data.world

WHITE PAPER

An Actionable Framework for Governing the Data Mesh

Transform data into knowledge

data.world

Solving for scale in the modern enterprise

Two decades into the twenty-first century, the majority of enterprise organizations continue to struggle in their quest to derive true value from their ever-growing stores of data. Bottlenecks created by centralized data teams, monolithic architectures, and restrictive governance policies make it exceedingly difficult to access and use the data needed to solve critical business problems.

The rise of the data lake and cloud data warehouses have helped to address the issue, but technology alone cannot solve such a monumental challenge. A paradigm shift in how enterprise data is managed is the only way to move forward.

Data mesh emerged in 2019 with the potential to revolutionize enterprise data management as we know it. Popularized by Thoughtworks' Zhamak Dehghani, data mesh is a socio-technical approach that marries product thinking with a move towards domain-driven data management.

To date, much has been written about the philosophy behind the approach, but very little has been published in regard to establishing a framework for governing the data mesh. This white paper seeks to change that.

Read on for actionable advice that will enable you to:

- Establish a framework for treating data as a product
- Find the right balance of decentralization and centralization
- Transform data into knowledge that will benefit the whole of your organization

We hope this information will help kickstart your data mesh journey.

TABLE OF CONTENTS

What is data mesh?

Before diving into our unique perspective on data mesh governance, it is important to level-set on how the concept is defined by the market.

Data mesh is a new approach for organizations to collect, manage, and share data — a method of data management that empowers domain experts to own the data they create and make it available to consumers across business lines.

It is defined by four pillars:

- Domain ownership
- Data as a product
- Self-serve data infrastructure as a platform
- Federated computational governance

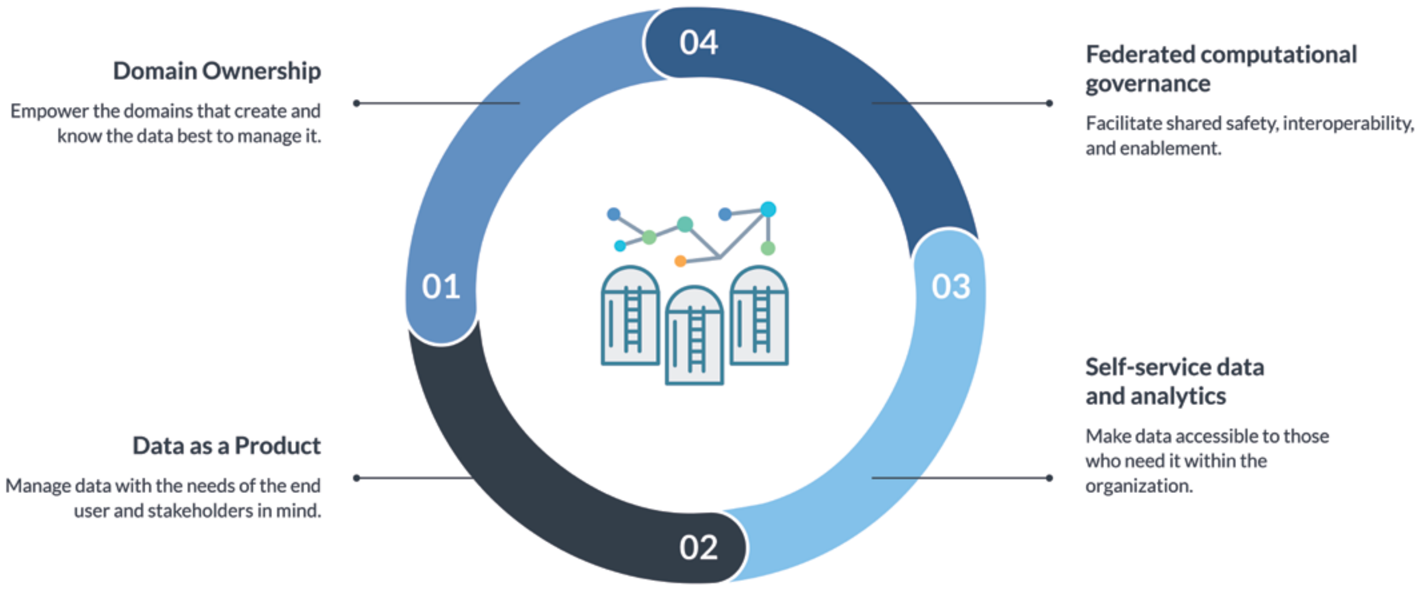
The first pillar, data ownership by domain, decentralizes data ownership and gives business domains (Sales, Marketing, Finance, etc.) ownership of the data they create. The benefit is that the domain's familiarity with the data will provide deeper insight into where, why, and how it should be used.

