inferencing	IT staff	Inferencing and scoring of AI models within the transactional landscape, often in real time

*(continued)*¹⁰MDM stands for Master Data Management.

Table 10-1. (continued)

#	Category	Pattern	User	Description
8	Transactional	IoT, edge, 3G	End customers	Inferencing and analytics at edge and mobile devices, integrated into the application and transactional landscape
9	Hybrid	Data Mesh	Business	Establishing a self-service data marketplace, enabling access to data-as-a-product
10	Hybrid	Governance	Governance	Access to data and AI artefacts for governance purposes, for example, data curation, data quality, etc.

Categorizing these patterns strictly as either transactional or analytical is not always possible, especially for the Data Mesh and governance patterns. For instance, the Data Mesh solution pattern (#9 in Table 10-1) should enable business users to deal with data-as-a-product, which includes search, discovery, and exploration of data. However, it may also include the usage of data and AI artefacts within an application and transactional context. You therefore see the term *Hybrid* as a category for some patterns listed in Table 10-1.

The AI pattern (#2 in Table 10-1) is primarily concerned about developers. However, business users create the business framework and communicate the business requirements. You therefore see both users listed. The trustworthy AI pattern (#5 in Table 10-1) may engage a variety of different users: Business users obviously have an interest in trustworthy AI. However, IT staff and AI governance personnel need to be engaged as well to enable trustworthy AI. The Data Fabric needs to support these various users with their corresponding roles and responsibilities.

The MDM pattern (#6 in Table 10-1) is concerned about delivering a single version of the truth for core information (master data) within a transactional and application landscape in real time, which includes providing 360-degree customer insight. In today’s enterprises, data emanates at various points of customer interaction.

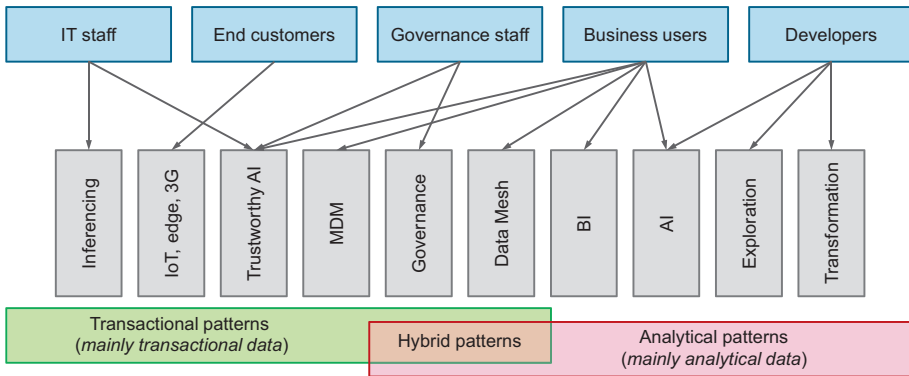


Figure 10-2. Data Consumption Patterns

The Data Fabric architecture needs to guarantee this single version of the truth within the application and transactional landscape, which – depending on the deployment option of an MDM solution – could also mean to assemble this single version of the truth based on core information that is dispersed and maintained in various data stores.¹¹

¹¹ See Reference [5] for more information on MDM solutions and deployment options.

Figure 10-2 summarizes our discussion as a conceptual illustration. The key message of this subsection is to understand the broader application scope of a Data Fabric architecture, reaching far beyond the pure analytical data and application domains.

Data Fabric for a Data Mesh Solution

The interrelationship between both concepts has already been discussed in Chapter 2. This subsection examines further Data Mesh solution functions and how they are intertwined with the Data Fabric capabilities. Since a Data Mesh enables line of business and product owners to build and deliver data-as-a-product in a self-service fashion, we intend to focus on this self-service layer as one of the key integration points between a Data Fabric architecture (or framework) and a Data Mesh solution.

In addition, we introduce a high-level architecture overview diagram, depicting a Data Mesh intertwined with the Data Fabric framework.

Data Mesh Self-Service Capabilities

Both Data Fabric and Data Mesh concepts include self-service capabilities; however, the Data Mesh solution requires additional self-service capabilities that are imposed by the Data Mesh solution imperatives.

Data Fabric self-service capabilities are technology-centric; they are enabled by semantic search and discovery and information in the knowledge catalog, such as active metadata, information regarding access methods, etc.

Data Mesh self-service capabilities are business- and domain-centric; they are primarily geared toward building, delivering, and managing data products in a concrete business, domain, or industry context, enabling the shopping-for-data experience.

Figure 10-3 illustrates this interconnectedness of Data Mesh and Data Fabric self-service capabilities. The boundaries we have drawn in Figure 10-3 may be perceived as arbitrary and, in some way, even artificial. Some readers may even ask themselves what makes the Data Mesh self-service capabilities¹² unique and different in comparison with the ones that are already present in a Data Fabric architecture.

To clarify this, let us examine the specific Data Mesh self-service capabilities in detail.

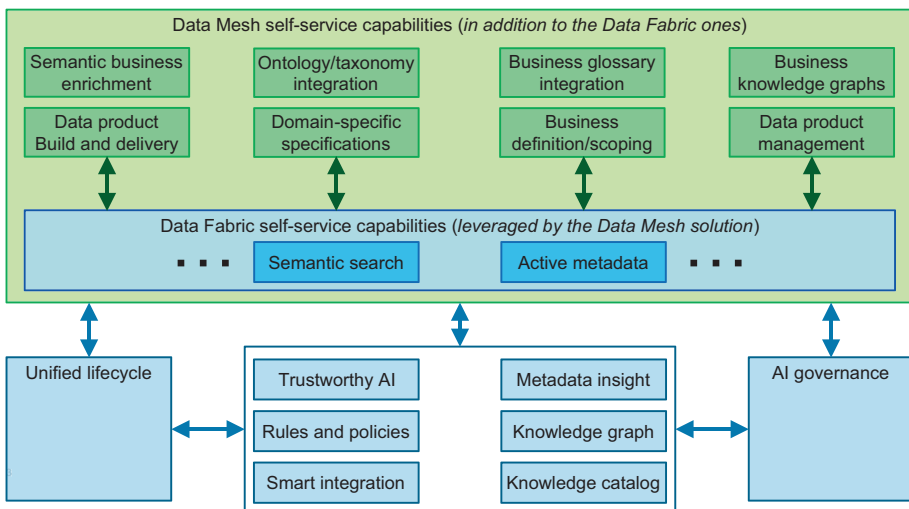


Figure 10-3. Data Mesh Self-Service Capabilities

¹²See Reference [6] for more details on self-service data platforms.

The following is a list of self-service capabilities that are specifically related to building a Data Mesh solution:

- **Semantic business enrichment:** Semantic enrichment, which is undoubtedly strongly linked to the Data Mesh characteristics, has been prominently discussed in Chapter 7. In the context of a Data Fabric, semantic enrichment simplifies data discovery, access, and consumption. In the context of a Data Mesh, however, semantic enrichment needs to embrace a specific business, domain, or industry context. Data and AI artefacts must be interpretable and enriched above and beyond the pure technical aspects; they need to be understood and consumable in a specific business context, which is what we refer to as *semantic business enrichment*.
- **Ontology/taxonomy integration:** Again, in Chapter 7 we have seen domain-specific ontologies to serve as input to the semantic enrichment engine of a Data Fabric. The ontology/taxonomy integration for a Data Mesh solution needs to serve the data product build and delivery service and needs to support the domain-specific specifications and business definition and scoping needs.
- **Business glossary integration:** The business glossary is part of the Data Fabric knowledge catalog; however, it needs to be integrated into the data product build and delivery processes. Thus, the Data Mesh solution leverages an existing business glossary, may add new terms, and will link relevant business terms to the data products or services.

- **Business knowledge graph:** In Chapter 5, we have introduced the semantic knowledge graph as an AI/ML-enriched knowledge graph. If you revisit our discussion from Chapter 5, the semantic knowledge graph within the Data Fabric framework is indeed focusing primarily on technical and organizational relationships. Further enriching this graph with business-, domain-, or industry-specific knowledge and relationships with corresponding data products is called a business knowledge graph, which represents and enriches data products for Data Mesh solutions.
- **Data product build and delivery:** One of the key objectives of a Data Mesh solution is to build and deliver data products in self-service fashion. This requires a GUI-based Data Mesh application module that supports data owners, business users, etc. to perform these tasks, where existing Data Fabric self-service and other capabilities are utilized. Data product delivery needs to include the provisioning to other data owners or business domains as well to enable interoperability across business domains.
- **Domain-specific specifications:** Data products may have to be defined and delivered within a domain-specific context or event, which requires data and AI artefacts to be understood holistically and in relationship to domain-specific events or processes. A Data Mesh solution needs to facilitate this specification of domain-specific events or processes to the underlying data and AI artefacts, which serves as input to the data product build process.

- **Business definition/scoping:** Like domain-specific specifications, a Data Mesh solution requires self-service capabilities regarding the definition of business events and scoping of the business context, which must be configurable and mappable to relevant data and AI artefacts or objects. The business glossary, the integration of the ontology and taxonomy, and the business knowledge graph may be leveraged by the business definition and scoping services.
- **Data product management:** Once data products have been built and made available to the data consumers, business users, or AI governance staff, they need to be managed (for instance, addressing updates, product versions, ownership, access rights, etc.) as additional artefacts in the Data Fabric knowledge catalog.

Like any data and AI artefact, data products are likewise stored in the Data Fabric knowledge catalog, which we may then consider as a *Data Mesh knowledge catalog*. The preceding list of Data Mesh self-service capabilities makes it inevitably clear that specific Data Mesh tools and services are required that leverage the Data Fabric capabilities to build a Data Mesh solution. These tools and services need to include REST APIs and data exchange protocols or standards (e.g., JSON,¹³ Apache Avro¹⁴) and must be easy to use by data owners, data consumers, or business users.

¹³JSON stands for JavaScript Object Notation and is a lightweight data interchange format. See <https://www.json.org> for more details.

¹⁴Apache Avro is a data serialization framework. See <https://avro.apache.org> for more details.

Data Mesh Architecture Overview Diagram

In addition to the self-service interconnectedness, we introduce a high-level architecture overview diagram of a Data Mesh solution, implemented via key Data Fabric capabilities – as depicted in Figure 10-4.

As you can see in the lower part of Figure 10-4, there are several data domain owners, who are responsible for their corresponding data products. As an example, *data domain owner A* manages the corresponding data in a Google cloud, whereas *data domain owner B* has the corresponding data stored on-premises, which could be an x86 Intel cluster or an IBM mainframe system. Building and delivering these data products is based on Data Fabric capabilities, such as data preparation and data integration tasks, exploiting active metadata stored in the knowledge catalog, and defining new or using existing data policies and rules – to just mention a few. Data Fabric and Data Mesh self-service capabilities – as discussed in the previous subsection – are used to build, manage, and deliver data products.

It is essential to realize that the Data Fabric architecture enables the Data Mesh solution via its rich knowledge catalog, semantic search and discovery, smart integration capabilities, and semantic knowledge graphs. Trustworthy AI, for instance, is enabled via the Data Fabric as well.

Thus, the Data Mesh solution establishes a data marketplace with shopping-for-data characteristics (data-as-a-product). This is depicted in the upper half of Figure 10-4, where you see several data consumers, who could be business users as well as end customers, depending on the data consumption pattern. As an example, *data_consumer_1* is using BI services of MS Azure, *data_consumer_2* is using an on-prem data science platform, *data_consumer_3* is using BI services of a private cloud, and *data_consumer_4* is using data science services of IBM Cloud.

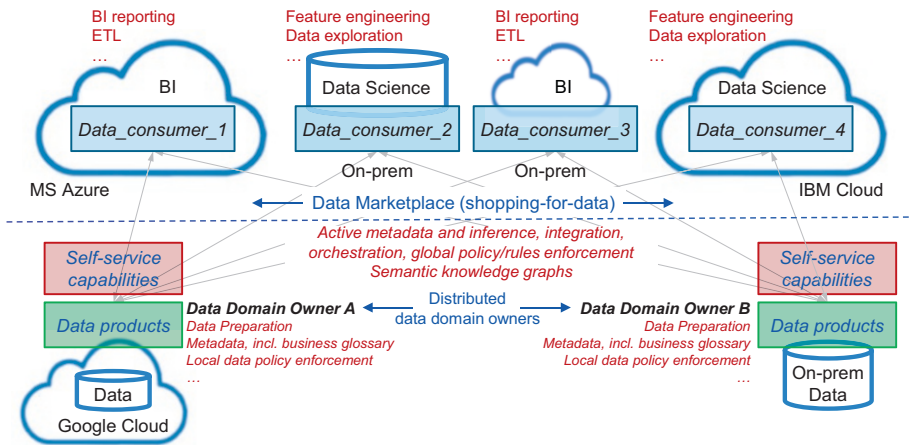


Figure 10-4. *Data Mesh Architecture Overview Diagram*

All data consumers may access and use data products from the various data domain owners and consume those data products for their particular purpose, for example, traditional BI reporting, data exploration and feature engineering, AI model development, etc. This is depicted via the arrows between the various data consumers and the data products. Thus, data products that are owned by one data domain owner may be accessible and consumable by several data consumers. Access rights, rules, and policies regarding the consumption of these data products are captured and managed via the Data Mesh knowledge catalog.

Intelligent Information Integration Styles

A discussion about Data Fabric architecture patterns remains incomplete without touching on intelligent information integration styles. Since much has been published about information integration,¹⁵ we limit

¹⁵ See Reference [7] for more on information integration and [8] for a recent magic quadrant from Gartner on data integration tools.

our examination to a well-known list of information integration styles, providing recommendations on when to use a certain pattern or style and when not.

In Chapter 5, we have introduced the term *intelligent information integration* as an AI-infused information integration layer, which constitutes a vital capability of a modern Data Fabric architecture and Data Mesh solution.

Table 10-2 inherently prerequisites this AI-infused Data Fabric view of intelligent information integration styles.

Table 10-2. *Intelligent Information Integration Styles*

Style	Description	When to Use and When Not
ETL, ELT, etc.	Traditional DWH-style ETL, ELT, etc.	Required for complex data aggregation and transformation jobs to prepare data for specific consumption purposes, like traditional BI/DWH scenarios. ETL vs. ELT depends primarily on data size and complexity of transformation requirements.
CDC ¹⁶	CDC-based enterprise replication	Allows near-real-time movement of mainly relational data via log-based CDC, has limited impact on source systems, enables additional SQL statements to be executed on an identical DB schema without impact on the source systems.

(continued)

¹⁶ CDC stands for change data capture.

Table 10-2. *(continued)*

Style	Description	When to Use and When Not
Federation/Virtualization	Data federation and virtualization with pushdown function	Simplifies access to heterogeneous and distributed data sources by generating one virtual data view, leverages pushdown functions, leaves data in place, only access to data that is needed.
REST API	API conforming to the REST architecture style	Compared with SQL, REST APIs provide more flexibility in accessing data, which can be stored in DBMS or in resources addressable by URLs, use JSON, and can be used to easily connect applications or microservices.
Microservices	Distributed, loosely coupled architecture framework for building apps	Microservices are smaller, usually container-based services that can be “stitched” together via REST APIs to function as an integrated application.
Streaming	Transmitting a continuous flow of data	Mainly for real-time data consumption and real-time analytics on continuously streamed data, for example, video, social media feeds, traffic monitoring, real-time stock trades, real-time cybersecurity, etc.

(continued)

Table 10-2. (continued)

Style	Description	When to Use and When Not
Messaging	Messaging describing events, requests, replies, etc.	Especially useful for feeds of business-related events, requests, replies, etc., advantage above raw data streaming due to business-relevant messaging.
JDBC/ODBC	SQL-based API to access DBMS	Used for native SQL-based access to a wide variety of RDBMS, consumes resources of the source systems. BI/analytics workload may interfere with transactional workload.

As we have outlined in Chapter 5, intelligent information integration is an AI-infused information integration layer that comprises AI-based capabilities, which should shield the various users from the complexity of these information integration styles; AI capabilities should automate and *learn* information integration – meaning to automatically choose, adjust, optimize, and improve the information integration pattern over time, further reducing human intervention.

Providing AI-infused intelligence and automation to the information integration layer is one of the most demanding areas to implement a Data Fabric architecture and Data Mesh solution.

Key Takeaways

We conclude this chapter with a few key takeaways as summarized in Table 10-3.

Table 10-3. *Key Takeaways*

#	Key Takeaway	High-Level Description
1	AI should be infused into the Data Fabric.	An AI-infused Data Fabric constitutes the most essential recent innovation.
2	A Data Fabric needs to turn insight into automated action.	Moving toward a modern autonomous Data and AI Fabric requires turning insight into automated actions, for example, to implement recommendations and proposed optimizations autonomously.
3	There are analytical and transactional data consumption patterns.	A Data Fabric needs to serve analytical and transactional data consumption patterns to, for instance, address MLOps, trustworthy AI, MDM, inferencing, IoT, edge, and 5G.
4	New technologies, for example, IoT, edge, etc., have a profound impact on a Data Fabric.	Data Fabric must power industrial IoT, mobile devices, edge computing, and 5G, which means for the Data Fabric to become more dispersed and move geographically closer to end customers.
5	Data Mesh–specific self-service capabilities are needed.	Data Mesh self-service capabilities are business- and domain-centric; they are geared toward building, delivering, and managing data products in a concrete business, domain, or industry context.
6	Data product build, delivery, and management are key.	To build, deliver, and manage data products is one of the key capabilities that the Data Mesh has to implement in self-service fashion.

(continued)

Table 10-3. (continued)

#	Key Takeaway	High-Level Description
7	Data product specifications are also stored in the knowledge catalog.	We call a Data Fabric knowledge catalog that also captures information about data products a <i>Data Mesh knowledge catalog</i> .
8	A Data Fabric architecture enables a Data Mesh solution.	A Data Fabric architecture enables the Data Mesh solution via its rich knowledge catalog, semantic search and discovery, smart integration capabilities, semantic knowledge graphs, etc.
9	Several intelligent information integration styles are needed.	A Data Fabric needs to support multiple intelligent information integration styles, such as federation and virtualization, microservices with REST APIs, streaming, messaging, etc.

References

- [1] IBM Institute for Business Value, *Weaving data fabric into hybrid multicloud*, <https://www.ibm.com/downloads/cas/Y5A968XV> (accessed August 6, 2022).
- [2] IoT Business News, *The Interoperable Growth of Data Fabric and IoT*, <https://iotbusinessnews.com/2021/08/18/33020-the-interoperable-growth-of-data-fabric-and-iot/> (accessed August 7, 2022).

- [3] Carr, L., Actian, *What's an Edge Data Fabric?*, <https://www.actian.com/blog/data-management/whats-an-edge-data-fabric/> (accessed August 7, 2022).
- [4] McKendrick, J., *The Move to Modern Data Architecture: 2022 Data Delivery and Consumption Patterns Survey*, 2022, file:///Users/eberhardhechler/Downloads/2022%20Data%20Delivery%20and%20Consumption%20Patterns%20Survey%20-%20ChaosSearch.pdf (accessed August 11, 2022).
- [5] Dreibelbis, A., Hechler, E., Milman, I., Oberhofer, M., Van Run, P., Wolfson, D., *Enterprise Master Data Management, An SOA Approach to Managing Core Information*, IBM Press, 2008, ISBN-13: 978-0132366250.
- [6] Dehghani, Z., *Chapter 4. Principle of the Self-Serve Data Platform*, <https://www.oreilly.com/library/view/data-mesh/9781492092384/ch04.html> (accessed August 13, 2022).
- [7] Giordano, A., *Data Integration Blueprint and Modeling, Techniques for a Scalable and Sustainable Architecture*, IBM Press, 2013, ISBN-13: 978-0137084937.
- [8] Gartner, *Magic Quadrant for Data Integration Tools*, 2021, <https://www.gartner.com/doc/reprints?id=1-27E7HBB5&ct=210909&st=sb> (accessed August 14, 2022).

CHAPTER 11

Data Fabric Within an Enterprise Architecture

Any data architecture, and therefore also our Data Fabric architecture, needs to be looked at in conjunction with the implemented application architecture in an existing enterprise landscape. Many organizations are in the process to modernize and digitalize their application and data landscape. Applications have different requirements with respect to data characteristics, which may recommend a particular data architecture implementation over another, for example, characterized by data access based on data virtualization or data replication and transformation.

In this chapter, we briefly revisit the key imperatives of an enterprise and application architecture before we elaborate on key aspects of a Data Fabric within an enterprise architecture, which includes challenges and benefits. Using a conceptual architecture model as a representation of a Data Fabric, we illustrate how the most essential components of a Data Fabric play in concert with imperatives that are derived from the application and business domains.

The integration of a Data Fabric architecture within a given enterprise and application architecture provides a basis to build a Data Mesh solution.

Introduction

In Chapter 1 we saw that a Data Fabric architecture is an evolution from previous data architectures influenced by new business-driven data requirements exploiting new technologies. From a pure IT perspective, almost everything seems to be possible today. Especially in larger organizations, well-defined architectures with IT implementation recommendations based on functional and non-functional requirements are necessary to avoid an unreliable, unmanageable, and overly expensive IT landscape. An architecture is a clear representation of a conceptual framework of components and their relationship at a specific point in time.

It is an acknowledgment that organizations make better IT implementation decisions when they take a broader view and describe the need for data architecture capabilities as part of an enterprise architecture with emphasis on business needs. Looking at data architectures in organizations over time, you can observe a shift from data architecture determining application architecture to the other way around, application architecture determining or at least strongly influencing the underlying data architecture, driving significantly different software and hardware delivery systems.

In this chapter we briefly review the levels of an enterprise architecture, followed by a discussion of the current three major approaches for an application architecture and its implications on the underlying data architecture. We revisit the main data architecture decision criteria and share a methodology on how to evaluate them based on specific business requirements. This methodology helps choose the most applicable implementation based on technical characteristics and not individual opinions and preferences.

Furthermore, we introduce the conceptual model and resulting components of a Data Fabric architecture, including its relevance for implementing a Data Mesh solution.¹

What Is Enterprise Architecture?

The enterprise architecture framework was first developed as a reference model by the National Institute of Standards and Technology (NIST) in the late 1980s.²

It took until the late 1990s before its relevance was widely recognized and enterprises used it as directions for their operations. Today, no organization can implement an efficient IT ecosystem without a well-defined enterprise architecture that is specific to its needs. The enterprise architecture consists of five layers as shown in Figure 11-1.

¹ Please, recall Chapter 3, where we have listed the relevance of those components for specific use cases.

² See Reference [1] for more details on the enterprise architecture framework from NIST.

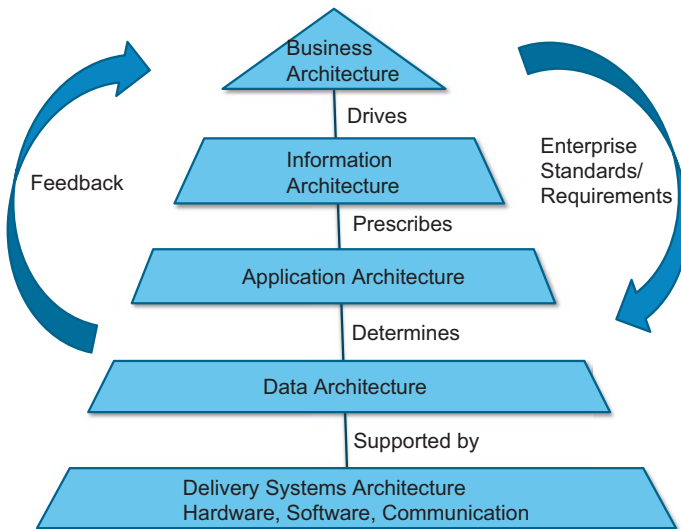


Figure 11-1. Enterprise Architecture Framework

Each level includes components that define enterprise standards and requirements. The feedback loop back to the business architecture is critical for the adoption of new industry trends and technologies, making the enterprise architecture³ a living document:

1. **Business architecture level:** Describes the entire business landscape of the enterprise or a specific line of business, which is in contact with external organizations, including business partners, and end customers. It includes the business practices of functional areas and resulting external and internal corporate reporting requirements. It unambiguously drives the information architecture level.

³See Reference [2] for more information on enterprise architecture management.

2. **Information architecture level:** Specifies the contents, presentation forms, and formats of information required for the entire enterprise or a line of business, for example, industry standards for business-to-business (B2B) data exchange such as SEPA⁴ or HIPAA.⁵ Compliance and security regulations as well as the business glossary are also defined on this level. It prescribes the information systems and application architecture level.
3. **Application architecture level:** Specifies the applications with their main components and functions, application integration, and software development standards, including application interfaces and programming languages. Later in this chapter, we will discuss three major approaches that are currently popular for the core system transformation. The application architecture strongly determines the main characteristics of the data architecture.
4. **Data architecture level:** Specifies the framework for access and use of data, its maintenance, as well as the communication between data. It includes data models and their elements and structures and defines standards for naming conventions and representations, typically including the data dictionary. Both application and data architectures are implemented and supported by the delivery system architecture.

⁴SEPA stands for Single Euro Payments Area.

⁵HIPAA stands for Health Insurance Portability and Accountability Act.

5. **Delivery system architecture level:** Represents the technical implementation details regarding software, hardware, and network and communication in support of the application and data architectures. It includes hardware platforms, operating systems, utilities and tools, and storage media. Cloud deployment standards (public or private cloud) belong in this level.⁶

An absolute most common mistake that architects make is to neglect the separation of application and data architectures from the delivery system architecture. Guided by the architecture decision criteria, multiple options for the technical implementation of a data or application architecture should be defined in the delivery system architecture and not in the preceding architectural layers. In addition, implementation options should be complemented with pros and cons to derive well-balanced decisions.

These guidelines are most relevant for implementing a Data Fabric architecture, especially since there is no single architectural approach that is applicable and advantageous for all use cases.

What Is Application Architecture?

An application architecture describes the behavior of applications, how they interact with each other and with users, and how data is produced and consumed. For a long time, the application development focus was on providing new functionality for a specific line of business. The internal application structure was monolithic, very interconnected, making any change to the user interface complex and expensive, as well as causing the level of code reusability to significantly decrease.

⁶See Reference [3] for more information on weaving in a Data Fabric into a multicloud environment.

To support many more user communication channels such as B2B, online, and mobile and to react to changed industry trends quickly, current application architectures include three major approaches. Each organization's environment, end state requirements, risk tolerance, financial justification, etc. define the business impact of each of those approaches – clearly suggesting hybrid approaches and variants as desirable approaches.

The following are the three major application architecture approaches:

- **API enablement:** Seems to be the simplest approach to transform existing applications, where application functionalities are wrapped in REST-based application programming interfaces (APIs) to make them easily consumable for other applications and user channels, hiding all platform and implementation details. A single API can have multiple implementations for different sets of users, for example, low-value consumer implementations may prefer commodity platforms vs. high-value consumer implementations may choose highly reliable and scalable configurations.
- **Containerization:** Is the systematic movement of application logic into a container, which is a pre-built stack of operating system, middleware, and application code. Containers have several advantages, that is, through open source and standardization efforts, containers are portable across platforms and can be deployed anywhere, including on-premises and in a cloud environment. Containers, once built, are unchangeable and signed, making them more secure from unintended or malicious changes. It typically preserves data and operational processes while quickly scaling its consumption.

- **Microservices segmentation:** Is the total restructuring of existing applications into cloud-native applications. While this approach is attractive for new applications, it can be extremely risky for transforming existing applications. Microservices⁷ segmentation has profound consequences on the underlying data architecture, which typically results in a highly partitioned data model.

Figure 11-2 illustrates the business impact vs. feasibility of the three application architecture approaches when transforming existing applications. API enablement of existing functionality delivers good business value for relatively low investment and is a widely adopted approach. It is followed by application containerization and incremental rearchitecture of applications adopting microservices.

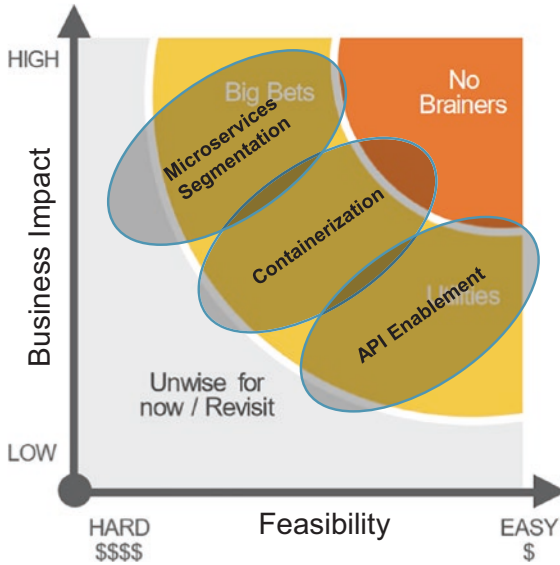


Figure 11-2. Application Transformation Approaches

⁷ See Reference [4] for more information on microservices architectural patterns.

Microservices have a potential of a big business impact but also come with new data architecture considerations.

The application modularity and data partitioning will drive data architecture decisions and has key implications on application logic:

- **Data access vs. data replication:** The first fundamental decision concerns data access of the original data vs. replicating the data for consumption, creating one or multiple copies. There are several advantages that favor data access, especially with availability of advanced data virtualization technologies. While it is straightforward to replicate data once, maintaining data replication pipelines over an extended period is expensive and time-consuming. It also typically creates data quality and data latency challenges for consuming applications. Accessing data in place accelerates the application transformation while preserving existing data management and recovery processes, therefore delivering value to the business faster.
- **Data consistency vs. eventual consistency:** Data consistency considerations are introduced with data segmentation. Microservices by architecture are stateless. There is no transactional boundary across multiple microservices implementing a consistent change across multiple data sources. Subsequently, microservices are eventually consistent,⁸ meaning that data consistency is reached when all microservices relevant for a particular use case execution succeed. If a microservice fails, previously executed logic may need to be undone, which may increase the amount of

⁸See Reference [5] for more information on the CAP (consistency, availability, partition tolerance) and eventual consistency.

compensation logic and/or business processes needed for systems that are out of sync. The main benefit is that eventual consistency enables applications to be developed independently of each other. However, it can result in data quality issues. Therefore, the majority of microservices implementations so far focus on use cases that consume data but do not update data.

