O'REILLY®

Data Mesh

Delivering Data-Driven Value at Scale

Free Chapter

Zhamak Dehghani

Data Mesh

Delivering Data-Driven Value at Scale

This excerpt contains Chapter 1. The complete book is available on the O'Reilly Online Learning Platform and through other retailers.

Zhamak Dehghani

Beijing • Boston • Farnham • Sebastopol • Tokyo O'REILLY®

Data Mesh
by Zhamak Dehghani

Printed in the United States of America.

Published by O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472.

O'Reilly books may be purchased for educational, business, or sales promotional use. Online editions are also available for most titles (*http://oreilly.com*). For more information, contact our corporate/institutional sales department: 800-998-9938 or *corporate@oreilly.com*.

Acquisitions Editor: Melissa Duffield
Development Editor: Gary O'Brien
Production Editor: Beth Kelly
Copyeditor: Charles Roumeliotis
Proofreader: Kim Wimpsett

Indexer: Potomac Indexing, LLC
Interior Designer: David Futato
Cover Designer: Karen Montgomery
Illustrator: Kate Dullea

March 2022: First Edition

Revision History for the First Edition
2022-03-08: First Release

See *http://oreilly.com/catalog/errata.csp?isbn=9781492092391* for release details.

978-1-098-11276-9

[LSI]

Table of Contents

CHAPTER 1
Data Mesh in a Nutshell

> *"Think in simples" as my old master used to say—meaning to reduce the whole to its parts in simplest terms, getting back to first principles.*
>
> —Frank Lloyd Wright

Data mesh is a decentralized sociotechnical approach to share, access, and manage analytical data in complex and large-scale environments—within or across organizations.

Data mesh is a new approach in sourcing, managing, and accessing data for analytical use cases at scale. Let's call this class of data analytical data. Analytical data is used for predictive or diagnostic use cases. It is the foundation for visualizations and reports that provide insights into business. It is used to train machine learning models that augment business with data-driven intelligence. It is the essential ingredient for organizations to move from intuition and gut-driven decision making to taking actions based on observations and data-driven predictions. Analytical data is what powers the software and technology of the future. It enables a technology shift from human-designed rule-based algorithms to data-driven machine-learned models. Analytical data is becoming an increasingly critical component of the technology landscape.

The term *data* in this book, if not qualified, refers to analytical data. Analytical data serves reporting and machine learning training use cases.

The Outcomes

To get value from data at scale in complex and large-scale organizations, data mesh sets to achieve these outcomes:

- Respond gracefully to change: a business's essential complexity, volatility, and uncertainty
- Sustain agility in the face of growth
- Increase the ratio of value from data to investment[1]

The Shifts

Data mesh introduces multidimensional technical and organizational shifts from earlier analytical data management approaches.

Figure 1-1 summarizes the shifts that data mesh introduces, compared to past approaches.

Data mesh calls for a fundamental shift in the assumptions, architecture, technical solutions, and social structure of our organizations, in how we manage, use, and own analytical data:

- *Organizationally*, it shifts from centralized ownership of data by specialists who run the data platform technologies to a decentralized data ownership model pushing ownership and accountability of the data back to the business domains where data is produced from or is used.
- *Architecturally*, it shifts from collecting data in monolithic warehouses and lakes to connecting data through a distributed mesh of data products accessed through standardized protocols.
- *Technologically*, it shifts from technology solutions that treat data as a byproduct of running pipeline code to solutions that treat data and code that maintains it as one lively autonomous unit.
- *Operationally*, it shifts data governance from a top-down centralized operational model with human interventions to a federated model with computational policies embedded in the nodes on the mesh.
- *Principally*, it shifts our value system from data as an asset to be collected to data as a product to serve and delight the data users (internal and external to the organization).

1 XREF HERE unpacks the expected outcomes of data mesh, with a high level of description of how it achieves those outcomes.

- *Infrastructurally*, it shifts from two sets of fragmented and point-to-point integrated infrastructure services—one for data and analytics and the other for applications and operational systems to a well-integrated set of infrastructure for both operational and data systems.