The second, treating data as a product, makes data discoverable, understandable and usable in the same way you search for and purchase products using your favorite e-commerce platform. Data is considered a product by each domain that publishes it. This approach empowers domain owners to become wholly responsible and accountable for their data, including its quality, representation, and cohesiveness. Under this model, other domains or parts of the business become internal customers of your data.

Self-service data and analytics — the third pillar — makes data accessible to members of the organization who need it to make informed business decisions. It simplifies data discovery and enables data democratization, making it quick and easy for anyone to surface relevant insights.

The fourth pillar, federated computational governance, is the least understood. Simply put, it establishes governance policies for each decentralized domain, ensuring all domain owners operate within a consistent framework. These policies should be computable — in code — in order for data products to be easily consumed.

Data Mesh Pillars



Domain Ownership
Empower the domains that create and know the data best to manage it.
Data as a Product
Manage data with the needs of the end user and stakeholders in mind.
Federated computational governance
Facilitate shared safety, interoperability, and enablement.
Self-service data and analytics
Make data accessible to those who need it within the organization.
01
02
03
04

Data mesh essentials: data.world's point of view

Now that we've addressed the market definition of data mesh, let's dive into our point of view.

First, as a socio-technical approach, we believe data mesh is as much about people and culture as it is technology. Yes, technology is integral, but it is not the defining characteristic.

Second, we believe that the two most important aspects of data mesh are:

1. Treating data as a product
2. Finding a balance between centralization and decentralization that works best for your organizational culture

Why you should treat data as a product

Treating data as a product means bringing product thinking to data management. This is already a common practice applied to software development: a product manager gathers requirements and use cases from end users and defines a roadmap and plan. The product team then executes on the plan, and tests, releases, and iterates in an agile fashion to improve that product. Product management teams care deeply about whether end users are getting value from the product, and consider factors like feasibility, usability, viability, maintainability, reusability, and scalability in their decision making. Data product management teams can serve the same role for data.

By establishing standards for data products, data product management enables domains and those serving in data ownership roles to curate high-quality data assets that are consumable by business and technical users alike.

For those new to product thinking, the Data Product ABCs framework provides insights into the types of questions data leaders should be asking when developing data products.

Accountability	• Who is the owner that is responsible for the data? • Who defines the requirements? • Who fixes it when it breaks?
Boundaries	• What is the data? • What isn't it? • Where will it live? • What are the inputs and outputs? • How do you balance that roadmap against other organizational priorities and considerations?
Contracts & Expectations	• What are the data constraints, definitions, and tests? • What are the SLAs and SLOs? • What are the sharing agreements, consented uses, and policies? • What is the purpose? • What is the performance and scale? • What are the quality and maintainability details? • How is it being kept secure and private? Who can see it? Use it?
Downstream Consumers	• Who are the current consumers? • Who are potential consumers? • What are the use cases that have been considered? • What is the value? • What is the roadmap of the data product? • How will it evolve to provide more value for consumers over time? • What is the user experience of the data? APIs, shape, access point?
Explicit Knowledge	• What is the meaning? • What is the schema? • How is it related to other data products? • Where is the documentation with examples?

Additionally, domain teams must maintain a consistent and usable interface to their data. Consumers should agree on the style of the interface as it pertains to their needs: well-defined tabular structure, API endpoint, SQL or SPARQL interface, Parquet, Graph, etc.

What's most important, regardless of the interface, is that the semantics – underlying logic – of the data products are the same. This includes keeping the Contracts and Expectations in place and notifying producers and consumers if something goes wrong.

While this may seem like a lot of effort, it's worth it to explicitly understand who is consuming which data and for what reason. Think of it in the context of ROI: just like an e-commerce platform tracks what you click and when you buy, you can use data products to identify which assets are the most popular and drive the greatest usage.

