MONTE CARLO

The Ultimate Guide To: Data Mesh Architecture

An introduction to key concepts and implementation guide.

Table of Contents

I. Taking The Data World By Storm

If implementing data mesh is high on your list of priorities in 2023, you're not alone. The concept of the data mesh has taken the data world by storm since 2019, when Zhamak Deghani first coined the term in her landmark publication. Data mesh is a proposed way of life for data-first companies, and successfully implementing it within an organization requires both structural and cultural buy-in.

At its core, data mesh is a type of data platform architecture that both acknowledges the limitations of traditional data infrastructure and anticipates an inevitable future state in which companies embrace data democratization and scalability.

Data mesh supports distributed, domain-specific data consumers and views data as a product. In a data mesh architecture, each domain team handles its own data pipelines with a self-serve design underpinned by federated data governance.

As a growing number of teams within an organization regularly utilize high volumes of data, fissures in a centralized approach to data architecture can emerge. Companies that adopt a centralized architecture often find their data teams become a bottleneck as the company grows.

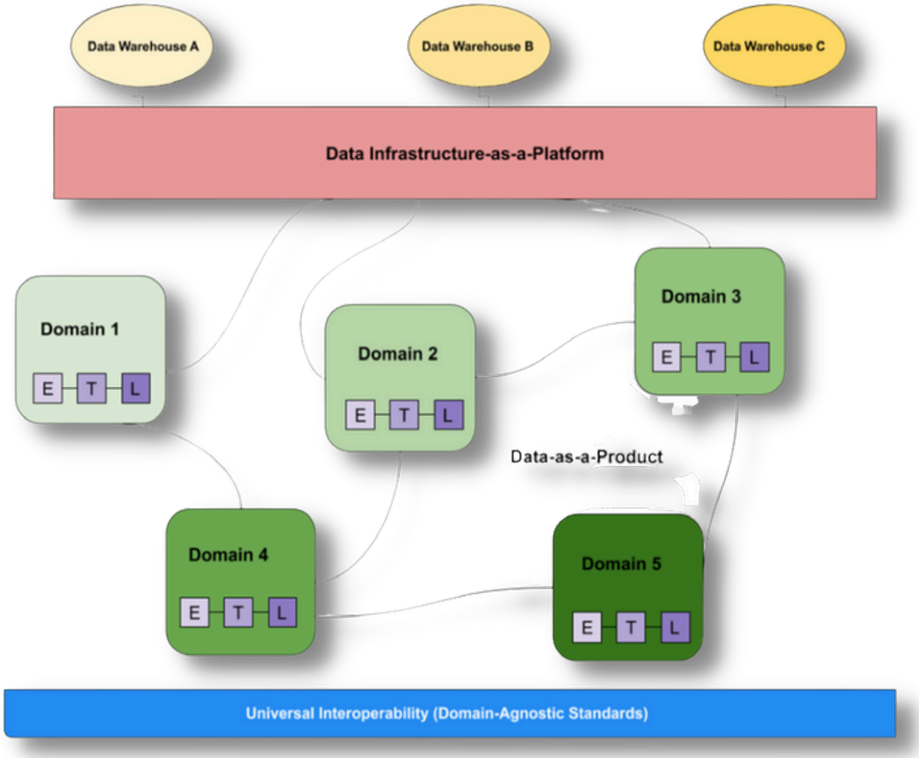
Data mesh can help solve this problem, with domain-oriented decentralization that empowers domain teams, relieves pressure from the data team, and fosters easier collaboration and connection with federated data governance across the organization. As more and more companies embrace data democratization, the flexibility and scalability of the data mesh is becoming increasingly attractive.

Implementing data mesh effectively, though, takes careful planning, foresight, and buy-in from various stakeholders across the company. A successful data mesh deployment is as much about mindset as it is about infrastructure.

In this guide, we'll discuss what it takes to plan, design, and execute on your data mesh strategy, through the lens of successful implementations at Intuit, Zalando, BlaBlaCar, and others.

II. What Is - And Isn't - A Data Mesh?

What data mesh is



In engineering language, data mesh is essentially the platform version of microservices. It's a type of platform architecture that embraces the ubiquity of data in the enterprise by leveraging a domain-oriented, self-serve design.

Data meshes differ from traditional monolithic data infrastructures that handle all an organization's data consumption, storage, transformation, and output within one central data platform by decentralizing data ownership. The data mesh concept views "data as a product" and empowers each domain team to handle its own data pipelines by giving each team ownership over its underlying platform or data storage layer.