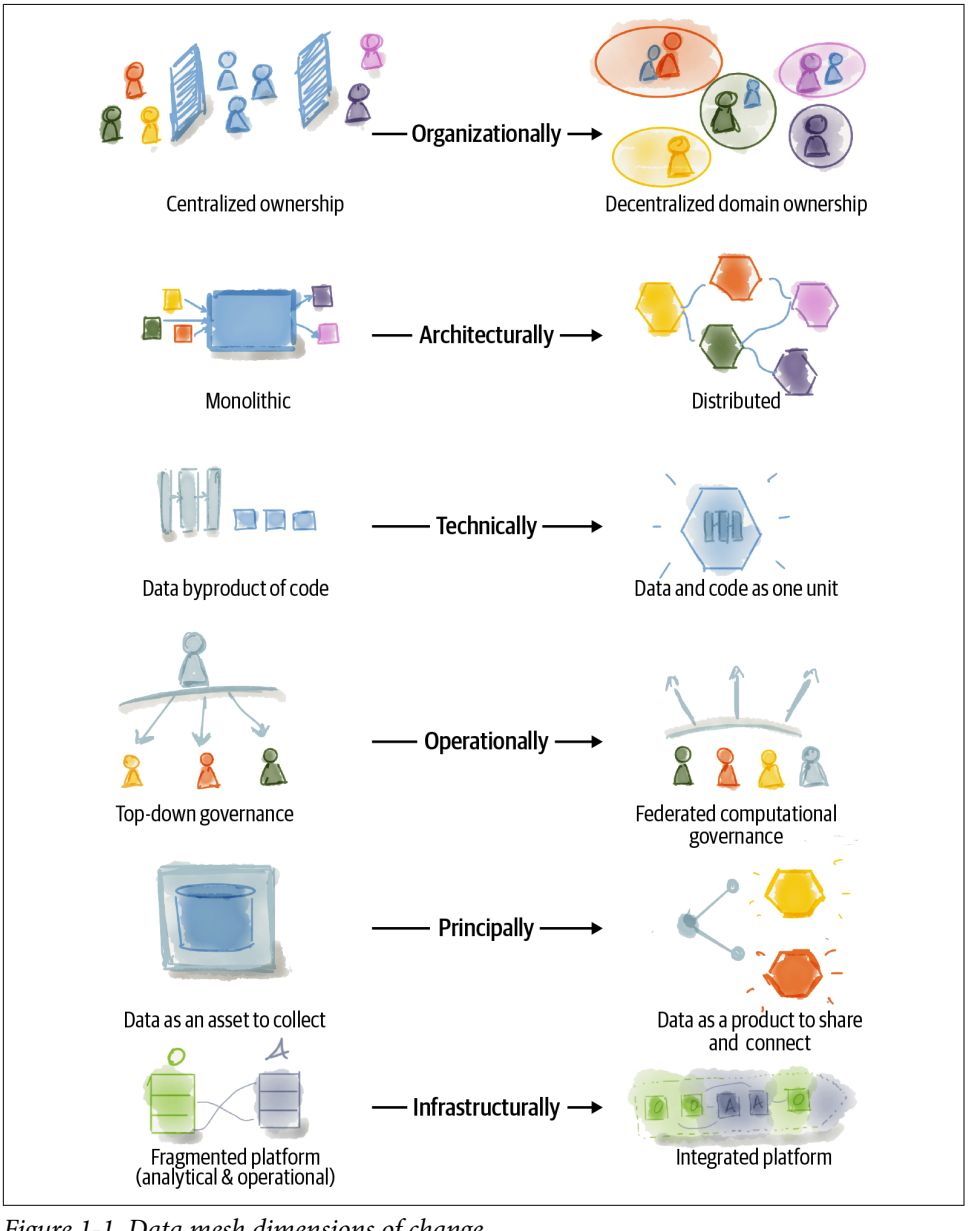

Figure 1-1. Data mesh dimensions of change

Since the introduction of data mesh in my original blog post (kindly hosted by Martin Fowler), I have noticed that people have struggled to classify the concept. Is data mesh an architecture? Is it a list of principles? Is it an operating model? After all, we rely on the *classification of patterns*[2] as a major cognitive function to understand the structure of our world. Hence, I have decided to classify data mesh as a *sociotechnical* paradigm: an approach that recognizes the interactions between people and the technical architecture and solutions in complex organizations. This is an approach to data management that not only optimizes for the technical excellence of analytical data sharing solutions but also improves the experience of all people involved: data providers, users, and owners.

Data mesh can be utilized as an element of an enterprise *data strategy*, articulating the target state of both the *enterprise architecture* and an *organizational operating model* with an iterative execution model.

In the simplest form it can be described through four interacting principles. In this chapter I give a very brief definition of these principles and how they work together.

