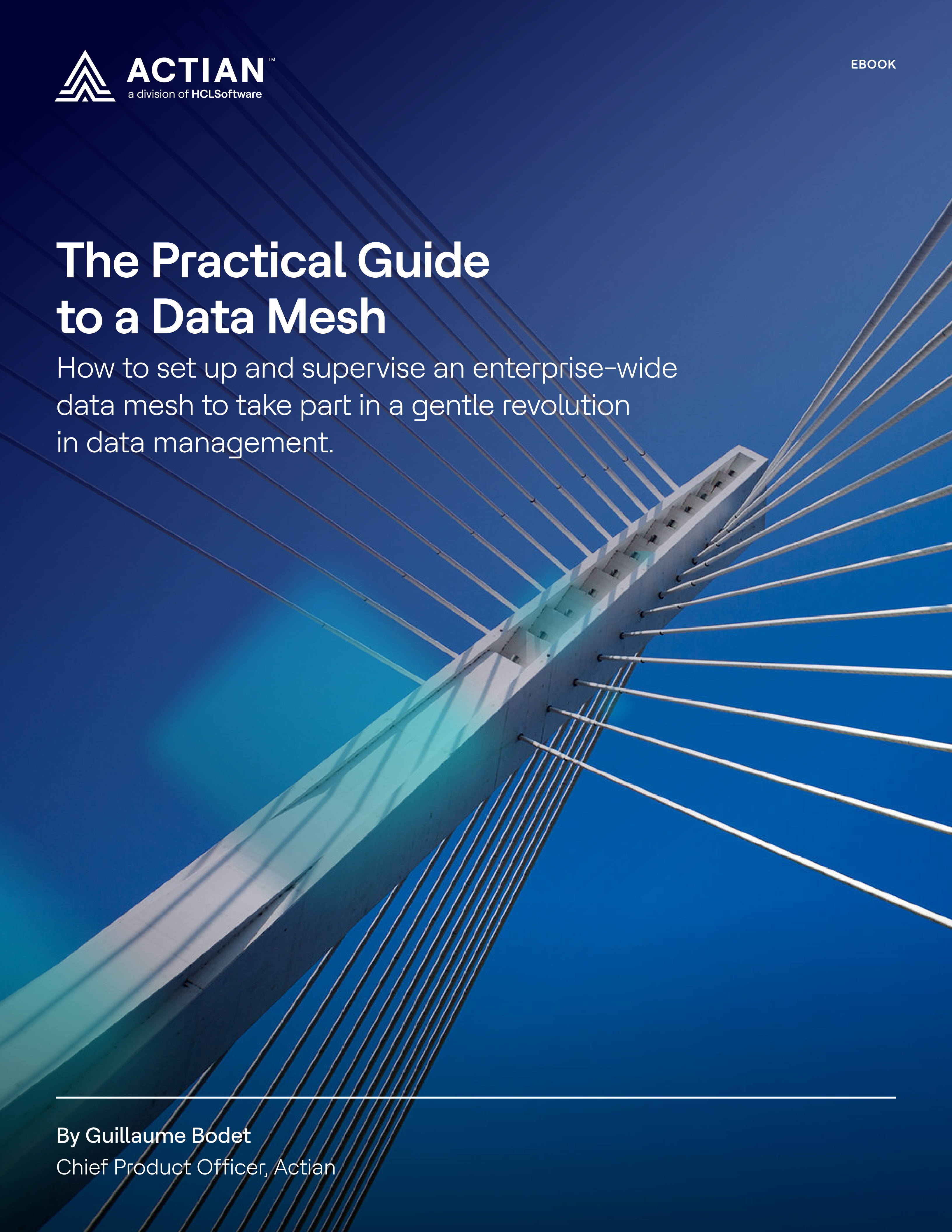
ACTIAN™
a division of HCLSoftware

EBOOK

The Practical Guide to a Data Mesh

How to set up and supervise an enterprise-wide data mesh to take part in a gentle revolution in data management.



By Guillaume Bodet
Chief Product Officer, Actian

Table of Contents

About The Author

Guillaume Bodet is a visionary product leader with an entrepreneurial spirit, combining deep technical expertise with a passion for building and scaling innovative businesses. As Chief Product Officer at Actian, he leads the vision, strategy, and roadmap for Actian's data management and intelligence solutions, ensuring organizations can effectively harness and govern their data at scale.Guillaume's career began as a software engineer. Over the years, he has held significant roles, including Principal Consultant on Distributed Architectures at Borland, CTO and Associate Partner at Xebia, and CTO – Head of Product & Engineering at Finance Active. His journey culminated in co-founding Zeenea, a pioneering data catalog company acquired by Actian in 2024, where he served as CPTO and played a critical role in shaping modern metadata management.With over 15 years of experience in data architecture, analytics, and machine learning, Guillaume is committed to delivering market-leading solutions that bridge technology and business needs.

Executive Summary

Approximately 20 years ago, Amazon set off a true technological and organizational revolution at its workplace. At the time, Amazon.com was already one of the most visited sites globally, running on a massive technical infrastructure.

However, it relied on a monolithic codebase with more than 500 developers, following a highly centralized development structure. This setup posed a significant risk for the company. Subject to demanding constraints for load, performance, availability, and security, the platform could only progress through very strict change controls. This resulted in the company moving at a pace incompatible with the ambitious goals of its founder.

To tackle this challenge, the company didn't just redefine the technical architecture of the platform. It embraced a systemic approach, covering architecture, organization, and culture. The approach relied on four key pillars:

1. **Platform modularization.** Following the principles of hexagonal architectures meant establishing a set of distributed components and services, developed and operated autonomously by small, agile teams. This provided assurances in terms of interoperability, backward compatibility, security, and performance.
2. **An infrastructure designed and managed by specialized teams.** Infrastructure components were provided to development teams as services for storage, processing, databases, networking, analytics, and more.
3. **Organization by domains.** These domains were structured according to the major operational capabilities of the company, such as sellers, catalog, customers, search, recommendations, reviews, cart, payment, and delivery (Figure 1). Each domain became responsible for the development of its components and services.
4. **Internal product culture.** The culture was instilled at all levels to foster innovation and enhance value creation. Services and components developed by the domains were treated and managed as digital products, with their performance continuously measured and optimized by dedicated managers. Some internal products were even commercialized at Amazon, particularly the infrastructure, which led to Amazon Web Services (AWS).

The comprehensive and successful transformation has yielded remarkable results, especially considering Amazon's growth, diversification, and innovation over the past two decades.