A data mesh is not, however, a fully decentralized siloed data architecture. Rather, the data mesh connects multiple autonomous domains via a universal interoperability layer that applies standard rules and syntax across the organization.

The "mesh" is an integral part of the infrastructure, as it allows teams to collaborate and cross-utilize their data. Facilitating this type of sharing can lead to a whole that is greater than the sum of its parts, as teams can leverage aggregated data to build better, more comprehensive data products.

Ultimately, data meshes are implemented in service of an overarching goal: to drive flexibility and scalability when it comes to data use within an organization.

What data mesh isn't

It’s relatively new, and it’s relatively complex—so, naturally, the data mesh concept is ripe for misinterpretation. We’ve covered what a data mesh is—but here are some things a data mesh is not.

It is not synonymous with data virtualization

Data virtualization is an approach to data management that allows an application to retrieve and manipulate data across many silos. According to data mesh founder Zhamak Dehghani:

> *“Whether it’s a data mesh or an API base, you’re trying to expose a database that has been optimized for a transactional purpose for analytical purposes. And predictive analytics or historical trend modeling both require a very different view of the data…if you think about using virtualization on top of your microservice database and expose them and call that a mesh, that is probably a bad idea.”*

It is not a technology

Data mesh relies on multiple technologies to operate, and it facilitates the effective and efficient use of an organization’s data tech stack. It is not, however, a standalone technology. Rather, data mesh is the overarching infrastructure and strategy that enables various technologies within an organization.

It is not a self-serve data platform

We know the data mesh concept arose partly in response to the ubiquity of centralized data platforms that were unwieldy to manage and difficult to scale. Does it stand to reason, then, that a data mesh is the same thing as a decentralized self-serve data platform?

In short, no. While large organizations have already begun to implement self-serve platforms for infrastructure management, the self-serve aspect of the data mesh is different in a few key ways. According to Zhamak:

> *“The majority of the service or data platforms built today are built for centralized data teams—they’re built to help data specialists move through their backlog faster.”*

Self-serve data platforms are built for different purposes than the data mesh—i.e., they’re not optimized for both the autonomy of domain teams and the oversight of data generalists.

Modern data mesh design principles

While each data mesh is unique in its own way, a set of four design principles unites every data mesh platform. A data architecture is a data mesh if it:

1. Supports a domain-oriented architecture, which is inspired by the concept of domain-oriented microservice architecture in engineering and which distributes ownership of data from one central team to domain-specific teams.

2. Views data as a product and gives each domain-oriented team the accountability, product thinking, and tools it needs to do so.

3. Empowers teams with a self-serve data infrastructure, which simplifies data product creation and management.

4. Employs federated data governance, which maintains order and enables cross-team collaboration and consistency.

What does each of these principles look like in practice, and how have real companies wielded them to successful data mesh implementations? Let’s take a look.

III. Domain-Oriented Architecture

What it means

This fundamental principle of the data mesh arose from an overarching problem with the centralized data architecture model: the tendency of the data team to become a bottleneck.

When multiple teams need to create multiple data products across an organization, it's challenging for a single central data team to keep apace with the myriad questions and problems posed by a disparate group of data product owners. Domain-oriented architecture aims to solve that problem by shifting data ownership away from the central team and toward domain teams.

What it looks like

In theory, establishing a domain-oriented architecture is an easy strategy to follow. In practice, doing so will look a little different for each company—and that's okay.

The first step in establishing a domain-oriented architecture is to define what "domain" means in the context of your specific organization.

According to Zhamak: "Domains can be classified by data characteristics and data product usage," and she recommends classifying data products in three ways:

1. Source-aligned, in which most published data products correspond closely to the domain events and entities generated in the operational system. For example, an online shop divided into domains along the customer journey (product search, browsing, checkout, and payment).

2. Aggregate, a more efficient way for a team to focus s olely on delivering a data product that is the aggregate of various data products from other domains. For example, a 360-degree customer view that includes relevant data from multiple domains such as account data, orders, shipments, invoices, returns, account balance, internal ratings, etc.

