**VIETTEL GROUP - VIETTEL DIGITAL TALENT PROGRAM**
VIETTEL DIGITAL TALENT RESIDENCY 2023
SOFTWARE & DATA ENGINEERING SECTOR

**PERIOD 1**

MINI-PROJECT REPORT

Theo cách của bạn

# DATA MESH ARCHITECTURE

## MINI-PROJECT REPORT - TOPIC 8

DATA ENGINEERING SPECIALIZATION

**RESIDENT: Tạ Ngọc Minh**
BSc. student in Data Science & AI @ HUST

**SUPERVISOR: Mr. Nguyễn Chí Thanh**

**Hanoi - 2023**

# PREFACE

Dear the board of directors of Viettel Group, the mentors, and all of my friends,

This document is a part of my research mini-project as a data engineer intern for the first period of Viettel Digital Talent 2023 Program. The completion of this research report does not mean the end of my duties as learners. In fact, this report is a first step towards consistently studying what I have reviewed and written here.

Data has grown up so quickly recently. The term 'big data' has become familiar with every tech-lovers is the clearest evidence of the explosion of data. This leads to many requirements of upgrading the current system and management, along with finding new architecture to store, process and analyze data. Once the central data lake has been overloaded, we need to design a decentralized architecture. This report is written under the research about the Data Mesh architecture, a new design that was first defined by Zhamak Dehghani in 2019.

To write this document, I would like to give many thanks to Viettel Group, and Viettel Digital Talent program for giving me chance to do my research. I want to send a big thanks to Mr. Nguyễn Chí Thanh, and all the mentors of the first period of the program, who gave me many useful lectures and guidance. Also, I want to say thanks to professor Huỳnh Thị Thanh Bình, professor Đỗ Tuấn Anh for their initial supports, and many other professors at the School of Information and Communication Technology, Hanoi University of Science and Technology about their public documents about data engineering. Last, I would like to say thanks to all of my friends, who always read my documents, comment about them and support me whenever I need.

Because this document is written in such a short time and some of the concepts are so new and hard to understand for a fresher like me, it is so hard to make sure it is completely correct, including the grammar errors and formatting errors in LaTeX. I would love to receive your contributions via **ngocminhta.nmt@gmail.com**.

Thank you for reading this report and looking forward to getting many comments from you.

**Ngoc-Minh Ta**

# CONTENTS

# LIST OF FIGURES

# LIST OF TABLES

# Chapter 1

# Problems and Initial Approach

## 1.1 Real-life Problem

### 1.1.1 A problem from history

Many organizations have invested in building a central data lake. To make data-driven business, there is a central data administration team take the responsibility for this.