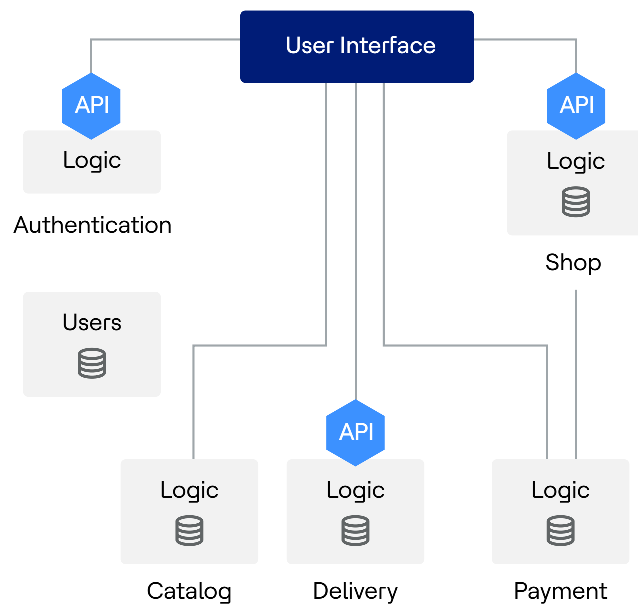


Figure 1. Example of a Service Architecture

Birth of a New Data Management Paradigm

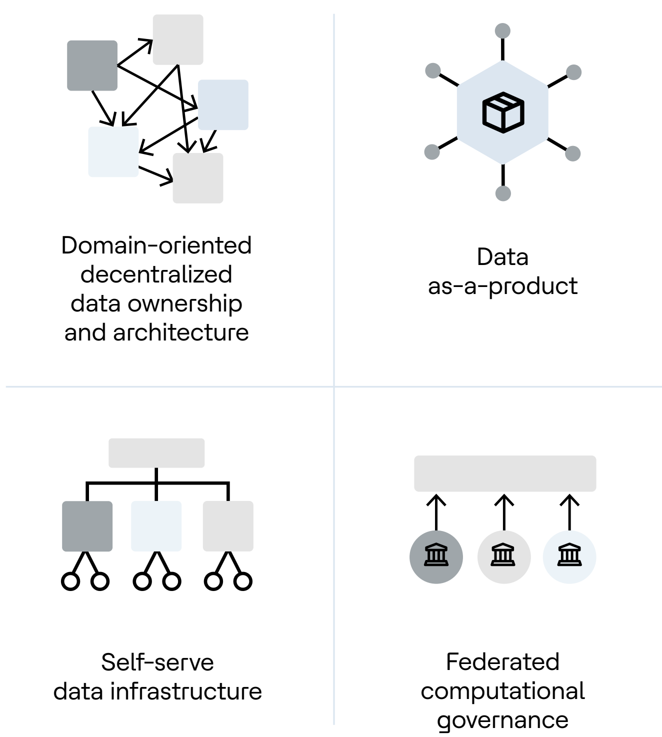
The world of data management is undergoing a similar gentle revolution to what Amazon, and subsequently many other organizations, experienced in the early 2000s. The paradigm driving this transformation over the past 20 years now has a name—the data mesh.

Centralized and monolithic data management, focused on a data lake or data warehouse, creates a massive bottleneck. It stifles innovation and severely limits or even eradicates the ability of data teams to meet the increasingly urgent demands of the business.

Eliminating this bottleneck and, above all, restoring an organization's ability to rapidly innovate requires rethinking data management practices. Four fundamental principles come into play:

Data Mesh Principles

Empowers domain experts to create meaningful data products within a decentralized framework



According to a study by research and consulting firm Eckerson Group from 2023, the adoption of data mesh principles is widespread. Nearly 70% of companies surveyed were implementing a data mesh or planned to do so. Other studies corroborate this number, with a majority incorporating the principles of data management decentralization into their strategic roadmap.

Key Factors Driving the Rise of Data Mesh

Two converging factors explain the enthusiasm for data mesh and the decentralization of data management. These factors are related to economic pressures and the competitiveness weighing on organizations. The factors are related to the very birth of the data mesh itself.

First, there's a growing frustration among executive teams struggling to discern the returns on often substantial investments made in data infrastructure and data management over the past decade. This frustration is compounded by the fear of losing competitiveness due to the inability to take advantage of democratizing opportunities offered by artificial intelligence (AI).

Until recently, developing AI models was a long and risky process with uncertain outcomes. The rapid development of highly performant, inexpensive, and easy-to-integrate off-the-shelf models has changed the game entirely. It's now possible to prototype an AI application in a few days by adjusting and combining shelf models.

However, scaling requires feeding these models with quality data that's traceable, secure, and compliant. In short, ensuring well-managed data adds additional pressure on centralized data management teams.

These initial factors aren't directly related to the data mesh, but they outline a context in which data-driven organizations are pressured to reform to improve their performance and address current strategic challenges. Other factors are more directly related to the data mesh itself.

A data mesh is not an architecture, language, method, or even a technology—all of which are often complex and can be divisive subjects. Data mesh simply lays out a few easy-to-understand principles, and these principles aren't prescriptive. They can be implemented in a thousand different ways.

ACTIAN™
a division of HCLSoftware

These principles are also not purely academic. They transpose to the world of analytical data the practices that allow large organizations to master the complexity of their systems while continuing to innovate at a rapid pace. Data mesh is based on strong theoretical and empirical foundations.

Data mesh has the rare quality of easily gaining support, even enthusiasm, from data teams, including those at the decision-making level. This unanimity limits resistance to change, ensures strong sponsorship, and partly explains the speed of its adoption worldwide.

The last factor is probably the most important one—the principles of data mesh are easy to implement, without significant investments, simply by reallocating existing resources. When transforming a monolithic software platform into a plethora of loosely coupled and tightly integrated distributed services, the process will be lengthy, costly, and risky.

For data, the situation is very different. Data is already, by its nature, distributed. And organizations typically already have the necessary technologies to extract, process, store, and consume their data in higher-level applications. Implementing the basics of the data mesh primarily involves transforming an organization and practices, not making massive new technological investments.

In their eagerness to reform their practices, data leaders have found in the data mesh a convincing and accessible framework, and have included it in their strategic roadmap. It goes without saying, however, that the transition from a centralized data management model to an operational data mesh can only be done gradually—there is no magic wand. And each organization begins this transition in its own context with its own strategic challenges, personnel, organization, processes, culture, or even its technological stack.

From Literature to Practice: Implementing the Data Mesh

While literature on the data mesh is extensive, it is often described in a final state. There's rarely details on how to achieve it in practice. The question then arises: What approach should be adopted to transform data management and implement a data mesh?

A best practice is to take an iterative approach that's structured around the four principles of data mesh and informed by certain principles of lean manufacturing, particularly the elimination of waste in production chains. This approach is based on the idea that building a data mesh is a learning process and this learning can start very quickly by leveraging existing human and technological resources.

The overall approach is based on the following five steps:

1. Identify an initial use case and develop it by implementing the four principles of data mesh using existing resources.
2. Measure the production and consumption cycle times of data products.
3. Iterate the production and consumption processes of data products to reduce production cycle times.
4. Iterate on use cases to increase the reuse of data products.
5. Iterate on domains to expand and then generalize the data mesh.

To illustrate this iterative approach for implementing a data mesh, a fictional company is used in this eBook. Premium Offices represents a commercial real estate company whose business involves acquiring properties to lease to businesses.

Building a Pilot Project: The Embryo of a Data Mesh

The initial step to transform data management and implement a data mesh involves selecting a pilot use case. This will be developed based on the four principles of a data mesh using existing resources, and without impacting the organization.

Before embarking on the development of this initial use case, it's vital to focus on several essential prerequisites to ensure a successful start to the data management decentralization initiative.

Identifying and Selecting Domains

The primary prerequisite for launching the pilot project is the identification of domains—the federation of autonomous domains is at the core of the data mesh. This step generally poses no difficulty. Indeed, the concept of domains is already widely understood, and the division into domains is often stable. This is true whether it's structured according to value chains, major business processes, or organizational operational capabilities.

Domains sometimes have their own technical teams and operational systems that generate the majority of the data. The transition often involves reallocating data ownership according to an existing structure.

Characteristics and Selection of the First Use Case

The choice of a use case for the pilot project is relatively arbitrary. It could involve revamping an existing dashboard, creating a new dashboard, adding AI capabilities to an application, or commercializing certain data.

However, this first use case must possess specific characteristics to facilitate optimal learning conditions:

- **It must focus on usage, not just one or more data products.** The intrinsic value of a data product is null, and its value is realized through its uses.
- **It should have a limited scope.** The use case should consume data from no more than one or two domains—ideally just one.
- **It should not be overly simplistic.** It should consume more than one data product; two or three are sufficient. Combining data products is a fundamental learning process.
- **It should not be overly experimental.** The goal is to achieve concrete results quickly.

For the Premium Offices example, the company is already structured around domains that reflect its major capabilities. Here are three of its domains:

1. **Assets.** A domain responsible for acquiring and managing real estate assets. It primarily relies on asset management software.
2. **Brokerage.** A domain that manages the commercialization of properties for rent and tenant management. It utilizes tenant management software and is responsible for the commercial website and posting offers on specialized marketplaces.
3. **Capital markets.** A domain responsible for loans to finance purchases and optimize the loan portfolio. It uses another specialized software.

Premium Offices already has a modern data platform. It's managed by a centralized team supported by a centralized data office.

For the pilot project, Premium Offices chose to build a credit risk dashboard for its tenants to better anticipate and prevent potential defaults. This dashboard needed to combine tenant data from its software and credit data acquired from a specialized provider. These datasets are already used operationally in the process of evaluating a new tenant.

ACTIAN™ a division of HCLSoftware

Building the Pilot Project Development Team

Forming the team responsible for developing the pilot will help implement the first principle of the data mesh—domain-oriented decentralized data ownership.