In a Data Fabric, based on intelligent metadata, data consumers should get recommendations on intelligent information integration technology that is most appropriate for a specific use case.

Data Fabric as a Data Architecture

A data architecture defines data standards in an organization, including how data is accessed and consumed. It furthermore describes the data structures used by the business units. Data integration also depends on the defined data architecture standards since data integration requires interaction between data.

Besides new technologies, there are various other constraints and influences that will impact the data architecture design, also called data architecture decision criteria. Those can be grouped in three categories:

- **Capabilities or functional requirements:** This includes a description and prioritization of functional characteristics such as data consistency, data latency, quality and granularity of data, and data access and consumption requirements.
- **Resilience or non-functional requirements:** It includes a description and prioritization of characteristics such as availability, maintenance, performance, simplicity, scalability, and security and data governance.

- **Financial or business requirements:** It includes description and prioritization of characteristics such as cost of infrastructure (cost of acquisition), service-level agreements, operational cost, time to value, and risk to the business.

The whole organization or a specific business unit defines requirements for particular use cases that lead to the relative prioritization of the data architecture decision criteria. Different implementations of the delivery system architecture, of course, satisfy those criteria to a varying degree.

For example, accessing the original data via data virtualization provides more current and accurate data for targeted marketing campaigns in comparison with accessing ETL-transformed data in an EDW that may be a couple of days old. In addition, data virtualization guarantees data currency to any consumer. In turn, ETL-transformed data may better deliver on the data granularity needs for specific reporting needs, such as sales trending or reporting applications. However, ETL-transformed data may lag behind in data currency.

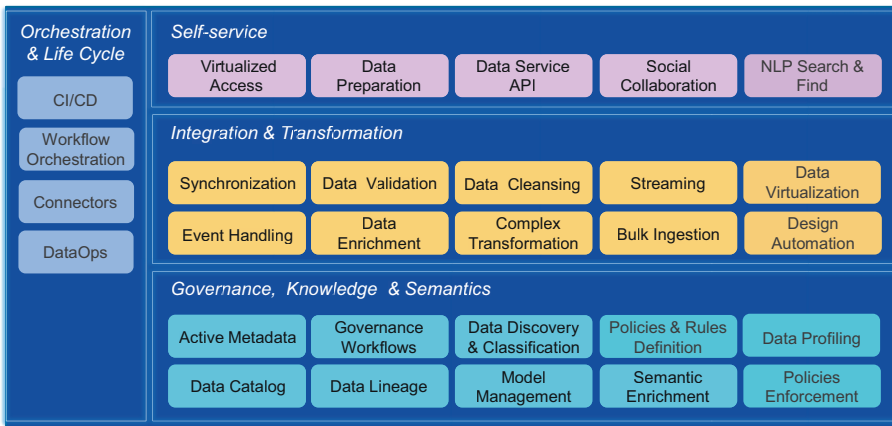


Figure 11-3. *Data Fabric Architecture Layers*

The strength of the Data Fabric architecture lies in utilizing AI for reasoning and optimization,⁹ active metadata, knowledge graphs, and semantic enrichment, combining intelligent information integration and transformation technologies to support data consumers.

Figure 11-3 shows key capabilities grouped into distinct Data Fabric architectural layers:

1. **Governance, knowledge, and semantics layer:**

The foundational layer includes the knowledge catalog as an inventory of data structures. Data discovery and classification provides functionality to discover, classify, and label data in the data sources. Additionally, active metadata and data profiling increase the value of metadata by analyzing and reviewing the data, augmented with human knowledge and processed with ML. For example, it can assess accuracy, completeness, consistency,

⁹ See Reference [6] for more information on AI for integrated learning, reasoning, and optimization.

timeliness, and accessibility of metadata also in relation to other data sources. Semantic enrichment further extends metadata by tagging, indexing, and cataloging the data. Besides rich active metadata, data lineage functionality and impact analysis is critical; it visualizes data pipelines from the original data source to the data consumer over time. Model management applies similar intelligent metadata management to ML models as they go through their lifecycle of data preparation, model development, model training, and model deployment. Governance policies and rules definitions, enforcement, and workflows ensure that available data is only consumed by authorized consumers in an authorized way. The overall goal of the layer is to get a unified view of data and metadata with actionable insights and unified enforcement of data governance policies.

2. **Integration and transformation layer:**

Provides all capabilities needed for data pipeline implementations. Data replication, streaming, and synchronization as well as event handling move changed data from the original data source through defined data pipelines to keep consuming applications aware of those changes for further processing and consumption. Alternatively, bulk ingestion creates copies of entire data sources. Data validation, data enrichment, data cleansing, and complex transformation implement complex ETL pipelines. Besides copying data from the original data sources with the preceding functionalities,

data virtualization as technology to access original data becomes more important as part of a data pipeline implementation, especially if data virtualization includes computational mesh and intelligent pushdown capabilities.¹⁰ For example, data virtualization techniques can be used to read data from multiple data sources as input to data preparation, where complex transformation can be applied through SQL logic and executed while accessing the data sources, eliminating data movement. This integration and transformation layer provides a range of policy-driven integration styles with intelligent automation.

3. **Self-service layer:** Provides functionality to the data consumer for virtualized access to data sources or for data preparation as well as social collaboration and search and find functionality besides custom data service APIs. The self-service layer makes the trusted and governed marketplace available for the data consumers to easily search, find, understand, collaborate, and gain access to data assets and data products. For example, a global search for customer data could simply type the word *customer* and find related data assets that they have access to. Self-service shopping-for-data represents huge time savings for business users and data consumers in general. This is the overarching goal of a Data Fabric architecture and Data Mesh solution.

¹⁰ See Reference [7] for more on data virtualization and computational mesh in IBM Cloud Pak for Data.

4. **Orchestration and lifecycle layer:** Provides change management capabilities such as APIs for CI/CD integration and workflow orchestration as well as connectors to the different supported data sources and DataOps and MLOps capabilities. The orchestration and lifecycle layer provides the end-to-end lifecycle to build, test, deploy, and manage all data and AI assets in a Data Fabric, equally for data structures and data pipelines as well as AI model assets.

Sample of a Data Fabric Within an Enterprise Architecture

Many vendors develop and offer products in support of a Data Fabric architecture in an organization. As an example, Figure 11-4 shows IBM Cloud Pak for Data¹¹ as an architectural foundation for a Data Fabric in an enterprise architecture.¹²

¹¹ See Reference [8] for more information on IBM Cloud Pak for Data.

¹² See Reference [9] for more information on using cloud deployments for analytical insight.

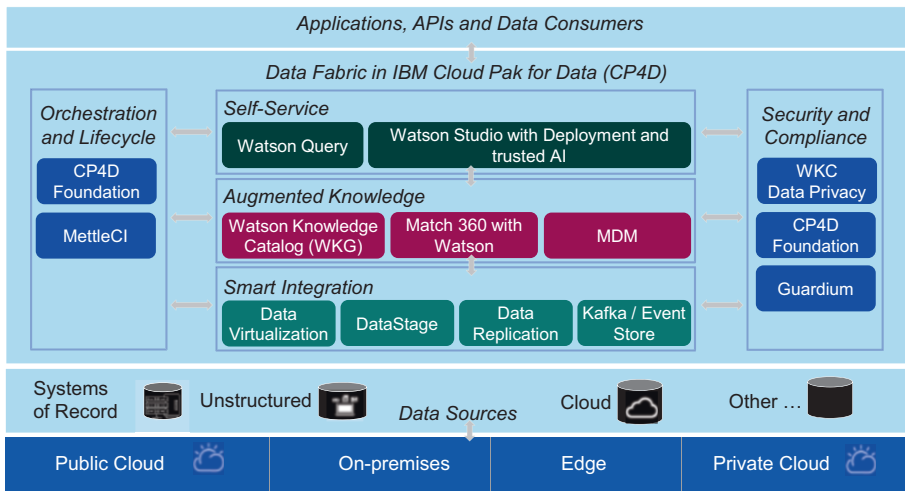


Figure 11-4. Sample of a Data Fabric with IBM Cloud Pak for Data (CP4D) Within an Enterprise Architecture

At the bottom all variations of the delivery system architecture, public cloud, private cloud, edge computing, and on-premises computing environments are supported. Having data stored in multiple computing environments is called hybrid cloud, which we will discuss in more detail in Chapter 12. As part of the data architecture, supported data structure types need to be listed such as structured data (e.g., systems of records), semi- and unstructured data, and cloud data stores. IBM Cloud Pak for Data Foundation and Watson Studio pipelines provide the orchestration and lifecycle functions as well as integrated security and compliance capabilities. At the core is the augmented knowledge with IBM Watson Knowledge Catalog (WKC), MDM, and IBM Match 360 with Watson services for comprehensive view of customers and other parties.

The smart integration is provided through Kafka/Event Store, data replication, IBM DataStage for ETL pipelines, and data virtualization. IBM Watson Query is the virtualization service, which makes queries across data sources faster and easier without moving the data. IBM Watson Studio

is the collaboration tool for users to build and train AI and ML models and prepare and analyze data. Applications, APIs, and data consumer interfaces become part of the application architecture of the organization.

Figure 11-4 is an illustration of the Data Fabric architecture within a larger enterprise architecture.

Key Takeaways

We conclude this chapter with a few key takeaways as summarized in Table 11-1.

Table 11-1. *Key Takeaways*

#	Key Takeaway	High-Level Description
1	Data Fabric is part of an enterprise architecture.	Data Fabric architecture needs to be looked at in conjunction with the implemented application architecture in an enterprise.
2	Application architecture determines the data architecture.	The current application architecture pattern such as microservices, containerization, and API enablement determines underlying data architecture characteristics.
3	Data Fabric as a data architecture needs to be defined in the enterprise architecture.	A Data Fabric architecture as a data architecture needs to be defined in the enterprise architecture for an organization.
4	Data Fabric combines data integration technologies.	Data Fabric architecture utilizes active metadata, knowledge graphs, and semantic enrichment, combining intelligent information integration and transformation technologies to intelligently support data consumers, for example, business users.

(continued)

Table 11-1. (continued)

# Key Takeaway	High-Level Description
5 Microservices are eventually consistent.	Text microservices are eventually consistent, meaning that data consistency is reached when all microservices relevant for a particular use case execution succeed.
6 Data product build, delivery, and management are key.	To build, deliver, and manage data products is one of the key capabilities that the Data Fabric architecture and Data Mesh solution has to implement in self-service fashion.

References

- [1] National Institute of Standards and Technology (NIST), Rigdon, W.B., *Architectures and Standards*, p. 135–150, in Fong, E.N., Goldfine, A.H. (eds.), *Information Management Directions: The Integration Challenge*, 1989, www.nist.gov/publications/information-management-directions-integration-challenge (accessed September 13, 2022).
- [2] Jung, J., Fraunholz, B., *Masterclass Enterprise Architecture Management*, Springer, 2021, ISBN-13: 978-3030784942.
- [3] IBM Institute for Business Value, Sarkar, S., Bijlani, V., Warrick, R., *Weaving data fabric into hybrid multicloud*, www.ibm.com/downloads/cas/Y5A968XV (accessed August 6, 2022).

- [4] Christudas, B., *Practical Microservices Architectural Patterns*, Apress, 2019, ISBN-13: 978-1484245002.
- [5] IBM, *What is the CAP theorem?*, www.ibm.com/cloud/learn/cap-theorem (accessed September 14, 2022).
- [6] Heintz, F., Milano, M., O'Sullivan, B. (eds.), *Lecture Notes in Artificial Intelligence, Trustworthy AI – Integrating Learning, Optimization and Reasoning*, Springer, 2021, ISBN-13: 978-3-030-73958-4.
- [7] IBM, *IBM Cloud Pak for Data, Eliminate data silos: Query many systems as one, Data virtualization in IBM Cloud Pak for Data*, 2019, www.ibm.com/downloads/cas/97AJPYNN (accessed September 14, 2022).
- [8] Manda, H., Srinivasan, S., Rangarao, D., *IBM Cloud Pak for Data: An enterprise platform to operationalize data, analytics, and AI*, Packt Publishing, 2021, ISBN-13: 978-1800562128.
- [9] Forbes, CIO Network, Davenport, T., *Is The Cloud Slower For Analytical Insights?*, www.forbes.com/sites/tomdavenport/2022/01/21/is-the-cloud-slower-for-analytical-insights/?sh=41194bc66dca (accessed September 13, 2022).

CHAPTER 12

Data Fabric and Data Mesh in a Hybrid Cloud Landscape

In this chapter, we investigate additional facets of both concepts that arise specifically from hybrid cloud deployments. We briefly review the term *hybrid cloud* and how it relates to on-premises. Although cloud services are rated very highly in importance, they nevertheless create new challenges regarding access, integration, and consumption of data and AI assets across the organization. What does *Data Fabric architecture and Data Mesh solution* mean in the context of a hybrid cloud landscape, and what are the key challenges, for instance, related to addressing data and AI governance and trustworthy AI?

Even more important though, what are the benefits of having a Data Fabric and Data Mesh for a hybrid cloud environment, and how does this differ from an approach that is targeted for a traditional on-premises IT and application landscape?

This chapter gives adequate attention to the specifics of building a Data Mesh solution in a hybrid cloud, highlighting the building and consumption of data-as-a-product in a data marketplace that is delivered via a hybrid cloud.

Introduction

In this chapter, we look at the intersection of two initiatives of the digital business transformation, (a) implementing a Data Fabric architecture and Data Mesh solution to address today's inability to extract business insight from the huge amount of data that is available in each organization and (b) adopting hybrid cloud computing. Chapter 3 introduced the unified view of data across a hybrid cloud as one of four entry use cases.

The requirement to view, analyze, and process data across a hybrid cloud architecture is becoming a high priority as organizations have projects ongoing to adopt and use cloud-native applications and to modernize existing applications exploiting public or private cloud environments next to running applications in their traditional processing environments.

In this chapter we revisit what is a hybrid cloud and what are the key challenges for data and AI capabilities in an organization. We intend to demonstrate how both concepts can address data and AI needs in a distributed, hybrid cloud environment across transactional and analytical data operations.

What Is Hybrid Cloud?

Hybrid cloud integrates public cloud services, private cloud services, and on-premises infrastructure and provides orchestration, management, and application portability across all three.¹ The vision is an agile, distributed compute environment where an organization can optimize its existing workload and adopt new workloads quickly:

- **Public cloud services:** Provide on-demand IT resources over the Internet with a consumption-based pricing model. Instead of buying and installing hardware

¹ See Reference [1] for more information on hybrid cloud.

and software for a specific solution, which may take many months in bigger organizations, services such as compute power, storage, databases, analytics tools, etc. can be used on an as-needed basis without the up-front cost and installation time. A major benefit of a cloud deployment option is agility in adopting new technologies. Public cloud providers offer a wide range of services such as AI and ML, data lakes, and analytics. It promises the freedom to test new ideas and verify business value quickly. Another benefit is elasticity. Especially in x86 processing environments, overprovisioning of compute power is very common, which leads to increased cost of hardware and software licenses as well as sustainability concerns in respect to floor space and power consumption. Public cloud service providers are responsible for owning and administering the data centers where user workloads run and implementing orchestration and virtualization software to scale service resources up and down based on demand. The IT consumption model can be compared to a *utility model* for consuming electricity.

- **Private cloud services:** Use similar cloud infrastructure exclusively for one company. Typically, it is operated behind an organization's firewall but can also be hosted on dedicated third-party infrastructure. In a private cloud, the organization's IT team is responsible for installation and maintenance of the IT infrastructure and can control resources, data security, and regulatory compliance as well as look into potential performance optimization. The application development teams and business units are the consumers of the private

cloud services and expect similar agility and elasticity as advertised by public cloud service providers. A big advantage of a private cloud compared with the public cloud deployment model is the ability to customize applications and infrastructure leading to substantial cost savings beyond initial test and adoption phases.²

- **Traditional on-premises IT infrastructure:** Consists of hardware and software products and is developed over a longer period of time to fit the organization's needs and goals. The organization's IT team operates the data centers, networking, and applications and develops custom tools and procedures for system and application deployment and operation as well as charge-back models. A well-managed IT infrastructure and application portfolio implements the enterprise architecture; is highly optimized, reliable, and secure; and therefore can be used as a competitive advantage. In some organizations, the rigorous processes created to ensure the scalable, efficient, reliable, and secure IT infrastructure slowed down the adoption of new IT and led to the perception that on-premises IT infrastructure is less agile and scalable compared with the cloud deployment model.

Today's hybrid clouds are architected by focusing on building applications through loosely coupled services across one or many of public cloud, private cloud, and on-premises IT infrastructure.

²See Reference [2] for more information on the cloud lifecycle cost.

Key Challenges for Data Architecture

Review data challenges as discussed in Chapter 1, now with added complexity of data stored and accessed using a different format and flexibility for data sources to be portable across IT environments. The agility of a hybrid cloud architecture allows for new locations where data can proliferate. Data silos make moves between cloud service providers unattractive. The public cloud service provider fee structure makes it very attractive to move data into the cloud and assume data stays in there. However, taking data out of a public cloud can be prohibitively expensive. Therefore, traditional data architectures such as directly connecting multiple data sources as done with traditional ETL pipelines to maintain EDW or consolidate data onto a single platform as done in most data lake environments create technical and cost challenges in a hybrid cloud environment. The concept of loosely coupled services of a hybrid cloud application architecture is critical throughout a Data Fabric architecture and Data Mesh solution as well. Data or analytics is also exposed as a service or data product with an access URL.

The heterogeneity of data access and data formats increase the pain point of finding relevant and consumable data and AI assets faster with consistent data and AI governance. It becomes even harder to understand data characteristics such as data currency, data quality, and trustworthiness of AI and manage data and AI governance rules and processes as required by the SLA of a data consumer use case.

Let us now explore what it means to deploy a Data Fabric architecture and Data Mesh solution in a hybrid cloud landscape.

Data Fabric and Data Mesh in Hybrid Cloud

Applying a Data Fabric architecture and implementing a Data Mesh solution in a hybrid cloud environment requires similar capabilities as

we have discussed throughout this book, namely, intelligent cataloging, active metadata, semantic knowledge graphs, self-service capabilities, etc. While costs may be reduced by leveraging cloud services, the complexity, however, can increase significantly introducing imperatives for both concepts to bridge additional boundaries and ensuring integration across disparate systems and public cloud providers.

In this section we examine these imperatives, both for a Data Fabric architecture and a Data Mesh solution deployed in hybrid cloud landscapes.

Data Fabric Architecture in Hybrid Cloud

We introduced the Data Fabric architecture in Chapter 11 as a data architecture that defines data standards in an organization. Let us look how it provides the mechanism to manage the added data complexity in a heterogeneous hybrid cloud environment.

Please, review the right side of Figure 3-7 in Chapter 3, which illustrates a sample configuration for data stores in a hybrid cloud environment. It is key to establish a horizontal data management layer across all data stores that need to be *woven* together. This horizontal layer can be materialized based on an enterprise-wide knowledge catalog that includes active metadata from data and AI assets collectively from all systems (including on-premises and private cloud systems) and public cloud providers.

Alternatively, it may also follow a distributed paradigm of services-based knowledge cataloging considering individual knowledge catalogs for data and AI assets stored, managed, and governed per individual system or cloud service provider, as depicted in Figure 12-1.

Both Data Fabric architecture deployment options (distributed and centralized) require integration capabilities to bridge boundaries between systems and public cloud service providers. The distributed deployment option (left side of Figure 12-1) requires orchestration and metadata

exchange capabilities across all catalogs (*Catalog_A*, *Catalog_B*, and *Catalog_C*) as illustrated by the dashed arrows between the catalogs, to enable consumption of data and AI assets and services across systems and public cloud providers.

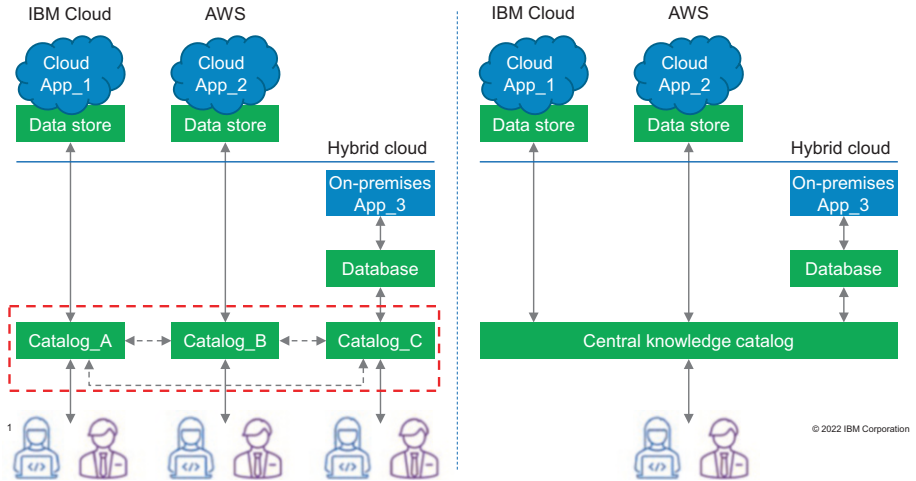


Figure 12-1. Central and Distributed Cataloging in Hybrid Cloud

This can, for instance, be achieved via an Egeria cohort³ implementation, as described in the “Data Product” section of Chapter 2. The centralized deployment option (right side of Figure 12-1) may appear as a more straightforward and simple solution; however, maintaining and leveraging a central knowledge catalog requires connections across all systems and public cloud service providers. Especially performing the rich set of knowledge catalog functions, such as discovery and data profiling, quality assessments, and metadata enrichments, as well as using a central knowledge catalog for DataOps and ModelOps purposes addressing the entire lifecycle of data and AI assets, makes a centralized deployment option challenging as well.

³ See Reference [3] for more information on Egeria.

Figure 12-1 is focusing on the cataloging aspects only. However, additional aspects have to be taken into consideration when discussing a Data Fabric architecture in a hybrid cloud environment, such as data access and consumption by data engineers, data scientists, business users, etc., data and AI governance including trustworthy AI, DataOps and ModelOps, and so forth.⁴

Let us continue the discussion focusing on the deployment of Data Mesh solutions in hybrid cloud environments.

Data Mesh Solution in Hybrid Cloud

Examining the implementation of a Data Mesh within a hybrid cloud environment, and understanding how these two concepts are interrelated, requires us to describe how data products are produced and consumed. Furthermore, we need to detail out the role of the distributed knowledge catalogs in the context of various data sources, their corresponding metadata, and the specifications for the various data products. In addition, the data marketplace – we may rather use the term *data and data product marketplace* – needs to be depicted in this hybrid cloud environment. This discussion assumes familiarity with the concepts as we have described them in Chapter 2 and the section “Data Fabric for a Data Mesh Solution” in Chapter 10.

We are assuming a distributed, organizational, and federated approach, where each LoB or organization is pursuing their Data Mesh initiative, building data products based on their corresponding business imperatives and use case scenarios. As you can imagine, a knowledge catalog enables each LoB to build data products, as well as making the data products discoverable for consumption by business users.

⁴See Reference [4] for more information on Data Fabric in a hybrid cloud environment.

As you can see in Figure 12-2, we are assuming dedicated knowledge catalogs (*Catalog_A*, *Catalog_B*, *Catalog_C*, and *Catalog_D*), which are dedicated to the four LoB organizations using on-premises infrastructure and services from three public cloud providers (IBM Cloud, AWS, and MS Azure). For simplification purposes, we are furthermore assuming a relationship between data domain owners and LoBs who are tasked with building their corresponding data products. Each catalog is capturing the metadata, which is related to various data sources, owned by corresponding data domain owners.

For instance, *Data_domain_owner_1* owns *Data_source_1*, which serves as a base to automatically generate the *Data_source_1 metadata*. In addition to this metadata, the data product developers (including data engineers) in conjunction with data product owners are generating data product specifications that are equally stored in the catalog.

These data product specifications are generated during the product build process and stored in the catalog using, for instance, XML, JSON, or any other data exchange standard.

Once data products are registered in the data product marketplace, they can be discovered through access methods, locations (i.e., URI endpoints), SLAs, etc. Generating the data product specification requires access to the metadata of the corresponding data sources, as well as to the data sources themselves. The data products themselves are not stored in the data product marketplace; they are rather registered there using XML, JSON, etc. and can therefore be searched for, discovered, and consumed.⁵

There are two particular aspects regarding Data Mesh solutions in hybrid cloud environments that are related to *cross-LoB scenarios*. The first scenario relates to data products that are derived from data sources that belong to different data domain owners, whereas the second scenario relates to data products that are derived from data sources from

⁵ See Reference [5] for more information on Data Mesh deployments in hybrid cloud environments.

their corresponding data domain as well as other data products that are produced by a different LoB. Let us examine these two scenarios in detail, as they have been depicted in Figure 12-2 as well:

- **Scenario 1** – data products that are derived from data sources that belong to different data domain owners: As depicted in Figure 12-2, Data_product_B is based on Data_Source_2 belonging to Data_domain_owner_2, as well as Data_Source_1 belonging to Data_domain_owner_1. Thus, in order to build Data_product_B, complementary access to the metadata generated from Data_Source_1 (dashed arrow 1) as well as to Data_source_1 itself (dashed arrow 2) is required. The Data_product_B specification stored in Catalog B is thus derived from artefacts that are owned by Data_domain_owner_1 and Data_domain_owner_2.
- **Scenario 2** – data products that are derived from data sources from their corresponding data domain as well as other data products: As illustrated by Figure 12-2, Data_product_D is based on Data_Source_4 belonging to Data_domain_owner_4, as well as Data_product_C. Thus, in order to build Data_product_D, complimentary access to Data_product_C (dashed arrow 3), as well as to the Data_product_C specification (dashed arrow 4) and Data_Source_3 (dashed arrow 5), is required. The Data_product_D specification stored in Catalog D is thus derived from artefacts that are owned by Data_domain_owner_3, Data_domain_owner_4, and the owner of Data_product_C.

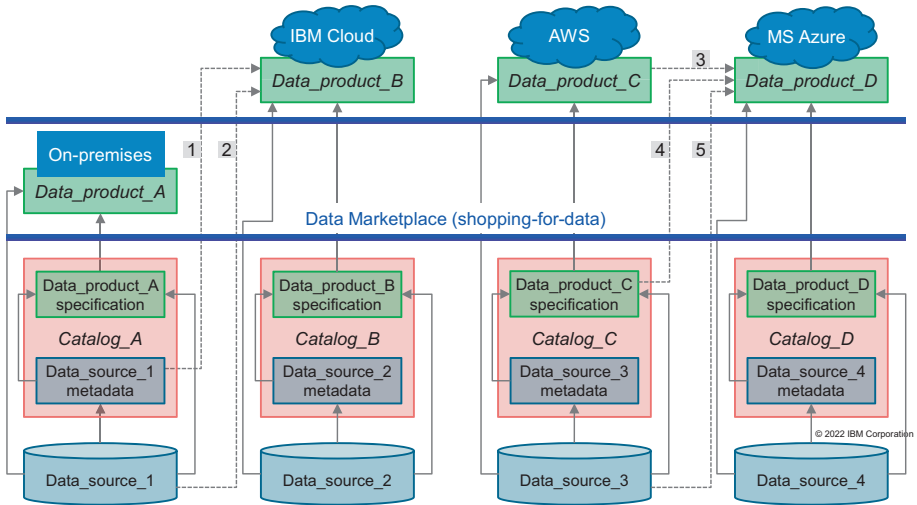


Figure 12-2. Data Mesh in a Hybrid Cloud Environment

The challenges of these cross-Lob Data Mesh scenarios are presumably derived from the need to integrate cataloging including metadata exchange and consumption and even data product build and consumption across different on-premises infrastructure and public cloud service providers.⁶ This requires data authorization and security measures and data exchange standards to be in place, implemented via hybrid cloud-wide data and AI governance, risk, and control. Furthermore, to adhere to Data Mesh self-service capabilities, DataOps, ModelOps, AIOps, trustworthy AI, etc. need to be implemented across on-premises and public cloud service providers addressing the end-to-end lifecycle requirements of data products, imposing additional challenges for an already difficult endeavor.

In the following section, we describe just a few benefits of Data Fabric and Data Mesh for a hybrid cloud environment.

⁶ See Reference [6] for more KPMG’s view on Data Mesh for hybrid cloud.

Benefits of Data Fabric and Data Mesh for Hybrid Cloud

Although it may be challenging to implement a Data Fabric architecture and Data Mesh solution for a hybrid cloud environment, key benefits can doubtless be materialized. With the inevitable trend moving data and AI to the cloud and leveraging public cloud services, a Data Fabric architecture and Data Mesh solution augments the cloud value by enabling cross-organizational and cross-domain data and AI governance and enables data consumers and business users to build data products for a data product marketplace with data and AI assets regardless of where they reside and whom they are owned by.

Entangling a Data Mesh with hybrid cloud provides agility and flexibility regarding deployments and consumption of data products across public cloud service providers; data products may be deployed in one public cloud (i.e., IBM Cloud) and used as input to produce another data product that is deployed in a different public cloud (i.e., MS Azure). For instance, data may be stored in one public cloud, leveraged for training, validation, and testing of an AI model on a second public cloud, and eventually deployed and operationalized as a data product on-premises, where it is used for inferencing within a transactional application.

Building a Data Mesh solution for hybrid cloud provides flexibility in using technology and products from different public cloud service providers, mapping their services and capabilities to the development- and deployment-related requirements of a particular data product. For instance, data products may be associated with different business domains and purposes, for example, AI models, BI reports, ETL processes, 360-degree customer data, etc. These different domains require different services and data product development and operationalization environments. A hybrid cloud landscape is particularly well suited to address these specific needs.