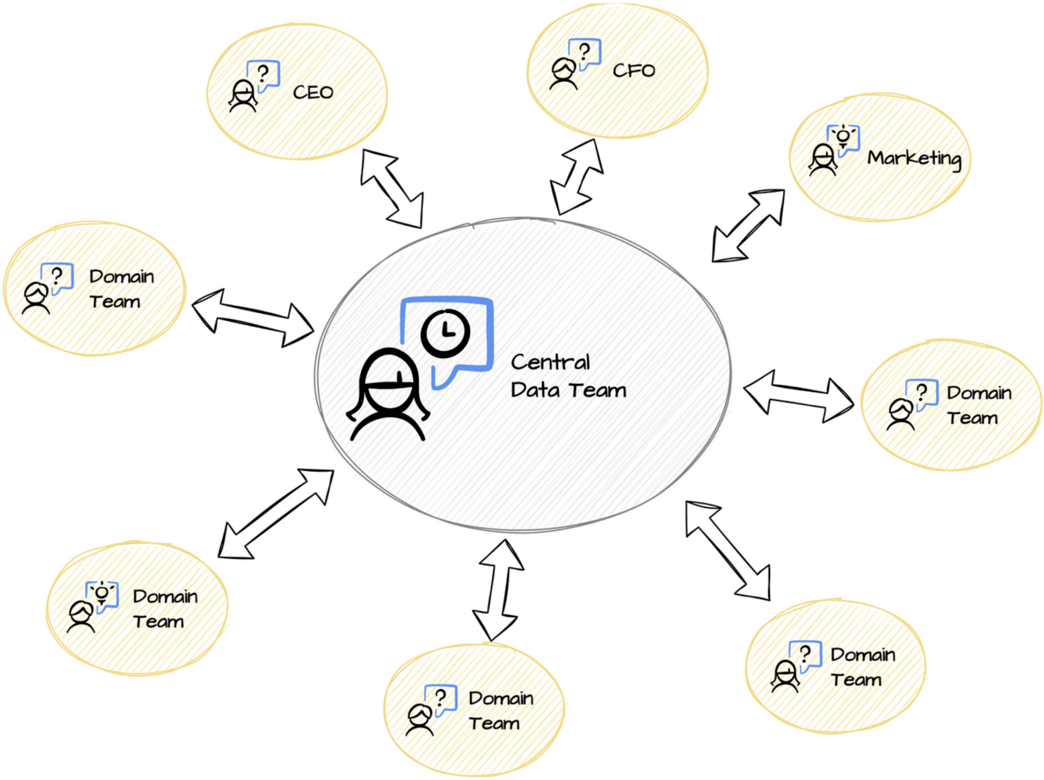

Figure 1.1: Central data team in a data-driven business

This model works quite well at the first time, however, after a while, they noticed that **the team often became bottleneck**. The team were overloaded with thousands of tasks and questions from multiple teams, along with the requests of time and accuracy. This lead to a massive problem since the competitiveness of business depends much on the speed of analysis. For example, when we create a landing page for a new product, how does this change the influence clicking and buying rate?

But, why the response of the central data team is so slow and struggle? Actually, once the operational database changes, the team has to spend much of their time fixing the broken data pipelines. The little remaining time is insufficient for them to discover and understanding all the necessary domain data **for each question**. Getting the required domain expertise is a daunting task. [1, 2]

In contrast, some firms employ domain-driven design strategies that include independent domain teams and a decentralized micro-service architecture to relieve burden on the central data team. These teams exclusively control and understand their domain, including the company's information demands. The issue hasn't entirely been resolved, though. The domain teams must contact the overloaded central data team despite being aware of the key information needs and the domain, in order to obtain the essential data-driven insights. [1]

### 1.1.2 Demands from recent developments

Let's consider the scale-up of the software development over time. When one task grow bigger and bigger, we have to decentralize it into many smaller tasks, otherwise, all the organization will overload and lost of control. Take into consideration what we have done for the scale-up of software development:

- Decentralize business into domains;
- Decentralize engineering into autonomous teams;
- Decentralize monolith into micro-services;
- Decentralize operations into DevOps teams.

Hence, the next thing we need to do is scaling up data analytics by decentralizing data lake into data mesh.[2]

The position between domain teams and the central data team gets worse as the organization eventually grows. Transferring data management responsibilities from the central data team to the domain teams can help. Domain-oriented decentralization for analytical data is the central notion of the data mesh concept. Similar to APIs in a micro-service design, a data mesh architecture allows domain teams to conduct cross-domain data analysis on their own and connects data.

## 1.2 INITIAL APPROACH

### 1.2.1 What is Data Mesh?

The term data mesh was first stated by Zhamak Dehghani in 2019 and is based on four fundamental principles, which will be discussed in depth later:

- Principle of Domain Ownership;
- Principle of Data as a Product;
- Principle of the Self-Serve Data Platform;
- Principle of Federated Computational Governance.

### 1.2.2 What will changes after data mesh?

Data mesh is a fresh method for business intelligence, data analysis and management that is based on an innovative distributed architecture. Along with that, data lake and data warehouse do not disappear they just become nodes in the mesh. [3, 4]

Data mesh ensures organizations to continue to apply some data lake principles, such as making immutable data available for exploration or analytical use, and data lake tooling for internal implementation of data products or as part of the shared data infrastructure. Data lake would not be the centerpiece anymore.