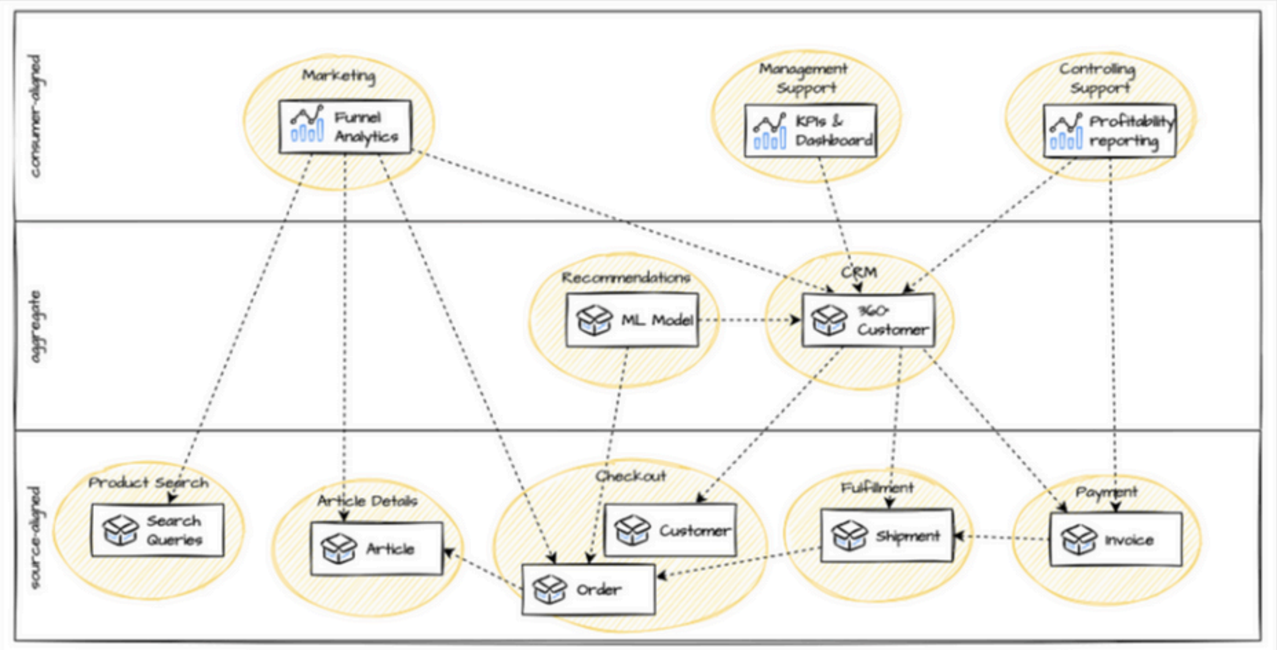
3. Consumer-aligned, a model that is optimized for a specific department's needs. In this model, teams can focus on fulfilling the data needs of a specific business domain, which engenders deep domain knowledge and facilitates better analytical results.

Making a mesh

Transferring ownership to the team responsible for creating each data product can streamline operations and alleviate pressure from the central data team. But what happens to data that affects multiple domains?

That's where the "mesh" part of data mesh comes in. Data mesh architecture actually facilitates the collaborative use of data products across domains.

According to Zhamak: "Using data from upstream domains simplifies data references and lookups…while data from downstream domains enables analyzing effects."



datamesh-architecture.com

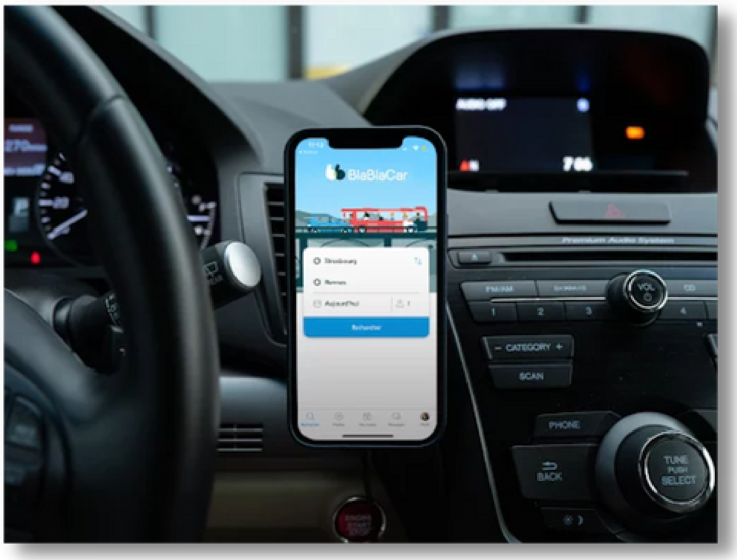
The end results of this cross-functional collaboration are often more comprehensive reports and better data products.