A Data Fabric architecture enables such a distributed Data Mesh solution for a hybrid cloud environment. Implementing intelligent information integration, generating and leveraging active metadata, deploying federated cataloging with metadata exchange, and establishing self-service capabilities for data and AI asset consumption are essential capabilities to underpin such a Data Mesh solution in a hybrid cloud landscape.

Key Takeaways

We conclude this chapter with a few key takeaways as summarized in Table 12-1.

Table 12-1. *Key Takeaways*

#	Key Takeaway	High-level Description
1	Hybrid cloud is an agile, distributed compute environment.	Hybrid cloud integrates public cloud services, private cloud services, and on-premises infrastructure to provide an agile and flexible distributed compute environment.
2	Hybrid cloud increases data access complexity.	Heterogeneity of data access and data formats in a hybrid cloud increase the pain point of finding relevant and consumable data faster with consistent data governance.
3	Data Fabric adds a data management layer.	Data Fabric architecture provides the mechanism to manage the added data and AI complexity in a hybrid cloud landscape.
4	There are two Data Fabric deployment options.	Two Data Fabric architecture deployment options (distributed and centralized) can be chosen, depending on preferences and use case requirements.

(continued)

Table 12-1. (continued)

#	Key Takeaway	High-level Description
5	There are two Data Mesh scenarios in hybrid cloud.	The two Data Mesh scenarios enable cross-LoB and cross-organizational data source and data product deployments.
6	Hybrid cloud and Data Mesh combined represent additional value.	Data Mesh with hybrid cloud provides agility and flexibility regarding deployments and consumption of data products across public cloud service providers.
7	A Data Fabric is required to enable a Data Mesh.	A Data Fabric architecture enables and underpins a distributed Data Mesh solution for a hybrid cloud environment.

References

- [1] IBM, IBM Cloud Learn Hub, *What is Hybrid Cloud?*, www.ibm.com/cloud/learn/hybrid-cloud (accessed October 14, 2022).
- [2] Andreessen.horowitz, Wang, S., Casado, M., *The Cost of Cloud, a Trillion Dollar Paradox*, <https://a16z.com/2021/05/27/cost-of-cloud-paradox-market-cap-cloud-lifecycle-scale-growth-repatriation-optimization/> (accessed October 14, 2022).
- [3] The Linux Foundation Projects, *Egeria – open metadata and governance for enterprises*, 2022, <https://egeria-project.org/> (accessed October 18, 2022).

- [4] IBM, Lighthouse, *Weaving data fabric into hybrid multicloud: Doing more with connected data*, <https://w3.ibm.com/services/lighthouse/documents/168319> (accessed October 21, 2022).
- [5] Cloudera, Data Architecture Series, *The Data Mesh Paradigm*, www.cloudera.com/content/dam/www/marketing/resources/whitepapers/the-data-mesh-paradigm.pdf?daqp=true (accessed October 22, 2022).
- [6] KPMG, *KPMG Data Mesh for Hybrid Cloud – Modernize your enterprise data operations*, <file:///Users/eberhardhechler/Downloads/data-mesh-ibm.pdf> (accessed October 22, 2022).

CHAPTER 13

Intelligent Cataloging and Metadata Management

Suppose you are a business analyst and you need to find the customer purchase records of a certain market in the last quarter from 1.3 million tables and billions of records to make a predictive analysis of the consumption trend in this region for the next quarter. How are you going to do it? This task is like looking for a needle in a haystack. What is even more frustrating is that when you finally find the relevant data after spending weeks on source data exploration, it is out of date, and new essential data is available. This example illustrates that it is not sufficient for companies to make data accessible; they also need to make it discoverable, understandable, and consumable in near real time to gain timely and relevant insights from the data.

Introduction to Metadata Management

In the digital era, enterprises need to know all aspects of data:

- What data they have, where it is, how much is there, and what data needs to be protected

- What are the business values of data, what is the quality of data, and when does the data need to be archived
- Where does the data come from, who owns the data, who uses the data, for what business purposes, and so on

To address the preceding questions effectively, enterprises need to put metadata management into place. As we have described before, metadata¹ is the data used to describe the data. Let's look at a real-life example. On many occasions, we need to do self-introductions. When we introduce ourselves, we usually introduce name, position, the company we're working for, the school we graduated from, etc. We may also introduce family, hobbies, etc. Name, occupation, working experience, education, family background – all this can be seen as metadata used to define our identity, which enables others to know who we are and how they connect with us in social life, as depicted in Figure 13-1. In short, metadata helps users comprehend and communicate meaning associated with data.

¹See Reference [1] for more details on the metadata definition.

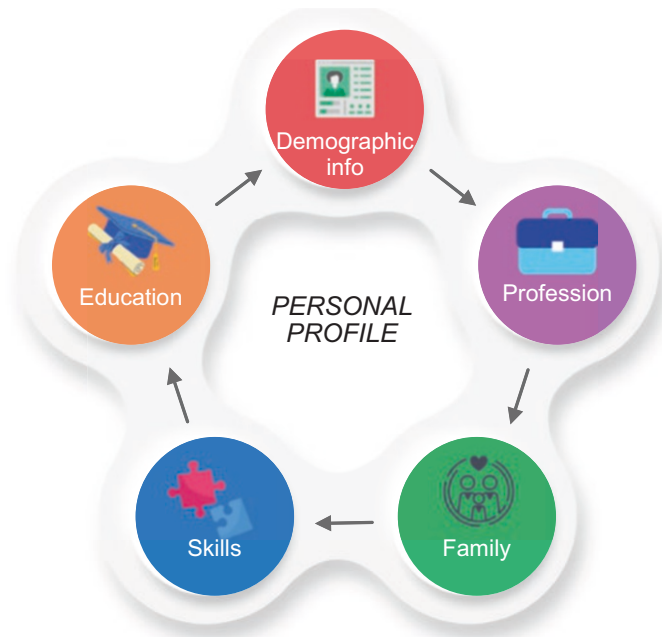


Figure 13-1. *Metadata for Personal Profile*

Metadata can be broadly classified into three categories: business metadata, technical metadata, and operational metadata.² Business metadata describes the meaning of the business that the data represents in the real world. For example, this nine-digit string *111-22-3344* can represent a person's Social Security number (SSN), an account number, or an application confirmation number. The validation rules and protection rules vary depending on the business context. For example, an SSN belongs to Personal Confidential Information (PCI), which is classified as personal sensitive data and requires the application of high-level security and privacy protection. In summary, business metadata facilitates the understanding of data; it furthermore links data with corresponding business rules.

² Please review Chapter 5, where we have introduced metadata in the context of generating active metadata.

In contrast, technical metadata describes the access information of data in an IT environment, for example, where data is stored, in which format, and how to access it. For example, an SSN is stored in a column named *ID* of a table in a Db2 database. The column name, table name, database name, and database connections are all technical metadata, which can serve data engineers to get access to data. In addition, data types, constraints, and dependencies are technical metadata too.

Operational metadata is concerned about the creation and transformation of data and AI assets. Table 13-1 lists some examples of metadata.

Table 13-1. *Business Metadata, Technical Metadata, and Operational Metadata*

Metadata	Purpose	Examples
Business metadata	Business meaning in the real-world for understanding data	Business definitions, the explanations of business terms; KPIs, calculated KPIs, and derived KPIs; business identification rules, quality rules, protection rules, security and privacy level, etc.
Technical metadata	Technical information of IT platforms for retrieving data	Physical database table names, column names, field lengths, field types, constraint information, data dependencies, SQL script information, ETL conversion, data update frequency, etc.
Operational metadata	Data concerning transformation of data and AI assets	ETL stages, I/U/D logs, pipelines, job execution logs including runtime parameters, SQL query execution logs (for instance, access path information), AI asset usage logs, etc.

Let us go back to metadata management.³ An effective metadata management platform should have the following capabilities:

- **Hierarchy:** Establish a complete hierarchy and system of business metadata, technical metadata, operational metadata, policies and rules, and ultimately active metadata.
- **Quality:** Create a continuous quality inspection framework to understand and improve data integrity and accuracy through metadata information.
- **Insight:** Enable users to clearly understand and visualize details of the entire data flow across enterprises to improve data traceability, addressing the entire lifecycle of data and AI assets.

Metadata management is not a one-time job; metadata needs to be administered and maintained reiteratively. Therefore, from a technical point of view, the metadata management platforms usually include components such as metamodel management, metadata audit, metadata maintenance, metadata change management, and metadata version management. There are many mature commercial products available in the market, such as IBM Watson Knowledge Catalog, Informatica, Alation, etc.

Key Aspects of Intelligent Cataloging

The enterprise knowledge catalog is a system for organizing and storing data according to well-established metadata systems. Usually, the library catalog is used to explain the concept. Suppose you go to the library and want to find a book. You can find the shelf number by genre: fiction,

³ See Reference [2] for more details on metadata management.

nonfiction, textbooks, and so on. Or you can find it by author and the year of publication. The title, the author, and the year of publication are all considered metadata. The place that stores this information and the location of the book is the library catalog. The process of cataloging is a nontrivial endeavor. When a new batch of books comes in, the first task librarians need to tackle is to enter the information of the books into the knowledge or library catalog, according to the metadata framework.

Now imagine if the incoming data is stored in tens of thousands of tables with billions of records. It would be a time-consuming and labor-intensive task to catalog these data records with accurate metadata. That's where AI comes into play. Intelligent cataloging⁴ is to use ML models that automate the cataloging process to the maximum extent.

It generally has the following capabilities:

- **Automated data discovery and enrichment:** Responding to the complex and diverse data environment of enterprises, the intelligent knowledge catalog automatically discovers data and enriches data assets with business definitions based on AI-infused technologies, including profiling data, assigning business terms, classifying data, analyzing the data quality, correlating data assets, and enforcing the privacy and security rules for data. The data discovery process is the foundation of intelligent cataloging; it quickly creates a knowledge catalog and then makes data ready for use.
- **Semantic search and recommendation:** To help business users quickly identify and locate data, intelligent cataloging provides powerful semantic search capabilities. The difference between semantic search and full-text search is that instead of just

⁴See Reference [3] for more information on intelligent data catalog tools.

matching for textual similarity, semantic search aims to understand the intent and contextual meaning of your search and thereby to provide for more relevant content. In addition, the intelligent knowledge catalog will also make recommendations of similar data assets based on the user's search results, which can also greatly improve the efficiency of finding data.

- **Data lineage and provenance:** Data lineage is the end-to-end flow of data across the enterprise, and as part of the data asset catalog, it provides tracking and tracing throughout the data lifecycle to understand where the data came from, how it was transformed, and who is using it for which purpose. Typically, data lineage diagrams visualize the relationships between tables, views, fields, etc. using a directed acyclic graph model. It shows whether data comes from a trusted source and if there are any malicious attempts on the data. Data provenance sheds light on the origin of data or any AI asset, for example, who owns the data or AI asset, who has created it, etc.

Build an Intelligent Catalog by Automating Data Discovery and Enrichment

The intelligent data catalog first connects to various data sources in the enterprise and extracts metadata from them. The process is known as data discovery. Typically, users specify the connection to the data source, and data discovery scans the catalog tables of data sources with connection information to retrieve various technical metadata, for example, field name, data format, etc. On the other hand, data enrichment refers to the

process where intelligent cataloging assigns business terms, analyzes the quality, and adds relationships between data assets based on metadata.

For instance, the format of a column may indicate the data is an SSN. In this case, the column will be assigned the business term *ID*, which elevates the privacy level to PII information. The name of the field may also indicate that it is related to an address. In this case, the field will be linked to other fields, and it is tagged as contact information.

The volume of data in the enterprise has proliferated, and performing data discovery and enrichment tasks manually seems to be impossible. Automated discovery and enrichment tasks do not require human invention; they are – to a maximum extent – performed by ML algorithms.

To better illustrate this process, let's go back to the library example, as illustrated in Figure 13-2. Suppose there are several books that need to serve as input to the library catalog. The location of these books can then be seen as the connection information of data sources, while the book is regarded as the data record. With automated data discovery, the scattered books are then automatically identified and organized in the catalog.

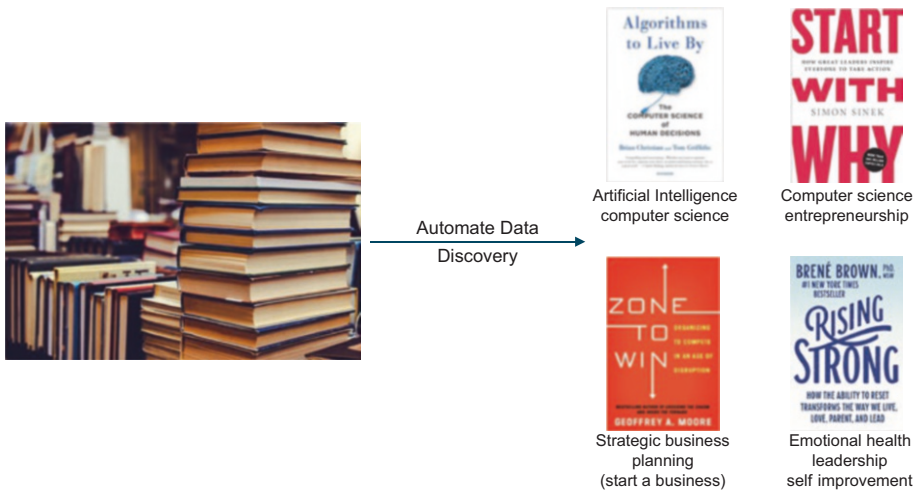


Figure 13-2. Illustration of the Data Discovery Process

In addition to the basic information as author, ISBN, and publisher, the topics of the books are also recognized by analyzing book review abstracts and comments in websites by NLP technology.⁵ Judging from the title of the book, it is easy to categorize *Rising Strong* to the area of emotional intelligence. But it's not straightforward just by reading the name *Start with Why* that this is the title of a computer book.

That is where sophisticated ML algorithms can help. The higher the level of AI applied to the knowledge catalog is, the more comprehensive the derived knowledge graph behind the catalog will be. AI can support a lot of the heavy lifting work, but nonetheless it cannot ultimately replace subject matter experts to deal with complex situations. After all, the accuracy of AI models depends on historical training of models and corresponding AI algorithms. Indeed, it may not always yield the correct results.

In this case, AI models can provide a suggestion, which subject matter experts need to confirm for acceptance or rejection, as illustrated in Figure 13-3, where *Transaction* is a table in a Db2 for z/OS database. After discovering and enriching the *Transaction* table via a Db2 for z/OS connection by intelligent cataloging, the column *Transaction_TS* is suggested as the business term *Transaction_ID* with an 82% confidence level.

⁵ Please, see Chapter 6 for more details on NLP.

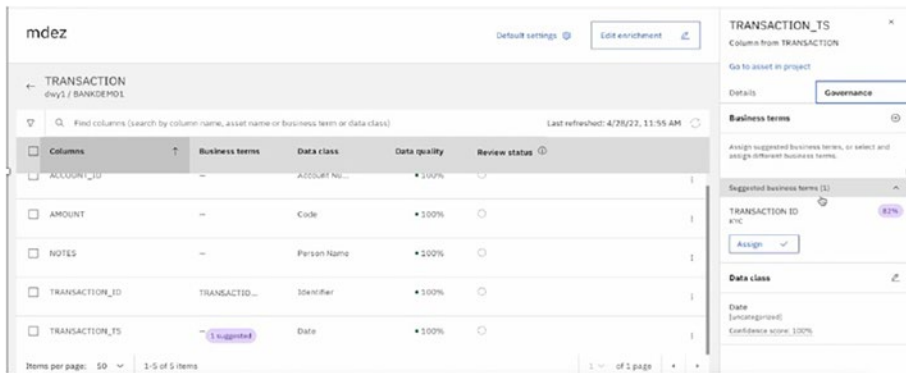


Figure 13-3. *Enrichment of Data with Business Terms*

The data curator could review the suggested business term and decide whether it should be assigned to this column or not. Once the *Transaction_ID* is assigned, the policies and rules that are defined with *Transaction_ID* will be enforced on the data of the column *Transaction_TS*. For example, the data protection rule or data locality rule will mask the data in *Transaction_TS*.

Find Data Assets with Semantic Search and Recommendation

In real-world situations, very likely business analysts and data scientists are struggling to discover and access relevant data sources and to fully understand the structure and content of a particular data source. Subsequently, they need an effective way to search for it and to perform data exploration tasks. However, using keyword searches requires a solid understanding regarding the most relevant keywords. If the keywords are not chosen correctly, users often cannot find the relevant data.

This is where semantic search comes into play. Semantic query⁶ attempts to capture the users' thoughts and relevant context, also referring to the relationship between search entities and providing more accurate and more comprehensive query results. For example, if you are browsing the menu of a restaurant website, you may search for *soap*. The semantic search will guess that you are looking for *soup* and may then return today's options. Assuming that you are using a supermarket app searching for *soap*, the semantic search will likely return products related to detergents and stain removers. Semantic search queries return the most suitable results based on a specific context, for example, what are you currently doing and why are you doing it, trying to understand your most likely intent, as illustrated in Figure 13-4.



Figure 13-4. *Illustration of Semantic Search*

Figure 13-5 shows another example of a semantic search, where users intend to find data assets that have contact information.

⁶ See Reference [4] for more use cases about semantic search.

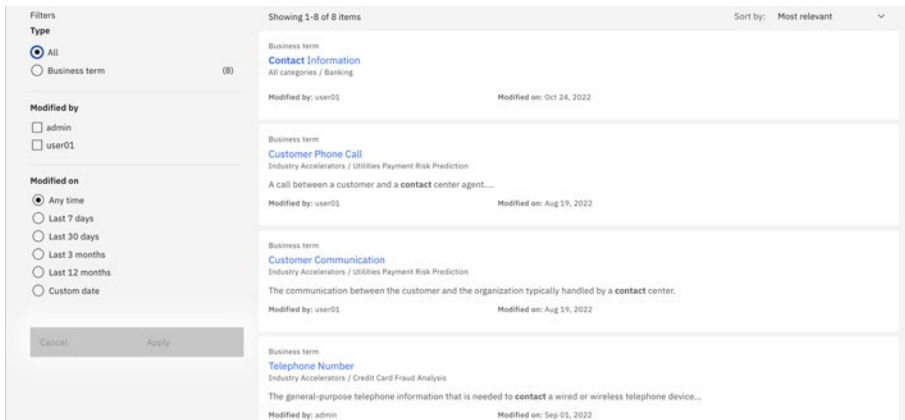


Figure 13-5. Example of Semantic Search Results for Contact Information

Users input *Contact* in the global search field, and multiple records are returned as search results including *Telephone Number*, *Customer Phone Call*, *Customer Communication*, etc. The search result does not contain a column that is labeled exactly as *Contact*. If using keyword search, these data records are least likely to be returned, but they are actual contact information that users are looking for. When zooming into *Telephone Number*, it is marked as having a relationship with *Communication Content*, as illustrated in Figure 13-6.

Telephone Number Published ⋮

Overview Related content

^ General

Description ✎

The general-purpose telephone information that is needed to contact a wired or wireless telephone device.

Primary category ✎

Credit Card Fraud Analysis
Categories / Industry Accelerators

Secondary categories ⊕

No secondary category added yet.

^ Related terms

Type relationships

Is a type of Show list ⊕

Communication Content
↳ Telephone Number

Has a type of ⊕

No business term added yet.

Part relationships

Is a part of ⊕

Customer
Industry Accelerators / Credit Card Fraud Analysis

Has a part of ⊕

No business term added yet.

Figure 13-6. *Semantic Search and Business Terms*

Recommendation is another powerful feature for intelligent cataloging in conjunction with semantic search. Recommendation may depend on usage metrics, historical searches, and peer insights. Metrics-based means that asset recommendations are based on usage data, such as popularity, top rated and quality scores, etc. Historical search leverages previous searches to infer what assets would be of interest. Peer insights are based on the usage history of others.

This is like our online shopping experience. In ecommerce, there are often targeted recommendations that are specifically customized for you that are based on other customers' purchase behavior, often linking one product to a set of others. This concept can be easily applied to searching for data, thus improving the overall shopping-for-data experience. This greatly improves the efficiency of business analysts and data scientists to find relevant and useful data, for instance, influencing targeted marketing campaigns that are geared toward efficient cross-selling and upselling initiatives.

Let us move on to data insight and provenance

Provide Data Insight and Provenance as Data Flows Across the Enterprise

One of the top data challenges customers are facing today is gaining trust in data and AI (trustworthy AI). How can I *prove* to data consumers that the data is trustworthy? How can I provide transparency in terms of data provenance and whether the data was changed inadvertently?

This is not surprising; with the explosion of information, there is a reason to question the credibility of data, as incomplete and outdated information does not benefit decision-making. Instead, it may lead to wrong decisions with significant business ramifications. Data lineage addresses this concern by documenting, tracking, and visualizing the data processing and transformation across disparate tools and sources, as illustrated in Figure 13-7 for the workflow of data lineage.

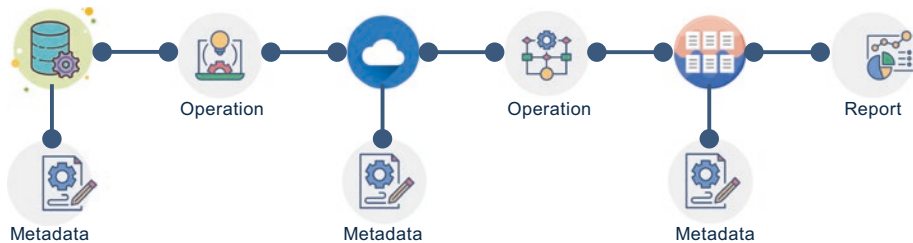


Figure 13-7. Illustration of Data Lineage

Data lineage and provenance⁷ are often used interchangeably. Both terms refer to the entire lifecycle of the data, including the five Ws: (a) where the data originates, (b) where the data has been and where is the destination, (c) who made changes to the data, (d) when the data was created or updated, and (e) where the data is stored and used. Knowing answers to these questions is critical to data consumers to trust analytics outcomes derived from data.

⁷ See Reference [5] for more details on data lineage and provenance.

Figure 13-8 shows an example of data lineage. There are three types of nodes: a data node, operation node, and report node. The data asset *borrow* comes from multiple data sources, *books*, *audiobooks*, and *person*, via the operations *borrow_book* and *borrow_audiobook*.

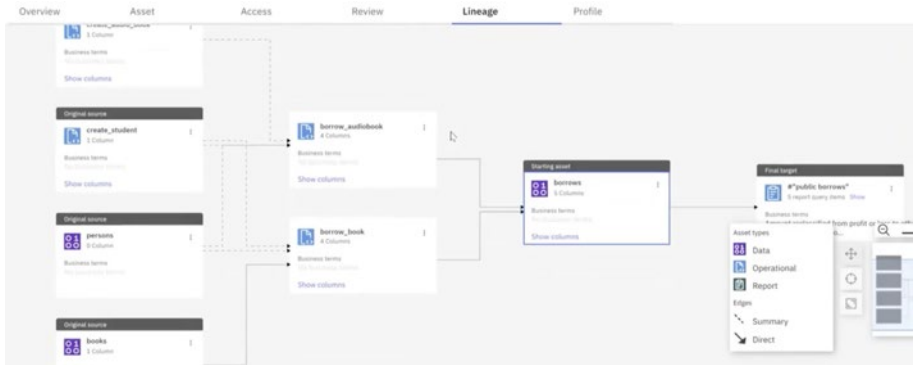


Figure 13-8. Example of Data Lineage

The lineage diagram can easily become very complex. Therefore, it is important to have zoom-in and zoom-out and expand and collapse capabilities to navigate through the diagram allowing users to focus on the specific part of the diagram and to advance step by step.

Let’s drill down to the next level of detail, referring to Figure 13-9. The table *students* is constructed by an insert statement with data fields: *student_name*, *student_surname*, *student_birthdate*, and *student_gender*.

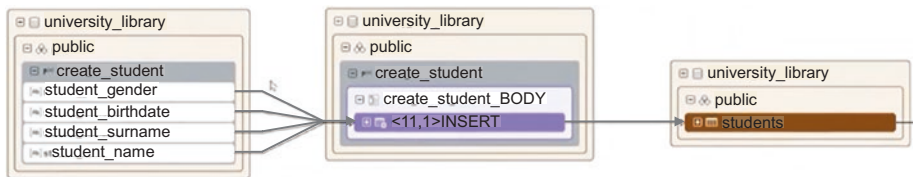


Figure 13-9. Lineage Details

Thus, users not just know where the data came from and what transformation processes were applied to it. With data lineage, enterprises can track the data as it flows through the organization and trace data quality issues back to specific processes or systems.

What is even more important, with impact analysis, organizations can assess the impact of data changes in one system on the entire enterprise.

Key Takeaways

We conclude this chapter with a few key takeaways as summarized in Table 13-2.

Table 13-2. *Key Takeaways*

#	Key Takeaway	High-Level Description
1	Metadata is the data used to describe data.	Metadata can be broadly classified into three categories: business metadata, which describes the meaning of the business that the data represents in the real world, and technical metadata, which describes the access information of data in the IT platforms, and operational metadata which describe the creation and transformation of data.
2	Intelligent cataloging contains three key capabilities.	Intelligent cataloging uses AI to automate the data discovery and enrichment process to improve efficiency, and provide semantic search and recommendation for easy consumption, and provide data lineage to improve trust in data.
3	Data discovery and enrichment are critical to building the catalog.	The intelligent data catalog connects to all enterprise data sources, extracts metadata from the data source, uses ML algorithms to assigns business terms, analyzes the quality, and adds relationships between data assets.

(continued)

Table 13-2. (continued)

#	Key Takeaway	High-Level Description
4	Semantic search yields better results that fit your purpose.	Semantic search aims to understand the intent and contextual meaning of your search and thereby provide for more relevant content.
5	Data lineage is the process of tracking how data is moved, transformed, and consumed across disparate sources.	Data lineage addresses five Ws: where the data originates, where the data has been and where is the destination, who made changes to the data, when the data was created/updated, and where the data is stored and used. Knowing answers to these questions is critical for people to trust analytics outcomes from data.

References

- [1] TechTarget, Kranz, G., *Definition metadata*, www.techtarget.com/whatis/definition/metadata (accessed September 2, 2022).
- [2] TIBCO, *What is Metadata Management?*, www.tibco.com/reference-center/what-is-metadata-management (accessed September 2, 2022).
- [3] AI Outlook, Staff, A., *Intelligent Data Catalog Tools Summarized*, 2022, www.aixoutlook.com/intelligent-data-catalog-summarized/ (accessed September 2, 2022).
- [4] Bloomreach, Roberts, T., *Semantic Search Explained in 5 Minutes*, 2019, www.bloomreach.com/en/blog/2019/semantic-search-explained-in-5-minutes (accessed September 2, 2022).

- [5] Electrosoft, Proud-Madruga, D., *Provenance, Lineage, Pedigree: Are they the Same?*, 2020, www.electrosoft-inc.com/electroblog/provenance-lineage-pedigree-are-they-same (accessed September 2, 2022).

CHAPTER 14

Automated Data Fabric and Data Mesh Aspects

The vigilant reader of this book has certainly noticed the distinguished focus that we have put on applying AI with automation and intelligent augmentation and optimization to nearly all aspects of a Data Fabric architecture and Data Mesh solution. There are indeed numerous areas of both concepts, which are increasingly optimized with AI-infused automation, such as automated workload performance prediction and runtime adjustment, automated capacity planning and resource demand estimation (e.g., CPU capacity, network bandwidth, memory sizes, etc.), automated query generation, intelligent information integration, automated data curation, and automated creation of data products.

Especially automated capacity planning and forecasting has been a well-established topic for vendors¹ and in academia² as well. Applying AI and automation clearly amounts to a paradigm shift. In fact, we would not use the terms *Data Fabric* and *Data Mesh* in the context we do without the inherent assumption of applying AI, intelligent knowledge, and automation.

¹ See Reference [1] for an example from IBM about performance and capacity planning.

² See Reference [2] for more information on capacity planning for enterprise Data Fabrics.

The desire to create an organization- or enterprise-wide description of data and AI artefacts is not at all a new concept. It was already painfully considered a failure about two decades ago. This chapter describes the usage of intelligent automation to collect metadata information from different data sources and catalog them, automatically checking data quality and augmenting the metadata as well as automating data governance services as a foundation for both essential Data Fabric and Data Mesh concepts.

In this chapter, we thus concentrate on two distinct but nevertheless related areas: (a) intelligent automation of metadata and (b) automated data quality assessment.

Automated data and AI governance services will be addressed in the next chapter.

Introduction

Applying AI to metadata management and intelligent cataloging, data quality assessments, and especially data and AI governance seems to have lagged far behind the rest of data-related areas. In Chapters 5 and 7, we have already elaborated on the vital role of AI/ML to reimagine a modern approach, that is, to generate active metadata, to activate the digital exhaust, to build semantic knowledge graphs, and likewise for semantic enrichment and semantic search, entity matching, self-service capabilities, and automated data quality assessments and adjustments – well underpinned and empowered by must-have knowledge catalog capabilities.

In Chapter 13, you have seen how indispensable the knowledge catalog is for intelligent cataloging and enriching metadata and to enable, for instance, data lineage and provenance. As we have mentioned previously, the core content of this chapter is clearly on intelligent automation of metadata and automated data quality assessments.

Let us begin with intelligent automation of metadata.

Intelligent Automation of Metadata

We use the term *intelligent automated metadata* to portray the inclusion of AI/ML into the entire lifecycle of metadata management, including generation, enrichment, and consumption of metadata.

As mentioned, the knowledge catalog is the core component to enable intelligent automation of metadata. Figure 14-1 is a sample knowledge catalog,³ where four recently added data assets are visualized, including the owner, the date when the asset was added, and the type of asset. As you can see, three of these assets are relational database tables, and the fourth asset is a database connection based on data virtualization (right side of Figure 14-1).

The screenshot shows the 'Recently added data assets' section of the IBM Watson Knowledge Catalog. It displays four asset cards, each with details like name, owner, and date added. Below the cards is a table listing the assets with columns for Name, Owner, Tags, Business terms, Asset type, and Date added. Red boxes and arrows highlight the Owner and Asset type columns in both the cards and the table.