It is an implementation detail sub-serving the idea of domain data product as the first-class concern. The same applies to data warehouse in terms of business reporting and visualization. [4]

Implementing a data mesh is not a purely technical project that we can implement in isolation from the rest of the business. It's not some thing that we can just start with, and then it works from the first try, but we have to grow and develop with it. It also cause a cultural shift in the organization.

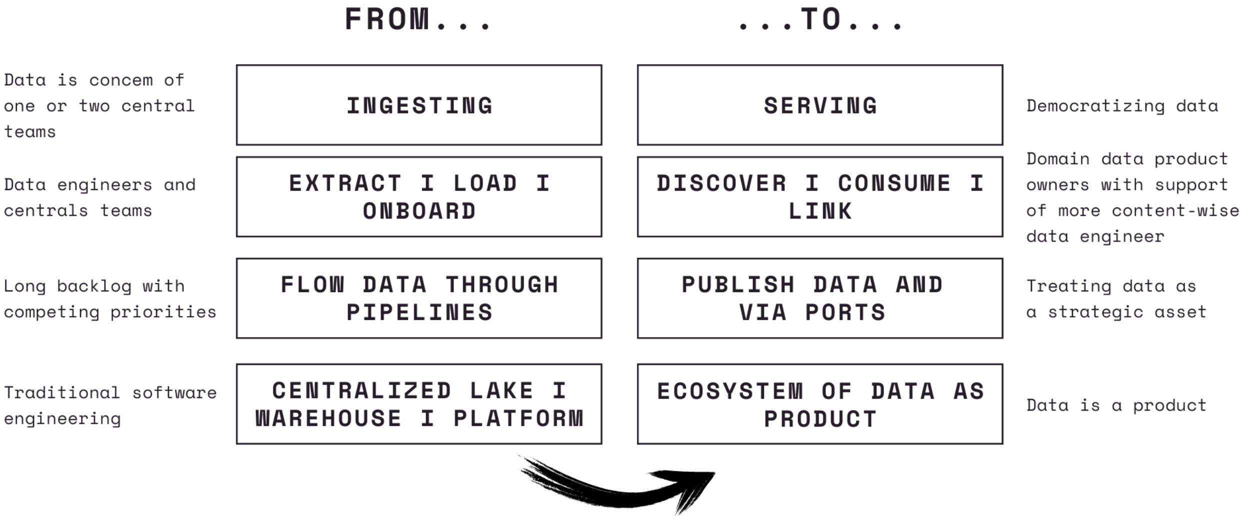
FROM...
...TO...
Data is concem of one or two central teams
INGESTING
SERVING
Democratizing data
Data engineers and centrals teams
EXTRACT I LOAD I ONBOARD
DISCOVER I CONSUME I LINK
Domain data product owners with support of more content-wise data engineer
Long backlog with competing priorities
FLOW DATA THROUGH PIPELINES
PUBLISH DATA AND VIA PORTS
Treating data as a strategic asset
Traditional software engineering
CENTRALIZED LAKE I WAREHOUSE I PLATFORM
ECOSYSTEM OF DATA AS PRODUCT
Data is a product

Figure 1.2: Cultural shift after implementing data mesh

Also, data mesh calls for a fundamental shift in the assumptions, architecture, technical solutions, and social structure of our organizations, that means, how we manage, use and own analytical data. [5]

Data mesh can be used as part of an enterprise data strategy, articulating the target state of the enterprise architecture as well as an organizational operating model with an iterative execution plan.

In its most basic form, it can be characterized by four interacting principles, which will be discussed in depth in section 2.1.