For example, with Premium Offices, the data required for the pilot belongs to the brokerage domain, where the team responsible for developing the pilot is created. This multidisciplinary team includes:

- **A data product owner.** This role should have both a good understanding of the business and a strong data culture to fulfill the following responsibilities:
  - Designing data products and managing their lifecycle
  - Defining and enforcing usage policies
  - Ensuring compliance with internal standards and regulations
  - Measuring and overseeing the economic performance and compliance of the product portfolio
- **Two engineers.** One is from the brokerage domain team, bringing knowledge of operational systems and domain software engineering practices. The other is from the data team and is familiar with the data technologies being used.
- **A developer.** This role should design and build the dashboard.

The composition of this development team will vary depending on the context, but it should meet two requirements. It should include a data product owner, and it should integrate all the necessary skills to develop and manage its products.

Developing the Pilot Project with an Initial Use Case

Once domains have been identified, an initial use case is defined, and the team responsible for its development is assembled, it's time to kick off the pilot project. This entails building the embryo of the data mesh.

The second principle of the data mesh is treating data as a product. Over the past decade, domains have often developed a product culture around their operational capabilities. They offer their products to the rest of the organization as APIs that can be consumed and used to develop new services and applications.

In some organizations, teams strive to provide the best possible experience to developers using their domain APIs, such as searching in a global catalog, comprehensive documentation, code examples, sandbox environments, and guaranteed and monitored service levels. These APIs are then managed as products that are born, evolve over time without compatibility breaks, enriched, and eventually deprecated, usually replaced by a newer, more modern, more performant version.

The data mesh proposes to apply this same product-thinking approach to the data shared by the domains. In some organizations, this product-oriented culture is already well established. In others, it needs to be developed or introduced.

It's important to remember that a data product is not a new digital artifact requiring new technical capabilities, like an API product. It's simply the result of a particular data management approach exposed by a domain to the rest of the organization.

Managing APIs as a product does not require a technological breakthrough—existing middleware does the job just fine. Similarly, data products can be deployed on existing data infrastructures, whatever they may be.

Technically, a data product can be a simple file in a data lake with a SQL interface; a small star schema, complemented by a few views facilitating querying, represented in a relational database; or even an API, a Kafka stream, or an Excel file. A data product is not defined by how it's materialized, but by how it's designed, managed, and governed, and by a set of characteristics allowing its large-scale exploitation within the organization. These characteristics often fall under the categories of being discoverable, addressable, trustworthy, self-describing, interoperable, and secure (DATSIS).

Obtaining a DATSIS data product does not require significant investments. Instead, it involves defining a set of global conventions that domains must follow, such as naming, supported protocols, access and permission management, quality controls, and metadata. The operational implementation of these conventions usually does not require new technological capabilities—existing solutions are generally sufficient to get started.

An exception is the data catalog. It plays a central role in the deployment of the data mesh by allowing domains to publish information about their data products, and data consumers to explore, search, understand, and utilize these products. In the data mesh, the data catalog plays a somewhat different marketplace role from its traditional usage in large organizations.

How Premium Offices Created Its Framework

To establish an initial framework for the governance of its data mesh, Premium Offices set the following rules:

- **A data product materializes as a dedicated project in BigQuery.** This allows setting access rules at the project level, or more granular if necessary. These projects will be placed in a "data products" directory and a sub-directory bearing the name of the domain to which they belong, such as "Brokerage."
- **Data products must offer views to access data.** These views provide a stable consumption interface and potentially allow for evolving the internal model of the product without impacting its consumers.
- **All data products must identify data using common references for common data.** These can include clients, products, suppliers, and employees. This simplifies cross-referencing data from different data products, such as product codes and email addresses.
- **Access to data products requires strong authentication.** The authentication for Premium Offices is based on its cloud provider's identity and access management (IAM) capabilities. Using a service account is possible, but each user of a data product must then have a dedicated service account. When access policies depend on users, the end user's identity must be used via OAuth2 authentication.
- **The norm is to grant access only to views.** It's not granted to the internal model.
- **Access requests are processed by the data product owner.** This is done through workflows established in ServiceNow.
- **Data built tool (dbt) is preferred for extract, transform, and load (ETL) for implementing pipelines.** Each data product has a dedicated repository for its pipeline.
- **Options for data production consumption.** A data product can be consumed either via the JDBC protocol or via BigQuery APIs, read-only.
- **A data product must define its contract.** This includes data update frequency, quality levels, information classification, access policies, and usage restrictions.
- **The data product must publish its metadata and documentation in a marketplace.** In the absence of an existing system, Premium Offices documented its first data products in a dedicated space on its company's wiki.

This initial set of rules will evolve, but it sets a pragmatic framework to ensure the DATSIS characteristics of data products by exclusively leveraging existing technologies and skills. For its pilot, Premium Offices chose to decompose the architecture into two data products:

- **Tenancy analytics.** This first data product offers analytical capabilities on lease contracts, such as entity, parent company, property location, lease start date, lease end date, lease type, and rent amount. It is modeled in the form of a small star schema, allowing analysis along two dimensions—time and tenant. These are the analysis dimensions needed to build the first version of the dashboard. It also includes one or two views that leverage the star schema to provide pre-aggregated data. These views constitute the public interface of the data product. It also includes a view of the most recent list of tenants.
- **Entity ratings.** This second data product provides historical ratings of entities in the form of a simple dataset and a mirror view to serve as an interface, in agreement with common rules. The rating is obtained from a specialized provider, which distributes them in the form of APIs. To invoke an API, a list of entities must be provided, obtained by consuming the appropriate interface of the tenancy analytics product.

To identify entities and allow data cross-referencing, Premium Offices uses the Legal Entity Identifier (LEI) and already knows that this data has quality issues. The LEI is not systematically filled in with its enterprise resource planning (ERP) system, and when it's filled in, it's sometimes incorrect. This quality issue does not need to be resolved immediately, but it must be documented and will be a matter for the product owner.

The general schema of this data mesh embryo at Premium Offices is shown in Figure 2.

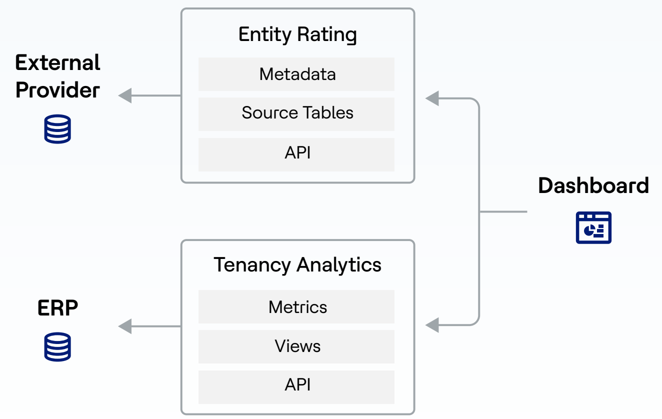


Figure 2. Data Mesh General Schema

Domain Tooling: The Data Platform of the Data Mesh

Designing a data product is not an exact science. Organizations can have one or multiple data products. To guide this choice, it's useful to leverage best practices from distributed architectures. A data product must:

- Have a single and well-defined responsibility.
- Be usable in several different contexts and therefore support polyglotism.
- Have stable interfaces and ensure backward compatibility.

One of the main barriers to decentralization is the risk of multiplying the efforts and skills required to operate pipelines and infrastructures in each domain. But in this regard, distributed architectures can provide a proven approach. The solution is to build a team responsible for providing domains with the technological resources and tools needed to extract, process, store, and serve data from their domain.

This model has existed for several years for application infrastructures and has gradually become generalized and automated through virtualization, containerization, DevOps tools, and cloud platforms. Although data infrastructure tooling is not as mature as software infrastructure, especially in terms of automation, most solutions are transferable, and capabilities are already present in organizations as a result of past investments.

Therefore, nothing preventing organizations from establishing a data infrastructure team, setting its roadmap, and gradually improving its service offering. Simplification and automation are the main drivers of this progression.

The data platform for a data mesh covers a wide range of capabilities, broader than infrastructure services. This platform is divided into three planes:

1. **Data infrastructure provisioning plane.** This provides low-level services to allocate the physical resources needed for big data extraction, processing, storage, real-time or non-distributed distribution, encryption, caching, access control, network, and co-location.
2. **Data product developer experience plane.** It provides the tools needed to develop data products—declaration of data products, continuous build and deployment, testing, quality controls, monitoring, and securing. The idea is to provide abstractions above the infrastructure to hide its complexity and automate the conventions adopted on the data mesh scale.
3. **Data mesh supervision plane.** This provides a set of global capabilities for discovering data products, lineage, governance, compliance, global reporting, policy, and control.