Name	Owner	Tags	Business terms	Asset type	Date added
Data Virtualization	user11			Connection	Aug 28, 2022
USER11.CREDITCARDS_DV	user11			Data	Aug 28, 2022
USER11.CREDITTRANS_DV	user11			Data	Aug 28, 2022
USER11.CREDITUSERS_DV	user11			Data	Aug 28, 2022

Figure 14-1. Sample Knowledge Catalog

³ All samples in this chapter are derived from the IBM Watson Knowledge Catalog.

Additional information can be displayed by clicking the various assets. The power of the knowledge catalog with its corresponding function – above and beyond visualization – is based on a set of intelligent automation capabilities.

These intelligent automation capabilities are described in the following list:

1. **Automated metadata generation:** Data and AI assets, which may span across a multitude of sources or formats, need to be automatically discovered; corresponding metadata must be auto-generated and moved into the knowledge catalog, which applies to changes of the source data and AI assets as well. This comprises the first step of the end-to-end lifecycle of intelligent metadata management, which we may call automated cataloging of data and AI assets or automated metadata generation. This phase may be complemented with automated quality assessments of all data and AI assets. Quality assessments and improvements must be performed repetitively; we are dealing with this topic in a separate section further in the following.
2. **Automated metadata enrichment:** Once metadata has been auto-generated, it needs to be enriched, meaning that the metadata needs to be analyzed and profiled using AI/ML methods to gain additional insight to be added to the metadata. Furthermore, the metadata needs to be tagged and annotated and further optimized and prepared for consumption by data product owners or business users, for instance, by building and adding semantic

knowledge graphs, leveraging industry- or domain-specific ontologies, etc. This phase of metadata enrichment represents by far the most complex stage in the end-to-end metadata lifecycle and may also include transformation and adherence to metadata standards.⁴ This metadata enrichment process may have to be performed repetitively.

3. **Automated metadata extraction and exploitation:**

Once metadata has been generated, captured in the knowledge catalog, and further enriched, it is ready to be extracted for consumption and exploitation by business users. This may sound like a simple undertaking; nevertheless, it requires a rich and easy-to-use GUI-based knowledge catalog, which lists the available data and AI assets in context, meaning type and format of the assets (i.e., relational database tables, JSON or XML documents, Parquet and Avro files, CSV files, connection methods, AI models, pipelines, ETL stages, etc.), ownership, correlation with other assets (i.e., via semantic knowledge graphs), and access rights and access methods (i.e., SQL, NoSQL, REST API).

Figure 14-2 is an illustration of intelligent automated metadata management⁵ with the three phases discussed previously.

⁴ See Reference [3] for a comprehensive list of metadata standards and Reference [4] for Egeria, an open metadata standard from the Linux Foundation.

⁵ See Reference [5] for mor information on automated metadata management.

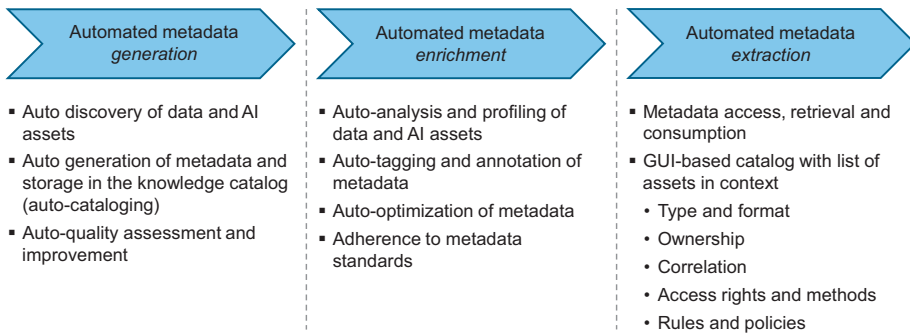


Figure 14-2. *Intelligent Automation of Metadata*

Intelligent automation of metadata must also address the needs of data engineers and data product developers.

The following two subsections provide a glance on automated analysis and profiling of data and automated tagging and annotation of data. By addressing these issues, we provide some basic examples.

Let us begin with automated analysis and profiling of data.

Automated Analysis and Profiling of Data

Data analysis and profiling has been around for decades. What made this notably different in a modern Data Fabric architecture and Data Mesh solution, however, is the infusion of AI/ML and the introduction of automated processes. Let us explore further what this AI/ML infusion really means.

As we have pointed out in Chapter 7, data profiling is the process of examining, analyzing, and checking the content of all data attributes of relevant data sources to get a first path of statistical insight and an initial understanding regarding data quality.⁶ This data analysis and profiling process can broadly be categorized into the following three areas – (a)

⁶See Reference [6] for more information on profiling data assets.

structure discovery, (b) content discovery, and (c) relationship discovery – where each area generates metadata and statistics about its content, which is automatically captured in the knowledge catalog.

Figure 14-3 is an example visualizing statistics of some columns in a database table, called *CREDITUSERS*. We have included four columns – *USER_ID*, *PERSON*, *CURRENT_AGE*, and *RETIREMENT_AGE* – including the corresponding quality score and value frequency distribution for each column. Figure 14-3 also contains some basic statistics for each column, such as number of unique values, minimum and maximum values, mean value, standard deviation, etc. It is essential to realize that these values are calculated directly from real database table data, not statistical data that may be included in the database catalog.

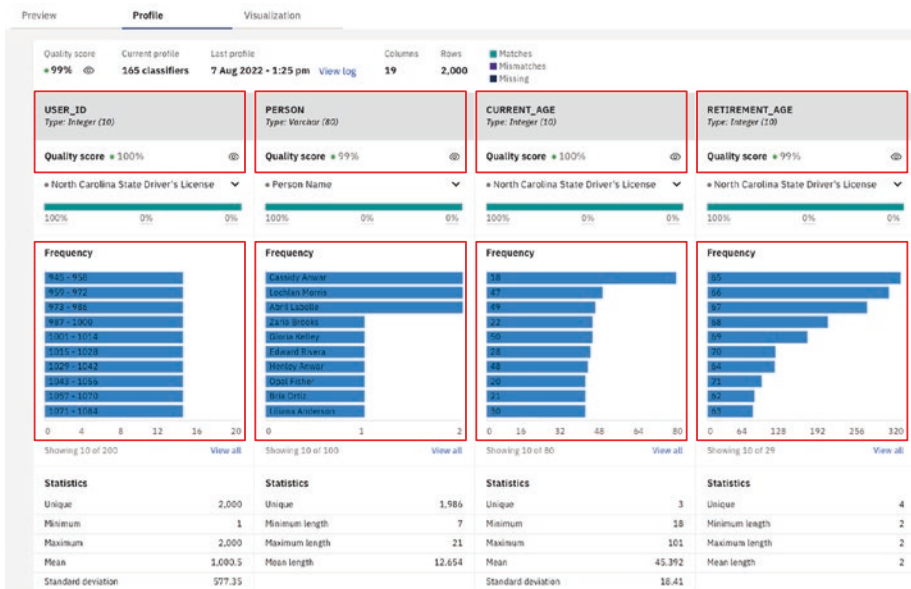


Figure 14-3. Statistics of Each Column in a Database Table

In Chapter 7, we have also mentioned that ML can be used to accelerate and automate data profiling tasks; even DL techniques can be deployed to calculate quality measures, for instance, of text data. Yet we still need to detail out the crucial factors of how and for what specific purpose to use ML and DL.

Before moving on to discuss the usage of ML and DL, we include another example of user counts of each US state in the *CREDITUSERS* table, as depicted in Figure 14-4, with the US states of New York (NY), California (CA), Florida (FL), and Texas (TX) among the highest counts and the US states of Wyoming (WY), Arkansas (AK), and the Federal District of Columbia (DC) among the lowest counts. Of course, this is a rather traditional depiction; a more sophisticated depiction would, for instance, be based on ML-based clustering, meaning to identify and visualize various clusters of US states to, for instance, discover and depict demographic similarities of US states.

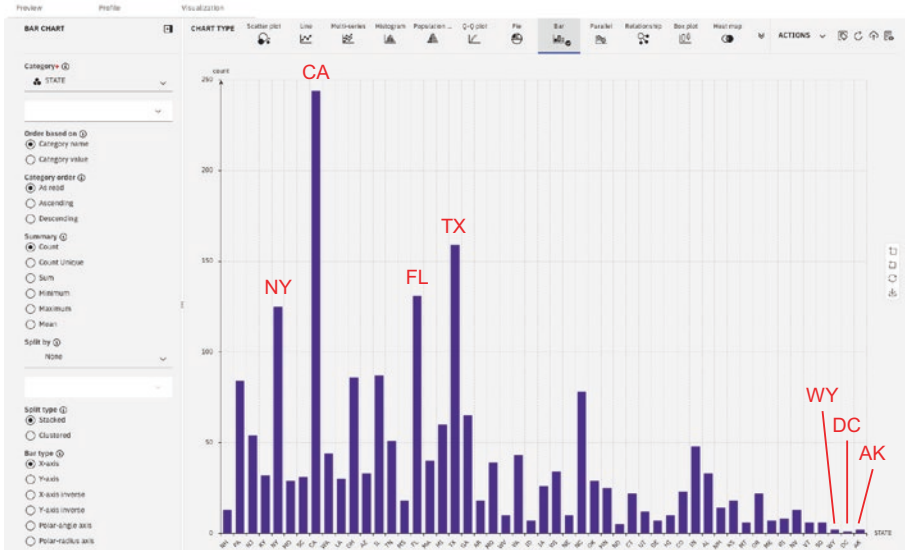


Figure 14-4. User Counts of Each US State on the *CREDITUSERS* Table

This leads us to the endeavor of detailing out the role of AI in performing data analysis and profiling tasks.⁷

The following list includes key capabilities of data analysis and profiling, which are applied via ML and DL:

- **Determining representative sample datasets:**
ML algorithms and techniques should be used to determine a representative subset of records in a database table that consists, for instance, of only 10–15% of the original data records. Statistical measures should be determined on these representative subsets, which have the potential to reduce time and compute resources by factors. Depending on the values of the underlying dataset, a larger representative dataset may yield more accurate predictive results for the entire dataset.
- **Extrapolating statistics:** Once key statistical measures have been determined based on those representative subsets, ML methods should be applied to extrapolate (predict) corresponding statistical values for the entire dataset. This applies to all statistical values that are depicted in Figure 14-3, for example, frequency distribution, quality score, mean value, standard deviation, etc.
- **Understanding structure of data and AI assets:**
ML and DL methods are particularly well suited to discover the structure of data and AI assets, including non-structured data, pipelines, text, images, etc. Discovering structure includes determination of data

⁷ See Reference [7] for more information on applying ML to profiling.

types (e.g., XML, JSON, etc.), ML/DL model types (e.g., ONNX, Scikit-learn, PMML, XGBoost, TensorFlow, etc.), number of levels in random forest ML models, number of hidden layers in ANN models, identification of queues or ETL stages, etc.

- **Clustering of data:** A much deeper and more relevant profiling is done with clustering of data (as we have suggested previously when discussing Figure 14-4), particularly values of key columns in database tables, for example, salary income, credit card spending, purchasing volume, types or categories of goods purchased, etc. This enables data engineers and data consumers to gain additional insight that is not so obvious and easily visible by just looking at statistical distribution values. It specifically supports the transformation from pure customer data to gaining more insight regarding customer behavior.
- **Relationship discovery:** Data and AI assets need to be visualized in context and in relationship to each other, which requires ML-based discovery methods and visualization and access methods for ease of consumption. For instance, AI models need to be presented in regard to associated data pipelines and applications for scoring and inferencing.

The preceding analysis and profiling tasks must be performed autonomously and repetitively, depending on availability of new assets or updates to existing data and AI assets. Results of these analysis and profiling tasks should be pushed to the knowledge catalog autonomously as well.

Before we move on to automated tagging, annotation, and labeling of data and metadata, we would like to point out the difference between data profiling and data mining. With data profiling, we essentially limit our analysis to structure discovery, content discovery, and relationship discovery, whereas data mining is geared toward finding *golden nuggets* in data primarily via exploration, pattern discovery, etc. Although there may be an overlap between data mining and profiling, data mining is done for a particular business purpose, that is, targeted marketing campaign or customer care, whereas data profiling has a more general purpose.⁸

Automated Tagging, Annotation, and Labeling

Before diving into automated tagging, annotation, and labeling of metadata, let us look at another example⁹ depicted in Figure 14-5, where several columns of the *CREDITUSERS* table are listed on the left side. For the column ADDRESS, three business terms are automatically suggested, including a probability measure for a match: *Work Address* (79%), *Email Address* (77%), and *Postal Address* (76%).

⁸ See Reference [8] for a comparison of data profiling and data mining.

⁹ Please, refer to the example in Chapter 13, depicted in Figure 13-3.

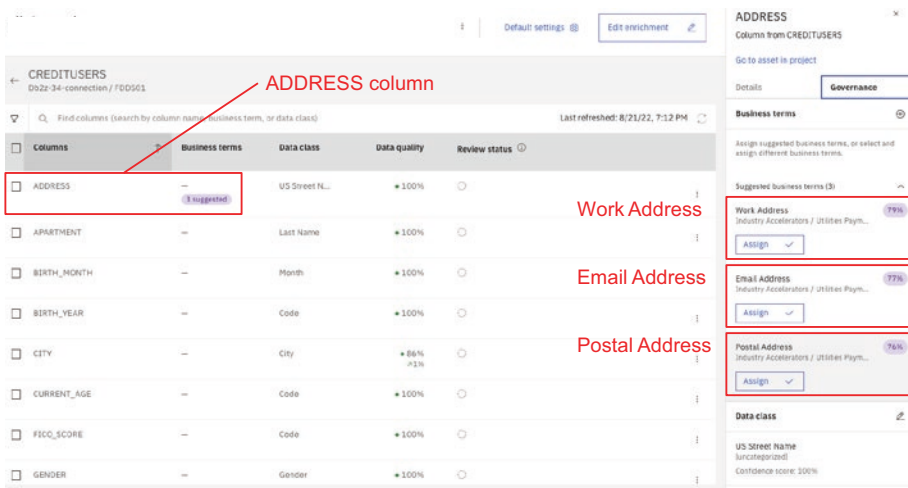


Figure 14-5. Business Term Assignment

These business terms are automatically taken from the business glossary and assigned to the column *ADDRESS*, giving the data steward a choice to select one of these terms as the most suitable one. In this example, the data steward has the option to either choose one of the proposed terms or even reject all three, depending on their own judgment.

Let us now clarify the terms *tagging*, *annotation*, and *labeling*, which are often wrongly used interchangeably. Comparing these terms, we limit the discussion to the Data Fabric and AI scope. Quite a few terms exist to further enrich data, such as labels, annotations, tags, comments, attributes, explanations, etc.

In general, *tagging* means to add additional keywords, phrases, or terms to an asset to further explain what the asset is all about and to support searches, for example, a column of a database table may be tagged as *PII*, meaning it contains sensitive information. Some readers may refer to this as *comments* or *explanations*. An *attribute* is a characteristic of an asset, for example, an AI model may have the following characteristics

assigned to it: *trained*, *validated*, or *deployed*. Like tags, *annotations* refer to additional information about assets that may comprise figures and text, for example, explanatory text as comments added to a Java program.

In the context of AI with ML and DL though, *annotation* refers primarily to text, image, or video annotation, meaning to capture an object of interest, for instance, vehicles, bicycles, and pedestrians, in videos to support learning and training of an ANN model for autonomous driving.

Likewise, *labeling* refers to adding additional information to all records of a dataset to support training of AI models, for example, adding *TRUE* or *FALSE* to each passed banking transaction to train an ML model to predict whether a future transaction is fraudulent or not. Thus, annotation and labeling in AI¹⁰ are similar concepts to create datasets for AI model training. They differ by style and method: annotations are capturing or marking objects of interest (primarily in a text, image, or video), whereas labeling adds data to individual records.

The crucial factor here is to apply AI/ML methods to add meaningful and relevant tags or annotations to data and metadata autonomously or to provide actionable recommendations for the data steward or data consumers. This facilitates a closer interlock and linkage of data and AI assets to business domains and thus supports self-service Data Mesh initiatives. Autonomous annotations and labeling in an AI context are essential tasks to address trustworthy AI as well in which, for instance, retraining and revalidation of AI models need to be performed autonomously, in case bias, drift, or worsening quality metrics have been detected for some AI models.

We would like to point out that autonomous tagging, annotation, and labeling needs to be done on data and AI assets as well as on metadata itself. Tags, annotations, and labels added to data and AI assets represent metadata, whereas tags, labels, and annotations to metadata yield active metadata, further enriching the existing set of metadata.

¹⁰ See Reference [9] for more information on labeling and Reference [10] for more information on tagging for autonomous driving.

Some examples of the latter are the autonomous assignment of a business term to a database column; the relationship of a pipeline to an AI model; determining the impact on the business by, for instance, removing a supposedly unused dataset; or simply adding a descriptive label or a quality score to a database column that is already captured in the knowledge catalog.

Let us now move on to automated data quality assessments.

Automated Data Quality Assessment

Understanding and solving data quality issues is still the most common challenge for any data owner and data consumer; it goes far beyond just profiling data where some basic statistics for each column, such as frequency distribution, number of unique values, minimum and maximum values, mean value, standard deviation, etc., are calculated as depicted in Figure 14-3. A comprehensive data quality assessment requires a deeper look at data to understand, for instance, the number of data class violations, data type violations, missing values, format violations, values out of range, etc.

The outcome of a comprehensive data quality assessment is a detailed summary of findings that is represented as a *data quality score* for each data asset as illustrated in Figure 14-6.

Assets	Source	Business terms	Data quality	Review status	Er
CREDITCARDS	Db2z connection / DWASYNC	Credit Card	99%		
CREDITTRANS	Db2z connection / DWASYNC	Transaction	98%		
CREDITUSERS	Db2z connection / DWASYNC	Customer	99%		

Figure 14-6. Data Quality Assessment

Figure 14-6 lists three tables – *CREDITCARDS*, *CREDITTRANS*, and *CREDITUSERS* – with their corresponding quality scores, which are 99%, 98%, and 99%, respectively. These quality scores are derived by calculating data quality dimensions for each individual column of a database table.

The overall quality score for a data asset is then derived from the quality scores of each column from each of the tables. Thus, a set of well-defined data quality dimensions are used to calculate a quality score for each chosen column, where the set of quality scores for each chosen column are used to calculate the overall quality score of the whole data asset.

Let us briefly discuss the data quality dimensions¹¹:

1. **Data class violations:** Is counting the violations of data classes defined in the knowledge catalog (e.g., Passport Number, New York State Driver’s License, Organization Name, VISA Card, etc.) for each column.
2. **Data type violations:** Is counting violations of data types (e.g., SMALLINT, REAL, DATE, TIMESTAMP, etc.) of each chosen column as it has been determined during the prior data profiling phase.
3. **Inconsistent capitalization:** Is looking for patterns regarding usage of upper- and lowercase in strings and is counting the number of violations taking the identified usage pattern as a base – assuming a pattern has been identified for the majority of values in a column.

¹¹ See Reference [11] for more information on data quality assessments and data quality dimensions.

4. **Values out of range:** Is looking for violations of a minimum and/or maximum constraint, which has been defined for values of a column.
5. **Rule violations:** Is looking for violations of any data or quality rule for a row in a database table that has been defined by the data steward.
6. **Suspect values:** Is looking for mismatches of values and format and properties of values, compared with the identified most likely pattern of a column.
7. **Suspect values in correlated columns:** Similar to the *suspect values* dimension, this data quality dimension detects mismatches of likely patterns or correlations of values across columns within a table.
8. **Missing values:** Is looking for non-expected NULL or empty values in columns.
9. **Inconsistent representation of missing values:** Is looking for NULL, empty, or values containing only spaces in a column.
10. **Format violations:** The data steward may have declared certain values of a column as invalid; the number of these invalid values for each column is counted by this dimension.
11. **Duplicated values:** This dimension may already be captured via the data profiling phase, where the number of duplicate values is calculated for columns that should contain unique values, that is, primary or unique keys.

Figure 14-7 shows a subset of the data quality dimensions for the table *BANK_CUSTOMERS* and the overall data quality score for this table, which is 96%. It also depicts the quality score for some of its columns, like *CUSTOMER_ID*, *NAME*, *ADDRESS*, *ZIP*, *CREDIT_RATING*, and *AGE*. As you can see, the quality score for most of the columns is either 100% or very close to 100%, whereas the *ZIP* column is only 66%, which is a key reason for the overall data quality score of the table *BANK_CUSTOMERS* to be 96%.

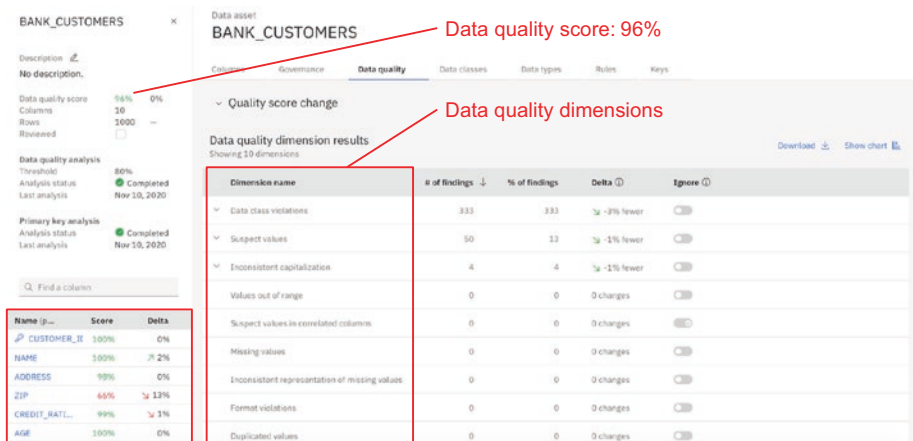


Figure 14-7. Data Quality Dimensions

This example is presumably derived from rather traditional data quality assessment methods of database tables. With the exception of *suspect values* and *suspect values in correlated columns*, which require AI techniques (i.e., pattern discovery), all other data quality dimensions are primarily calculated via conventional methods. Future methods are inevitably affected by broadening the scope to include AI assets, meaning to assess quality of AI models, ETL stages, pipelines, etc. For instance, measuring the quality of an AI model is related to what we have discussed in Chapter 5 in the section on trustworthy AI, namely, to measure quality metrics, for example, the areas under the ROC or PR curves while models are in production.

Even today's data with its unprecedented diversity in terms of text, images, and videos is calling for innovative methods to measure the quality of these data in the context of a certain usage purpose. For instance, what are the quality imperatives and measurement methods for medical texts to be declared as suitable to serve as reliable input for DL model development to predict medical treatment, promising recovery with high confidence? What are the methods to measure the quality and completeness of annotated video sequences to serve as input for training DL models used for autonomous driving?

The treatment of this topic cannot end without discussing the need for automated actions to correct quality issues. Similar to what we have seen with metadata enrichment where business terms are suggested to be assigned to a column, data quality assessments can only be seen as the beginning of a journey where corrections need to autonomously performed, using AI/ML: quality assessments in the context of a modern Data Fabric and Data Mesh need to be developed into *intelligent automated quality management*, where corrections must be implemented autonomously or suggested to the data steward.

Key Takeaways

We conclude this chapter with a few key takeaways as summarized in Table 14-1.

Table 14-1. Key Takeaways

#	Key Takeaway	High-Level Description
1	Intelligent automated metadata is key to a Data Fabric and Data Mesh.	The term <i>intelligent automated metadata</i> does portray the inclusion of AI/ML into the entire lifecycle of metadata management, including generation, enrichment, and consumption of data.
2	A knowledge catalog is based on intelligent auto-capabilities.	A knowledge catalog for a Data Fabric architecture and Data Mesh solution needs to manage automated metadata generation, automated metadata enrichment, and automated metadata extraction and exploitation of the underlying data and AI assets.
3	Data profiling is the first step to gain insight into data assets.	Data profiling is the first process of examining, analyzing, and checking the content of data attributes of data assets to get a first path of statistical insight and an initial understanding regarding data quality.
4	AI/ML can augment data profiling.	AI/ML can determine representative sample datasets, extrapolate statistics for the entire dataset, understand structure of data and AI assets, and perform clustering and relationship discovery.
5	Tagging, annotation, and labeling.	AI/ML should be applied to tagging, annotation, and labeling, which can be seen as vital methods for metadata enrichment.
6	Automated data quality assessments are based on data quality dimensions.	A number of data quality dimensions are used, such as data class violations, data type violations, inconsistent capitalization, values out of range, rule violations, suspect values, etc., to perform automated data quality assessments.

(continued)

Table 14-1. (continued)

#	Key Takeaway	High-Level Description
7	AI/ML needs to be leveraged for improving quality assessments.	Comprehensive quality assessments of data and AI assets in the context of a modern Data Fabric and Data Mesh need to be infused with AI/ML and developed into intelligent automated quality management processes.

References

- [1] IBM, IT Infrastructure, *IBM Z Performance and Capacity Analytics*, www.ibm.com/products/z-performance-and-capacity-analytics (accessed September 3, 2022).
- [2] Kounev, S., Bender, K., Brosig, F., Huber, N., Okamoro, R., *Automated Simulation-Based Capacity Planning for Enterprise Data Fabrics*, <https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.1064.6434&rep=rep1&type=pdf> (accessed September 3, 2022).
- [3] Digital Curation Centre, University of Edinburgh, *List of Metadata Standards*, www.dcc.ac.uk/guidance/standards/metadata/list (accessed September 3, 2022).
- [4] The Linux Foundation Projects, *Egeria*, <https://egeria-project.org/> (accessed September 3, 2022).
- [5] Alation, Zumpano, A., *How to Build a Successful Metadata Management Framework*, 2022, www.alation.com/blog/metadata-management-framework/ (accessed September 4, 2022).

- [6] IBM, IBM Cloud Pak for Data, *Profiling data assets*, <https://dataplatform.cloud.ibm.com/docs/content/ws/j/catalog/profile.html> (accessed September 4, 2022).
- [7] Van Otterlo, *A Machine Learning View on Profiling*, 2013, <https://martijnvanotterlo.nl/cpdp11-draftversion-ProjectedWorlds-MartijnVanOtterlo-2011.pdf> (accessed September 4, 2022).
- [8] GeeksforGeeks, *Difference between Data Profiling and Data Mining*, 2021, www.geeksforgeeks.org/difference-between-data-profiling-and-data-mining/ (accessed September 4, 2022).
- [9] Bekos, M., Niedermann, B., Nöllenburg, M., *External Labeling: Fundamental Concepts and Algorithmic Techniques (Synthesis Lectures on Visualization)*, Springer, 2021, ISBN-13: 978-3031014819.
- [10] Label Your Data, Kniazieva, Y., *The Role of Data in Constructing Autonomous Vehicles*, 2022, <https://labeleyourdata.com/articles/data-annotation-for-autonomous-driving> (accessed, September 5, 2022).
- [11] Towards Data Science, SAILLET, Y., *Data quality dimensions in IBM Watson Knowledge Catalog*, <https://towardsdatascience.com/data-quality-dimensions-in-ibm-watson-knowledge-catalog-79cd0aaf0af2> (accessed September 6, 2022).

CHAPTER 15

Data Governance in the Context of Data Fabric and Data Mesh

Companies generate and manage large amounts of sensitive information about their employees, customers, and business during their operational and analytical activities. This information gives companies a competitive advantage and at the same time brings great risks. The exposure of sensitive information can lead to serious consequences, such as lawsuits. Therefore, companies need to implement a purposeful and well-planned data governance platform.

As we have seen throughout this book, a Data Fabric architecture and Data Mesh solution are emerging data and AI concepts enabling flexible, reusable, proactive, and enhanced platforms to operationalize data and AI. As you recall from Chapter 2, data governance and privacy is one of the four entry points for a Data Fabric and Data Mesh journey. While accelerating the enterprise's data management efforts, a Data Fabric architecture also provides a Data Mesh solution to optimize enterprise data and AI governance implemented in an organizational federated fashion.

This chapter will first discuss why data management and data and AI governance are critical to a data-driven strategy and then further

explore how capabilities of both concepts can help build a solid data and AI governance foundation for organizations and ultimately the entire enterprise. Using the term *data governance* is inherently referring to data and AI assets.

Introduction

Data is constantly generated while we live and work. This data can have incredible value, which makes it a target for theft and misuse. Companies have obligations to respect both the privacy of personal information and the confidentiality of business information. The protection needs to be throughout the entire data management and AI lifecycle. Data management and data and AI governance are like two strands of DNA that are inextricably interdependent. First and foremost, let us distinguish two important concepts, namely, data management and data and AI governance.

According to DAMA-DMBOK2,¹ data management is the process of developing, implementing, and monitoring plans, systems, procedures, and practices to deliver, control, protect, and enhance the value of data and assets throughout their lifecycle, while data and AI governance is defined as the exercise of authority and control during the management of data and AI assets. Subsequently, data management focuses more on execution issues; data and AI governance, on the other hand, has its focus more on the oversight perspective, ensuring that specific data and AI-related management tasks are under control and are carried out in an orderly, effective, and sustainable manner.

Often, when considering data management and data and AI governance, we tend to overemphasize the importance of technology and tools. For example, we may ask which DBMS should be chosen,

¹ See Reference [1] for more information on DAMA-DMBOK2.

whether to use ETL or ELT to integrate data, and what tools to use for data quality profiling and monitoring. Of course, these are vital topics to discuss; however, we should also recognize the importance of people and processes in addition to technology and tools.

Companies also need to answer whether they have established a culture that recognizes the value of data-as-a-product. Furthermore, have processes and business rules for established data and AI governance goals been defined? Are standards and policies established to ensure effective compliance with regulations and laws?

As you can see in Figure 15-1, there are a number of questions to consider in the context of data and AI governance.