If establishing domain-oriented architecture sounds too overwhelming for your centralized team, consider starting with a structure that leverages a Center of Excellence.

In this type of structure, a single "Center of Excellence" composed of data engineers devises a set of central standards that then guide the infrastructure, product development, and governance of each domain.

This is the approach the Sanne Group took to implementing data mesh across the organization. Sanne's Head of Data Engineering, Martin French, cites needing space for organic, decentralized evolution amid a period of transition as key reasons to pursue a Center of Excellence en route to a fully decentralized data mesh architecture.

Case Study: BlaBlaCar



BlaBlaCar, one of the world's leading carpool networks, relies on high-quality, trustworthy data to execute and scale its operations. To address fundamental issues of data quality and poor data visibility, BlaBlaCar implemented a data mesh architecture.

"We realized many of the problems we were experiencing could be solved in this new socio-technical concept," says Kineret Kimhi, Data Engineering and BI Manager at BlaBlaCar.

In establishing a data mesh, Kineret and his team learned how to empower multiple domains while retaining universal governance. "If you're going to separate your central team, you must have a strong set of governance policies so everyone is doing things the same way," she says.

"We got a lot of positive feedback from our internal teams on our transition to the data mesh, but one of the downsides was individuals working in domains worried they are the only practitioner and don't have someone to consult with. So we created horizontal groups we call chapters that we strengthen and build collaboration and reporting structures around."

IV. Treating Data Like A Product

What it means

Organizations that effectively implement data mesh think of data not as a tool or an input, but as a product. That means domain teams apply product thinking to their data to ensure it is easily discoverable, easily read, and easily understood while also applying product principles such as versioning, security, monitoring, logging, and alerting.

What it looks like

Just as it's important to begin establishing a domain-oriented architecture by defining "domain," the first step in treating data like a product is to identify what a data product actually is at your organization. Your organization's data products could include enriched datasets, dashboards, analytical apps, algorithms, APIs—the list goes on (lucky for you we wrote an eBook focused entirely on data products).

Ultimately, the products you define should have clear, quantifiable business value, and the domain teams that support them should have the skills they need to build and manage those products from day one.

> *"When we started to develop our data platform, we already had domain-oriented ownership in place," says Matheus Espanhol, Data Engineering Manager at BairesDev. "But as we thought about treating data as a product, we wanted to prioritize what would be a good solution for data producers and consumers. For example, not all the domains had the roles needed to implement the full self-service concept at first. And we need to be patient with that and make sure they aren't relying on the data platform team instead of finding their own autonomy."*

Ultimately, it’s the data team’s job to understand what the business needs.

According to Max Schultze, Data Engineering Manager at Zalando, “This is where I often see people stumbling. They focus so much on how to build a product and lose sight of what they’re building it for. But when you start with the business need and build to that, it becomes much easier to measure, display, and showcase the right requirements—which creates trustworthiness.”

As you begin planning your strategy, it might make sense to start small and pilot a data mesh approach to a single data product. You probably don’t want to overhaul your entire tech stack at once, especially if you’re working in a more established organization—data mesh is, after all, as much about cultural buy-in as it is about structural change.

When in doubt: focus on building trustworthy products

A data mesh is only as good as the data it leverages and the product development it enables. As you build yours, keep this overarching principle in mind: focus on building trustworthy products. That, of course, requires the utilization of trustworthy data.

The concept of data product raises the bar for the availability, reliability, and usability of enterprise data, but it also represents a profound shift in perspective.

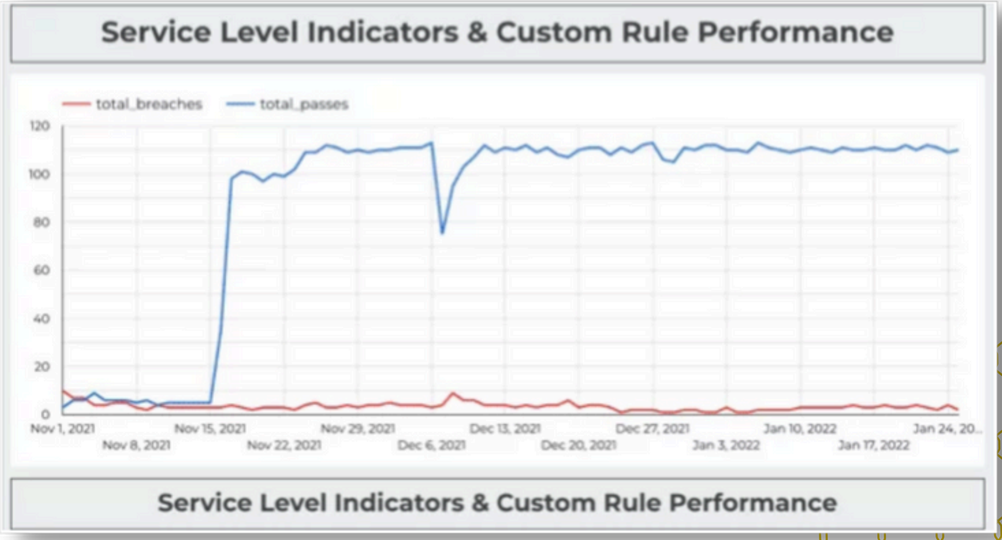
Waving a magic wand and declaring an executive dashboard a data product without any real changes to the data's underlying governance or reliability won’t do your team - or customers - any favors.