On the infrastructure side, the data mesh does not require new capabilities. The vast majority of organizations already have a data platform. The implementation of the data mesh also does not require a centralized platform. Some companies have already invested in a common platform, and it makes sense to leverage the capabilities of this platform to develop the data mesh.

It's entirely possible to deploy the data mesh on hybrid infrastructures—as long as the data products respect common standards for addressability, interoperability, and access control. The technical details of their execution are of little importance.

It's entirely possible to deploy the data mesh on hybrid infrastructures—as long as the data products respect common standards

Premium Offices Uses a Cloud Platform

Premium Offices invested in a shared cloud platform. The platform includes experts in a central team who understand its intricacies. For its pilot project, Premium Offices simply chose to integrate one of these experts onto the project team. This person is responsible for finding solutions to automate the deployment of data products as much as possible, identifying manual steps that can be automated later, and uncovering any missing tools.

Developing the First Data Products

The developer experience is a fundamental aspect of the data mesh, with the ambition to converge the development of data products and the development of services or software components. It's not just about being friendly to engineers, but also about responding to a certain rationality—the decentralization of data management implies that domains have their own resources to develop data products.

In many organizations, the centralized data team is not large enough to support distributed teams. To ensure the success of the data mesh, it's essential to be able to draw from the pool of software engineers, which is often larger.

State-of-the-art software development relies on a high level of automation. This includes the declarative allocation of infrastructure resources, automated unit and integration testing, orchestrated build and deployment via CI/CD tools, Git workflows for source and version management, and automatic documentation and publishing.

The development of data products should converge toward this modern approach. The organization's maturity, teams, and technological stack will determine how long this convergence takes. The best approach is to automate as much as possible using existing and mastered tools, then identify operations that are not automated to gradually integrate additional tooling.

In practice, this is what constitutes a data product:

- **Code first.** This is for pipelines that feed the data product with data from different sources or other data products, for any consumption APIs of the data product, and for testing pipelines and controlling data quality.
- **Data.** Often, the data exists in systems and is simply extracted and transformed by pipelines. Therefore, it is typically not present in the source code.
- **Metadata.** Some metadata will document the data product, such as schema, semantics, syntax, quality, and lineage. Other metadata is intended to ensure product governance at the mesh scale, such as contracts, responsibilities, access policies, and usage restrictions.
- **Infrastructure.** This is the declaration of the physical resources required to bring the data product to life, such as deploying and executing code, deploying metadata, and allocating storage resources.

Premium Offices Develops Two Data Products

For its pilot project, Premium Offices must develop two data products, which are entity ratings and tenancy analytics. The company's development teams use GitHub to manage the development process and automate the integration and deployment of software.

The development of data products therefore relies on the same tools. The source code of each data product will be managed in a repository on GitHub, and deployment on a cloud platform automated via CI/CD.

However, using the cloud platform to deploy software components is unfamiliar, and the central data team has not yet invested in DevOps tools to automatically create resources. The pilot team manually creates the necessary resources using the cloud platform admin console. Setting up Terraform is on the roadmap of the "Infra & Tooling" team.

Other steps are relatively simple. The entity ratings product is mainly composed of a dbt job that queries the list of tenants via the view provided by the tenancy analytics product. It invokes the data provider's API to obtain the ratings, feeds a dataset with timestamped ratings, and creates a view to expose them.

The rating data is subject to a license that states that the information can only be used for internal purposes and not distributed to third parties. This usage restriction will be clearly documented and governed by the product owner.

The tenancy analytics product is also composed of a dbt job that reads raw data from the ERP. This raw data has already been extracted and integrated into the cloud platform, and builds the small star schema and associated views.

Deployment via CI/CD, scheduling, and execution of dbt jobs on the cloud platform are well documented and pose no particular problem for the pilot team. The dbt features automatically test jobs and document views and tables.

Transition to a Federated Governance Model

The final fundamental principle of a data mesh is federated computational governance. In the data mesh, governance has a federated structure. This governance model is well known and already operational in certain highly decentralized groups.

In a federated structure, the central body defines the rules and standards that domains must adhere to. Department leaders are responsible for implementing these rules in their domain and providing the central body with evidence of their compliance, usually in the form of reporting.

Although the model is theoretically simple, its implementation often faces internal cultural challenges. This is particularly the case for organizations operating in heavily regulated sectors, where centralized governance teams are reluctant to delegate all or part of the controls they historically had responsibility for managing.

Federated governance also faces a reality that's rarely favorable—data governance is closely linked to risk management and compliance, which are two areas that do not excite operational teams. Consequently, it becomes difficult to identify local responsible parties or to transfer certain aspects of governance to data product owners who, for the most part, must already learn a new skill set.

Therefore, in most large organizations, the federated structure will likely be emulated by the central body and then gradually implemented in the domains as their maturity progresses. To avoid an explosion of governance costs or fragmentation, a data platform can automatically support entire aspects of governance.

So, what aspects of governance are likely to be automated?

- **Quality controls.** Many solutions for this already exist.
- **Fine-grained access policy management.** There are already solutions available, all of which rely on tagging information.
- **Traceability.** Development teams can automatically extract complete data lineage information from their data products and document transformations.

The road is long, but decentralization allows for iterative progress, domain by domain, product by product. It's important to remember that any progress in automating governance, in whatever aspect, relies on producing and processing metadata.

Being Mindful of Company Culture

At Premium Offices, the data office has a very defensive governance culture. Because the company operates in the capital market, it is subject to very strict regulatory constraints.

As part of the pilot, the company decided to not impact the governance framework. Quality and traceability remained the responsibility of the data office and will be addressed retroactively with their tools and methods. The office will also be responsible for access control, which is a process already in place in the form of a ServiceNow workflow.

The only concession is that the workflow will be modified so access requests are verified by the data product owner before being approved and processed by the data office. In other words, the company is taking a small step toward federated governance.

Regarding metadata, the new tables and views in BigQuery must be documented at both the conceptual and physical levels in the central data catalog, which is unaware of the concept of a data product. It's a declarative process that the pilot team already knows. Any column tagging will be done by the data office after evaluation.

For the rest, user documentation for data products will be disseminated in a dedicated space on the internal wiki. It will be organized by domain, which allows for rich and structured documentation.

Scaling Up the Data Mesh

Premium Offices has successfully completed its pilot project and laid the groundwork for a future, and still hypothetical, data mesh. It's now time to capitalize on this success by industrializing it at the organizational scale and including it in the company's strategic roadmap.

The strategic objective of decentralizing data management complements two other operational objectives set by the data office, which itself is pressured by regulatory authority recommendations. The objectives are to automate quality controls on critical data, and to extract technical and functional lineage from the same critical data.

First, at the strategic level:

- **The pilot project generally concerns only one domain.** This is usually the most mature one. It's a best practice to extend the data mesh across this domain to capitalize on its success before extending it to other areas.
- **Understand the architecture work.** The design and combination of data products require architectural work that should not be underestimated.
- **Prioritize based on usage.** This entails prioritizing new uses over old ones. In practice, this means that existing uses, such as dashboards, reports, and applications, should only be connected to the data mesh as part of a major evolution of these uses. In short, prioritize by value.

Then, on the operational level, the data mesh can be used to handle everything. This includes resource allocation, build, deployment, and monitoring of data products, access management, quality controls, lineage, compliance, and performance analysis. Although many organizations are not fully optimizing automation, moving in this direction can enable new benefits.

Recent history offers assurance. The first large distributed architectures relied on manual effort. Gradually, solutions became available to automate entire aspects. These solutions have often been the product of web giants that have made some of them available via open source.

No data platform covers all of the capabilities an organization may want. However, many free open-source and low-cost solutions exist, and they're generally easy to integrate into various existing tools. Many come from the world of distributed architectures and are already very robust.

The right approach is to move forward in small steps, automating one aspect of one data product, and then generalizing the practice. The next question is how to prioritize these automation projects.

Optimizing the Production and Consumption of Data Products

The introduction of data products into data management has an interesting side effect. It allows organizations to consider the development of data products as an activity of producing a digital object, just like a service, component, or application.