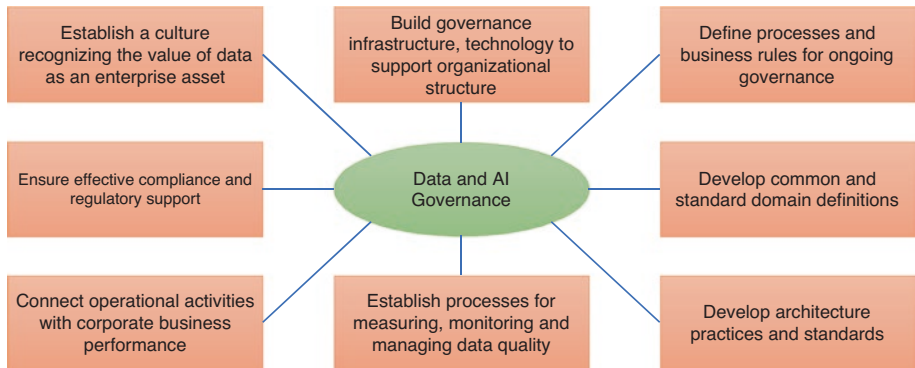


Figure 15-1. *Data Governance Aspects*

Importance of Data and AI Governance

Why should enterprises take data and AI governance seriously? Let's see some survey results and predictions from Forrester and Gartner. Nearly one-third of analysts spend more than 40% of their time vetting and validating their analytical data before it can be used for strategic decision-making.² By 2024, 30% of organizations will invest in data and analytics

² See Reference [2] for more information on Forrester's report.

governance platforms, thus increasing the business impact of trusted insights and new efficiencies. By 2025, 80% of organizations seeking to scale their digital business will fail because they do not pursue a modern approach to data and analytics governance. By 2024, organizations that utilize active metadata to enrich and deliver a dynamic Data Fabric and Data Mesh will reduce time to integrate data delivery by 50% and improve the productivity of data teams by 20%.³

Therefore, the most intuitive reason to establish data and AI governance policies is to accelerate decision-making while ensuring that the insights derived from data are credible and of high quality. Another reason to establish data and AI governance is that it helps break down data silos, which are not only caused by data being stored on different physical storage devices or databases but may also be caused by inconsistent data standards and definitions across different organizations or different requirements for data sharing.

A unified knowledge catalog may help solve this problem technically; however, breaking down business silos between various user roles and responsibilities and organizations still requires trust and a high-level governance based on guidelines and policies.

Regulatory compliance is another key driver for conducting data and AI governance. Although 70% of organizations plan to spend at least \$1 million per year on compliance, this may not completely prevent data security and compliance issues from occurring. Good data and AI governance best practices and systems need to be put in place to reduce the likelihood of risks and mitigate those risks in accordance with laws and regulations.

³See Reference [3] for more information on Gartner's report.

Key Aspects of Data and AI Governance

As data governance must ensure that data management and AI-related tasks are well under control, you may want to know what specific management tasks are involved. Based on the DAMA-DMBOK2 knowledge area wheel, as depicted in Figure 15-2, there are 11 key knowledge areas. Although the scope of the DAMA-DMBOK2 knowledge area wheel is on data, we suggest to broaden the scope to data and AI by taking well into account emerging regulations and laws that are specifically geared toward AI.⁴ Addressing these data and AI governance knowledge areas requires the implementation of a modern Data Fabric architecture.



Figure 15-2. DAMA-DMBOK2 Knowledge Area Wheel

⁴ Please, review the section on trustworthy AI in Chapter 5.

The following list explores a few areas that are most relevant to both concepts:

- **Data and AI governance:** First and foremost, data and AI governance sits at the center of data management, responsible for planning, oversight, and control over the management of data and AI and the use of data and data-related resources.
- **Data architecture:** As the name implies, data architecture⁵ defines the blueprint to meet the data needs of the enterprise; it is an integral part of the enterprise architecture. It guides the integration between data sources and aligns data investments with the overall IT and business strategy. The design of the data architecture needs to adhere to regulations as well as corporate policies.
- **Data storage and operations:** Storage is the foundation for data residency. The storage infrastructure must support the data architecture and data model. If an enterprise chooses a hybrid cloud as their architecture, they will use cloud storage to supplement their on-premises storage. Enterprises can also house their data or move their data to the storage of different data access speed, based on requirements regarding the frequency of data usage. Typical operations include replication, archiving, and disaster recovery to prevent data loss.
- **Data and AI security and privacy:** Data privacy is concerned about whether an organization protects sensitive data and provides appropriate authentication, authorization, access, and auditing to ensure

⁵See Chapter 11 for more information on data architecture.

compliance. The main consideration is to reduce the risk of data leakage and unauthorized use. Data and AI governance establishes a hierarchy of sensitive data and AI asset classification, as well as a set of data and AI protection and access rules (Create, Read, Update, and Delete), resulting in a permission matrix for enterprises to protect data and AI assets and meet audit requirements. Differing from data privacy, data security protects data from compromise by external hackers and malicious insiders. It's a vital aspect of enterprise security. Due to the specialized nature of this field, it will not be elaborated in this book. Please refer to books on data security if you are interested.

- **Data and AI integration and interoperability:** Defines how data is acquired, extracted, transformed, moved, replicated, and virtualized for access. The integration needs to follow the business rules, government regulations, and industry standards defined by the governance platform. For example, PII protection laws do not allow personal information to leave the country. Therefore, integration will not be allowed for this type of information.
- **Data quality:** Considers whether an organization can consistently define and measure data quality and mitigate quality issues. Each enterprise may define different dimensions to measure data quality. Some may focus more on the timeliness of the data, and others might emphasize the completeness and consistency of the data. No matter what dimensions

are considered, an end-to-end workflow to periodically detect data quality issues and provide mitigation solutions is always required.

Metadata⁶ and master data⁷ are part of data governance.⁸ Reference data should also be mentioned in this context, which is a set of data for complex hierarchies or classifications. For example, the country code *US* represents the United States, and *CN* represents China. Databases store country codes to ensure better quality through standardized definition.

To use an analogy, let us assume that the data is the community's public facilities, such as parking lots, mailboxes, swimming pools, gyms, and other facilities. Each tenant has their own parking space and mailbox and needs the appropriate keys to unlock them. At the same time, all tenants have access to the pool, the gym, as well as the gate into the community parks. The property management is defining these rules and documenting these rules for all tenants. In this example, the data architecture, model, and storage refer to how many of these facilities need to be added to the community and where they should be located for the convenience of tenants. Metadata, reference data, and documents refer to the maps of these facilities and the instructions of using the facilities, that is, referencing *opening hours for the pool* and *reservation required* or *first come first serve* for tennis courts. Data security means that the tenants' assets are well protected and cannot be accessed by others. The property management team is the governance team that ensures all the tenants are familiar with and follow the rules. They should take action for any violations of the rules, that is, if a car stays overnight in a community premise without a permit, it will be towed away, and violators will have to pay a fine.

⁶ See Reference [4] for more information about metadata.

⁷ See Reference [5] for more information about master data.

⁸ Chapter 8 and Chapter 13 explain these concepts in more detail.

Establishing a Data Governance Foundation with a Data Fabric Architecture

Enterprises are increasingly recognizing the importance of data and AI governance, and many of them start to build a data and AI governance platform. However, at present, data governance activities in enterprises are still at department level, and the lack of top-level design of enterprise data and AI governance and the coordination of data and AI resources can lead to the delay or even failure of data and AI governance projects.

The Data Fabric architecture can help enterprises address the challenges of data and AI governance effectively, including the orchestration and exchange of metadata across organizational implementations. First, Data Fabric pulls data from disparate data sources and orchestrates metadata exchange across organizational systems, thus providing a holistic view of data and AI at the enterprise level, which lays a solid technology foundation for a consistent and unified enterprise-level data and AI governance. Likewise, a Data Fabric architecture serves as a foundation for a Data Mesh solution, which is supporting organizational or departmental data and AI governance initiatives.

Second, the advanced automation and AI technologies employed by a Data Fabric architecture can greatly simplify the implementation of data and AI governance at the enterprise or organizational level, enabling organizational federated Data Mesh initiatives, where orchestration and exchange of metadata across organizations need to be implemented as well.

The following are some key aspects derived from advanced automation capabilities of a Data Fabric architecture:

1. Automating the import of industry regulations and company policies, converting them into global or local data protection rules by business rule type, thus creating a system of governance regulations

2. Automating rules and policy enforcement to data assets, such as data masking or obfuscation, which ensures that data assets are accessible by certain users
3. Periodically scanning and analyzing data quality, reporting the data assets that do not meet quality requirements, and triggering a quality improvement process

Establishing Automated Regulation with a Data Fabric Architecture

Since the introduction of the GDPR⁹ in the EU in 2018, other countries around the world have enacted or proposed post-modern privacy and data protection legislation. Brazil, for instance, introduced the Lei Geral de Proteção de Dados (or LGPD), while the US state of California introduced the California Consumer Privacy Act (CCPA). More than ten countries in Asia-Pacific including China, Korea, and New Zealand have proposed, adopted, or enforced new privacy laws. By 2023, more than 80% of companies worldwide will face at least one privacy-focused data protection regulation.

This resurgence of data privacy regulations puts companies in a challenging position to interpret and implement data privacy strategies to comply with these complex regulatory requirements in order to protect organizational and individual claims to their data privacy.

Companies are recognizing that the implementation of regulations is a complex endeavor that requires not just strong subject matter expert support but also a corresponding technical infrastructure. How to digitize

⁹ See Reference [6] for more details on GDPR in the EU.

regulations and get business units to comply with these regulations is the top concern for companies. The good news is that advancement of AI technology, especially NLP technology, can largely solve this problem. A Data Fabric architecture should employ NLP¹⁰ to automate regulatory compliance.

Please see Figure 15-3 for an example, which is based on IBM's Data Fabric approach, where the regulation accelerator makes GDPR policies and regulations ready for use in minutes.

The screenshot shows the IBM Data Fabric interface for GDPR. It features a search bar, a list of subcategories, and a table of governance artifacts. The table has columns for Name, Description, Primary category, and Type. The 'Article and Rule' artifact is highlighted with a red box.

Name	Description	Primary category	Type
Article and Rule	Specific Article and Rule of GDPR requirements	GDPR	Policy
Cross-border data	GDPR contains provisions that address the transfer of personal data from EU member stat...	GDPR	Policy
Data controller and processor	If you access personal data, you do so as either a controller or a processor, and there are differen...	GDPR	Policy
GDPR Policy	GDPR will affect (1) all organizations established in the EU, and (2) all organizations involved in...	GDPR	Policy
Use of third-parties to process data	Prior to engaging with any third party, evaluates their GDPR position, and executes an agreem...	GDPR	Policy

Figure 15-3. GDPR

Figure 15-4 is another illustration of GDPR-related governance artifacts, such as *consent*, *personal data inventory*, *data protection by design*, etc., which can be displayed and selected (left side of Figure 15-4). One of the governance artifacts depicted in Figure 15-4 is *processing children's data* (dashed line in Figure 15-4).

¹⁰ See more details about NLP in Chapter 6.

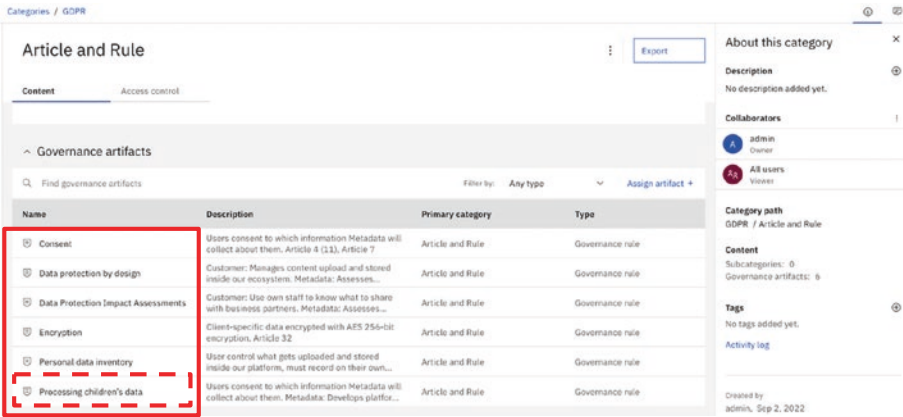


Figure 15-4. Governance Artifacts

This *children's data* can further be displayed (e.g., with its effective dates) and adjusted (e.g., defining an end date), as seen in Figure 15-5.

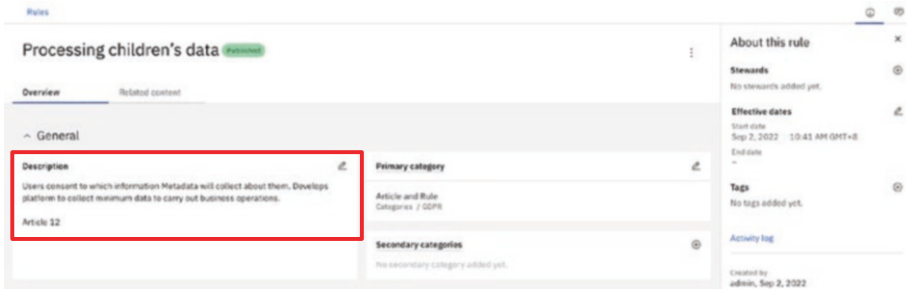


Figure 15-5. Governance Rules

Automatic Enforcement of Data Regulations in Data Fabric

The privacy laws and regulations create obligations for organizations processing their personal and confidential data, which impose severe penalties for noncompliance. Sometimes personal data and confidential

data are seen as identical. However, personal data is referred to as PII including name, date of birth, mobile number, etc. Confidential data is the information regarding the organization, for example, trading price, intellectual property, and customer information.

Establishing and enforcing regulations to corresponding data assets are equally important. To fulfill these responsibilities, the data governance platform needs to classify and assign data to corresponding categories. Whether data belong to personal information or even sensitive personal information or confidential information determines the level of protection. For example, processing PII is explicitly prohibited under the GDPR unless it has been consented to or is otherwise lawful.

How does technology enable the enforcement of data protection rules? In a Data Fabric platform, the data protections rules can be defined by the characteristics of corresponding data and AI assets. As depicted in Figure 15-6, *address* is PII information and needs to be protected via the *protect address rule*.

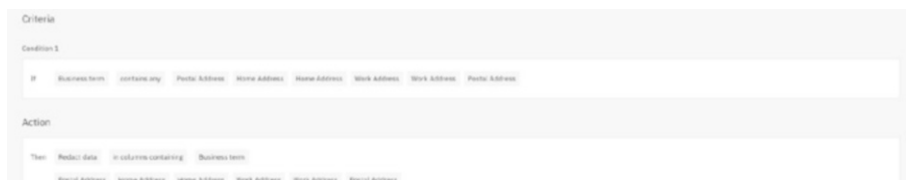


Figure 15-6. *Data Protection Rules – Protect Address Rule*

If a business term containing any *address* tags is recognized at the discovery stage, the data will be masked, and the data protection rule will be applied. When business users access these data assets, the data on the address column are redacted, as depicted in Figure 15-7.

Schema: 19 Columns | 1000 Rows | 3 Columns masked

The preview includes only a limited set of columns and rows.

USER_ID String	PERSON String	CURRENT_... String	RETIREMENT_... String	BIRTH_... String	BIRTH_MC_... String	GENDER String	ADDRESS String	APARTM... String	CITY String	STATE String	ZIPCODE String	LATITUDE String	LONGITUDE String	PER_CAFF String
1	0000000000	53	66	1956	11	Female	0000000000	0000000000	La Verne	CA	91750	34.15	-117.76	29279
2	0000000000	53	68	1956	12	Female	0000000000	0000000000	Little Neck	NY	11363	40.76	-73.74	37891
3	0000000000	81	67	1938	11	Female	0000000000	0000000000	West Covina	CA	91792	34.32	-117.87	22605
4	0000000000	63	63	1957	1	Female	0000000000	0000000000	New York	NV	10569	40.71	-73.99	163145
5	0000000000	43	70	1976	8	Male	0000000000	0000000000	San Francisco	CA	94117	37.76	-122.44	53797
6	0000000000	42	70	1977	10	Male	0000000000	0000000000	Evansport	IA	52803	41.35	-90.60	20599
7	0000000000	26	67	1933	12	Female	0000000000	0000000000	Louisville	KY	40299	38.22	-85.74	23250
8	0000000000	26	67	1933	12	Male	0000000000	0000000000	Portland	OR	97214	45.51	-122.64	26790
9	0000000000	81	66	1938	7	Female	0000000000	0000000000	Tellurid	PA	13969	40.32	-75.32	26273
10	0000000000	34	60	1956	1	Female	0000000000	0000000000	Aldouise	LA	70510	29.97	-92.52	18730

Figure 15-7. Enforce Data Masking Rules – CREDITUSERS

Automate Quality Analysis with Data Fabric

The establishment of regulations and the enforcement of regulatory rules ensure that only authorized users have access to the data. There is another aspect that should also be taken into consideration, and that is ensuring access to *trusted* data, which relates to complete, high-quality, and consistent data. If the quality of data does not meet defined standards, the use of this data may even generate potentially wrong results with negative business impact, loss of customers, degrading reputation, etc.

Therefore, analyzing and evaluating data quality, as depicted in Figure 15-8, is a prerequisite for subsequent usage. How does a company define the metrics of data quality?

Columns	Business terms	Data class	Data quality	Review status
<input type="checkbox"/> IS_HOLD	—	Boolean	+100%	<input type="radio"/>
<input type="checkbox"/> MCC	—	Code	+85%	<input type="radio"/>
<input type="checkbox"/> MERCHANT_CITY	—	City	+94%	<input type="radio"/>
<input type="checkbox"/> MERCHANT_NAME	—	Person Name	+100%	<input type="radio"/>
<input type="checkbox"/> MERCHANT_STATE	—	US State Code	+98%	<input type="radio"/>
<input type="checkbox"/> TIME	—	Date	+100%	<input type="radio"/>
<input type="checkbox"/> TRANS_ID	—	Identifier	+83%	<input type="radio"/>

Figure 15-8. Quality Analysis – CREDITTRANS

Usually there are six dimensions¹¹ of data quality:

- **Accuracy:** Data must reflect the true business content. Accuracy could be a problem due to mistakes in entry or errors in data conversion. For example, let's say the cost should be \$1000. Due to a staff error, it is \$100 in the system, an error that, if not caught, would result in wrong decisions.
- **Consistency:** The type and meaning of the data elements must be consistent. For example, the organization code in a system may be completely different from the organization code in another system, which creates significant difficulties in performing analytical tasks.
- **Timeliness:** The data should be updated in a timely manner based on the user's currency requirements, especially if replicated multiple times. Some data transformation processes may take several days, which is unacceptable for some real-time applications.

¹¹ See Reference [7] for more information about quality dimension.

- **Uniqueness:** There should be no duplicate data values for the same entity. Duplicate data¹² needs to be removed or merged, which is typically addressed by MDM systems.
- **Completeness:** Completeness of data should be measured. For epidemiological surveys, for instance, the disease origin is required; if missing, the degree of data completeness is relatively low, which may prevent data scientists from including this data in their project.
- **Validity:** The value of data must meet requirements of the data or business definition, such as the format of certain telephone or mailbox numbers. Otherwise, it will give a low validity score.

Figure 15-9 is an example of quality dimensions and quality score, where the column *MERCHANT_CITY* is recognized as data class *City*. Any data in this data class needs to be in the reference dataset *Cities.cls*, as illustrated in Figure 15-10. However, 6% of the data is not there.¹³ Either users accept this and make an informed decision to use it, or users need to go to the reference dataset and compare the disparity of this dataset with the reference dataset.

¹² See more information about duplicate record removal in Chapter 8.

¹³ Please, review the section “Automated Data Quality Assessment” in the previous chapter.

Findings: 79 | Quality score: 94%

Dimension	Findings	Percentage of records
Data class violations	60	6%
Inconsistent capitalization	19	2%
Suspect values	0	0%
Inconsistent representation o...	0	0%
Unexpected missing values	0	0%
Format violations	0	0%
Unexpected duplicated values	0	0%
Values out of range	0	0%
Data type violations	0	0%

Figure 15-9. Data Quality Dimensions for MERCHANT_CITY

The screenshot shows a configuration page for a data class named 'City'. The page is divided into several sections:

- General:**
 - Description:** A place name such as a city or town.
 - Examples:** Los Angeles
 - Primary category:** [uncategorized]
 - Secondary categories:** for secondary category selected yet.
- Data matching:**
 - Matching method:** Match based on valid values.
 - Details of matching method:** Text: Max diff: 1; min: 0; reference: City.cls; Text: Matching: Exact spacing; Threshold: 5%
 - Other matching criteria:** Create data type: Text.
 - Matching priority:** Data class priority: 7.

The 'Data matching' section is currently disabled, as indicated by a green 'Disabled' toggle.

Figure 15-10. Data Class

In addition to these standard dimensions, companies can customize the dimensions or adjust the weights of these standard dimensions. The goal is to set data quality scores that accurately reflect the extent to which

that data can be used for business analytics. Also, companies need to set up a data quality monitoring framework that must trigger processes to fix quality issues should they occur.

Key Takeaways

We conclude this chapter with a few key takeaways as summarized in Table 15-1.

Table 15-1. *Key Takeaways*

#	Key Takeaway	High-Level Description
1	Data management and data and AI governance are like two strands of DNA.	Data management is the process of developing, implementing, and monitoring systems, procedures, and practices to deliver and enhance the value of data and assets throughout their lifecycle, while data and AI governance is defined as the exercise of authority and control during the management of data and assets.
2	Companies often lack a top-level design of enterprise data governance that leads to the delay or failure of data and AI governance projects.	The Data Fabric architecture can help enterprises address data and AI governance challenges effectively by two key levers, (a) pulling data from disparate data sources to create a holistic view and (b) using AI technology to simplify the implementation of data and AI governance.
3	By 2023, more than 80% of companies worldwide will be faced by at least one privacy-focused data protection regulation.	Digitizing the regulation and getting the business units to comply with regulations and laws is a matter of urgency. Use NLP technology with a Data Fabric architecture to accelerate the establishment of regulations.

(continued)

Table 15-1. (continued)

#	Key Takeaway	High-Level Description
4	Policy and rule enforcement contains two key elements.	First, the platform needs to recognize or classify data to the correct categories, that is, PII or confidential data; second, the corresponding protection rules need to be applied to the data according to the outcome of the first step when data gets accessed.
5	Data quality ensures that only authorized users have access to the data.	Data quality is measured by six dimensions: completeness, accuracy, timeliness, validity, uniqueness, and consistency. Businesses need to constantly monitor the data quality. Once the quality is below a defined bar, corrective actions need to be taken.

References

- [1] Dama International, *DAMA International DMBOK Version 2*, www.dama.org/cpages/dmbok-2-image-download (accessed September 27, 2022).
- [2] Forrester, Goetz, M., *Data Performance Management Is Essential To Prove Data's ROI*, 2018, www.forrester.com/report/Build-Trusted-Data-With-Data-Quality/RES83344 (accessed September 27, 2022).
- [3] Gartner, Duncan, A., D., *Over 100 Data and Analytics Predictions Through 2026*, www.gartner.com/en/webinars/4011950/100-data-analytics-predictions-through-2026 (accessed September 27, 2022).

- [4] Opendatasoft, *What's metadata and why is it as important as the data itself?*, 2016, www.opendatasoft.com/en/blog/what-is-metadata-and-why-is-it-important-data (accessed September 27, 2022).
- [5] Profisee, *MASTER DATA MANAGEMENT - WHAT, WHY, HOW & WHO*, <https://profisee.com/master-data-management-what-why-how-who/> (accessed September 27, 2022).
- [6] GDPR.EU, *Complete guide to GDPR compliance*, <https://gdpr.eu/> (accessed September 27, 2022).
- [7] Smartbridge, *Data Done Right: 6 Dimensions of Data Quality*, <https://smartbridge.com/data-done-right-6-dimensions-of-data-quality/> (accessed September 27, 2022).

PART IV

Current Offerings and Future Aspects

CHAPTER 16

Sample Vendor Offerings

As we have mentioned in Chapter 2, Gartner named Data Fabric as one of the top ten technology trends in data and analytics for 2019 and 2021 and tipped it as one of the top ten emerging technology trends for 2022. Meanwhile, Forrester said that of the 25,000 reports the company published last year, reports on Data Fabric ranked in the top ten downloads for 2020.

As emerging technology trends, Data Fabric and Data Mesh have been in the spotlight since their inception. IBM, Amazon, and Microsoft, the world's largest information technology companies; Denodo, the leader in data virtualization; Informatica, the established leader in data integration; and many other global top vendors like Snowflake, NetApp, and Talend have responded to Data Fabric and Data Mesh requirements and provided enterprise-ready solutions.

Let us look at some sample offerings from the major vendors, as well as their strengths and limitations.

Introduction

The goal of Data Fabric¹ is to provide a flexible, seamless, and automated approach to data access, enabling self-service consumption of data at any time, and to keep data and AI trustworthy through proactive, intelligent, and sustainable data and AI governance. How does Data Fabric differentiate from other architectures as the best solution to deal with the diversity, complexity, and heterogeneity of data? It's primarily due to three key features of the architecture design²:

- **Connecting data, not centralizing it:** One of the key principles of a Data Fabric architecture is the flexibility of data integration approaches, that is, the solution automatically chooses the best integration strategy, by either virtualizing, transforming, replicating, or providing direct access for users based on workload requirements and the policies of the enterprise, eliminating the need for users to manually build data pipelines and selecting compute storage solutions. What kind of data sources and data types are supported and in what way they are connected are all important decision factors in examining Data Fabric implementations.
- **Self-service, not expert service:** A Data Fabric architecture and especially a Data Mesh solution are democratizing data and AI, allowing business users to easily discover and consume data and AI assets and

¹ See Reference [1] on Gartner's view on the role of Data Fabric modernizing data management and integration.

² Please, review Chapter 2, where we introduced Data Fabric and Data Mesh concepts.

enabling agile delivery of data products. In the existing centralized data provisioning model, data engineering teams have become the biggest bottleneck affecting the efficiency of data-driven decisions. Self-service data and AI consumption increases productivity of analysts and business users to meet exuberant data-driven requests. Self-service could be done through searching from the enterprise knowledge catalog to obtain connectivity information for data and AI assets, using it in conjunction with other tools, or directly accessing data assets using SQL, REST, etc.

- **Intelligent governance, not manual operation:** Traditional data governance initiatives are driven by top-down governance mandates and often start after problems have occurred, which is an unsustainable and inappropriate approach to cope with the rapidly expanding data volume and domain complexity. Data Fabric and Data Mesh approaches, on the other hand, suggest that data and AI governance should be considered from the beginning and be infused with intelligence, building governance capabilities through an augmented knowledge base and integrating them into every stage of the data and AI lifecycle.

In summary, both concepts emphasize distributed data management with its core idea to deliver trusted data from all relevant data sources to all relevant data consumers in a flexible way by optimizing the discovery and access of heterogeneous data allowing data consumers to achieve agile data product delivery in self-service fashion. At the same time, infusing AI in all aspects enables semantic exploration, analysis, and usage recommendation of data and AI to make both approaches as automated as possible.

In addition to these core capabilities that are connecting data, self-service, and intelligent data and AI governance, the *deployment option* and the ability to *integrate with external services* are also criteria for selecting vendor offerings.

IBM Cloud Pak for Data

IBM is recognized as a leader of enterprise Data Fabric³ by the Forrester Wave Q2 2022. IBM Cloud Pak for Data is IBM's implementation for a Data Fabric architecture, implementing Data Mesh solutions. It simplifies the whole information supply chain from collecting and organizing to analyzing and infusing data and AI by dynamically and intelligently orchestrating governed data and AI across a distributed landscape to provide a common data foundation for data consumers. Let us take a closer look at how IBM Cloud Pak for Data delivers on the Data Fabric and Data Mesh promise.

With regard to connecting data, IBM Cloud Pak for Data provides a wealth of integration options. IBM Watson Query, a service available on IBM Cloud Pak for Data, uses a single distributed query engine across clouds, databases, data lakes, data warehouses, and streaming data without copying or moving data. In addition, IBM Db2 for z/OS Data Gate allows users to access current Db2 for z/OS data without accessing and consuming Db2 for z/OS resources.

Furthermore, IBM DataStage is an industry-leading ETL tool that helps users design and transform data. The choice of integration technology depends on policy, latency, and performance requirements. For example, data location regulations do not allow data generated in a particular country or region to be transferred abroad. Therefore, the available options include data virtualization or replication to data stores residing in the same

³See Reference [2] for more details on IBM Cloud Pak for Data.

country or region. IBM Cloud Pak for Data also supports extensive data sources⁴ such as AWS S3, Cloud Object Storage, Db2, Snowflake, generic JDBC, and many more.

The data sources supported by each service may vary slightly, and new data sources are added with each new release, so please refer to IBM's website for the latest list of support.

IBM Cloud Pak for Data provides the capabilities of self-service and intelligent governance through IBM Watson Knowledge Catalog, which is the knowledge core of Cloud Pak for Data. It has provided its full breadth of self-service discovery of data, data cataloging, data profiling, data quality management, and semantic search capabilities. Data stewards use IBM Watson Knowledge Catalog to curate metadata, define data policies for privacy, capture data lineage, and perform other tasks related to security and compliance. Once data assets are published in the knowledge catalog, business analysts and data scientists with corresponding authorization can find and consume these assets in self-service fashion.

Please see Figure 16-1 for details. Moreover, their activities will also be tracked per data asset in preparation for future audits.

The screenshot shows a search interface with a search bar containing 'account'. Below the search bar are filter options: 'Filter by: Any asset type', 'Any source', and 'Any tag', along with a 'Clear all' button and a 'Hide featured assets' link. The results section shows 'Showing 4 of 4 items' and a table with the following data:

<input type="checkbox"/>	Name ↑	Owner	Tags	Business terms	Asset type	Date added	
<input type="checkbox"/>	ACCOUNT	user02	source		Data	Oct 05, 2022	
<input type="checkbox"/>	ACCOUNT	user02	target		Data	Oct 05, 2022	
<input type="checkbox"/>	TRANSACTION	user02	source		Data	Oct 05, 2022	
<input type="checkbox"/>	TRANSACTION	user02	target		Data	Oct 05, 2022	

Figure 16-1. *Self-Service Capabilities with Cloud Pak for Data*

⁴ See Reference [3] for more details on data sources supported by each service on IBM Cloud Pak for Data.