Takeaway: Consuming data to solve the crucial business problems should be as easy as buying a product on your favorite e-commerce platform.

Balancing decentralization and centralization

To cash in on the full value of a data mesh, it's imperative that your organization establish a documented method of federated data governance that balances centralization and decentralization. Let's first review the two ends of the spectrum.

"Top-down centralized governance" puts the responsibility for establishing and enforcing principles on senior-level employees who determine the form the organization's data governance should take. There are benefits to this approach, but the drawbacks include:

- Front-line data employees feeling excluded from the process
- Leadership enacting policies that don't work for data teams
- A lack of understanding from senior management who are several steps removed from the data itself
- Bottlenecks around data access and use that delays creation of crucial business insights

By contrast, "bottom-up decentralized governance" depends on solutions developed by the people and teams working most closely with the data. "Bottom-up" helps focus on concrete improvements to practical, day-to-day processes. However, it potentially removes the connection to business stakes understood by senior management. It also increases the risk of inconsistencies in governance between domains, negating the impact of governance efforts entirely. This reinforces silos between groups rather than breaking them down.

Data mesh disrupts the status quo of delegating ownership of all your organization's data to one team of highly specialized people who struggle to understand its value. Instead, decentralization gives ownership of the data to the domain team that knows and understands it best.

But let's be honest. We can't swing the pendulum from a centralized world to a decentralized world and expect that everything will work perfectly. It all depends on the current process and culture within your organization and the ones you aspire to have.

We believe the solution lies somewhere in the middle: conjunctive governance — encouraging input from experts working hands-on with your organization's data and developing global, inclusive, and resilient policies that work bottom-to-top. Execution of the process should be distributed across the domains, removing bottlenecks and breaking down silos.

Defining minimally viable governance

Domain decentralization only adds business value if all domains within an organization are governed according to documented global interoperability standards. That sounds like a big IF, but don't worry, we're not asking you to "boil the ocean." Instead, you should start by defining the minimal, core, and essential criteria that make up a data product across all decentralized domains.

This should start small and improve incrementally to assure safety and interoperability. The best ideas graduate up from the individual domains to become more broadly established standards. Without this process, there will likely remain a significant danger of miscommunication and misalignment in how your organization works with your data, and the result will inevitably lead to more and more data silos.

Putting the ABCs framework to the test

Following the Data Product ABCs framework for data products, we believe that A, B, and D are aspects that should be managed by each domain. C and E should be informed by global standards in order to ensure agreement on semantics, syntax, contracts, policies and access to data products.

The goal is to establish a standardized means of defining policies, contracts, and schemas instead of imposing them. This is done by making sure that everything is computable, i.e. it's code. **We believe in using SQL and well-established semantic web standards based on RDF, OWL, SHACL.** Of course, there are scenarios where imposition is necessary for security or regulatory purposes, as in GDPR.

Here are a few examples of what global standards can look like when applied to C and E in the ABCs framework:

Contracts and Expectations

- Select a language to define all the contracts and expectations: SHACL, Data Quality Vocabulary, Great Expectations
- Choose a standard architectural style (like Star Schemas or Data Vault for example)
- Define contracts (ex: telephone Number, must have a permission to send voicemail and text messages)

Explicit Knowledge

- Have a language to define schemas: OWL, JSON Schema, annotated or templated SQL DDL. Schemas defined for the core business concepts may appear in multiple domains. This way different domains won't define schemas for the same thing differently. Each domain can extend it if required.
- Separate data products for the core business concepts from data products of metrics. The latter are going to be combinations of the former. User data product + Activity data product = Metric data product. Each domain can have different definitions for a metric (ex: a user, may have mandatory and optional attributes).