Instead of the wand, try:

- Monitor all of your pipelines as well as the data itself: To build data trust you need to minimize data downtime–measured as the number of incidents multiplied by the average time to detect and resolve those incidents. You especially need to catch bad data before it hits your stakeholders. Data observability solutions like Monte Carlo leverage machine learning powered monitors to understand when your pipelines are broken or when the data flowing through them is bad so you can do just that. Data testing is inefficient and can't provide sufficient coverage at scale.

- Optimize incident triage and resolution: Data lineage–a key feature of data observability platforms–enables data teams to immediately understand the blast radius of a potential incident and triage accordingly. It can also be used to trace incidents to the most upstream impacted table for quick root cause analysis alongside other features such as query change detection, high correlation insights, and anomalous row detection.

- Define data SLAs, or service-level agreements, and access requirements early. These SLAs can foster a culture of transparency between the data team and domain teams, and they should address questions such as: what is the business need for this data? Where is it located? What are the expectations for it? What does the data need to meet the expectations? Who is affected? How will the team know if the SLA is met, and how should they respond if it isn't?



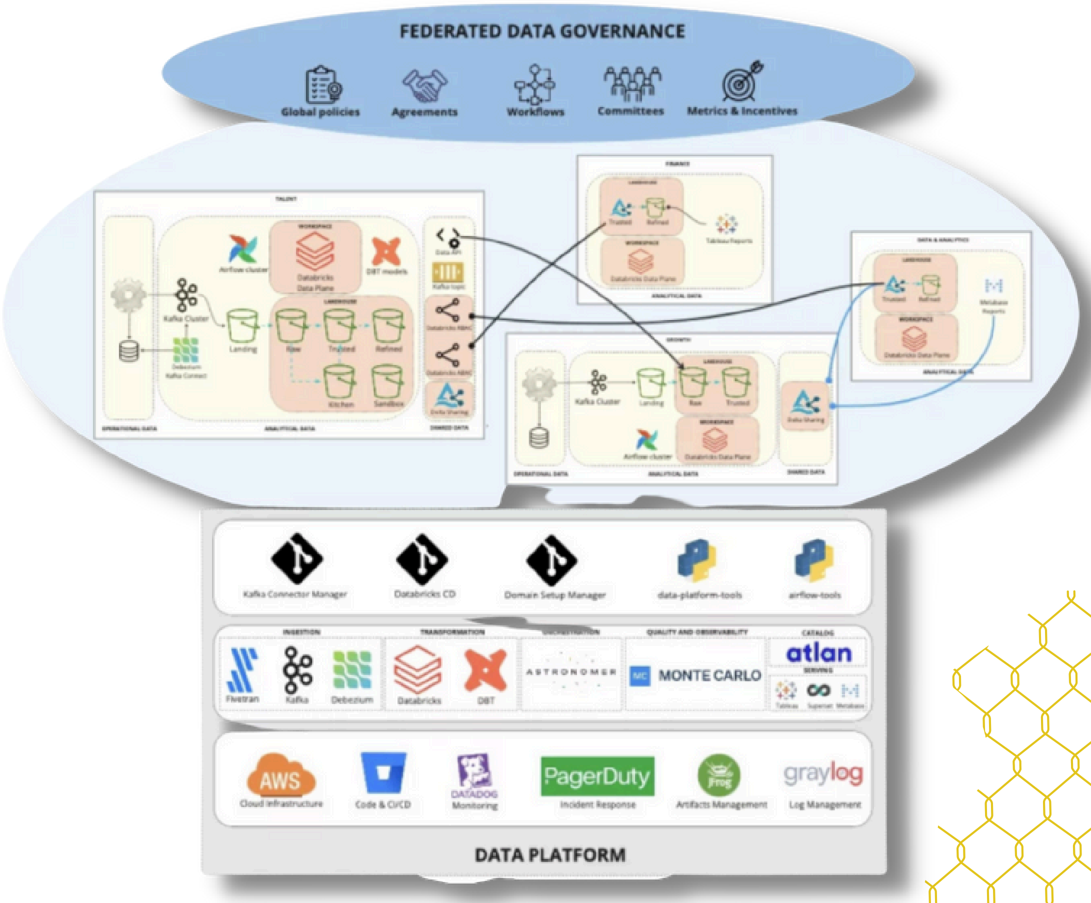
Case Study: BairesDev

Global software solutions and staffing provider BairesDev is one of the fastest-growing outsourcing companies in tech.

With 4,000+ employees located across 40+ countries, BairesDev lacked cohesion when it came to the development and management of data products. Data performance couldn’t keep up with the company’s growth, and employees lacked trust in the company’s data.

BairesDev’s data engineering team decided to tackle these issues with a data mesh model that would identify each domain’s roles and responsibilities and work to solve the specific problems each team faced, with solutions that could then be repurposed across domains.

The company already had strong domain ownership, so the data team was able to work with domain-specific data producers to establish capabilities for sharing data across domains. That included making data available to end consumers such as other applications, other Databricks warehouses, or BI tools.



V. Self-Serve Infrastructure

What it means

One thing to anticipate as you begin your data mesh journey? Pushback from various constituents, who may be daunted by the thought of managing, staffing, overseeing, and paying for distributed data product development and management across multiple domains.