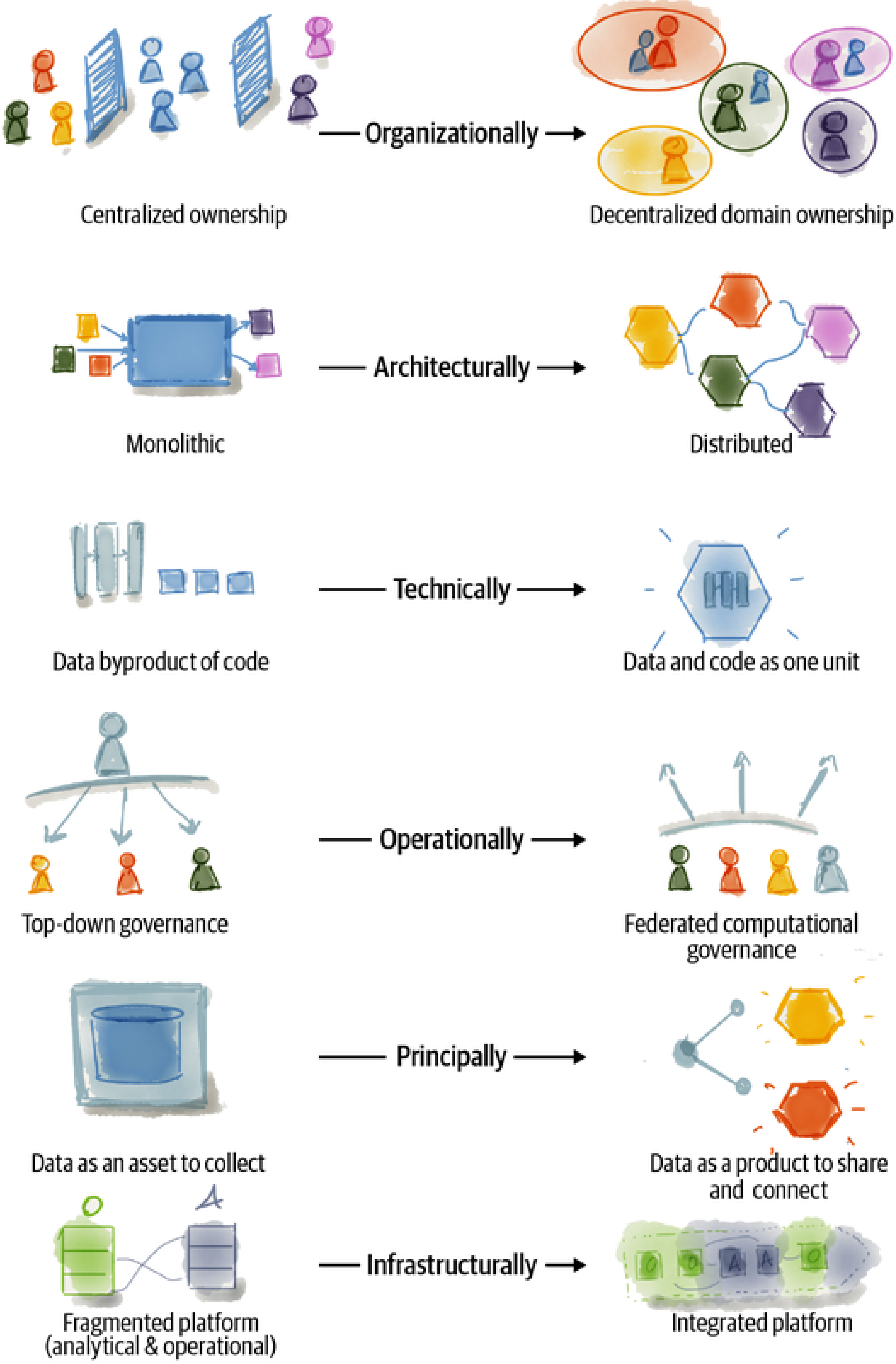
Centralized ownership
Organizationally
Decentralized domain ownership
Monolithic
Architecturally
Distributed
Data byproduct of code
Technically
Data and code as one unit
Top-down governance
Operationally
Federated computational governance
Data as an asset to collect
Principally
Data as a product to share and connect
Fragmented platform (analytical & operational)
Infrastructurally
Integrated platform

Figure 1.3: Data mesh dimensions of organizational changes

# Chapter 2

# Data Mesh Architecture Design

## 2.1 Four Fundamental Principles of Data Mesh

Four simple principles can represent the logical architecture and operating model of data mesh. They are intended to move us closer to the goals of data mesh: increasing the value of data at scale, maintaining agility as an organization grows, and embracing change in a complex and turbulent business context.