The "domain-driven design" that finds the right balance between decentralization and centralization provides improvements in various areas:

1. **Scalability -** By distributing ownership of domains, you empower multiple teams to "own" specific functional areas and smaller data products. When your organization reaches a certain scale, and is producing and working with a large amount of data, a single team — or *gulp* person — holding the keys to all of it can become overwhelmed by your data's volume or complexity. Multiple teams working within a specific domain are more nimble and much more likely to have a complete understanding of the data within their purview.
2. **Efficiency -** You can quickly combine data products to create new insights, which can then lead to new data products. Say yes to reusing data and no to more bottlenecks.
3. **Resilience -** A domain-driven approach naturally means every domain is capable of managing the changes that may occur to specific operational systems and update data products accordingly, without having to depend on a central structure.
4. **Accountability -** When a specific domain has ownership of data, they take responsibility and respond to incentives. They naturally want their data products to be the most used and valued by the rest of the organization.
5. **Reliability -** Similar to the above, when a team or person is accountable for the quality of a data set — particularly when they're the expert — that data is much more likely to be organized, accurate, and error free.
6. **Usability -** When your data is organized, accurate, error-free, and "owned" by people who understand it completely, it's vastly more likely that it's easy to use. And if you do have questions, you know exactly which domain owners have the answers.

Takeaway: Centralize the things that are core to your business. Start small and iterate. Push everything else to the domains.

Leading with people

You can't have a socio-technical paradigm shift without people, right? So let's introduce two very important roles: Data Product Developers and Data Product Managers. These people are part of a domain team and take ownership of the data products.

Data product managers understand your business's data ecosystem as a whole, facilitate the conversations between all data consumers and data producers (those who own and run operational systems), and define the strategy, governance, and roadmap. They reduce the burden of data management across individual contributors and help data become more consistent, accessible, and accurate. A **data product developer** implements the data product within a domain with a key focus on knowledge and semantics.

Now you may be thinking that at some point, this level of decentralization will generate friction between data consumers and producers. Should they be using the data product of Domain A or Domain B? What happens if data is duplicated and we don't know which data to use? **We believe if this friction is occurring, it's a sign of success!**

Friction implies energy, and it's an opportunity to embrace complexity. For example, now you can improve the communication between different domains, or maybe you stumbled upon something that needs to be standardized globally. This is why the global standards must be developed in consensus by data product managers from each domain.

If there is no friction, that can mean two things: everything is working well so leave it alone, or no one is using it, so there's no need to spend time on it.

Takeaway: Focus your energy where there is organizational energy. People are what make data mesh a socio-technical approach. You can't, nor should you want to, automate people away.

Applying agile data governance to the data mesh

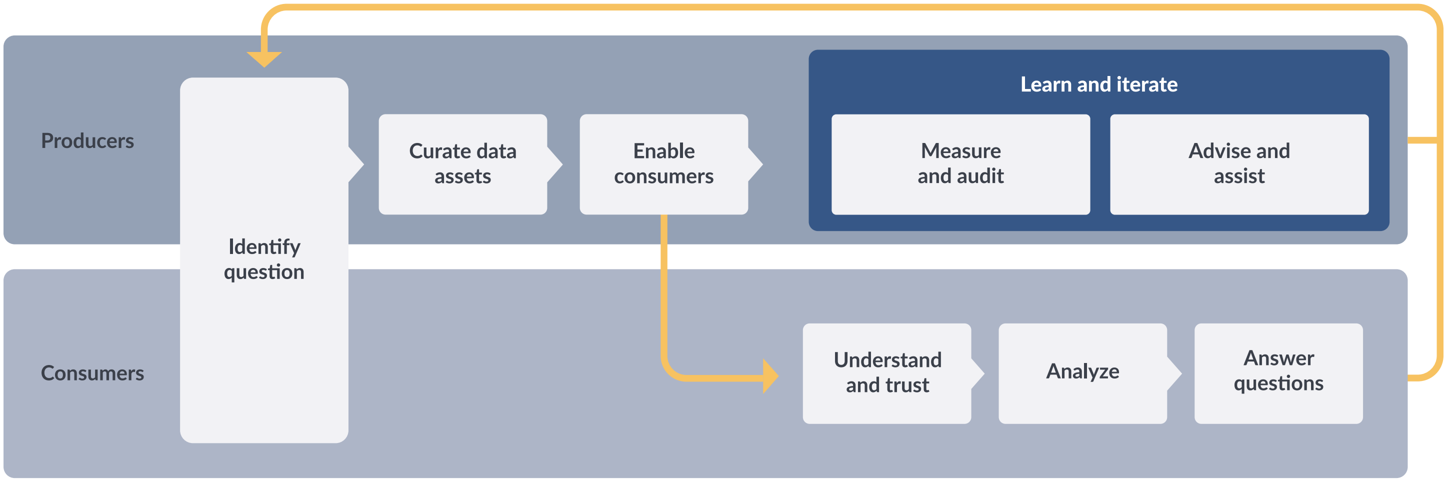
Agile Data Governance is the process of improving data products by iteratively capturing knowledge from data producers and consumers as they work together so everyone can benefit.