Implementing data mesh should streamline operations, both by empowering domain teams and by reducing pressure on the central data team.

According to this New Stack article, “a self-service data platform should serve the producers, consumers, and maintainers of data products. Each of these individuals has a different set of needs that overlap with one another.

The self-service platform must provide the tools and interfaces that simplify this lifecycle of creating, finding, using, and possibly deleting a data product.”

What it looks like

First things first: define what “self-serve” means in your organization. Who is actually accessing the data? How are they using it?

According to Zhamak, “In designing the platform, it is helpful to consider the roles of platform users and their journey in sharing and using data products. The platform can then focus on how to create a frictionless experience for each journey.”

Take stock of your teams and identify the roles that will be leveraging the data mesh. Data product owners and data product users are two integral personas, and the self-serve platform should empower them in different ways.

According to Zhamak, "A self-serve approach exposes a set of platform APIs for the data product developer to declare their infrastructure needs and let the platform take care of the rest."

Similarly, the platform should facilitate the frictionless discovery, analysis, and use of data for the data user.

Components of a self-serve data platform

Take a page from the New Stack's book and keep these three key features of a self-serve platform in mind:

1. Discovery: Your platform should make it easy for users to discover and use available data products (and to ask for help when they need it).

2. Data product management: Your platform should facilitate the simple sharing of data products across the organization.

3. Access control: Your platform should both allow and restrict data access as needed.

Don't wait on the perfect platform

Here's a secret: even with the best of intentions, the most thoughtful planning, and the highest levels of communication, your self-serve platform probably won't be perfect when you get started. That's okay—and imperfection shouldn't hinder progress.

Zalando's Max Schultze cautions teams against unrealistic ambition.

"Data mesh always describes this beautiful greenfield self-serve setup for data platform and infrastructure," he says. "But that's rarely where any company is starting. You usually have 10-15 years of legacy systems of data warehouses, governance, and functions. You have to understand how to evolve your tech step-by-step into the self-service direction to make it scalable."