It's possible to exploit certain tools of lean manufacturing to gradually improve the production and consumption of data products and increase the overall throughput of the system. One principle of lean manufacturing is to look at a system through its value chains, measure the cycle times, and then reduce them. The principle may seem a bit abstract, but it is quite simple in practice.

The production of data products includes three value chains, or cycles (Figure 3):

1. **Create a new data product.** Its lead time can be defined as the time elapsed between the moment the development of the data product is validated and the moment the first version of the product is completed.
2. **Update of a data product.** This lead time is the time elapsed between the validation of the evolution and its deployment.
3. **Consumption of a data product**. Its lead time separates the moment when a consumer defines their need and the moment when they can use the data from a data product that meets their needs.

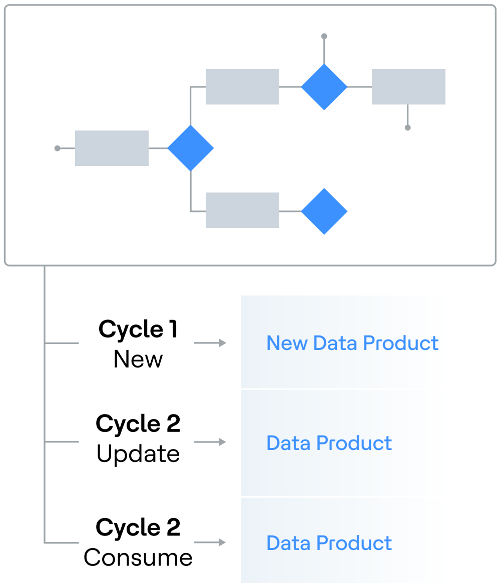


Figure 3. Production Cycles of Data Products

To define Cycle 1, it's necessary to agree on what a completed data product is. Otherwise, steps may be missed when trying to reduce the cycle time. The definition may vary from one environment to another.

Adopting a Data Product Definition

In the case of Premium Offices, the following definition was adopted and can be revised later:

- The data product is deployed in the production environment and complies with the technical and security standards of the data mesh.
- If the data product involves critical data or is involved in the data production chain of critical data, the concerned data must undergo quality controls as defined by the data office.
- Under these same conditions, the technical lineage of the data product must be transmitted to the data office in an agreed upon format.
- The tables of the data product are documented in the catalog following global conventions.
- The functional documentation of the data product is available on the wiki.

Once the start and end points of each cycle are well defined, it's possible to break them down into stages. This breakdown is called a value stream map. The concept can be illustrated by decomposing Cycle 3.

Here's the breakdown of the steps of Cycle 3 at Premium Offices (Figure 4):

- The consumer searches for the data product in the wiki.
- If the product is found, the consumer consults its documentation to understand how to use it.
- If it's not located, the consumer submits a request to the domain owner or data office.
- The domain owner reviews the request, with four possible outcomes:
  - They inform the consumer that a data product that meets their needs already exists and provides the URL in the wiki for easy access to documentation.
  - They consider it a new data product, which will be prioritized. Once planned, it enters Cycle 1. Upon completion of this cycle, the URL of the data product documentation is provided to the consumer.
  - They believe the need can be met by evolving an existing data product. They provide its URL to the consumer and submit the evolution request. Once planned, the evolution enters Cycle 2 and the consumer regularly checks for updates in the wiki.
  - They simply refuse the request, which interrupts the cycle.

Once the data product is identified and understood, the consumer triggers an access request. It is processed by a ServiceNow workflow. Once the request is processed and the appropriate permissions are set up in BigQuery, the consumer can use the data, marking the end of the cycle.

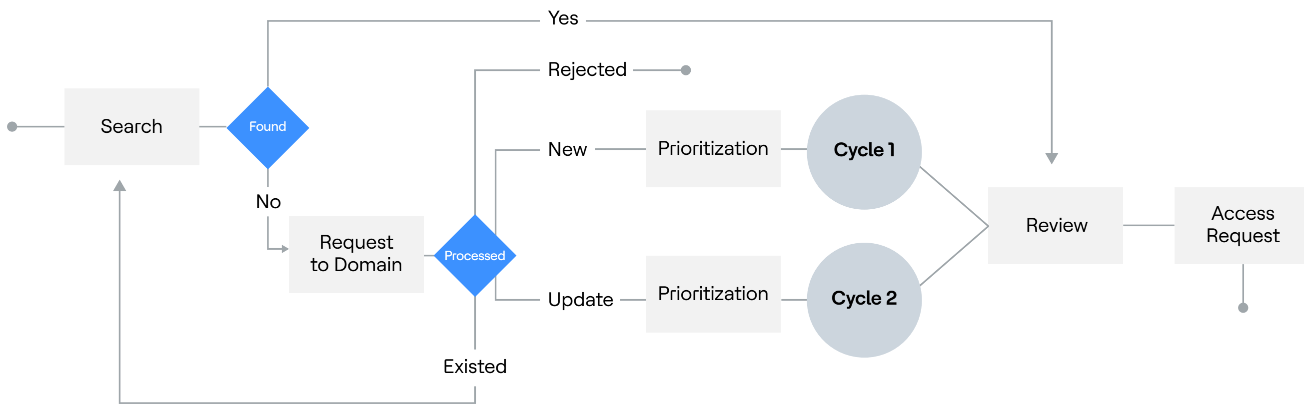


Figure 4. Breakdown of Cycle 3 for a Data Product

Reducing Production Cycle Times

The objective of this stage in setting up a data mesh is to reduce the production cycle times of data products. For each step, it's possible to measure or estimate the time needed to complete it. It's also possible to estimate the delays between steps, which lengthen the cycle time. The frequency of certain loops or paths can also be measured. Additionally, the frequency of cycle failure, which can reveal other sources of waste, can be measured.

The goal is to identify the most significant time losses and act on them, sometimes by improving certain processes. This can require introducing new automation tools.

Not all cycles are equivalent. Some are triggered more often than others. It's natural to prioritize optimizing these cycles first.

In the initial phase of implementing the data mesh, it's often useful to optimize Cycle 1. Indeed, in a developing mesh, creating a new data product is common. However, quite quickly, as with all digital products, the need to update operations will multiply. In the long run, maintaining and evolving a digital product is notoriously much more costly than its initial development. A good approach is therefore to prioritize optimizations that concern steps common to Cycles 1 and 2.

Taking Steps to Improve Decision Making

After the production and consumption process analysis, Premium Offices makes two important decisions:

1. **Automating physical resource allocation in the cloud platform using Terraform is not a priority.** The manual creation of resources is certainly tedious, but it's infrequent and low risk because it's supported by already established processes.
2. **Production of lineage information on data products is identified as a significant time loss**. Manually performed, it is a lengthy operation with a high risk of error. Moreover, it is necessary to review this lineage with each product evolution. Therefore, an initiative is launched to propose tools to automate the extraction of the technical lineage of a data product and integrate this tooling into CI/CD. The project is entrusted to the "Infra & Tooling" team.

Analyzing the cycle times of value chains is a valuable tool for supporting the progressive deployment of the data mesh and guiding the roadmap of the "Infra & Tooling" team, especially for Cycles 1 and 2. This analysis must be continuous because each optimization shifts the constraint and may challenge the previous prioritization.

Improving the Processes of Consumption and Reuse of Data Products

Cycle 3 is unique. Even if this cycle is initially infrequent—the data product offering must first expand before being multiplied—it becomes critical for scaling. After all, a data product has no inherent value until it contributes either directly or indirectly to a valid use. The reuse rate is also a good indicator of the value of a data product.

Initially, because available offerings are limited, the performance of Cycle 3 relies heavily on the other two. In practice, this means existing data products rarely meet new needs without modification. As a result, creating new products or adapting existing ones accounts for the largest share of the cycle time.

Quite rapidly, the cycle will be reduced to these three steps:

1. Searching for data products.
2. Consulting the data product documentation to understand its content, structure, and how to use it.
3. Requesting access to the data product.

This workflow constitutes the customer experience of the data mesh. An experience, on paper, that's very similar to that of a data marketplace or e-commerce system—search, consult, choose, order, and then delivered.

Drawing the Data Mesh Supervision Plane

There are pragmatic approaches that lay the foundations for a data mesh and guide its deployment in a large organization. Its performance and usability for both data product producers and consumers gradually improve over time.