The Principles

Four simple principles can capture what underpins data mesh's logical architecture and operating model. These principles are designed to progress us toward the objectives of data mesh: increase value from data at scale, sustain agility as an organization grows, and embrace change in a complex and volatile business context.

Here is a quick summary of the principles.

Principle of Domain Ownership

Decentralize the ownership of analytical data to business domains closest to the data—either the source of the data or its main consumers. Decompose the (analytical) data logically and based on the business domain it represents, and manage the life cycle of domain-oriented data independently.

Architecturally and organizationally align business, technology, and analytical data.

The motivations of domain ownership are:

- The ability to scale out data sharing aligned with the axes of organizational growth: increased number of data sources, increased number of data consumers, and increased diversity of data use cases

2 Jeff Hawkins and Sandra Blakeslee (2005). *On Intelligence* (p. 165). New York: Henry Holt and Co.

- Optimization for continuous change by localizing change to the business domains
- Enabling agility by reducing cross-team synchronizations and removing centralized bottlenecks of data teams, warehouses, and lake architecture
- Increasing data business truthfulness by closing the gap between the real origin of the data, and where and when it is used for analytical use cases
- Increasing resiliency of analytics and machine learning solutions by removing complex intermediary data pipelines

Principle of Data as a Product

With this principle in place, domain-oriented data is shared as a product directly with data users—data analysts, data scientists, and so on.

Data as a product adheres to a set of usability characteristics:

- Discoverable
- Addressable
- Understandable
- Trustworthy and truthful
- Natively accessible
- Interoperable and composable
- Valuable on its own
- Secure

A data product provides a set of explicitly defined and easy to use data sharing contracts. Each data product is autonomous, and its life cycle and model are managed independently of others.

Data as a product introduces a new unit of logical architecture called *data quantum*, controlling and encapsulating all the structural components needed to share data as a product—data, metadata, code, policy, and declaration of infrastructure dependencies—autonomously.

The motivations of data as a product are to:

- Remove the possibility of creating domain-oriented data silos by changing the relationship of teams with data. Data becomes a product that teams share rather than collect and silo.
- Create a data-driven innovation culture, by streamlining the experience of discovering and using high-quality data, peer-to-peer, without friction.

- Create resilience to change with built-time and run-time isolation between data products and explicitly defined data sharing contracts so that changing one does not destabilize others.
- Get higher value from data by sharing and using data across organizational boundaries.

Principle of the Self-Serve Data Platform

This principle leads to a new generation of self-serve data platform services that empower domains' *cross-functional* teams to share data. The platform services are centered around removing friction from the end-to-end journey of data sharing, from source to consumption. The platform services manage the full life cycle of individual data products. They manage a reliable mesh of interconnected data products. They provide mesh-level experiences such as surfacing the emergent knowledge graph and lineage across the mesh. The platform streamlines the experience of data users to discover, access, and use data products. It streamlines the experience of data providers to build, deploy, and maintain data products.

The motivations of the self-serve data platform are to:

- Reduce the total cost of decentralized ownership of data.
- Abstract data management complexity and reduce the cognitive load of domain teams in managing the end-to-end life cycle of their data products.
- Mobilize a larger population of developers—technology generalists—to embark on data product development and reduce the need for specialization.
- Automate governance policies to create security and compliance standards for all data products.

Principle of Federated Computational Governance

This principle creates a data governance operating model based on a federated decision-making and accountability structure, with a team composed of domain representatives, data platform, and subject matter experts—legal, compliance, security, etc. The operating model creates an incentive and accountability structure that balances the autonomy and agility of domains, with the global interoperability of the mesh. The governance execution model heavily relies on codifying and automating the policies at a fine-grained level, for every data product, via the platform services.

The motivations of federated computational goovernance are:

- The ability to get higher-order value from aggregation and correlation of independent yet interoperable data products

- Countering the undesirable consequences of domain-oriented decentralizations: incompatibility and disconnection of domains
- Making it feasible to build in cross-cutting governance requirements such as security, privacy, legal compliance, etc., across a mesh of distributed data products
- Reducing the overhead of manual synchronization between domains and the governance function

Interplay of the Principles

I intended for the four principles to be collectively necessary and sufficient. They complement each other, and each addresses new challenges that may arise from others. Figure 1-2 shows the interplay of the principles.