Powered by the superior capabilities of the IBM Watson Knowledge Catalog, IBM Cloud Pak for Data automatically applies industry-specific regulations and rules to data assets to secure data access across the entire enterprise, as depicted in Figure 16-2.

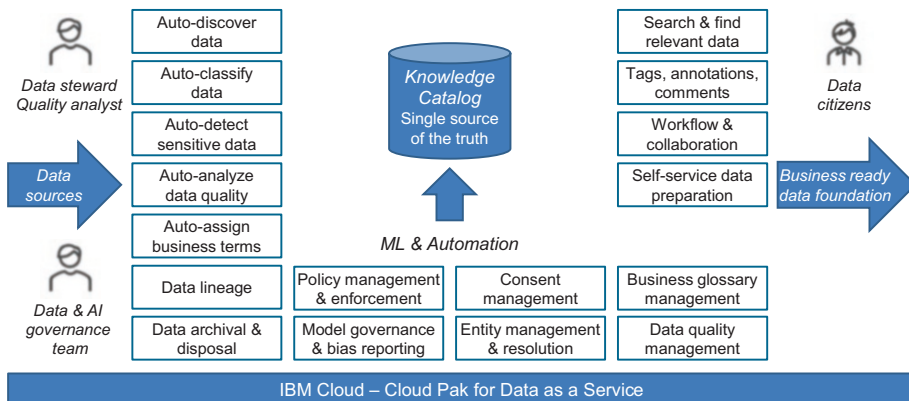


Figure 16-2. *Watson Knowledge Catalog*

It instills intelligence in all aspects of data and AI governance. First, it automatically classifies, profiles, and analyzes the quality of data assets and assigns or recommends business terms and data categories with built-in semantic models. Second, when business analysts and data scientists access data assets, protection rules and policies are autonomously enforced to ensure the compliance with data privacy regulations and laws.

IBM Cloud Pak for Data has an on-premises software version that is built on the Red Hat OpenShift container platform and a fully managed version built on IBM Cloud. It offers a wide selection of IBM and third-party services spanning the entire data lifecycle, for example, EDB Postgres Enterprise with IBM and MongoDB Enterprise Advanced in data management and Palantir for infusing AI into decision-making.

Both the on-premises and public cloud versions are suitable for large enterprises to implement either an enterprise-wide Data Fabric architecture or organizational Data Mesh solutions that may be federated

across organizations. It achieves full return on investment as more business units onboard the platform enabling the value of data and AI to be further realized.

Amazon Web Services

Amazon Web Services (AWS) offers a wide variety of data services, which are like Lego building blocks that can be assembled to implement a Data Fabric architecture. Of all the services, AWS Glue is the heart of data integration, including data discovery, data catalog, data enrichment and ETL, etc., as depicted in Figure 16-3.



Figure 16-3. AWS Glue

Connecting data is being implemented by AWS Glue crawlers. They scan data from different data sources; retrieve metadata for structured data and unstructured data, such as instance schema, location, etc.; and store this information in a centralized repository called AWS Glue Data

Catalog. The supported sources⁵ include data lakes in S3, data warehouses in Amazon Redshift, and other databases that are part of the [Amazon Relational Database Service](#). Crawlers not only connect to data but run classifiers to infer the schema, format, and data types of your data. Once the data is cataloged, it is immediately available for search and query. AWS doesn't have a virtualization solution, but Glue Elastic Views (in preview) uses SQL to provide a materialized view across many databases and data stores. It supports Amazon DynamoDB, Amazon Redshift, Amazon S3, and Amazon OpenSearch Service. In addition, AWS Data Migration Service allows customers to handle data capture for ongoing replication or changes.

AWS provides not just self-service by searching in AWS Glue Data Catalog, but direct SQL service via AWS Athena,⁶ which is an interactive query interface for data analysts to run analytics queries on various data sources. The query engine behind is Presto,⁷ which is an open source distributed query engine optimized for low-latency analysis over big data. A special feature worth mentioning is that Athena allows running federated queries to access multiple data types in multiple data sources without moving data. The supported data types by Athena include CSV, JSON, Apache Parquet, etc. Moreover, Athena and Glue have seamless integration of all data sources that Glue crawlers support and other JDBC-compatible databases.

AWS Lake Formation is yet another service that can be used to compose a Data Fabric. It implements a very important aspect – governance. With AWS Lake Formation, users can define fine-grained permissions at database, table, and column levels for a data lake. Lake Formation centrally manages security policies and enforces them across analysis services, eliminating the need for individual configuration

⁵ See Reference [4] for more details on supported data sources.

⁶ See Reference [5] for more details on Athena.

⁷ See Reference [6] for more details on Presto.

of access controls for each service. It automatically filters data and displays only the data allowed by the defined policy to authorized users, without replicating it. Lake Formation has built-in ML models for entity resolutions⁸ to link records from disparate data sources or remove duplicates from the same data source, increasing overall data quality.

Many AWS services are serverless services that can be easily scaled out. They are also optimized for S3 but not integrated with other third-party databases or S3. This section only covers a few selected AWS core services in the Data Fabric context. AWS offers flexibility but also requires users to have advanced technical skills on AWS. A steep learning curve is to be expected. Finally, AWS does not have an on-premises version. All services are only available on AWS. If you already have built your application with AWS and data in S3, AWS is a great choice for building a Data Fabric.

Microsoft Azure

Microsoft provides a unified experience and seamless integration across multiple services on Microsoft Azure, where customers can easily purchase new services as needed and get an integrated and consistent user experience. Microsoft has also developed several services for building a Data Fabric. Let us have a deeper look at these services.

For connecting data, Microsoft Purview is a unified data governance service that helps manage and govern users' on-premises, multicloud, and SaaS data. Most Azure data sources⁹ are supported through Microsoft Purview Data Map, including databases such as Amazon RDS, Db2, Oracle, MongoDB, Amazon S3, etc. In addition to that, Microsoft provides integration services via the Microsoft Azure Data Factory. It provides a no-code platform to construct ETL and orchestrate pipelines.

⁸ See Chapter 8 for more details on entity resolution.

⁹ See Reference [7] for more on data sources supported by Purview.

Microsoft also provides a data virtualization preview for Azure SQL Managed Instance, which currently only supports querying external files stored in Azure Data Lake Storage or Azure Blob Storage. Regarding a replicate data solution, Azure relies on the change data capture solution on the source, for example, based on Microsoft Azure SQL databases and SAP.

Coming to self-service, Microsoft Azure Synapse Analytics provides SQL service as well. Synapse Analytics, as depicted in Figure 16-4, is powered by the Data Factory¹⁰ and supports a variety of data lakes, repositories, NoSQL, files, common protocols, and other services. It also has a common ODBC connector to extend the support for even more databases. With Microsoft Azure Synapse Analytics, users gain insights across DWH and big data platforms with a unified user experience. It transparently brings together the best technologies from multiple domains, such as the best SQL technologies for enterprise data warehouse, Spark technologies for big data, data explorer for logging and time series analysis, and pipelines for data integration and ETL/ELT. It also has deep integration with other Microsoft Azure services such as Power BI, CosmosDB, and AzureML.

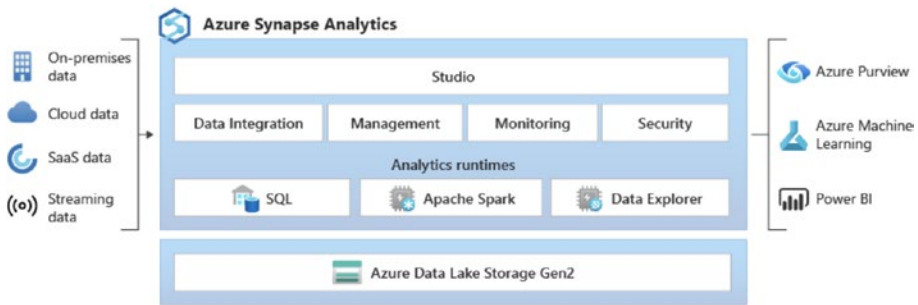


Figure 16-4. Azure Synapse

¹⁰See Reference [8] for more on data sources supported by Synapse Analytics.

The core service of Microsoft's Data Fabric approach is Purview, as depicted in Figure 16-5. It orchestrates with a few Microsoft Azure services to provide Data Fabric capabilities.

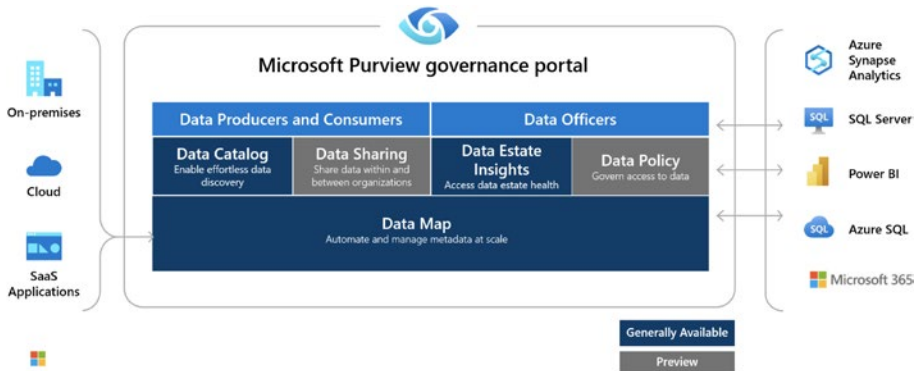


Figure 16-5. Microsoft Purview

First, Data Map automatically keeps the data catalog up to date with built-in automated scanning and classification systems. Second, Purview Data Catalog enables business and technical users to quickly and easily find relevant data using multiple dimensions, such as glossary terms, classifications, sensitivity labels, and more. Third, Microsoft Purview Data Sharing enables organizations to securely share data both within the organization or across organizations with business partners and customers. Lastly, Microsoft Purview Data Lifecycle Management (formerly Microsoft Information Governance) provides users tools to retain the content they need to keep and archive or remove the content that is obsolete. Regarding governance, Data Policy (currently in preview) can enforce policies through Microsoft Purview to the data sources that have registered for the policies. When the new policy is published, it will enforce the underlying data sources asynchronously.

A key differentiator of Microsoft Azure is its Microsoft Purview Data Estate Insights, which provides a C-suite, a complete view of their data estate with actionable insights to bridge the gaps discovered by the

governance process. Microsoft Purview and Azure Synapse are services on Microsoft Azure. As of today, Microsoft doesn't have an on-premises version. In terms of functionality, Microsoft just gets started in some areas, like virtualization, but the user experience and integration are phenomenal. Even across so many services, Microsoft still manages to deliver a consistent and quality user experience.

Denodo

Denodo is a niche vendor specialized in data virtualization, offering rich capabilities to build a logical Data Fabric focusing on data virtualization. It comprises a common semantic layer for provisioning data more quickly to the business, a dynamic data catalog for enterprise-wide data governance, and an industry-leading query engine powered by ML.

For connecting data, Denodo supports a variety of data sources,¹¹ including most databases on the market, data lakes with SQL interfaces, or allowing connections with generic JDBC. It also supports direct connections to Db2 for z/OS. Moreover, it runs queries directly on Salesforce with Salesforce wrappers and JSON data sources via connection services like ServiceNow and on CSV, Avro, Map files, sequence files stored in HDFS, and AWS S3 with Distributed File System Custom Wrapper. Denodo is specialized for virtualizing data, so it doesn't provide solutions for replication and transformation.

The Denodo platform provides comprehensive metadata and data discovery capabilities, including data governance, data lineage, change impact analysis, etc. The virtualization technology enables organizations to create unified data access and centralized governance policies across heterogeneous systems of structured and unstructured data sources.

¹¹ See Reference [9] for more details on data sources by Denodo.

Denodo has a self-service business glossary and information catalog that allow users to add their own business terms and find assets on users' needs. In addition, Denodo's data catalog uses AI/ML technology and provides interactive discovery features, collaboration capabilities, and personalized recommendations. Denodo also provides security and governance capabilities, including advanced data masking and attribute-based access control (ABAC), available for all data assets in the catalog, via a single point of control and administration.

Notably, Denodo's core technology is its data virtualization, as depicted in the middle of Figure 16-6, and built-in query acceleration features, such as aggregation awareness, flexible caching options, and query optimization. It has superior support for active metadata, which is captured from the source of static connections and dynamic data processing, such as statistics of data access and queries, latency of requests, etc. These metadata can be further used to optimize the Data Fabric.

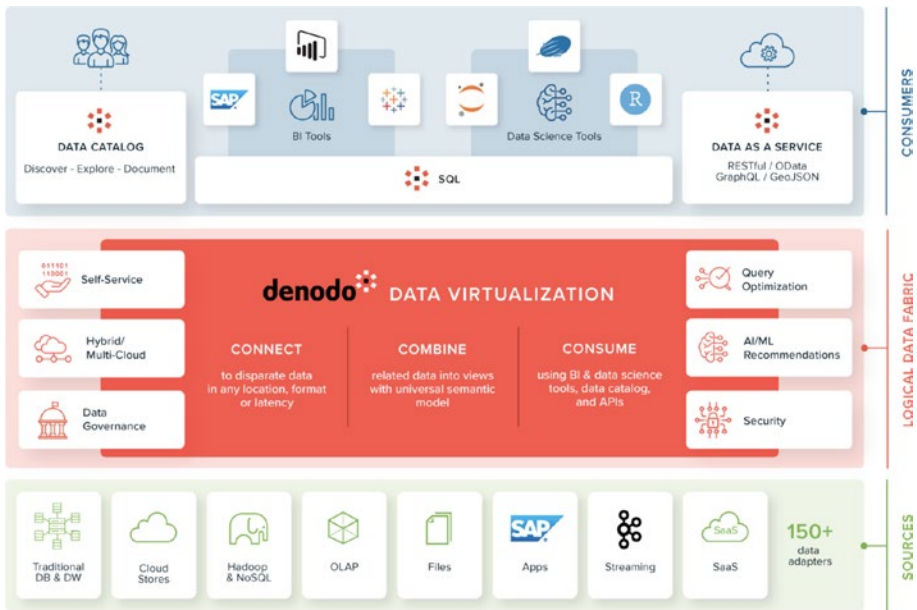


Figure 16-6. *Denodo*

Denodo has both an on-premises version and cloud services available on AWS, Microsoft Azure, and Google Cloud. It has more than 150 data adapters, supporting a broad spectrum of data sources. Denodo does not have a BI/AI solution, and customers need to integrate with other third-party solutions.

Informatica

Informatica is one of the leading providers in the Data Fabric market, ranking as the leader in data integration, data quality, and MDM in Gartner’s Magic Quadrant. Please, see Figure 16-7 for Informatica’s architecture.

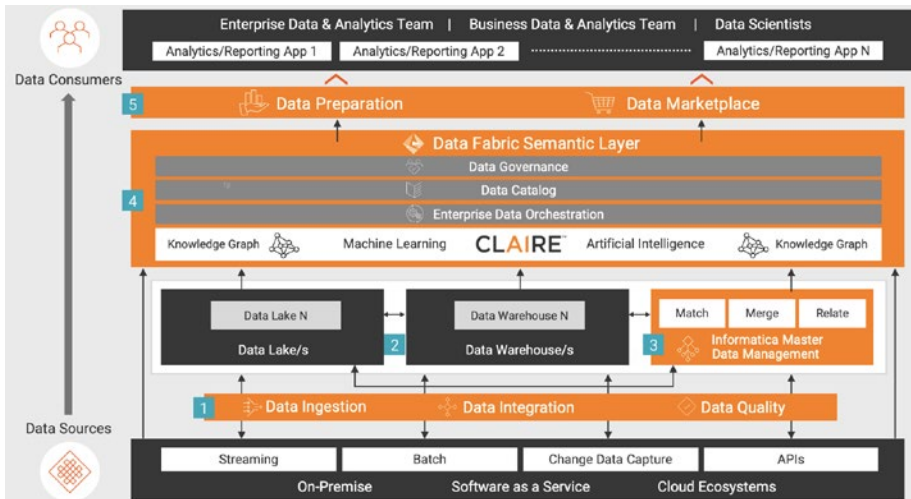


Figure 16-7. Informatica

In terms of connecting *data*, Informatica Data Ingestion and Data Integration products and services support replication, ELT, and change data capture for many data sources¹² to enable customers to build a data lake or DWH solution across different cloud providers. It supports not only most relational databases, data lakes, and file storage solutions but also messaging and social media. It currently supports HBase for NoSQL databases.

Both data integration and data quality products by Informatica provide low-code or no-code experience for citizen data teams. Data engineers can easily drag and drop on the canvas to define data integration and data analysis jobs. Data curators can perform various types of operations through the GUI such as cleaning data, creating dictionaries, parsing, and labeling data, creating rule specifications and verifiers. It also supports automatic dictionary generation from data analysis results. The newly generated dictionary can be used in later parsers, taggers, rules, or verifiers.

¹² See Reference [10] for more details on data sources supported by Informatica.

Informatica supports the use of AI/ML technologies to automatically capture and enhance metadata from disparate data sources and create a knowledge graph to establish connections between technical and business contexts. Its catalog provides a browsable and hierarchical view for users to easily find data assets with extended knowledge, such as data lineage, profiling results, tribal knowledge, etc. Informatica also provides powerful governance features. The augmented knowledge graph, which is established by linking technical data and business terms, enables the enforcement of business-relevant policies. Users can easily find which policies are related to a particular data asset.

Informatica supports both on-premises and cloud deployments partnering with AWS, Google Cloud, Microsoft Azure, Oracle, and Snowflake. The on-premises version remains a standalone installation-based product, and it does not offer data warehouse, data lakes, and BI or AI solutions, so users still need to integrate other vendor offerings providing these capabilities.

Key Takeaways

We conclude this chapter with a few key takeaways as summarized in Table 16-1.

Table 16-1. Key Takeaways

#	Key Takeaway	High-Level Description
1	IBM Cloud Pak for Data	<ul style="list-style-type: none"> • A wealth of integration options: direct access, transformation, virtualization, and replication. • Comprehensive self-service and governance capabilities: data discovery, intelligent cataloging, data profiling, data quality, and semantic search and recommendation. • Supports both on-prem deployment on a Red Hat OpenShift cluster and SaaS on IBM Cloud. • Can be used for Data Fabric architecture and Data Mesh solutions.
2	AWS	<ul style="list-style-type: none"> • Optimized for RDS and S3, requiring customization for other third-party data sources. • Crawlers scan data and retrieve metadata from different data sources. • No on-premises version. • Lego style, assembly required, steep learning curve.
3	Microsoft	<ul style="list-style-type: none"> • Unified user experience across multiple products, easy for self-serve. • Provides a completed view of their data estate and actionable insights to C-suites. • Gets started in virtualization areas. • No on-premises version.
4	Denodo	<ul style="list-style-type: none"> • Logic Data Fabric limited to data virtualization. • Specialized on virtualizing data, not moving or replicating data. • Supports both on-premises and SaaS.

(continued)

Table 16-1. (continued)

#	Key Takeaway	High-Level Description
5	Informatica	<ul style="list-style-type: none"> • Supports most popular data sources with ETL, change data capture, or replication. • No-code platform for data integration and data quality management. • No data warehouse, no data lake solution, no AI and BI tool. Needs integration with other vendor offerings. • Supports both on-premises and SaaS.

References

- [1] Gartner, Gupta, A., *Data Fabric Architecture Is Key to Modernizing Data Management and Integration*, 2021, www.gartner.com/smarterwithgartner/data-fabric-architecture-is-key-to-modernizing-data-management-and-integration (accessed August 24, 2022).
- [2] IBM, Limburn, J., *IBM named a leader in The Forrester Wave™: Enterprise Data Fabric, Q2 2022 – Driving end-to-end integration and reducing complexity across the data lifecycle*, www.ibm.com/blogs/journey-to-ai/2022/06/ibm-named-a-leader-in-the-forrester-data-fabric-wave-for-2022/ (accessed August 24, 2022).
- [3] IBM, *IBM Cloud Pak for Data*, www.ibm.com/docs/en/cloud-paks/cp-data/4.5.x?topic=data-supported-sources (accessed August 24, 2022)

- [4] AWS, *AWS Glue*, <https://docs.aws.amazon.com/glue/latest/dg/crawler-data-stores.html> (accessed August 24, 2022).
- [5] TechTarget Network, Gillis, A. S., Carty, D., *Amazon Athena*, www.techtarget.com/searchaws/definition/Amazon-Athena (accessed August 24, 2022).
- [6] Presto, *Fast and Reliable SQL Engine for Data Analytics and the Open Lakehouse*, <https://prestodb.io/> (accessed August 24, 2022).
- [7] Microsoft, *Azure*, <https://docs.microsoft.com/en-us/azure/purview/microsoft-purview-connector-overview> (accessed August 24, 2022).
- [8] Microsoft, *Azure*, <https://docs.microsoft.com/en-us/azure/data-factory/connector-overview> (accessed August 24, 2022).
- [9] Denodo, *Denodo Platform 8.0*, https://community.denodo.com/docs/html/browse/8.0/en/vdp/administration/appendix/supported_jdbc_data_sources/supported_jdbc_data_sources (accessed August 24, 2022).
- [10] Informatica, *Developer Tool Guide*, <https://docs.informatica.com/data-integration/powercenter/10-2/developer-tool-guide/connections/connections-overview/connection-types.html> (accessed August 24, 2022).

CHAPTER 17

Data Fabric and Data Mesh Research Areas

Imagine a hyper-automated Data Fabric or Data Mesh that is self-acting, self-improving, and self-optimizing, meaning that it can operationalize intelligence and AI insight in real time without human intervention or significantly reducing human-driven data and AI management tasks. Such a modern Data Fabric architecture or Data Mesh solution is not only infused with AI to gain more relevant insight, discover assets, or activate the digital exhaust; it is action-oriented and capable to auto-tune and auto-correct operations of both concepts. Applying this to AI governance, imagine a new international law or regulation, which affects your company in terms of establishing new or adjusting existing data-related processes, for example, guaranteeing bias-free AI models. A modern Data Fabric or Data Mesh should be able to apply ontology-based semantic searches to identify relevant data and AI assets that are affected by the law or regulation and autonomously infuse these assets to be processed by the trustworthy AI component¹ of your Data Fabric architecture.

In this chapter we examine trends and directions and research areas, which are based on many concepts that we have elaborated on in this book, such as active metadata, semantic knowledge graphs, activating the digital exhaust, intelligent information integration, etc.

¹ Review Chapter 5.

Introduction

It must have become obvious in this book that there is no unified Data Fabric architecture, which serves all purposes, use cases, and business scenarios. There is, for instance, a Data Fabric architecture that is specifically geared toward building a Data Mesh solution. However, there is also a *Data and AI Fabric*,² which addresses a broader set of assets than just data. Furthermore, a *Communication Fabric* may specifically address the needs of an omni-channel digital experience in the financial service sector, or an *Edge Fabric* is geared toward the needs for mobile- and 5G-based scenarios. Each of these Data Fabric architectures may require specific functions and features.

However, in this chapter, we take a holistic view and introduce trends and directions to augment a generic Data Fabric architecture by further infusing mainly AI techniques, for example, ontology-based semantic search, semantic knowledge graphs, etc. In addition, we elaborate on an AI-infused automated AI governance scenario, taking trustworthy AI as an example.

Finally, we introduce the concept of a hyper-automated Data and AI Fabric or Mesh that autonomously improves processes itself, limiting human intervention to approve, adjust, or also reject proposed actions.

AI-Based Augmented Insight

As we have seen throughout this book, the modern Data Fabric or Data Mesh DNA is distinctly driven by AI, which constitutes indeed a paradigm shift in how metadata management, entity matching, information integration, information governance, semantic search, semantic knowledge graphs, etc. are seen today. In this section, we present a few research areas that are highly relevant for both approaches.

² Review Chapter 10.

Generating a semantic knowledge graph is already a challenge by itself, especially if the graph is interspersed with business data and ontologies. However, exploiting and interpreting a semantic knowledge graph and transforming its inherent knowledge into automated recommendations and actionable insight represents another challenge. Both areas are dealt with in the research.³ For instance, trustworthy and explainable AI is addressed by applying ML and related interpretation algorithms, as we have seen in Chapter 5. Applying semantically enriched knowledge graphs improves the insight and explainability of AI.⁴ ML-based trustworthy AI approaches are limited by explaining and interpreting the influence of features for the predictive outcome or calculating the confusion matrix, which by itself limits the type of ML models that can be evaluated.

The attentive reader has certainly noticed that model fairness, drift detection, and the quality metrics discussed in the “Trustworthy AI” section of Chapter 5 were limited to multiclass classification and regression problems. However, exploiting semantic knowledge graphs can support interpretability and explainability of nearly all AI model types by discovering and depicting semantic and non-obvious relationships or depicting an ML or DL model in a simplified and more readable, explainable way.

Let us add a short note regarding terminology: we refer to the term *interpretation algorithms* in the context of model fairness, drift detection, etc. as dealt with in Chapter 5, whereas *interpretability* is geared toward the explainability of the model itself and its inferred outcome, for example, its predictive outcome.

A particularly hard problem to tackle is the trustworthiness and explainability of DL models, which relates to the semantics of the input, output, and middle layers of the Artificial Neural Network (ANN) or Deep Neural Network (DNN), the entanglement of the ANN or DNN nodes and

³ See Reference [1] for more information on recent trends in knowledge graphs.

⁴ See Reference [2] for more information on the role of knowledge graphs in explainable AI.

their weights in the middle layer(s), and the continuous adjustments and learning that the ANN or DNN is subject to, which contributes to the black-box perception of DL models. Developing interpretation algorithms or self-interpretable models, where trustworthiness is an intrinsic capability of the model itself, is a subject in the research.⁵ Semantic knowledge graphs⁶ are particularly well suited to improve understanding, interpretability, and explainability of the DL model outcome and decision-making process. Current Data Fabric or Data Mesh approaches have not yet embraced these aspects. Applying ontology-based semantic knowledge graphs to AI tasks in general – going above and beyond trustworthy and explainable AI – is a key area in the research.⁷

In Chapter 7, we have discussed semantic enrichment and a few enhancements that are broadening the scope of standard SQL, such as AutoSQL, confidence-based query matching, and semantic SQL, which are all key Data Fabric and Data Mesh capabilities. We have furthermore suggested domain- or industry-specific taxonomies or ontologies to serve as input to the semantic enrichment engine. This leads us to the topic of ontology-based semantic search, which is another key Data Mesh-related topic that is dealt with in the research. SPARQL (Simple Protocol and RDF Query Language) is a well-established RDF (Resource Description Framework) semantic query language for databases, able to retrieve and manipulate data stored in the RDF format. RDF is a method to represent information as triples, where a set of triples defines a knowledge graph.⁸ OWL (Ontology Web Language) is a knowledge representation and ontology language that can be used to describe RDF data. The RDF, OWL,

⁵ See Reference [3] for more information on trustworthiness and interpretability of DL models.

⁶ See Reference [4] for November 2021 conference proceedings on semantic knowledge graphs.

⁷ See Reference [5] for more information on ontology-based, large-scale knowledge graphs of AI tasks.

⁸ Please, review the section “Semantic Knowledge Graphs” in Chapter 5.

and SPARQL⁹ standards have already a long history in ontology-based semantic search.

In the context of a Data Fabric and Data Mesh, however, a key challenge in applying ontology-based semantic search is the heterogeneity of the data format and structure used in the knowledge catalog, particularly as it relates to the digital exhaust metadata and the varied AI artefacts (e.g., AI models, pipelines, ETL stages, etc.), which may not be represented in RDF format. Furthermore, using OWL to represent existing industry-specific ontologies and making them searchable for semantic enrichment via SPARQL may still be a relatively straightforward process. Nevertheless, domain-specific ontologies that relate to specific processes, for example, intelligent information integration and activating the digital exhaust, may first have to be developed using OWL. In addition, the relevant metadata, digital exhaust, and other input data for the Data Mesh semantic enrichment engine need to be transformed into the RDF format to make them accessible via SPARQL. Specific interfaces and tooling need to be developed to guide users through this process as well. Alternative approaches, partly based on open source initiatives, are currently investigated,¹⁰ for example, Apache Jena,¹¹ an open source Java framework for building semantic web and linked data applications.

Most of the current research in semantic knowledge graphs and ontology-based semantic search applies to the semantic enrichment tasks, such as generating active metadata, activating the digital exhaust, and searching for relevant data and AI assets in a self-service fashion within the context of a specific business or industry domain.

⁹ See Reference [6] for more information on SPARQL, RDF, and OWL.

¹⁰ See Reference [7] for more information on an ontology-based semantic search framework for disparate datasets.

¹¹ See Reference [8] for more information on Apache Jena.

AI-Infused Automated AI Governance

As we have seen in Chapter 15, AI governance and ethics¹² have become a reality that enterprises need to embrace. Indeed, existing regulations published by the European Commission and the US Department of State¹³ have already addressed trustworthy AI.

However, implementing AI governance with its many facets, like trustworthy and explainable AI and ethics, and addressing the AI lifecycle end to end – specifically as it relates to AIDevOps – still represents quite a challenge, especially with the claim to automate the implementation of AI governance as far as possible, for instance, by infusing AI into relevant processes.

The ever-increasing need of understanding these regulations and above all transforming and implementing the imperatives into the existing Data Fabric architecture or Data Mesh solution to guarantee compliance is often nontrivial.

AI-infused automated AI governance capabilities, as depicted in Figure 17-1, should support interpretation, transformation, and implementation of these regulations – at least as far as possible. With all the Data Fabric vendor offerings and their functions and features as we have discussed in Chapter 16, this sophisticated automated AI governance scope, however, is not a reality yet and subject to R&D.

¹² See Reference [9] for more information on AI governance.

¹³ Please, review Chapter 5.