Organizations shouldn't underestimate the difficulty of introducing a product culture, which should complement a data culture that's often still in its infancy, or convincing governance leaders to delegate some of their responsibilities to the domains. Decentralizing and automating data management are not projects. They are long-term programs whose foundations can be quickly laid, but whose maturity will take years.

These programs also support data platform innovation, with new tools expected to gradually enhance and extend existing capabilities. The example of Premium Offices highlights this trend—some features, particularly those related to the data mesh supervision plane, remain difficult to implement using current tools.

The data mesh supervision plane, which only makes sense on a global scale, provides a set of capabilities for exploiting and governing the mesh as a whole. These capabilities can be grouped into three categories to serve the needs of data mesh users:

1. **Data consumers.** They need a simple system to search, understand, and order the data products they want to use.
2. **Domain data producers.** These producers need a system to publish information about their data products, announce new products or new versions, manage access requests, manage evolution or new product requests, provide evidence of regulatory compliance, and monitor their product's performance.
3. **Governance leaders.** They want to control data product compliance with common or regulatory rules and supervise the overall performance of the data mesh to guide its development.

Governance Support at Premium Offices

At Premium Offices, data product documentation was placed partly in the group catalog, essentially to support the data office's governance processes, and partly in a wiki, whose search capabilities will likely not withstand the multiplication of data products.

For data producers, creating and maintaining documentation is a manual process, lengthening the cycle times. Ultimately, metadata is scattered across different systems, in various structured formats, making it difficult to access and impossible to activate for automating certain processes, especially access control.

Deploying a Marketplace to Facilitate Data Product Consumption

A data product is the cornerstone of the data mesh and the first step in transforming data management. This is at the heart of the data fabric approach.

Sharing and exploiting data products through metadata is critical. A data product is a governed, reusable, scalable dataset offering data quality and compliance guarantees to various regulations and internal rules. This definition is quite restrictive. It excludes other types of products such as machine learning algorithms, models, and dashboards.

While it's desirable for these artifacts to be managed as products, they are not data products. They are other types of products, which could be generally termed "analytics products," and data products are simply one subset.

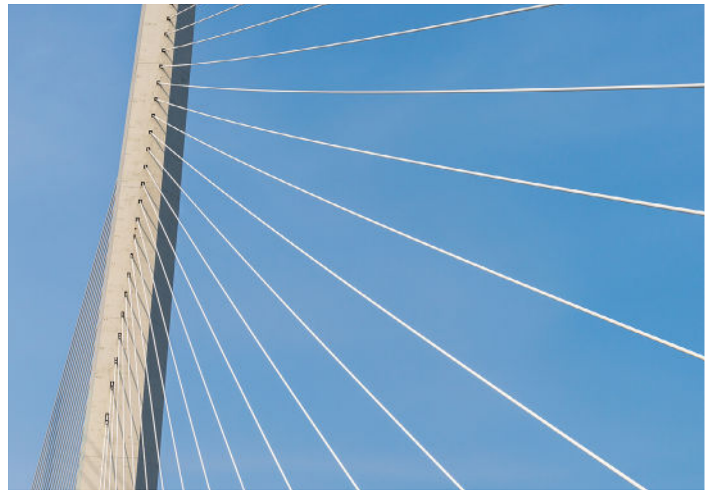
In practice, an operational data product consists of two things:

1. **Data.** Managed on a centralized or decentralized data platform, guaranteeing data accessibility, interoperability, and access security.
2. **Metadata.** Providing all the necessary information for sharing and using the data.

Metadata ensures that data consumers have all the information they need to use the data product. It typically covers these aspects:

- **Schema.** Providing the technical structure of the data product, data classification, samples, and their origin, or lineage.
- **Governance.** Identifying the product owner, the product's successive versions, its possible deprecation, and more.
- **Semantics.** Providing a clear definition of the information, ideally linked to the organization's business glossary, and comprehensive documentation of the data product.
- **Contract.** Defining quality guarantees, consumption modalities including protocols and security, potential usage restrictions, and redistribution rules.

In the data mesh logic, metadata is managed by the product team and deployed according to the same lifecycle as data and pipelines. This brings up a fundamental question—where can metadata be deployed?



Using a Data Marketplace to Deploy Metadata

Most organizations already have a metadata management system, usually in the form of a data catalog. But data catalogs, in their current form, have major drawbacks:

- They don't always support the notion of a data product.
- They can be complex to use. They are designed to catalog a large number of assets with sometimes very fine granularity. They often suffer from a lack of adoption beyond centralized data management teams.
- They mostly impose a rigid and unique organization of data, decided and designed centrally. This fails to reflect the variety of different domains or the organization's evolution as the data mesh expands.
- Their search capabilities are often limited, particularly for exploratory aspects. It's often necessary for users to know what they're looking for in order to find what they need for their use cases.
- The experience they offer sometimes lacks the simplicity users prefer, such as searching with a few keywords, identifying the appropriate data product, and then triggering the operational process of an access request or data delivery.

Given these shortcomings of data catalogs, a new concept is gaining popularity—the enterprise data marketplace (EDM). Like a general-purpose marketplace, the EDM aims to provide an easy shopping experience for data consumers.

ACTIAN™ a division of HCLSoftware

Setting up an Enterprise Data Marketplace

The EDM is a simple solution in which data consumers can easily search among data product offerings. They can find one or more products for a specific use case, access information related to these products, and then request them.

There are three options for setting up an EDM:

1. **Develop it.** This is a lengthy and costly option, but it holds the promise of a user experience optimized for the organization.
2. **Integrate a solution that's already on the market.** There are several options available, initially designed to ensure data commercialization or exchange outside the organization.
3. **Use existing systems.** Organizations can combine current solutions, such as a data catalog and corporate wiki, to deliver an EDM.

Although commercial marketplaces can offer a satisfying user experience and native support for the data product concept, they often have significant drawbacks. For example, they're highly focused on transactional aspects, such as distribution, licensing, contracting, purchasing or subscription, and payment. They're often poorly integrated with internal data platforms and access control tools.

These commercial solutions generally require data to be distributed by the marketplace. This involves a new infrastructure component, requiring data to be transferred to the marketplace in order to be shared. The system is sometimes called a data sharing platform. Introducing a new infrastructure component to deploy a data mesh is not a best practice. Instead, it's highly preferable to leverage existing capabilities as much as possible.

Scaling the Data Mesh with an EDM

An enterprise data marketplace is an essential component to optimize the data mesh on a large scale. It allows data consumers to have a simple and effective solution to search for and access data products from various domains.

Feeding the EDM via Domain-Specific Data Catalogs

Structuring data management around domains and data products is an organizational transformation that does not change the operational reality of most organizations. The reality is that data is available in large quantities, from numerous sources, evolves rapidly, and managing it is complex.

Data catalogs traditionally serve to inventory all available data and manage a set of metadata to ensure control and establish governance practices. A data mesh does not eliminate this complexity. It allows certain data, managed as data products, to be distinguished and intended for sharing and usage beyond the domain to which they belong.

In the data mesh, the need for a data catalog does not disappear. In fact, each domain should have a data catalog, allowing it to efficiently manage its proprietary data, support domain governance, and accelerate the development of robust and high-value data products. Metadata is managed at two levels:

1. **Domain level.** A data catalog allows for documenting and organizing the domain's data universe. Because the data catalog is a proprietary component, it's not necessary for all domains to use the same solution.
2. **Mesh level.** An EDM shares data products by all registered domains. The EDM is naturally common to all domains.

The general architecture for metadata management with a dedicated EDM is shown in Figure 5.

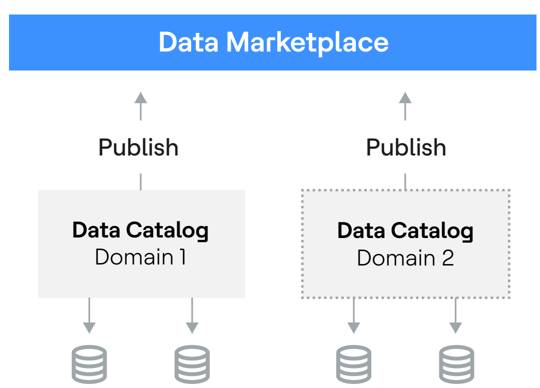


Figure 5. General Architecture for Metadata Management

In this architecture, each domain maintains its own data catalog, which may be built using a single or shared solution. By establishing a dedicated catalog for each domain, organizations can better structure their data and avoid the limitations of a one-size-fits-all metadata approach.