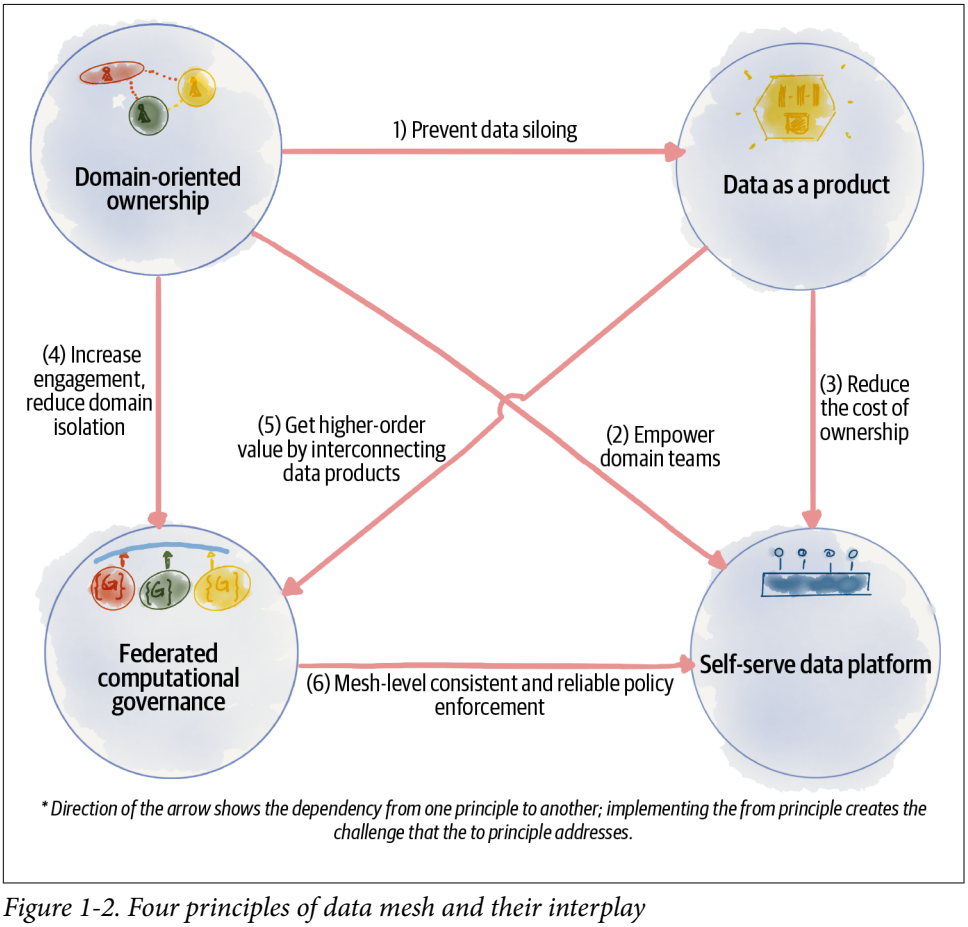

Figure 1-2. Four principles of data mesh and their interplay

For example, decentralized domain-oriented ownership of data can result in data siloing within domains, and this can be addressed by the data as a product principle that demands domains have an organizational responsibility to share their data with product-like qualities inside and outside of their domain.

Similarly, the domain ownership of data products can lead to duplicated effort, increased cost of data product ownership, and lowered data sharing productivity. In this case, the self-serve data platform empowers the cross-functional domain teams in sharing and using data products. The platform objective is to lower the domain teams' cognitive load, reduce unnecessary effort, increase domains' productivity, and lower the total cost of ownership.

Data Mesh Model at a Glance

Operationally you can imagine the principles in action as demonstrated in Figure 1-3.