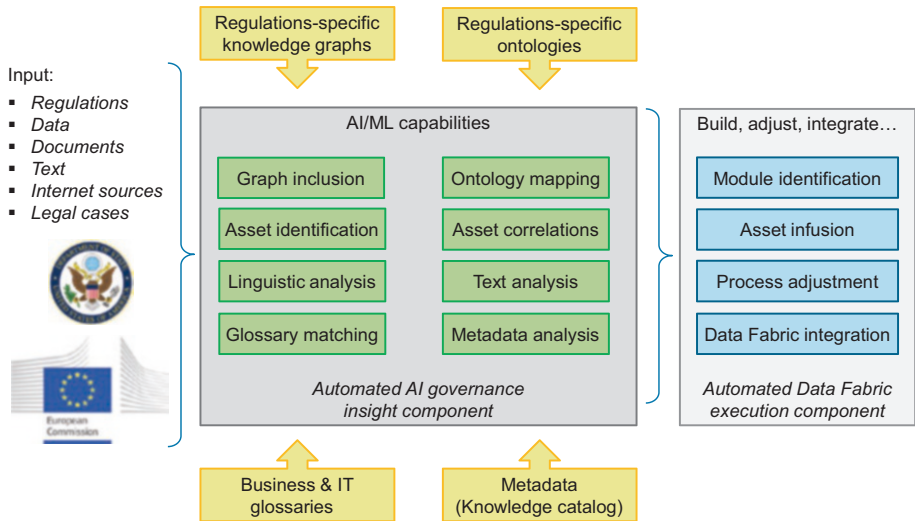


Figure 17-1. AI-Infused Automated AI Governance

Looking at Figure 17-1, let us discuss the various components and their functions. To illustrate the required functionality of an AI-infused *automated* AI governance platform, we are using trustworthy AI regulations from either the European Union or the US Department of State as an example.¹⁴

The AI-infused automated AI governance platform comprises the following two main components:

1. **Automated AI governance insight component:**

This component is essentially generating actionable insight based on regulation-specific input data (e.g., regulations, documents, Internet sources, legal assets, etc.), regulation-specific ontologies and knowledge graphs, business and IT glossaries, and relevant active metadata from the knowledge

¹⁴ See References [10] and [11] for details on guidelines of the European Commission and the US Department of State regarding trustworthy AI.

catalog. This input data is analyzed via AI/ML capabilities to generate actionable insight, which can (or rather should) be autonomously processed by the automated *Data Fabric execution component*.

2. **Automated Data Fabric execution component:**

This component processes the insight generated from the *automated AI governance insight component* to build additional or adjust existing Data Fabric processes to comply with regulations. New processes may have to be integrated into the Data Fabric, and existing ones must be customized or updated accordingly.

Let us first take a closer look at the *automated AI governance insight component*, detailing out specifically the relevant AI/ML and other capabilities. Most of these capabilities are subject to R&D. Needless to mention that additional capabilities may be required, such as easy-to-use GUIs and transformation modules, for example, to prepare input datasets and to generate JSON or Avro output datasets for dataset exchange purposes and so on:

- **Knowledge graph inclusion:** Existing regulation-specific knowledge graphs must be factored in; if none is available, they may have to be built first prior to inclusion. Semantic knowledge graphs may have to be generated, as outlined in Chapter 5, to leverage augmented insight related to non-obvious relationships or correlations, that is, of regulations to legal cases or aspects.
- **Ontology mapping:** Regulation-specific ontologies must be leveraged and mapped to relevant AI assets and Data Fabric processes. In our example, this means

mapping to ML models, pipelines, and Data Fabric processes that are relevant to guarantee trustworthy AI. This could, for instance, require a mapping to the relevant IBM Watson Studio trusted AI functionality, which our examples for trustworthy AI in Chapter 5 are based on.

- **Glossary matching:** Relevant business and IT glossaries available via the knowledge catalog must be matched to the regulation-specific key terms; new terms should be discovered and added to the existing glossaries.
- **Metadata analysis:** Existing active metadata should be analyzed to gain further insight; new metadata may have to be discovered and added to the knowledge catalog. Furthermore, active metadata will be used as input for some other modules, for example, *asset identification* and *asset correlation*.
- **Asset identification:** A key task is the identification of assets that are relevant for the trustworthy AI regulation, for example, existing ML or DL models, corresponding pipelines, Jupyter notebooks, etc. This requires discovery of AI assets stored in the knowledge catalog (active metadata) and may require input from the *linguistic analysis* and *text analysis* modules.
- **Asset correlation:** AI assets do not exist in isolation; they are correlated, which is either already captured in the knowledge catalog, or the correlation still needs to be discovered, that is, via the semantic enrichment engine of the Data Fabric architecture. In addition, the correlation goes above and beyond just the IT domain;

it should also include, for instance, correlation to the business domain and semantics of the AI model, including its business outcome and meaning.

- **Text analysis:** The regulation-specific documents and text need to be analyzed by applying ML techniques to automatically classify and extract valuable insights from unstructured text data. This may require correlating the analysis step with the relevant ontologies and active metadata.
- **Linguistic analysis:** The relevant documents and text may have to be linguistically analyzed as well, especially if sophisticated and legally relevant, for instance, to better understand the semantics (meaning) of the regularity text and impact in case of noncompliance.

Let us continue by deep-diving into the *automated Data Fabric execution component*, detailing out the capabilities it is composed of. This is indeed an area where most R&D is required to augment a Data Fabric architecture to automatically adjust and integrate required changes. Whereas the *automated AI governance insight component* generates actionable insights, this component automatically transforms the insight into actionable insight and autonomously executes the identified steps to implement trustworthy AI to guarantee compliance with the corresponding regulation:

- **Module identification:** Corresponding Data Fabric components, modules, products, etc. need to be identified to understand where adjustments need to be implemented. This can refer to an existing tool or component that delivers trustworthy AI for other ML models already to, for instance, improve explainability in addition to its currently implemented functional scope.

- **Asset infusion:** Identified AI assets need to be infused and made available to corresponding Data Fabric processes, which may require new integration methods, leveraging asset exchange standards to be used. For instance, operationalized ML models and pipelines must be integrated into the existing trustworthy AI product or ecosystem.
- **Process adjustment:** Relevant Data Fabric processes themselves or parts of the process flow need to be adjusted, which could relate to customizing pipelines, Jupyter notebooks, or ETL transformation stages. This could include automatically discovering, accessing, transforming, cleansing, and integrating new data records that are stored in data stores or generated in transactional systems that have not been accessed before. This must be done while corresponding ML models are deployed and in operation.
- **Data Fabric integration:** Overall integration of required changes, new modules, assets, or even ML algorithms must be taken into consideration by the Data Fabric. Adopting trustworthy AI in the context of new regulations may, for instance, require new ML algorithms from vendors for their corresponding products to be integrated autonomously to calculate required measures related to AI interpretation or interpretability. This obviously relates to an established AIDevOps infrastructure to be in place.

In Chapter 14, we have seen several existing capabilities, such as to auto-discover and classify data, to auto-detect sensitive data, to auto-analyze quality data, and to auto-assign business terms. Nevertheless, the scope and aspiration of an AI-infused automated AI governance as

outlined previously remains subject to R&D. Some vendors are already referring to an automated Data Fabric.¹⁵ However, this relates to a subset of tasks and processes, for example, automated data lineage, automated data discovery, etc.

A Software as a Service (SaaS) cloud deployment model may impose additional requirements that consider a services-based Data Fabric or Data Mesh implementation.

Hyper-automated Data and AI Fabric

As we have seen previously, vendors are already referring to an automated Data Fabric, meaning that certain tasks are performed automatically. Indeed, the Data Fabric journey is certainly moving toward identifying and automating as many business and IT processes as possible to eliminate or at least to reduce human intervention.

We refer to the term *hyper-automation*¹⁶ to automate adjustments and optimization of Data and AI Fabric or Mesh processes, meaning to *auto-tune*, to *self-correct*, and to *auto-implement* improvements or changes to these processes – mainly implemented via infusion of AI/ML, which also includes auto-adjustments to already defined data products.

Figure 17-2 introduces a few examples of how this might look like once vendor R&D organizations have delivered corresponding capabilities in their product suite.

¹⁵ See References [12] and [13] for examples on automated Data Fabric mentions.

¹⁶ See Reference [14] for more on hyper-automation.

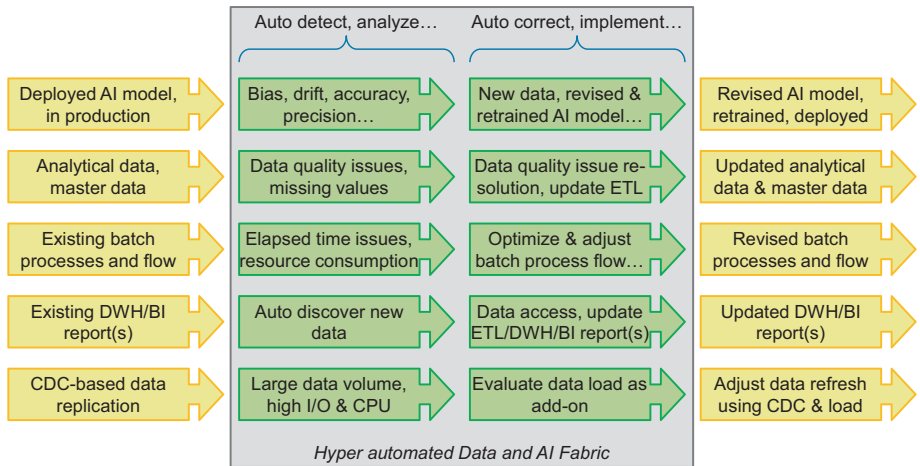


Figure 17-2. *Hyper-automated Data and AI Fabric Examples*

Needless to mention that AI/ML plays a pivotal role in delivering these hyper-automated Data and AI Fabric or Mesh capabilities. However, additional capabilities are required to address the integration into corresponding Data Fabric architectural components:

- AI model adjustments:** Once bias or drift or worsening accuracy or precision of operationalized AI models has been detected, this should be automatically corrected, by, for instance, taking new data for retraining, validation, and test of AI models. Revised AI models should be auto-deployed and operationalized.
- Data quality issue corrections:** Analytical or master data changes over time, which may cause new data quality issues to surface. Discovery and resolution of these quality issues should be performed autonomously and according to rules, policies, and business constraints, which could mean corrections to be auto-implemented in real time, delivering sustainable data quality.

- **Batch process flow optimization:** The elapsed time or resource consumption of batch processes themselves and the flow they are embedded in should be optimized and adjusted in an automated fashion, resulting in revised batch processes and flows.
- **DWH/BI report adjustments:** New data from new or even existing source systems that may be relevant for existing DWH/BI reports should be auto-discovered, resulting in automated updates of corresponding ETL stages and DWH/BI reports.
- **Data refresh adjustments:** Existing CDC-based data replication methods may result in large data volumes to be moved with high I/O or CPU cost. Introducing data load for some tables or table partitions may optimize the data refresh process by complementing the existing CDC-based replication.

The preceding examples illustrate the wide scope of Data Fabric-related processes; data quality (including data consistency, completeness, monitoring, preventing errors, etc.) as a particular Data Fabric area enjoys great popularity in the research and academia community.¹⁷ The same is true for hyper-automation with the caveat that the core research is focused on IoT, Robotic Process Automation (RPA),¹⁸ and specific industries.

However, specific research related to hyper-automation challenges seems to be left overall to vendor R&D organizations.

¹⁷ See References [15] and [16] for more information on data quality issues.

¹⁸ See Reference [17] for more information on hyper-automation.

Key Takeaways

We conclude this chapter with a few key takeaways as summarized in Table 17-1.

Table 17-1. *Key Takeaways*

#	Key Takeaway	High-Level Description
1	Exploiting semantic knowledge graphs.	Interpreting semantic knowledge graphs and transforming their inherent knowledge into automated and actionable insight to improve the Data Fabric or Data Mesh still represents a challenge.
2	Explainability of AI models above and beyond multiclass classification problems.	Exploiting semantic knowledge graphs can support interpretability and explainability of nearly all AI model types (including DL models) by discovering and depicting semantic and non-obvious relationships or depicting an ML model in a simplified and more readable, explainable way.
3	Developing domain-specific ontologies.	Domain-specific ontologies that relate to specific Data Fabric processes, for example, intelligent information integration and activating the digital exhaust, may have to be developed using OWL.
4	Building an AI-infused automated AI governance.	The need of understanding regulations and implementing the imperatives into the existing Data Fabric or Data Mesh to guarantee compliance is nontrivial; capabilities and specific Data Fabric architectural components are required to automatically generate actionable insight and auto-execute identified steps.
5	Hyper-automated Data and AI Fabric is largely subject to R&D.	The term <i>hyper-automation</i> refers to automating adjustments and optimization of Data and AI Fabric or Mesh processes: auto-tune, self-correct, auto-implement improvements or changes to these processes – mainly via infusion of AI/ML.

References

- [1] Tiwari, S., Al-Aswadi, F.N., Gaurav, D., *Recent trends in knowledge graphs: theory and practice*, Soft Comput 25, 8337–8355 (2021), <https://doi.org/10.1007/s00500-021-05756-8> (accessed August 23, 2022).
- [2] Lecue, F., *On The Role of Knowledge Graphs in Explainable AI*, 2020, www.semantic-web-journal.net/system/files/swj2198.pdf (accessed August 23, 2022).
- [3] Xiong, H., Liu, J., Bian, J., *Interpretable Deep Learning: Interpretation, Interpretability, Trustworthiness, and Beyond*, 2021, file:///Users/eberhardhechler/Downloads/Interpretable_Deep_Learning_Interpretations_Interp-1.pdf (accessed August 24, 2022).
- [4] Villazón-Terrazas, B., Ortiz-Rodriguez, F., Tiwari, S., Goyal, A., Jabbar, M. A., *Knowledge Graphs and Semantic Web*, Springer, 2021, ISBN-13: 978-3030913045.
- [5] Blagec, K., Barbosa-Silva, A., Ott, S. et al. *A curated, ontology-based, large-scale knowledge graph of artificial intelligence tasks and benchmarks*, Sci Data 9, 322 (2022), <https://doi.org/10.1038/s41597-022-01435-x> (accessed August 25, 2022).
- [6] W3C, *Web Ontology Language (WOL)*, www.w3.org/OWL/ (accessed August 24, 2022).

- [7] Kaur, P. et al., *Ontology-Based Semantic Search Framework for Disparate Datasets*, 2022, file:///Users/eberhardhechler/Downloads/TSP_IASC_45927.PDF (accessed August 24, 2022).
- [8] Apache, *Apache Jena*, https://jena.apache.org/about_jena/ (accessed August 25, 2022).
- [9] Liu, Z., Zheng, Y., *AI Ethics and Governance: Black Mirror and Order*, Springer, 2022, ISBN-13: 978-9811925306.
- [10] European Commission, *Ethics Guidelines for Trustworthy AI*, <https://ec.europa.eu/futurium/en/ai-alliance-consultation.1.html> (accessed July 26, 2022).
- [11] US Department of State, *Artificial Intelligence (AI)*, www.state.gov/artificial-intelligence/ (accessed July 26, 2022).
- [12] Informatica, *Data Fabric, the Transformative Next Step in Data Management*, www.informatica.com/se/resources/articles/data-fabric-the-transformative-next-step-in-data-management.html (accessed August 25, 2022).
- [13] NetApp, *Optimized and Automated Data Fabric, Optimize: Listen to that data fabric purr*, www.netapp.com/data-fabric/optimized-automated-data-center/ (accessed August 25, 2022).
- [14] IBM, IBM Cloud Education, *Hyperautomation*, www.ibm.com/cloud/learn/hyperautomation (accessed August 26, 2022).

- [15] McCord, S. E. et al., *Ten practical questions to improve data quality*, 2021, www.sciencedirect.com/science/article/pii/S0190052821000699 (accessed August 26, 2022).
- [16] McGilvray, D., *Executing Data Quality Projects*, 2nd edition, Academic Press, 2021, ISBN-13: 978-0128180150.
- [17] Madakam, S., Holmukhe, R. M., Revulagadda, R. K., *The Next Generation Intelligent Automation: Hyperautomation*, 2022, www.scielo.br/j/jistm/a/8BnnjHkvFGrmBFdtXmhNtC/?format=pdf&lang=en (accessed August 26, 2022).

CHAPTER 18

In Summary and Onward

Reaching the end of the book, it must have become obvious that both a Data Fabric architecture and Data Mesh solution are inevitably associated with applying AI, intelligent knowledge, and automation. Indeed, infusing AI is required to enable intelligent cataloging, to generate active metadata, to build semantic knowledge graphs, and to gain necessary and holistic insight to improve, optimize, and automate tasks and to enable self-service generation of data products that are ready for consumption. These capabilities are enabled via a knowledge catalog that stores active metadata and data product specifications as well.

While a Data Fabric is an architecture that facilitates the end-to-end integration of various data and AI pipelines across hybrid cloud environments through the use of intelligent and automated systems and applications, a Data Mesh should be seen as a solution, which is geared toward delivering data-as-a-product in an organizational federated approach. To understand the entanglement of these two concepts, we have presented the Data Mesh as a solution, which is underpinned and enabled by an AI-infused Data Fabric architecture.

A data product is based on semantically related raw data that is transformed into a meaningful business context and easily consumable by business users; it comes with data product ownership, defined SLAs, access methods, and policies and rules and is registered in the

knowledge catalog using, for instance, a JSON format. Once data product specifications are stored in the knowledge catalog, the data products are searchable, discoverable, and ready for consumption by business users.

Data Fabric and Data Mesh Summarized

There are four distinct but nevertheless related entry points or use case scenarios, which could support a consumable and manageable Data Fabric journey. These entry points could be pursued individually or even in parallel, depending on business objectives and priorities:

1. **Data and AI governance:** Automatically applies industry-specific regulatory policies and rules to your data and AI assets, to quickly establish an environment for highly automated and consistent AI governance and to automatically secure data across the enterprise.
2. **Hybrid cloud data integration:** Creates a unified view of all enterprise data and AI assets, enabling consistency between operational applications. It furthermore consolidates and simplifies IT infrastructures to be deployed anywhere (on-premises or any cloud) and automates data operations to deliver trusted data to business users. It can also prevent delays and disruptions of mission-critical data through data resilience and easy data access, enabling data and AI assets to be easily consumable as data products.
3. **Customer 360-degree insight:** Provides a comprehensive view of customers by integrating data across heterogeneous domains by spending

more time on applying AI and analytics to business challenges vs. wasting time on hunting quality data that is not well understood. It breaks down data silos with an integrated view of data and supports relevant business outcomes at pace with competitive needs.

4. **Trustworthy AI and MLOps:** Unlocks trustworthy, explainable AI starting with governed data access for data scientists. It furthermore enables automated MLOps infused with trust throughout the entire AI lifecycle and AI governance by introducing transparency and monitoring for each stage of the AI lifecycle.

The following list is a summary of the most essential Data Fabric and Data Mesh characteristics, which are all enabled through data and AI assets that are stored in the knowledge catalog. All of these characteristics have been elaborated on in this book. Some of these characteristics are interrelated with each other and subsequently treated in various chapters in this book; they have been discussed from different perspectives investigating different facets. This is especially true for metadata and data quality aspects:

- **Automated metadata enrichment:** Means infusing AI into the metadata enrichment process to discover non-obvious relationships, perform automatic data class assignments, generate semantic knowledge graphs, conduct data profiling and auto-correction of asset quality, etc.
- **Self-service information integration:** Data engineers and even business users need to perform information integration tasks, including data discovery, metadata

enrichment, data exploration, preparation, and transformation, in a self-service manner and regardless of the complexity of source systems. We have also addressed this intelligent information integration.

- **Automated workload distribution:** AI-enabled automated workload distribution needs to be enabled and supported, considering underlying system capabilities, resource consumption and constraints, SLAs, and performance needs, such as related to data throughput and latency, query elapsed time, etc.
- **Data product** (data-as-a-product, data marketplace): Clear ownership of data and AI assets, understanding semantically connected data, including discovery and access of data and AI assets, self-service consumption of data products without IT (or at least limited IT) involvement, SLAs, and enabling monetization of data and AI assets are key characteristics when producing and consuming data products.
- **Trustworthy and explainable AI:** Addressing emerging trustworthy AI regulations needs to be supported by automated detection and correction of model fairness (bias) and drift, explainability of AI models for business users, and measurement of key quality metrics.
- **Active metadata exploitation:** Business users should transparently leverage active metadata to gain pervasive and actionable insight regarding underlying data and AI assets, which includes search and discovery of assets, available access methods, asset ownership, enforced policies and rules, semantically connected assets, etc.

- **Leveraging the digital exhaust:** Information integration, workload distribution, data product build processes, and resource assignment should be improved over time by automatically activating the digital exhaust, that is, by optimizing resource allocation and data access methods for end-of-month integration tasks. Leveraging the digital exhaust means *learning* information integration and other tasks over time to improve, optimize, and simplify these tasks autonomously over time.
- **AI governance:** Distributed, federated, and organizational data and AI governance should address the entire scope of data and AI assets. AI-infused governance should support auto-mapping to regulations, limiting human intervention, and thus accelerating and simplifying adherence to regulatory compliance. This is obviously related to trustworthy and explainable AI.

Where to Go from Here

Looking at the dispersed systems and heterogeneous data landscape today on the one hand and the ever-increasing need for business agility and digitalization on the other results without a doubt in reimagining how to intelligently integrate and gain actionable insight from data. To build an intelligent fabric of data and AI enabling a mesh of data products that can easily be searched for and consumed by business users is the challenge enterprises are confronted with today.

In Chapter 17, we have examined a few Data Fabric and Data Mesh trends that are prevalent in the research and academia community. Today's focus may be on gaining more relevant and actionable insight

by enriching metadata (generating active metadata), activating the digital exhaust, or discovering non-obvious and semantic relationships across data assets; tomorrow's focus, however, will doubtless be to autonomously enhance Data Fabric and Data Mesh processes, which includes to auto-tune and auto-optimize and to auto-correct and auto-improve tasks, further reducing human intervention and thus improving business agility and time to value. This clearly affects how we need to look at DataOps, MLOps, ModelOps, and AIOps to intelligently entangle the operationalization of data and AI with addressing new business imperatives and circumstances.

Looking ahead, we should envision the Data Fabric architecture and Data Mesh solution to become more intelligent and autonomous over time, enabling business users and IT professionals to interact at an augmented level with increased efficiency and agility. As we further develop our passion for reinventing and improving capabilities, our hope is that this book may have given you useful ideas to get you started on your Data Fabric and Data Mesh journey.

Key Takeaways

We conclude this chapter with a few key takeaways that are collectively derived from all previous chapters, summarized in Table [18-1](#).

Table 18-1. Key Takeaways

#	Key Takeaway	High-Level Description
1	Data Fabric and Data Mesh are inevitably associated with applying AI, intelligent knowledge, and automation.	Augmenting Data Fabric and Data Mesh capabilities by infusing AI enables us to do intelligent cataloging, generate active metadata, build semantic knowledge graphs, and gain necessary and holistic insight to improve, optimize, and – most importantly – automate Data Fabric and Data Mesh tasks.
2	A Data Mesh solution is underpinned by a Data Fabric architecture.	A Data Mesh should be seen as a solution, which is geared toward delivering data-as-a-product (data marketplace) in an organizational federated approach; it is underpinned and enabled by an AI-infused Data Fabric architecture.
3	A data product is a semantically related set of datasets and ready for business consumption.	A data product is based on semantically related raw data that is transformed into a meaningful business context and easily consumable by business users; it comes with data product ownership, defined SLAs, access methods, and policies and rules and is registered in the knowledge catalog.
4	There are four key Data Fabric entry points.	The four key Data Fabric entry points are (a) data and AI governance, (b) hybrid cloud, (c) trustworthy AI and MLOps, and (d) 360-degree customer view.
5	There are distinct drivers to implement a Data Fabric architecture and Data Mesh solution.	The key drivers for a Data Fabric architecture and Data Mesh solution are the need to locate data faster, to simplify access and consumption of data, to apply consistent data and AI governance, and to move the data engineer from being the perceived <i>owner</i> of the data to being the enabler for the data source owners.

(continued)

Table 18-1. *(continued)*

#	Key Takeaway	High-Level Description
6	Trustworthy AI is an essential aspect of a Data Fabric.	Trustworthy AI means to detect AI model bias, to measure and ensure AI model fairness, to detect drift, to provide explainability, and to calculate model metrics that serve as input to enable trustworthy AI throughout the entire AI lifecycle, including AI operationalization.
7	ML-infused entity matching is a key aspect of a Data Fabric.	Entity matching is a classification problem for labeled data and a clustering problem for unlabeled data. Many problems in the entity matching process can be solved by ML methods, where SVM, K-means, and decision trees are common algorithms.
8	Data Fabric architecture implements DataOps, MLOps, ModelOps, and AIOps practices.	Data Fabric implements DataOps, MLOps, ModelOps, and AIOps practices via a unified enterprise data and AI architecture for consolidating dispersed data from a hybrid cloud environment through automated data discovery, intelligent information integration, and intelligent cataloging.
9	Several intelligent information integration styles are needed.	Depending on use case requirements, Data Fabric needs to support multiple intelligent information integration styles, such as data federation and virtualization or microservices with REST APIs, SQL and NoSQL, streaming, messaging, etc.
10	AI/ML needs to be leveraged to enable automated quality assessments.	Quality assessments in the context of a modern Data Fabric need to be infused with AI/ML and developed into intelligent automated quality management processes, including auto-correction of data quality issues.

(continued)

Table 18-1. (continued)

#	Key Takeaway	High-Level Description
11	Hyper-automated Data and AI Fabric is largely subject to R&D.	The term <i>hyper-automation</i> refers to automating adjustments and optimization of Data and AI Fabric processes: auto-tune, self-correct, auto-implement improvements or changes to these processes – mainly via infusion of AI/ML.