Case Study: Zalando

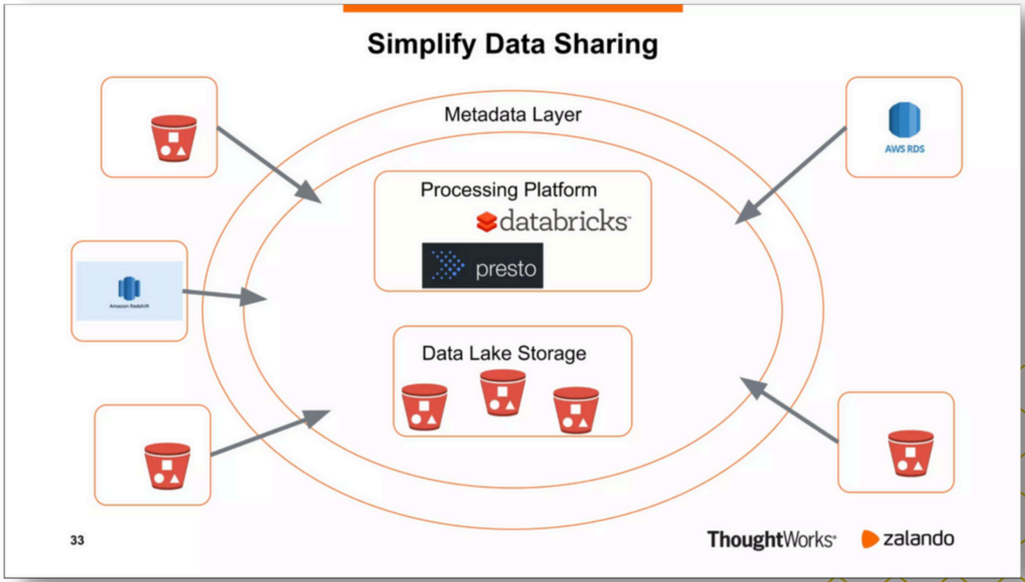
For Zalando, Europe's biggest online fashion retailer, data is key to success. The company collects huge volumes of data, which inform real-time decisions on a daily basis. As the company scaled, the central data team was becoming a bottleneck too often.

"We were coming from [a] situation where the infrastructure team was actually the bottleneck of the whole process," says Max Schultze. "We want to much rather be in the situation where infrastructure becomes the platform, and it's much easier to just use it without always having to involve the team."

Zalando implemented data mesh as a solution, and the data team deployed a concept they called "bring your own bucket," which allows employees to plug the datasets they created and stored personally into the company's central infrastructure.

"That's the great part about this, because the infrastructure team… [doesn't] even need to understand what the users are doing with these clusters. They are fully focusing on the infrastructure themselves, and they are offering this as a platform that people can just use from the outside," says Max.

The end result? Simplified data sharing—and a data team that became facilitators rather than inhibitors of innovation.



Simplify Data Sharing

VI. Federated Governance

What it means

In data mesh, federated data governance means a central data team defines and oversees overarching data governance standards, while distributed domains execute these standards however they see fit. This structure engenders independence and autonomy while ensuring cross-domain alignment and facilitating collaboration. It's a critical component of an ultimate benefit of the data mesh: its scalability.

What it looks like

In theory, federated data governance makes perfect sense. In practice, it can be challenging to strike the right balance between independence and oversight. Here are a few tips:

- Domain-first: This is especially important in a world where data regulations are increasing at an exponential rate at the regional (GDPR) and even state level (CCPA). A mesh structure enables teams to right size their level of governance as required by the domain or business group.

- Disciplined Documentation: Without good documentation it is difficult to enable discovery and without good discovery, self-service just doesn't work. The challenge is your ability to pipe data is virtually limitless, but you are constrained by the capacity of humans to make it sustainably meaningful. In other words, you can never document all of your data, and rarely will you have as much documentation as you'd like. Focus on your highest value assets and iterate.