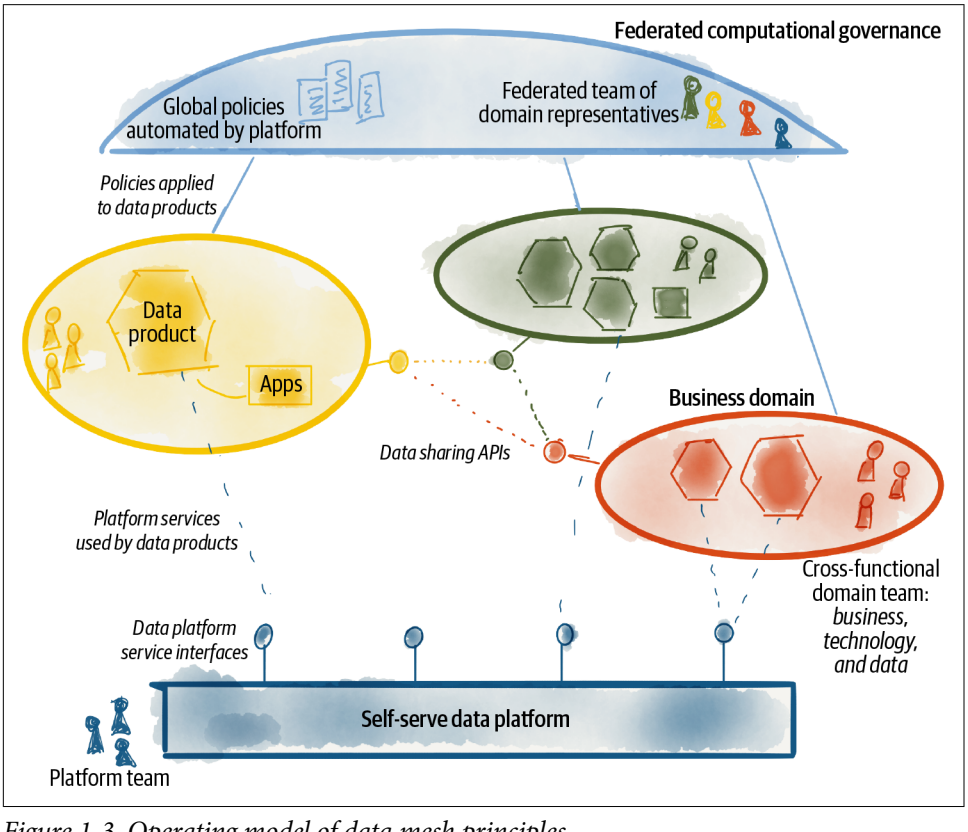

Figure 1-3. Operating model of data mesh principles

The domains with cross-functional teams are achieving the business domain's goals with digital applications and data products. Each domain shares its data and services through contracts. Data products can be composed and owned by new domains. The global policies are defined by a federated group composed of representatives of the domains. The policies along with other platform services are offered as automated capabilities.

This is a simplified operating model of data mesh.

The Data

Data mesh focuses on analytical data. It recognizes the blurry delineation of the two modes of data, introduces a new model of tight integration of the two, and yet respects the clear differences between them.

"What is operational data versus analytical data?" This has been a point of confusion for early enthusiasts of data mesh. Allow me to clarify what I mean by these terms.

Operational Data

Operational data supports running the business and keeps the current state of the business with transactional integrity. This data is captured, stored, and processed by transactions in real time, by OLTP (online transaction processing) systems.

Operational data sits in databases of microservices, applications, or systems of records that support business capabilities. It keeps the current state of the business.

Operational data modeling and storage are optimized for application or microservice logic and access patterns. It is constantly updated, with read and write access to it. Its design has to account for multiple people updating the same data at the same time in unpredictable sequences, which results in a need for transactions. The access is also about relatively in-the-moment activity.

Operational data is referred to as "data on the inside". It is the private data of an application or a microservice that performs CRUD (create, update, delete) operations on it. Operational data can be intentionally shared on the outside through APIs—e.g., REST, GraphQL, or events. The operational data on the outside has the same nature as the operational data on the inside: it is what we know about the business, *now*.

Operational data is recording what happens in the business, supporting decisions that are specific to the business transaction. In short, *operational data is used directly to run the business and serve the end users*.

Imagine Daff. Its **listener registration** service implements the business function of subscribing new users or unsubscribing them. The transactional database that supports the registration process and keeps the current list of subscribers is considered operational data.

Today, operational data is collected and then transformed into analytical data. Analytical data trains the machine learning models that then make their way into the operational systems as intelligent services.

Analytical Data

This is the historical, integrated, and aggregate view of data created as the byproduct of running the business. It is maintained and used by OLAP (online analytical processing) systems.

Analytical data is the temporal, historic, and often aggregated view of the facts of the business over time. It is modeled to provide retrospective or future-perspective insights. Analytical data is optimized for analytical logic—training machine learning models and creating reports and visualizations. Analytical data is part of the "data on the outside" category, data directly accessed by analytical consumers.

Analytical data has a sense of history. Analytical use cases require looking for comparisons and trends over time, while a lot of operational uses don't require much history.

Analytical access mode tends to include intensive reads across a large body of data, with fewer writers. The original definition of analytical data as *a nonvolatile, integrated, time-variant collection of data*[3] still remains valid.

In short, analytical data is used to *optimize* the business and user experience. This is the data that fuels the organization's AI and analytics aspirations.