Abbreviations

AI	Artificial Intelligence
ABAC	Attribute-Based Access Control
ACC	Accuracy
ACID	Atomicity, Consistency, Isolation, Durability
ANN	Artificial Neural Network
AOD	Architecture Overview Diagram
API	Application Programming Interface
AUC	Area Under the Curve
AWS	Amazon Web Services
B2B	Business-to-Business
BI	Business Intelligence
CAMS	Common Assets Managed Services
CCPA	California Consumer Privacy Act
CDC	Change Data Capture
CDO	Chief Data Officer
CI/CD	Continuous Delivery/Continuous Deployment
CICS	Customer Information Control System
CIO	Chief Information Officer
CNTK	Cognitive Toolkit
CP4D	Cloud Pak for Data

ABBREVIATIONS

CPU	Central Processing Unit
CRM	Customer Relationship Management
CSV	Comma-Separated Values
CTO	Chief Technology Officer
DaaS	Data as a Service
DL	Deep Learning
DNN	Deep Neural Network
DV	Data Virtualization
DVM	Data Virtualization Manager
DWH	Data Warehouse
EDW	Enterprise Data Warehouse
ELT	Extract-Load-Transform
ETL	Extract-Transform-Load
FN	False Negatives
FNR	False Negative Rate
FP	False Positives
FPR	False Positive Rate
GDB	Graph Database
GDPR	General Data Protection Regulation
GNN	Graph Neural Network
GPU	Graphical Processing Unit
GRU	Gated Recurrent Unit
GUI	Graphical User Interface
HDFS	Hadoop Distributed File System
HIPAA	Health Insurance Portability and Accountability Act

HPO	Hyperparameter Optimization
IBV	Institute for Business Value
ICA	Independent Component Analysis
IIS	InfoSphere Information Server
I/O	Input/Output
IoT	Internet of Things
I/U/D	Insert/Update/Delete
JDBC	Java Database Connectivity
JSON	JavaScript Object Notation
KPI	Key Performance Indicator
LDA	Linear Discriminant Analysis
LGPD	Lei Geral de Proteção de Dados
LoB	Line of Business
LSTM	Long Short-Term Memory
MAE	Mean Absolute Error
MDM	Master Data Management
ML	Machine Learning
MSE	Mean Squared Error
NLG	Natural Language Generation
NIST	National Institute of Standards and Technology
NLP	Natural Language Processing
NLU	Natural Language Understanding
ODBC	Open Database Connectivity
ODOi	Open Data Platform initiative
OMAG	Open Metadata and Governance

ABBREVIATIONS

OMRS	Open Metadata Repository Services
ONNX	Open Neural Network eXchange
OWL	Ontology Web Language
P	Precision
PCA	Principal Component Analysis
PCI	Personal Confidential Information
PFA	Portable Format for Analytics
PII	Personally Identifiable Information
PMML	Predictive Model Markup Language
PR	Precision/Recall
R	Recall
RDBMS	Relational Database Management Systems
RDF	Resource Description Framework
REST	Representational State Transfer
RMSE	Root Mean Squared Error
RNN	Recurrent Neural Network
ROC	Receiver Operating Characteristic
SaaS	Software as a Service
SDI	SQL Data Insights
SDK	Software Development Kit
SEPA	Single Euro Payments Area
SHAP	SHapley Additive exPlanations
SLA	Service-Level Agreement
SME	Subject Matter Expert
SOM	Self-Organizing Map

SPARQL	Simple Protocol and RDF Query Language
SQL	Structured Query Language
SSN	Social Security Number
SVM	Support Vector Machine
TN	True Negatives
TNR	True Negative Rate
TP	True Positives
TPR	True Positive Rate
URL	Uniform Resource Locator
VSAM	Virtual Storage Access Method
wFPR	weighted False Positive Rate
WKC	Watson Knowledge Catalog
wTPR	weighted True Positive Rate
YARN	Yet Another Resource Negotiator
XGBoost	eXtreme Gradient Boosting
XML	Extensible Markup Language
x86	Server with Intel Processors

Index

A

Acquire-transform-manage data
 pipeline, 75

Action problem, 129

Active learning, 187

AI algorithms, 195

AI and ML

 AI-infused data, 151

 business organizations, 147

 data access, 150

 data discovery, 149

 data Mesh solution, 164

 data profiling, 150

 data-rich enterprises, 147

 data steward, 156

 digital exhaust, 154

 governance, 152

 implications, 148

 knowledge/governance

 catalog, 149

 semantic enrichment, 152

 state-of-the-art entity, 164

 usage, 148

AI artefacts, 22

AI-based augmented

 insight, 376–379

AI-based generation, 156

AI-based intelligent search, 160

AI-based knowledge catalog, 24

AI-based pattern, 157

AI-based semantic enrichment, 25

AI engineering, 196, 200, 203, 211

AI-generated digital exhaust
 metadata, 25

AI governance, 24, 48, 204

AI-infused automated AI

 governance

 capabilities, 380, 381, 385

 components, 381, 382

 Data Fabric vendor offerings, 380

 data set exchange, 382–384

 trustworthy AI, 384, 385

 trustworthy and

 explainable AI, 380

AI-infused capabilities, 89

AI-infused Data Fabric, 23–25

AI-infused entity matching, 25

AI-infused topics, 154

AI lifecycle

 build models, 198

 business problems, 197

 data collection, 197

 data preparation, 198

 deploy models, 199

INDEX

- AI lifecycle (*cont.*)
 - goal, 200
 - govern models, 199
 - inference latency, 197
 - monitor models, 199
 - optimal solution, 196
 - service latency, 197
 - stages, 197, 225
- AI/ML algorithms, 136
- AI/ML-based automated
 - generation, 22
- AI/ML-based
 - capabilities, 166
- AI/ML-based matching engine,
 - 165, 167
- AI/ML methods, 167, 323
- AI/ML technology, 367
- AI model adjustments, 387
- AI models
 - drift detection, 104, 108
 - business reality, 109
 - data consistency, 109, 110
 - explainability, 104
 - model explainability, 111
 - LIME, 111
 - ML model, 112
 - SHAP, 112
 - model fairness, 104–106
 - bias, 106, 107
 - evaluations, 108
 - fairness score, 107
 - favorable outcomes, 106
 - ML model, 108
 - perfect equality, 106
 - quality metrics, 105
 - ACC, 113
 - confusion matrix, 115
 - FNR, 113
 - FPR, 113
 - harmonic mean, 114
 - logarithmic loss, 114
 - PR curve, 114
 - precision, 113
 - ROC curve, 114
 - TNR, 113
 - TPR, 113
- Amazon Redshift, 362
- Amazon Relational Database Service, 362
- Amazon Web Services (AWS)
 - data connection, 361
 - data services, 361
 - direct SQL service, AWS
 - Athena, 362
 - Glue Elastic Views, 362
 - Lake Formation, 362
 - serverless services, 363
 - supported sources, 362
- Analytics projects, 206
- Annotation, 321–324
- Anomaly detection, 129
- Apache Jena, 379
- Application architecture, 262
 - API enablement, 263, 264
 - approaches, 263, 264
 - business impact *vs.*
 - feasibility, 264
 - containerization, 263, 264

- data access *vs.* data replication, 265
- data consistency *vs.* eventual consistency, 265
- determination, 258
- internal application structure, 262
- microservices, 264, 265
- patterns, 273
- user communication channels, 263
- Application programming interface (API), 4, 263
- Architecture, 258
- Architecture overview diagram (AOD), 53
- Artificial intelligence (AI), 4, 123
- Artificial neural network (ANN), 320, 323, 377, 378
- Asset correlation, 381, 383
- Asset identification, 383
- Asset infusion, 385
- Assets
 - AI-Infused Understanding, 159
 - AutoSQL, 163
 - ETL stages and ML models, 158
 - metadata, 159
 - ML/DL techniques, 160
 - quality assessments, 161
 - quality scores, 161
 - and relationship needs, 160
 - semantic query, 163
 - SQL query performance, 162
 - standard SQL, 163
- Athena data types, 362
- Attribute, 322
- Attribute-based access control (ABAC), 367
- Attribute-level threshold, 184
- Auto-assignment, 157
- Auto-correct task, 398
- Auto-detecting sensitive data, 157
- Auto-improve task, 398
- Auto-mapping, 190
- Automated AI governance insight component, 381, 382, 384
- Automated Data Fabric, 386
- Automated Data Fabric execution component, 382, 384
- Automated data quality assessment
 - AI/ML, 328
 - challenge, 324
 - data asset, 325
 - database tables, 327
 - data class violations, 325
 - data quality dimensions, 327
 - data type violations, 325
 - duplicated values, 326
 - format violations, 326
 - inconsistent capitalization, 325
 - inconsistent representation, missing values, 326
 - innovative methods, 328
 - missing values, 326
 - outcome, 324
 - rule violations, 326
 - suspect values, correlated columns, 326
 - values out of range, 326

INDEX

Automated enrichment, 100
Automated tagging, 321–324
Automotive manufacturing industry, 131
AutoSQL, 378
AWS core services, 363
AWS Glue, 361
AWS Glue crawlers, 361
AWS Glue Data Catalog, 362
AWS Glue Elastic Views, 362
AWS Lake Formation, 362
Azure Data Lake Storage, 364
Azure SQL Managed Instance, 364

B

Batch process flow
 optimization, 388
Bias-free AI models, 375
Big data, 3, 9, 11
Binary classification model,
 98, 105, 113
Blocking, 186
Blocking algorithms, 183
Build models, 198
Business analysts, 293, 302
Business applications, 202
Business-critical applications, 220
Business domain, 47
Business-driven data
 requirements, 258
Business drivers, 73
 current and consistent data, 74
 data and AI asset, 73

 data pipeline, 75
 data quality, 74
 impact of change, 74
 knowledge catalog, 73
 multiple locations, 73

Business leaders, 209
Business metadata, 91
Business Term Assignment, 322
Business-to-business (B2B), 261
Business users, 25, 26, 28, 31,
 33, 52, 238

C

California Consumer Privacy Act (CCPA), 342
Cataloging process, 155, 298
Centralized data provisioning, 357
Change data capture (CDC), 250
Chief data officer (CDO), 43
Classification problem, 128
Cloud Pak for Data (CP4D), 272
Cloud providers, 55
Cloud services, 72, 277
 private, 279
 public, 278, 279
Cloud-style provisioning, 56
Clustering problem, 129
Common Assets Managed Services (CAMS) repositories, 38
Communication Fabric, 376
Composite attributes, 178
Computational complexity, 183
Computer vision, 130

Confidence-based query
 matching, 378
 Containers, 263
 Continuous Integration/Continuous
 Deployment (CI/CD), 203
 Credit scores, 208
 Cross-LoB scenarios, 285
 Customers, 306
 Customized model, 190, 191

D

DAMA-DMBOK2 knowledge area
 wheel, 337
 Data, 334
 Data analysis and profiling
 AI role, 319
 categorization, 316
 data clustering, 320
 data mining, 321
 DL, 318
 extrapolation statistics, 319
 ML, 318
 relationship discovery, 320
 representative sample
 datasets, 319
 structure, data and AI
 assets, 319
 user counts, US State,
 CREDITUSERS Table, 318
 Data and AI democratization, 204
 Data and AI governance, 312, 338
 AI-related tasks, 337
 analogy, 340
 analytical data, 335
 companies, 335
 DAMA-DMBOK2 knowledge
 area wheel, 337
 data and AI-related
 management tasks, 334
 data architecture, 338
 Data Fabric architecture,
 341, 342
 data management, 337
 data quality, 339
 data silos, 336
 DBMS, 334
 definition, 334
 DNA, 334
 enterprises, 335
 integration and
 interoperability, 339
 master data, 340
 metadata, 340
 organizations, 335
 property management
 team, 340
 reference data, 340
 regulatory compliance, 336
 security and privacy, 338
 storage and operations, 338
 Data architecture, 338
 big data, 9
 capabilities, 258
 categories, 266, 267
 challenges, 281
 data access, 267
 data lake implementations, 10

INDEX

- Data architecture (*cont.*)
 - data lakehouse, 12
 - data standards, 266
 - data virtualization, 267
 - decision criteria, 258
 - delivery system
 - architecture, 267
 - EDW architecture, 6–9
 - ETL-transformed data, 267
 - organizations, 258
 - requirements, 267
 - technologies, 266
 - values and challenges, 5, 6
- Data-as-a-product, 18, 21, 26, 29, 31, 37, 39, 103
- Data as a service (DaaS), 48, 59
- Data assessment, 58
- Data cataloging, 18
- Data characteristics, 281
- Data cleansing, 132
- Data consumer, 45
- Data consumption patterns, 242
 - AI models, 236
 - AI pattern, 242
 - analytical data, 236
 - categories, 239–241
 - data domains, 236
 - Data Mesh solution pattern, 241
 - differentiation, 237
 - MDM pattern, 242
 - orchestration, 237
 - requirements, 237
 - transactional landscape,
 - 237, 242
 - user landscape, 238
 - users, 238
- Data credibility, 306
- Data curation, 95
- Data domains, 203
- Data-driven business, 43
- Data-driven decisions, 357
- Data engineering team, 211
- Data engineers, 207
- Data exchange, 261
- Data exploration, 131
- Data Fabric, 3–5, 13, 14, 17
 - active metadata, 396
 - AI governance, 394, 397
 - AI-infused, 233, 234, 253
 - AI-infused Data Fabric, 23–25
 - AI-infused system, 17
 - AI/ML-based augmentation, 20
 - aspects, 234
 - automated action, 253
 - automated metadata
 - enrichment, 395
 - automated workload, 396
 - automate quality
 - analysis, 346–350
 - automatic enforcement, data
 - regulations, 344, 345
 - characteristics, 19
 - cloud data integration, 394
 - customer 360-degree
 - insight, 394
 - data architecture evolution, 20
 - and Data Mesh evolution,
 - 232, 233

- and Data Mesh
 - relationship, 30–32
- Data Mesh solution, 236
- data product, 32, 396
- data to AI artefacts, 234
- definitions, 19
- digital exhaust, 397
- explainable AI, 396
- framework, 21
- implementation, 231
- industry needs, 235, 236
- insight to automated action, 235
- intelligent information
 - integration styles, 254
- knowledge catalog, 247, 254
- MLOps, 395
- NetApp description, 18
- patterns, 231, 253
- scope, 233
- self-service, 243, 395
- solutions, 233
- tasks, 233
- tasks and capabilities, 21, 22
- technologies, 231, 234, 253
- technology scope, 235, 236
- Data Fabric architecture, 151, 154, 254, 273
 - AI techniques, 376
 - automated regulation, 342, 343
 - AWS, 361
 - Azure, 363
 - data and AI governance, 341, 342
 - data products, 274
 - DataOps, 225
 - Denodo, 366–370
 - deployment options, 289
 - determination, 231
 - goal, 205, 356
 - governance/knowledge/
 - semantics layer, 268, 269
 - IBM Cloud Pak for
 - Data, 358–361
 - integration, 231
 - integration/transformation
 - layer, 269, 270
 - key features, 356
 - knowledge catalog, 204
 - layers, 268
 - mechanism, 289
 - orchestration/lifecycle layer, 271
 - scope, 243
 - self-service layer, 270
 - strength, 268
 - technologies, 273
- Data Fabric/Data Mesh
 - AI and MLOps, 45
 - AI capabilities
 - bias, 66
 - drift, 66
 - explainability, 67
 - quality metrics, 67
 - customer, 45, 60, 360
 - data and AI identification, 51
 - data assessment, 52
 - data communication, 52
 - data governance and privacy, 44, 50, 51
 - data movement, 49

INDEX

- Data Fabric/Data Mesh (*cont.*)
 - deploy AI models, 49, 50
 - hybrid cloud data
 - integration, 44
 - MLOps, 65, 66
 - rules and policies, 51
 - task automation, 49
- Data Fabric integration, 385
- Data federation, 251
- Data fragmentation
 - credit card loans, 208
 - credit data pipelines, 209, 210
 - credit score, 208
 - data pipelines, 208
 - data preview, 207, 208
 - data quality, 207
 - implementation, 211
 - industries, 206
 - interest rate, 208
 - issues, 206
 - ML models, 209
 - mortgage application, 206, 207
 - organizations, 205, 206, 209, 211
 - restrictions, 206
 - technical challenges, 205
- Data governance, 50
- Data gravity, 222
- Data identification, 58
- Data integration, 5, 48, 266, 356
- Data/knowledge catalog
 - additional insight, 48
 - AI governance, 48
 - data lineage, 47, 48
 - store metadata, 47
- Data lake architecture, 57
- Data lakehouse, 4, 12, 13
- Data lakes, 4, 10–12, 14
- Data landscape, 45
- Data latency, 8, 9, 14
- Data lineage, 47
 - concern, 306
 - data lifecycle, 306
 - definition, 309
 - details, 307
 - diagram, 307
 - enterprises, 308
 - five Ws, 309
 - nodes, 307
 - users, 308
 - workflow, 306
- Data management, 334, 350
- Data management
 - functions, 19
- Data management policies, 58
- Data Map, 365
- Data Mesh, 3–5, 13, 17
 - AI artefacts, 28
 - business-focused data
 - products, 18
 - business requirements, 31
 - and Data Fabric
 - relationship, 30–32
 - definitions, 26
 - phrases, 18
- Data Mesh approaches, 357
- Data Mesh areas, 153
- Data Mesh semantic enrichment
 - engine, 379

- Data Mesh solution, 26–28, 168, 356, 375, 376
 - architecture overview diagram, 248, 249
 - data consumers, 248, 249
 - data consumption
 - pattern, 248
 - data domain owners, 248
 - Data Fabric capabilities, 248
 - data products, 249
- Data Fabric capabilities, 243
- data marketplace, 248
- DataOps, 225
- data products, 253
- enabling, 248
- goals, 205
- scenarios, 290
- self-service capabilities, 243–245, 253
 - business definition/scoping, 247
 - business glossary
 - integration, 245
 - business knowledge
 - graph, 246
 - data product
 - management, 247
 - data products, 246, 247
 - domain-specific, 246
 - ontology/taxonomy
 - integration, 245
 - semantic enrichment, 245
- tools/services, 247
- Data mining, 321
- DataOps
 - aspects, 203, 204
 - design, 200
 - implementation, 201
 - orchestration, 204
 - output, 202
 - stages, 201
- Data preprocessing, 182, 185
- Data privacy, 338
- Data product, 31, 393
 - access methods, 35
 - characteristics, 34
 - consumption in data marketplace, 34
 - consumption-ready, 36
 - data-as-a-product, 37
 - data domain owner and product owner, 32
 - data domain owner responsibilities, 32
 - data product owner responsibilities, 33
 - defined format, 35
 - description, 34
 - high-level depiction, 32
 - knowledge catalog, 36
 - marketing/customer care organization, 31
 - metadata exchange, 37, 38
 - policies and rules, 35
 - SLAs, 35
- Data product owner, 31–34
- Data provenance, 92, 155, 306
- Data quality assessment, 324

INDEX

- Data quality dimensions, 327, 351
 - accuracy, 347
 - completeness, 348
 - consistency, 347
 - data class, 348, 349
 - timeliness, 347
 - uniqueness, 348
 - validity, 348
- Data quality issue corrections, 387
- Data refresh adjustments, 388
- Data replication, 5
- Data science team, 207, 211, 213
- Data scientists, 112, 195, 197–199, 202, 213, 226, 302
- Data security, 339, 340
- Datasets, 198
- Data transformation, 132
- Data virtualization, 58, 251
- Deep learning (DL), 128, 378
- Deep Neural Network (DNN), 377
- Denodo
 - active metadata supports, 367
 - cloud services, 368
 - comprehensive metadata, 366
 - data catalog, 367
 - data discovery capabilities, 366
 - data sources supports, 366
 - data virtualization, 366–368
 - definition, 366
 - direct connection supports, 366
 - on-premises version, 368
 - self-service business glossary, 367
 - semantic layer, 366
 - virtualization technology, 366
- Deployment architecture, 226
- Deployment patterns
 - cloud-native, 222
 - advantages, 223
 - applications, 222
 - consideration factors, 223
 - data, 222
 - training phase, 222
 - edge deployment, 223, 224
 - factors, 219, 220
 - runtime environment, 220, 221
 - application, 221
 - core idea, 222
 - drawback, 222
 - inference requests, 221
 - integration, 221
 - masking, data, 220
 - model versions, 221
 - training data, 219, 220
- Deploy models, 199
- Developers, 238
- DevOps
 - model inference, 203
 - practices, 203
 - stages, 201
 - systematic software
 - development, 200
- Digital business
 - transformation, 278
- Digital exhaust, 169, 170
- Digital footprint, 168
- Digitalization, 397
- Distributed data
 - management, 357

Distributed File System Custom Wrapper, 366
 DWH/BI report adjustments, 388

E

Edge computing, 4
 Edge fabric, 376
 End customers, 238
 Enterprise architecture
 application architecture
 level, 261
 business architecture level, 260
 CP4D, 271
 data architecture level, 261
 Data Fabric architecture,
 272, 273
 data structure types, 272
 decision criteria, 262
 delivery system architecture
 level, 262
 development, 259
 feedback loop, 260
 hybrid cloud, 272
 IBM Cloud Pak, 272
 information architecture
 level, 261
 integration, 272
 IT ecosystem, 259
 layers, 259
 levels, 260
 mistake, 262
 variations, 272
 Watson Studio pipelines, 272

Enterprise Data Warehouse (EDW)
 architecture, 7-10
 Enterprise knowledge
 catalog, 297
 Enterprise master data
 cross-selling, 178
 customers, 178
 suppliers, 178
 upselling, 178
 Enterprises, 195, 199, 203, 215, 216,
 219-222, 293, 338, 341
 Enterprise-wide Data Fabric
 architecture, 360
 Entity matching, 179-181
 AI-based, 187-189
 MDM, 189, 192
 ML method, 185
 reference model, 182
 Extract-Transform-Load (ETL)
 pipelines, 5, 8, 9, 14, 156,
 157, 385

F

False negative rate (FNR), 113
 False positive rate (FPR), 113
 Favorable outcome, 106
 Feature construction, 133
 Feature engineering, 132, 217
 Feature extraction, 133
 Feature selection, 132
 Financial institutions, 220
 Financial service, 130
 Foundational technology, 181

INDEX

G

General Data Protection

Regulation (GDPR), 47,
342, 343

Glossary matching, 383

Governance

artifacts, 343, 344

Governance rules, 344

Governance staff, 238

Govern models, 199

Graph databases (GDBs), 97

H

Hadoop Distributed File System (HDFS), 10

Harmonic mean, 114

Hash-based blocking, 183

Heterogeneity, 289

Heterogeneous data, 357

Hybrid cloud, 55–57, 59, 278, 289

AI-infused entity matching, 62

building applications, 280

customer profile, 62

DaaS, 63

Data Fabric architecture

benefits, 288, 289

catalogs, 283

central/distributed

cataloging, 282, 283

data access/consumption, 284

data management layer, 282

DataOps/ModelOps, 283

data standards, 282

deployment options, 282, 283

functions, 283

knowledge catalog, 282

data identification, 62

data integration, 57

Data Mesh solution, 285

approaches, 284

benefits, 288, 289

cross-Lob, 287

data marketplace, 284

data product owners, 285

data products, 285

domain owners *vs.* LoBs, 285

knowledge catalogs, 284

metadata, 285

public service providers, 287

requirements, 287

scenarios, 285, 286

specification, 285

organizations, 278

policies and rules, 63

Hybrid cloud environments, 393

Hyper-automated data and

AI Fabric

AI/ML technology, 387

capabilities, 386, 387

challenges, 388

core research, 388

Data Fabric-related

processes, 388

Data Fabric architectural

components, 387

Hyper-automated Data Fabric, 375

Hyper-automation, 386

I

- IBM Cloud Pak for Data, 371
 - built-in semantic models, 360
 - data connection, 358
 - extensive data sources
 - supports, 359
 - IBM's implementation, 358
 - IBM Watson Knowledge Catalog, 360
 - on-premises software
 - version, 360
 - self-service and intelligent governance, 359
 - z/OS Data Gate, 358
- IBM DataStage, 358
- IBM Watson Knowledge Catalog, 38, 359, 360
- IBM Watson Query, 358
- IBM Watson Studio, 272
- IBM zSystems
 - data consumption, 53
 - governance artefacts, 54
 - knowledge catalog, 54
 - rules and policies, 54
- Independent Component Analysis (ICA), 133
- Industries, 206
- Industry-specific ontologies, 379
- Informatica
 - AI/ML technologies, 370
 - business-relevant policies, 370
 - catalog, 370
 - Data Ingestion and Data Integration, 369
 - data integration, 369
 - data quality, 369
 - dictionary generation, 369
 - leading providers, 368
 - on-premises and cloud deployments, 370
- Information integration, 115
- Institute for Business Value (IBV), 64
- Insurance companies, 72
- Insurance database, 180
- Intelligent automated metadata, 313
- Intelligent automation, metadata
 - annotation, 321–324
 - automated metadata enrichment, 314
 - automated metadata generation, 314
 - automated tagging, 321–324
 - data analysis and profiling, 316–321
 - extraction and exploitation, 315
 - labeling, 321–324
 - phases, 316
 - sample knowledge catalog, 313
- Intelligent cataloging
 - AI algorithms, 301
 - business term, 302
 - capabilities, 308
 - column format, 300
 - data discovery, 298–300, 308
 - data enrichment, 298, 299, 301, 302, 308
 - data lineage, 299

INDEX

Intelligent cataloging (*cont.*)

- data provenance, 299
- data sources, 299
- incoming data, 298
- library catalog, 300
- ML algorithms, 301
- NLP technology, 301
- recommendations, 299, 305
- semantic search, 298, 303
- subject matter experts, 301
- technical metadata, 299
- transaction, 301, 302
- volume of data, 300

Intelligent enrichment, 100

Intelligent governance, 357

Intelligent information

- integration, 116

Intelligent information

- integration styles

AI-based capabilities, 252

AI-infused intelligence/
automation, 252

AI-infused layer, 250

methods, 231

prerequisites, 250, 251

Internet of Things (IoT), 4

Interpretability, 377

Interpretation algorithms, 377, 378

IT staff, 238

J

JDBC/ODBC, 252

Jupyter notebooks, 215, 217

K

Knowledge catalog, 204

active metadata

AI/ML capabilities, 94

business, 91

operational, 92, 93

technical, 92

data curation, 95

business terms, 95

create metadata, 95

enriching metadata, 96

governance artefacts, 96

quality assessments, 96

refine assets, 96

self-service capabilities,

101, 102

semantic network, 96, 97

knowledge graph, 97–100

Knowledge graph inclusion, 382

L

Labeling, 321–324

Library

catalog, 297

Linear Discriminant Analysis

(LDA), 133

Lines of business (LoBs), 18

Linguistic analysis, 383, 384

Local Interpretable Model-

Agnostic Explanations

(LIME), 111

Logarithmic loss, 114

M

- Machine learning (ML), 123
- MapReduce, 10
- Master data, 340
- Master Data Management (MDM), 19
- Matching attributes, 165
- Matching decisions, 186
- Mean absolute error (MAE), 135
- Mean squared error (MSE), 135, 217
- Messaging, 252
- Metadata, 340
 - business metadata, 295, 296, 308
 - definition, 294, 308
 - hierarchy, 297
 - insight, 297
 - management, 297
 - operational metadata, 296
 - personal profile, 294, 295
 - quality, 297
 - self-introductions, 294
 - technical metadata, 296, 308
- Metadata analysis, 383
- Metadata exchange, 37
- Microservices, 251, 264, 265, 274
- Microservices architecture, 57
- Microsoft Azure
 - Data Fabric capabilities, 365
 - description, 363
 - integration services, 363
 - self-services, 364
 - services, 364
 - SQL technologies, 364
 - Microsoft Azure Synapse Analytics, 364
 - Microsoft Purview, 363, 365
 - Microsoft Purview Data Estate Insights, 365, 366
 - Microsoft Purview Data Lifecycle Management, 365
 - Microsoft Purview Data Map, 363
 - Microsoft Purview Data Sharing, 365
 - Miss rate, 113
 - ML algorithms, 166, 385
 - ML-based query optimization, 150
 - ML-based trustworthy AI approaches, 377
 - ML/DL-based assignment, 157
 - ML/DL models, 154, 155
 - ML-infused confidence-based query matching, 163
 - ML-infused data, 162
 - ML-infused entity matching techniques, 151
- MLOps
 - aspects, 203, 204
 - AutoAI, 217, 219, 226
 - automation, 217
 - data preparation, 216
 - data transformation, 217
 - definition, 215
 - employs tools, 202
 - encoding/scaling, 216
 - future selection, 216
 - goal, 202
 - metrics, 217

INDEX

MLOps (*cont.*)

- missing/inaccurate values, 216
- model creation, 216
- vs.* ModelOps, 202, 225
- success criteria, 216
- tasks, 202

Model deployment, 136

ModelOps

- aspects, 203, 204
- business application, 202
- and DataOps, 201
- framework, 201
- stages, 201
- tasks, 202

Module identification, 384

MongoDB Enterprise Advanced in data management, 360

Monitor models, 199

Mortgage business, 211

Most Azure data sources, 363

Multi-class classification, 217

Multiclass classification and regression, 377

N

National Institute of Standards and Technology (NIST), 259

Natural Language Generation (NLG), 139

Natural Language Processing (NLP), 139, 301

- audio transcript, 141
- grammar/spelling mistakes, 142

machine translation, 141

sentiment analysis, 142

text summarization, 142

virtual agents/chatbots, 141

Natural Language Understanding (NLU), 124, 139

NetApp, 18

Neural network, 187

O

Omni-channel environments, 61

On-premises IT infrastructure, 280

Ontology-based semantic knowledge graphs, 378

Ontology-based semantic searches, 375, 376, 378, 379

Ontology mapping, 382

Ontology Web Language (OWL), 378

Open Data Platform initiative (ODPi) Egeria, 37

Open Metadata and Governance (OMAG), 37

Open Metadata Repository Services (OMRS), 37

Open Neural Network eXchange (ONNX), 137

Operationalizing AI

- AI bias, 213
- challenges, 213, 215
- fact sheets, 215
- fairness, 213
- integration, 211
- model lineage analysis, 215

- mortgage business, 211
- organizations, 213
- pillars, 225
- prediction, plain English, 212, 213
- solution, 215
- status of models, 213–215

Operational metadata, 92, 93

Orchestration, 204

Organizational Data Mesh

- solutions, 360

Organizations, 205, 258

Overfitting, 135

P

Palantir for infusing AI into

- decision-making, 360

Passive metadata, 94

Perfect equality, 106

Personal Confidential Information (PCI), 295

Personal data, 345

Personally Identifiable Information (PII), 345

Portable Format for Analytics (PFA), 137

Predictive maintenance, 129

Predictive Model Markup Language (PMML), 137

Pre-training models, 140

Principal Component Analysis (PCA), 133

Privacy laws and regulations, 344

Probabilistic matching, 184

Process adjustment, 385

Process metadata, 92

Property management, 340

Public cloud providers, 279, 282, 283, 285

Python Scikit-learn models, 137

Q

Quality analysis

- CREDITTRANS, 347

Quality scores, 161

R

RBFOpt, 217

Receiver Operating Characteristic (ROC), 98, 135

Record-level threshold, 184

Recurrent neural networks (RNNs), 187

Red Hat OpenShift container platform, 360

Reference data, 340

Regression problem, 129

Regulation-specific ontologies, 382

Regulatory compliance, 336

Reinforcement learning, 127, 128

Relational Database Management Systems (RDBMS), 4–6

Resource Description Framework (RDF), 378

REST API, 251

Retail database, 180

INDEX

Robotic Process Automation
(RPA), 388
Root mean squared error (RMSE),
115, 217
Rule-based matching, 188
Rule-based method, 184
Rule-based systems, 124

S

Self-built systems, 181
Self-correction capabilities, 108
Self-interpretable models, 378
Self-Organizing Map (SOM), 185
Self-service, 356–358
Self-service capabilities, 90
Self-service fashion, 359
Semantic enrichment, 170–172
Semantic knowledge graphs,
376–379, 382
Semantic metadata, 172
Semantic search, 159, 303
 aims, 309
 business terms, 304, 305
 contact information, 303, 304
 data assets, 303
 keywords, 302, 304
 restaurant website, 303
 search entities, 303
 semantic query, 303
 specific context, 303
 supermarket app, 303
SHapley Additive exPlanations
(SHAP), 112

Shapley value, 112
Shopping-for-data, 17, 18, 23, 26,
28, 33, 103
Similarity computation, 186
Similarity discovery
 algorithms, 167
Simple Protocol and RDF Query
 Language (SPARQL), 378
Social Security number (SSN), 295
Software as a Service (SaaS), 386
Sophisticated automated AI
 governance scope, 380
Sort-based blocking, 183
Spectrum, 231
Stepwise approximation plus
 search strategy, 196
Store metadata, 47
Streaming, 251
Streaming API, 138
Structured Query Language (SQL),
4, 6, 9, 14
Supervised Learning, 125, 126
Support vector machine
(SVM), 186

T

Technical benefits
 accurate insight, 80
 compliance and security, 80
 self-service data shopping, 80
 spent time, 81
 value proposition, 82
Technical metadata, 92

Technical teams
 data access management, 77
 data delivery, 76
 infrastructure and storage, 78
 KPI, 78
 quality standards, 77, 78
 value proposition, 79

Telecommunications, 130

Text analysis, 383, 384

Threshold sensitivity, 166

Traditional data governance
 initiatives, 357

Traditional software, 196, 225

Transformation methods
 categorical values, 132
 missing values, 132
 outliers and duplicates, 132
 various date formats, 132

True negative rate (TNR), 105, 113

True positive rate (TPR), 105, 113

Trustworthy AI, 24, 377

Trustworthy and
 explainable AI, 377

U, V

Unstructured data, 140

Unsupervised learning, 126,
 127, 186

User communication channels, 263

W, X

Watson Knowledge Catalog
 (WKC), 272

Weighted false positive rate
 (wFPR), 115

Weighted true positive rate
 (wTPR), 115

Y, Z

Yet Another Resource Negotiator
 (YARN), 10