### 2.1.1 Principle of Domain Ownership

Data mesh, at its core, is founded in decentralization and distribution of data responsibility to people who are closest to the data. This is to support a scale-out structure and continuous and rapid change cycles.

However, in contrast to traditional data structures with technological partition (e.g. data warehouse, data lake), data mesh follows *the seams of organizational units*. It follows the lines of division of responsibility aligned with the business using *domain-driven design (DDD) strategies*. Data mesh also gives the data sharing responsibility yo each of the business domains, each domain is responsible for the data it is most familiar with.

**Domain-Driven Design (DDD) Strategies to Data**

DDD is an approach to decomposition of software design and team allocation, based on the seams of a business. It defines a *domain* as "a sphere of knowledge, influence or activity."

DDD’s Strategic Design embraces modeling based on multiple models each contextualized to a particular domain, called a bounded context[1]. As Z. Dehghani’s recommendation, data mesh adopts the boundary of bounded contexts to individual data products - data, its models, and its ownership. For some organizations have built services based on the domain bounded contexts, now, they apply the same decomposition and modeling to their analytical data in each domain.

Domain data ownership is the foundation of scale in a complex system like enterprises to-

[1] A bounded context is the delimited applicability of a particular model that gives team members a clear and shared understanding of what has to be consistent and what can develop independently. [6]

day. When we map the data mesh to an organization and its domains, we discover a few different archetypes of domain-oriented analytical data, then we defined them as:

- Source-aligned domain data (native data product): Analytical data reflecting the business facts generated by the operational systems.
- Aggregate domain data: Analytical data that is an aggregate of multiple upstream domains.
- Consumer-aligned (fit-for-purpose) domain data: Analytical data transformed to fit the needs of one or multiple specific use cases.

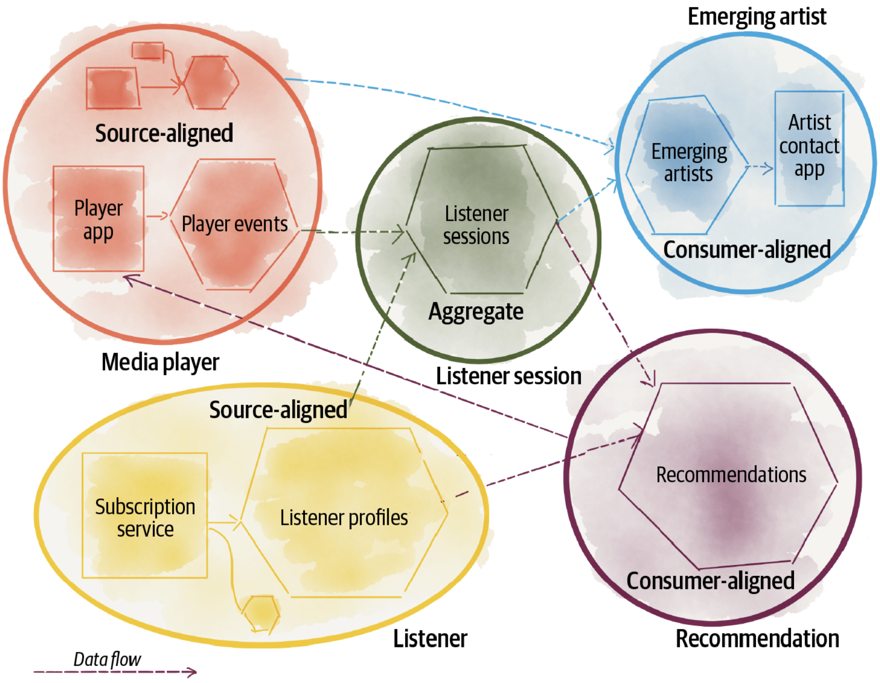

Figure 2.1: Decomposing the analytical data ownership and architecture, aligned with business domains, along with their data archetypes.