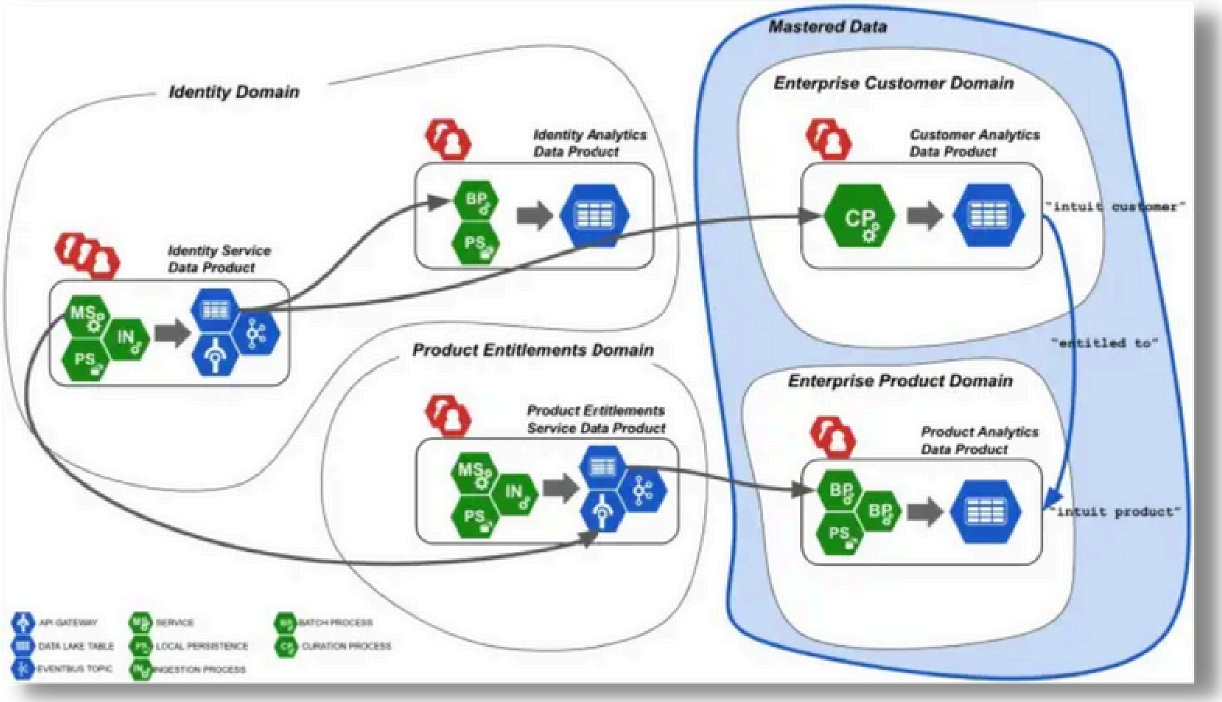
- Prioritize automation. It's hard to establish standardized processes and procedures when ownership is distributed across domains—and it's harder to monitor adherence to those processes and procedures manually. Whenever possible, automate policy monitoring and user access management. Having accurate object tagging in place can also make it much easier to identify and monitor higher-risk data (PII), and for compliance teams to implement additional security measures like dynamic data masking or row-level access restrictions.

Case Study: Intuit

Global financial technology platform Intuit relies on data-driven systems to enable the development of superior product experiences. The team, though, experienced issues of data discoverability, understandability, trust, consumption, and publication—and decided to solve those issues by implementing a data mesh strategy.

Intuit Engineer Tristan Baker says that Intuit supports “discovery, understanding, trust, consumption, and publishing of data systems” in distinct ways, including “ownership that ensures all teams understand and are accountable for a set of defined responsibilities in building and managing their solutions; including adherence to a set of defined best practices to produce only high quality data.”

Intuit implements clear rules and responsibilities for the scope of each domain team—which both promotes autonomy and maintains consistency across the organization.


Mastered Data
Identity Domain
Identity Analytics Data Product
BP
PS
Identity Service Data Product
MS
IN
PS
Enterprise Customer Domain
Customer Analytics Data Product
CP
"intuit customer"
"entitled to"
Product Entitlements Domain
Product Entitlements Service Data Product
MS
IN
PS
Enterprise Product Domain
Product Analytics Data Product
BP
BP
PS
"intuit product"
API GATEWAY
DATA LAKE TABLE
EVENTBUS TOPIC
SERVICE
LOCAL PERSISTENCE
INGESTION PROCESS
BATCH PROCESS
CURATION PROCESS

Source.

VII. The Future of Data Mesh Is Bright

When Zhamak Deghani first introduced the concept of the data mesh in 2019, it was exciting, innovative…and largely theoretical. Companies were only beginning to acknowledge the power and necessity of data, and centralized data lakes were the overwhelming standard for organizational data architecture.

Over the past few years, the environment has changed. While the fundamental principles of data mesh remain constant, their application has become more varied and nuanced, as an ever-increasing number of fast-growing companies implement data mesh strategies of their own.

New concepts such as data product containers and analytical data product APIs have also entered the data mesh lexicon thanks to Zhamak's startup nextdata, which debuted in early 2023.

As data mesh shifts from the theoretical to the practical, companies are adapting the concept in ways that make the most sense for their business and data needs. Thus, each data mesh strategy—and each resulting data mesh—looks a little different and will continue to change and evolve as the concept's popularity continues to grow.

One commonality among all companies that employ a data mesh strategy, though, is the need for reliable, high-quality data. Many teams that successfully implement data mesh do so in partnership with Monte Carlo's data observability platform, which automates data governance, expedites incident management, and ensures data trust.

If you're interested in learning more about building your own practical data mesh underpinned by high-quality, reliable data, reach out to Monte Carlo today.