For example, in the case of Daff it's important to optimize the listeners' experience with playlists recommended based on their music taste and favorite artists. The analytical data that helps train the playlist recommendation machine learning model captures all the past behavior of the listener as well as all characteristics of the music the listener has favored. This aggregated and historical view is analytical data.

Today, analytical data is stored in a data warehouse or lake.

3 Definition provided by William H. Inmon, known as the father of data warehousing.

The Origin

> To reject one paradigm without simultaneously substituting another is to reject science itself.
>
> —Thomas S. Kuhn, *The Structure of Scientific Revolutions*

Thomas Kuhn, an American historian and philosopher of science, introduced the *paradigm shift* in his at the time rather controversial book, *The Structure of Scientific Revolutions* (1962). He observed how science progressed in two main modes: *incremental* and *revolutionary*; science progressed through long stretches of legato *normal science* where the existing theories form the foundation of all further research, followed by the occasional disruption of staccato paradigm shifts that challenged and transcended the existing knowledge and norm. For example, the progress of science from *Newtonian mechanics* to *quantum mechanics* is considered a paradigm shift as scientists could no longer explain the governing laws of physics at the quantum level with the existing theories. Kuhn recognized that a prerequisite for a paradigm shift is identifying *anomalies*, observations that don't fit the existing norm, and entering the phase of *crisis*, questioning the validity of the existing paradigm in solving the new problems and observations. He also observed that people try, with increasing desperation, to introduce unsustainable complexities into the existing solutions to account for anomalies.

This almost perfectly fits the origin of data mesh and its principles. It came from the recognition of anomalies—failure modes and accidental complexities that I describe in XREF HERE—and moments of crisis where the characteristics of the existing data solutions didn't quite fit the realities of enterprises today. We are in a moment of Khunian crisis in the progression of our approach for data. Hence, there is a need for a new paradigm.

I wish I could claim that data mesh principles were novel and new and I cleverly came up with them. On the contrary, the principles of data mesh are a generalization and adaptation of practices that have evolved over the last two decades and proved to solve our last complexity challenge: scale of *software complexity* led by the *mass digitization of organizations*.

These principles are the foundation of how digital organizations have solved organizational growth and complexity, while delivering unprecedented digital aspirations: moving all of their services to the web, using mobile for every single touchpoint with their customers, and reducing organizational synchronizations through automation of most activities. They are an adaptation of what formulated the previous paradigm shift in software: the microservices and APIs revolution, platform-based Team Top-

ologies,[4] computational governance models such as Zero Trust Architecture,[5] and operating distributed solutions securely and across multiple clouds and hosting environments. In the last several years, these principles have been refined and adapted to the analytical data problem space.

Let's look more closely at each of the data mesh principles.

4 Matthew Skelton and Manual Pais (2019). *Team Topologies: Organizing Business and Technology Teams for Fast Flow*. Portland, OR: IT Revolution.

5 Scott W. Rose, Oliver Borchert, Stuart Mitchell, and Sean Connelly (2020). "Zero Trust Architecture", Special Publication (NIST SP), National Institute of Standards and Technology, Gaithersburg, MD.

About the Author

Zhamak Dehghani is a director of technology at Thoughtworks, focusing on distributed systems and data architecture in the enterprise. She's a member of multiple technology advisory boards including Thoughtworks. Zhamak is an advocate for the decentralization of all things, including architecture, data, and ultimately power. She is the founder of data mesh.

Colophon

The animal on the cover of *Data Mesh* is a great snipe (*Gallinago media*), the fastest migratory bird known to humans. Great snipes breed in northeastern Europe and migrate to sub-Saharan Africa. Researchers have recorded great snipes flying 4,200 miles at up to 60 mph without stopping.

A typical great snipe wingspan is 17–20 inches. They normally weigh under 7 ounces. The great snipe's plumage is mottled brown, an effective camouflage for the grasslands, marshes, and meadows they inhabit, with a dark stripe across their eyes. The beak is long for foraging in mud and wetlands for worms and insects.

The great snipe population is Near Threatened, according to the IUCN Red List. Many of the animals on O'Reilly's covers are endangered; all of them are important to the world.

The cover illustration is by Karen Montgomery, based on a black-and-white engraving from *British Birds*. The cover fonts are Gilroy Semibold and Guardian Sans. The text font is Adobe Minion Pro; the heading font is Adobe Myriad Condensed; and the code font is Dalton Maag's Ubuntu Mono.