When bringing an agile approach to data mesh governance, the first step is to establish and document the ABCDE framework that will guide your initiative. Once these are set, you're ready to embark on an agile data sprint, where you analyze strategic business initiatives and use cases.

Your first sprints should prioritize use cases involving known personas and their domain and a small number of data sources. This gets data into the hands of end consumers fast so you can immediately begin to measure the impact of your project and iterate for future use cases.

It's important to build a use-case backlog — a prioritized list of the common business questions you want your program to solve — and perform a retrospective and document what you learn after each use case implementation (i.e. where is the friction?). This methodology will provide fast and detailed insight into what went right, what went wrong, and what you can do to improve future use-case implementations.

The build-measure-learn loop of Agile Data Governance



The role of a modern data catalog in a data mesh

Up to now, we have focused on the people, process, and culture aspects of data mesh. Before we dive into the technical stuff, we should be clear about one thing: if someone tries to sell you a "data mesh platform," run away as fast as possible! You CANNOT buy a data mesh.

Remember, before you even think about technology, you first need to understand how your business culture is going to treat data as a product and find the balance between centralization and decentralization. Technology is needed for support, but it is not the solution.

We believe that the modern data catalog plays a pivotal role in a data mesh.

A modern data catalog must have two key attributes to support a data mesh:

- It must cater to data producers and data consumers
- It must be powered by a knowledge graph

Here's why:

- **Data producers and data consumers:** A modern data catalog should have different lenses for different personas. A data producer is a technical user on a data product development team who needs to understand the existing operational systems that have been cataloged. They will use technical features such as data lineage, sensitive data scanning, rest APIs, etc. Once they have generated a data product, it also needs to be registered in the data catalog. A data consumer is going to use the modern data catalog to discover data products, understand the policies, etc. Different personas need a catalog that caters to different experiences.

- **Powered by a knowledge graph:** Metadata is intrinsically connected and best represented as a knowledge graph.

  A knowledge graph makes it easier for data producers to represent the metadata, schemas, contracts, and policies. Additionally, a modern data catalog powered by a knowledge graph is powerful because it can be queried by any user.

Relationships between data products are automatically manifested in the knowledge graph which speeds the discovery process for data consumers. This is why knowledge graphs power the search and recommendation engines used by Google and Amazon. Remember, treating data as a product implies that you should strive to have the same experience you go through when you search and buy a product on your favorite e-commerce platform.

This is how Zhamak Deghani defines a 'knowledge-graph' interface on the mesh experience plane: "Browse the mesh of related data products' semantic models. Traverse their semantic relationship to identify the desired sources of data."

Conclusion

Striking the right balance between decentralization and centralization is a crucial step towards meeting your business goals. To find this balance, follow the ABCs framework for data products and use SQL, as well as established semantic web standards based on RDF, OWL, SHAC to ensure ease of adoption.

As mentioned above: there will be friction. Embrace it! Friction is a sign of success.

Applying the Agile Data Governance approach to Data Mesh will result in ongoing iterative learning across your organization. But in order to do so, you need a data catalog that supports data producers and consumers, and *is powered by a knowledge graph*.

When you treat data as a product and you find the right balance of decentralization and centralization, data is transformed into knowledge and opportunities for everyone in your organization.

Schedule a demo to learn more about how data.world supports data mesh implementation.

About data.world

data.world is the enterprise data catalog for the modern data stack. Our cloud-native SaaS platform combines a consumer-grade user experience with a powerful knowledge graph to deliver enhanced data discovery, agile data governance, and actionable insights. data.world is a Certified B Corporation and public benefit corporation and home to the world's largest collaborative open data community with more than 1.5 million members. Our company has close to 50 patents and has been named one of Austin's Best Places to Work six years in a row. Follow us on LinkedIn, Twitter, and Facebook, or join us.

data.world