With an EDM, each domain deploys metadata, or even data, for its data products. This approach requires the close integration of different modules:

- Domain catalogs must be integrated with the EDM to avoid duplicating efforts when producing certain metadata. This is especially true with lineage, but also data dictionaries and schema, and even business definitions that will be present in both systems.
- Domain catalogs potentially need to be integrated with each other. This enables sharing and synchronizing certain information, primarily the business glossary but also some repositories.

The respective capabilities of an EDM and data catalog are very similar, as shown in Figure 6.

Capacity	DC	EDM
Data Governance Classify, organize, document, define, ownership, retention and policies, etc.	●	●
Data Lineage Manage and display technical and business lineage	●	●
Metadata Modeling Define information structure and ontologies	●	●
Data Quality and Profiling Report data quality controls and metrics	●	●
Search and Explore Define information structure controls and metrics	●	●
Connectivity and Automation Automatically crawl data sources and operational system, data disctionary	●	●
Business Glossary Share business definitions linked to data assets	●	●
Collaboration Report issues, suggest changes, rate and comment	●	●
Data Shopping Trigger data access request and get data delivered with compliance	–	●

Figure 6. EDM and Data Catalog Capabilities

Yet there are differences, too. Factors that distinguish a data catalog from an EDM include:

- **Scope.** The data catalog is intended to cover all data, whereas the EDM is limited to the objects shared by domains, such as data products and other domain analytics products.
- **User experience.** The data catalog can be a fairly complex tool, designed to support governance processes globally. It focuses on data stewardship workflows. The EDM, on the other hand, typically offers very simple user experiences, heavily inspired by e-commerce platforms, and provides an experience centered on consumption, or data shopping.

The Data Mesh Supervision System: The Actian Solution

The Actian Data Intelligence Platform (formerly called Zeenea) covers both the need for a cross-domain marketplace and the need for a data catalog for each domain.

Technology and Architecture of the Actian Data Intelligence Platform

The platform offers unique characteristics to support the duality of EDM and data catalog capabilities:

- **Designed as an EDM.** The platform offers two distinct user experiences:
  - Studio serves as the back-office interface, enabling the definition of the metamodel, automation of data inventory, metadata feeding, and support for data producers' activities.
  - Explorer is a separate application that provides a search-focused experience and catalog exploration, inspired by leading e-commerce sites.
- **Knowledge graph.** The platform relies on a customizable knowledge graph that avoids relying on a single hierarchical structure. The knowledge graph seamlessly accommodates multiple classification schemes, enabling various domains to organize their assets differently to meet their unique needs.
- **Modern catalog capabilities.** The platform offers the capabilities of a modern data catalog, including a dictionary, governance, lineage, glossary, search, connectivity, automation, and quality.

To support EDM usage, the platform has a management layer that allows each domain to have a private catalog and to choose the objects they want to share with the other domains. The EDM becomes a subset of the knowledge graph, containing not only data products but also objects intended for sharing, such as business definitions, dashboards, and machine learning models. Figure 7 shows the solution's architecture.

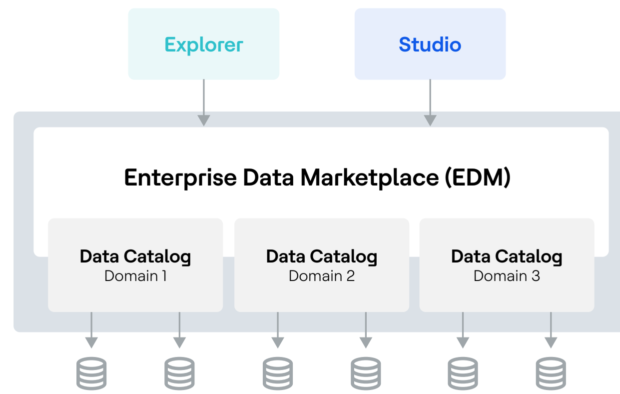


Figure 7. The Actian Data Intelligence Platform Architecture

With the architecture, each domain has its private space, where it can:

- Organize data as desired by defining a metamodel specific to the domain. Some elements of the metamodel can be shared or enforced by governance rules.
- Integrate and ensure the feeding of its data catalog from the data sources it owns.
- Manage its users and their permissions.
- Identify the objects it wants to share with other domains and control which information will be shared.

The level of delegation is highly configurable, allowing for more autonomy in mature domains while maintaining tighter control over less mature ones. The system's topology can be modified at any time to reflect the progression of the data mesh in the organization. Figure 8 shows the information structure.

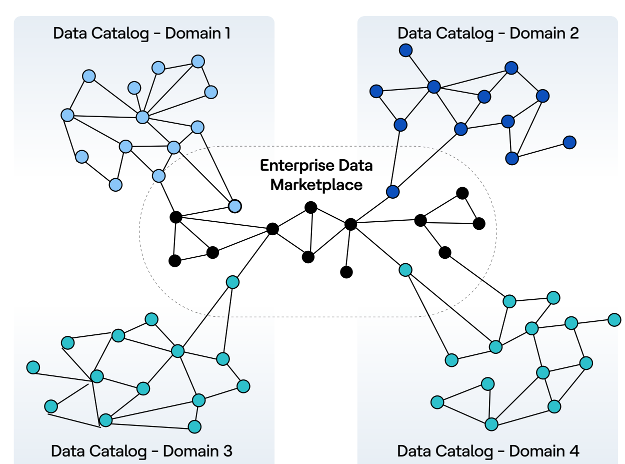


Figure 8. Private Data Catalogs per Domains and EDM Information Structure

The Federated Graph: Mirroring the Data Mesh at the Metadata Level

The information structure is made possible by what distinguishes the Actian solution from most other data catalog or metadata management offerings—an evolving knowledge graph. A knowledge graph is a data structure that adheres to a set of rules defined in an ontology.

The ontology describes the types of objects that the graph can contain, their attributes, and the relationships that these different types of objects can have with each other. This defines a semantic universe. Knowledge graphs have unique properties for organizing and searching for data:

- They allow for exploring the knowledge universe starting from any node, then traversing the relationships to navigate the graph through the lens of each user. The experience is vastly different from traversing a rigid and uniform hierarchical structure.
- The graphs form the basis of modern search engines. The engines can leverage not only the attributes of objects but also their relationships in the graph to rank them in the list of results.
- They support an associative exploration model, which dominates the web world and especially traditional marketplaces. Users conduct a search, then navigate the links between the search results and other associated objects.

In the Actian platform, each domain is responsible for a subset of the graph. Domains can model their subset according to their needs and evolve at their own pace.

Giving autonomy to domains to organize their data catalog is a fundamental capability. Too often, a unique catalog structure leads to excessive complexity in the hierarchical organization, making it incomprehensible and unusable. This is the shared drive syndrome, where the hierarchy gradually becomes so complex that it becomes very difficult to find anything, except the documents a user has classified themselves.

With a federated graph structure, the Actian platform mirrors the data mesh at the metadata level. The mirror can be continually adapted to reflect the evolution of the data mesh's topology and the platform's architecture as they deploy throughout the organization.

Structured as an evolving federation of interconnected graphs, the Actian Data Intelligence Platform provides the ideal foundation for building the data mesh supervision system and supporting its long-term deployment. In this structure, the EDM is a simple subset of the graph, containing objects shared by the domains, especially the data products.

In an EDM, the ability to search for and discover data products is fundamental. However, it is only the first step in the user journey—with the subsequent steps being ordering and delivering the data to the consumer.

The EDM Shopping Experience

All traditional marketplaces offer a similar "checkout" experience that's familiar to many users. Selected products are placed in a cart, then, when validating the cart, the buyer is presented with various delivery and payment options. The actual delivery is usually done outside the marketplace, which simply provides tracking functionalities.

Delivery can be immediate, like for digital products, or deferred, for physical products. Some marketplaces have their own logistics system, but most of the time, delivery is the responsibility of the seller. The delivery time is an important element of customer satisfaction—the shorter it is, the more satisfied users are.

How does this shopping experience translate to an enterprise data marketplace? Answering this question requires looking at what data delivery means in a business context and focusing on the data consumer.

The Delivery of Data Products

A data product offers one or more consumption protocols. These are its outbound ports. The protocols may vary from one data product to another, depending on the nature of the data. Real-time data, for example, may offer a streaming protocol, while more static data may offer a SQL interface along with instructions for using the interface from various programming languages or in-house visualization tools.

For interactive consumption needs, such as in an application, the data product may also offer consumption APIs, which in turn may adhere to a standard, such as REST, GraphQL,or OData. Or, the consumption may be to simply download the data in a file format.

Some consumers may integrate the data product into their own pipelines to build other data products or for higher-level uses. Others may simply consume the data once, for example, to train a machine learning model. It's up to them to choose the protocol best suited to their use case.

Whatever protocols are chosen, they all have one essential characteristic, which is that they are secure. This is one of the universal rules of governance—access to data must be controlled and access rights must be supervised.

With few exceptions, the act of purchase, therefore, simply involves gaining access to the data via one of the consumption protocols.

Access Rights Management for Data Products

In the world of data, access management is not a simple matter for one elementary reason—consuming data can be a risky act. Some data products can be desensitized by removing personal or sensitive data that pose the greatest risk.

But this desensitization cannot be applied to the entire product portfolio. Otherwise, the organization forfeits the opportunity to leverage data that's highly valuable, such as sensitive financial or HR data, commercial data, market data, and customers' personal data. In one way or another, access control is a critical activity for the development and widespread adoption of the data mesh.

In the logic of data mesh decentralization, risk assessment and granting access tokens should be carried out by the owner of the data product who ensures its governance and compliance. This involves not only approving the access request but also determining any data transformations needed to conform to a particular use case. This activity is known as policy enforcement.

Evaluating an access request involves analyzing three dimensions:

1. **The what.** Evaluating the data itself. Some carries more risk than others.
2. **The who.** Looking at the requester, their role, and their location. Geographical aspects can have a strong impact, especially at the regulatory level.
3. **The why.** Understanding the purpose of the request.

Based on this analysis, the data may be consumed as is, or it may require transformation before delivery. This could include data filtering, especially for data not covered by consent, anonymization of certain columns, and obfuscation of others. Sometimes, additional formalities may need to be completed. For example, this could involve joining a redistribution contract for data acquired from a third party, or complying with retention and right-to-forget policies.

Technically, data delivery can take various forms depending on the technologies and protocols used. For less sensitive data, granting read-only access may suffice. This involves simply declaring an additional user. For sensitive data, fine-grained permission control is necessary at the column and row levels.

Most modern data platforms support native mechanisms to apply complex access rules through simple configuration, usually using data tags and a policy enforcement engine. Setting up access rights involves creating the appropriate policy or integrating a new consumer into an existing policy.

For older technologies that do not support sufficiently granular access control, it may be necessary to create a specific pipeline to transform the data to ensure compliance, store it in a dedicated space, and grant the consumer access to that space. This is a lengthy and potentially costly approach, which can be optimized by migrating to a data platform supporting a more granular security model or by investing in a third-party policy enforcement solution that supports the existing platform.

The main stages of data delivery in an EDM are:

- The consumer submits an access request, describing precisely their intended use of the data.
- The data owner evaluates the request. In some cases, the owner may rely on risk or regulatory experts, or require additional validations, to determine the required access rules.
- A data engineer sets up access. The complexity of this operation depends on the technologies being used.

Data Product Shopping in the EDM

Data product shopping for the consumer involves triggering the data delivery workflow from the EDM. Actian's EDM does not integrate this workflow directly into the platform, but rather interfaces with external solutions.

The platform offers a uniform experience when triggering an access request. It also realizes that processing a request may be very different from one environment to another, or even from one domain to another within the same organization.

This principle is inherited from classic marketplaces. Most offer a unique experience for making a purchase, but connect to other systems for the operational implementation of delivery. Decoupling the shopping experience from delivery is essential for several reasons.

The main reason is the extreme variability of the processes involved. Some organizations already have operational workflows that rely on another solution, such as data access requests integrated into a general access request process, supported, for example, by a ticketing tool such as ServiceNow or Jira.

Others have dedicated solutions supporting a high level of automation, but deployment is not yet widespread. Still others rely on their data platform capabilities, while others grant access through direct requests to the data owner, who handles them without a formal process.

This variability is evident from one organization to another. It's also evident structurally within the same organization, when different domains use different technologies or when the organization invests in a more efficient or secure system and gradually migrates access management to this new system.

Decoupling, therefore, offers a consistent experience to the data consumer while adapting to the variability of operational methods. For an EDM customer, the shopping experience is very simple. Once a data product is identified, the consumer triggers an access request by providing the following information:

- **Who they are.** This information is already readily available.
- **Which data product they want to access.** This information is also already available, along with the metadata needed for decision-making.
- **Intended use.** What they intend to use the data for. This is crucial because it drives risk management and compliance requirements.

With Actian, once the access request is submitted, it's processed in another system. Its status can be tracked from the EDM. This is the direct equivalent of order tracking that's found on e-commerce sites.

From a data consumer's perspective, the EDM provides a catalog of data products, along with other digital products, and a simple, universal system for gaining access to these products. For the producer, the EDM plays a fundamental role in managing their product portfolio.

A traditional marketplace offers tools dedicated to sellers, allowing them to supervise their products, respond to buyer inquiries, and monitor the economic performance of their offerings. Other tools, intended for marketplace managers, analyze the overall performance of products and sellers.

Actian's EDM integrates these capabilities into a dedicated back-office tool, Studio. It manages the production, consolidation, and organization of metadata in a private catalog and allows stakeholders to decide which objects will be placed in the marketplace, which is a searchable space accessible to the widest audience.

These activities primarily fall under the production process. Metadata is produced and organized together with the data products. Organizations can monitor the use of each data product by providing a list of all its consumers and the uses associated with them.

This consumer tracking helps establish the two pillars of data mesh governance:

1. **Compliance and risk management.** Conducting regular reviews, certifications, and impact analyses during data product changes.
2. **Performance management.** The number of consumers, and the nature of their uses, are the main indicators of a data product's value. A data product that's not consumed has no value.

As a support tool for domains to control the compliance of their products and their performance, Actian's EDM offers comprehensive analysis capabilities of the data mesh. They include lineage of data products, scoring and evaluation of their performance, control of overall compliance and risks, and regulatory reporting elements. This is the magic of the federated knowledge graph, which allows exploiting information at all scales and provides a comprehensive representation of the entire data asset.

Key Takeaways and a Path Forward

The data mesh is no longer just an intellectual curiosity, but a mainstream practice undergoing massive adoption. Implementation is primarily an organizational and cultural transformation. It does not require immediate investments in infrastructure.

There is no canonical architecture for a data mesh. Each organization adopts solutions that are unique to them, and these solutions will evolve over time. The implementation approach should be gradual and use lean manufacturing tools to properly focus efforts and investments by identifying and addressing bottlenecks.

Ensuring data mesh scalability requires a comprehensive supervision system that:

- Connects data producers and consumers.
- Provides consumers with marketplace capabilities to order data with an e-commerce type shopping experience.
- Gives producers the means to control and evaluate the performance of their products and manage their private data assets.
- Offers tools for global analysis of the data mesh for risk management and strategic prioritization.

With an EDM powered by a federated knowledge graph, the Actian Data Intelligence Platform mirrors the data mesh at the metadata level. It also builds a comprehensive and scalable supervision system perfectly integrated into data production and consumption processes.

These capabilities empower organizations to scale their data mesh initiatives with confidence, agility, and control. By taking a deliberate, structured approach, organizations can unlock the full value of decentralized data management and drive innovation.

About Actian

Actian empowers enterprises to confidently manage and govern data at scale. Organizations trust Actian data management and data intelligence solutions to streamline complex data environments and accelerate the delivery of AI-ready data. Designed to be flexible, Actian solutions integrate seamlessly and perform reliably across on-premises, cloud and hybrid environments. Learn more about Actian, the data division of **HCLSoftware**, at **actian.com**.

Tel +1 512 231 6000 Fax +1.512.231.6010 710 Hesters Crossing Road, Suite 250, Round Rock, TX 78681 **Actian.com**

ACTIAN™ a division of **HCLSoftware**