### Transition to Domain Ownership

The shift toward domain-oriented data ownership leads to accepting and working with real-world messiness of data, particularly in high-speed and scaled environments:

- **Push Data Ownership Upstream:** Data can be consumable and useful right at the source analytical domain (source-aligned domain data). At a later point downstream, source-aligned domain data can be aggregated and transformed to create a new higher order insight (aggregate domain data or fit-for-purpose analytical data).
- **Define Multiple Connected Models:** We implement multiple models of polysemes[2]. In data mesh, each domain can model its data according to its context, share this data and its models with others, and identify how one model can relate and map to others.
- Data mesh does not enforce the idea of searching for the single source of truth for each shared business concept. However, it places multiple practices in place that reduces

[2] Polysemes are shared concepts across different domains. They point to the same entity, with domain-specific attributes. [7]

the likelihood of multiple copies of out-of-date data. Long-term domain-oriented ownership with accountability to share discoverable, high-quality, and usable data in multiple modes for analysts and scientists.
- **Hide the Data Pipelines:** Data pipelines are first-class architectural concerns in traditional data architectures, composing more complicated data processing and transportation. A data pipeline in data mesh is just an internal implementation of the data domain that is handled within the domain.

Domains are taking up extra data responsibilities with data mesh. To acquire agility and authenticity, responsibilities and efforts transfer from a centralized data team to domains. Don't try to design a business's domains ahead of time and then assign and model analytical data based on that. Instead, begin by working with your company's seams as they are. We should allow data evolution to shape the shape of the organization, and vice versa. [7]

### 2.1.2 Principle of Data as a Product

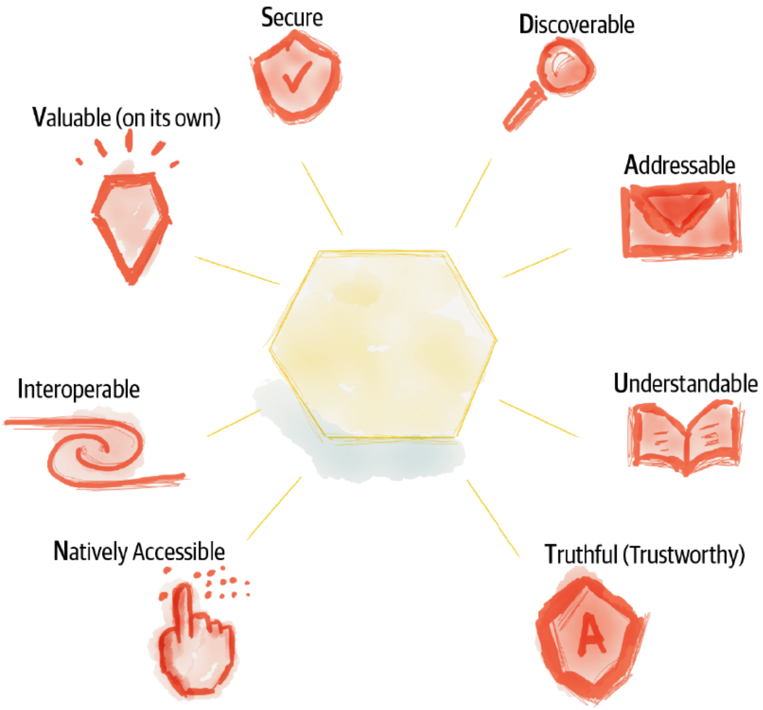

Figure 2.2: The baseline usability attributes of data products (DAUTNIVS)

The principle of data as a product is a response to the data siloing challenge that may arise from the distribution of data ownership to domains. It is also a shift in the data culture toward data accountability and data trust at the point of origin. The ultimate goal is to make data simply usable.

In approach to data product, there is a set of non-negotiable baseline characteristics in figure 2.2 that a data product incorporates to be considered useful. These characteristics apply to all data products, regardless of their domain or archetype. We call these baseline data product usability attributes. Every data product incorporates these characteristics to be part of the mesh. These are an addition to what has been known as **FAIR** data in the past—data that meets the principles of **D**indability, **A**ccessibility, **I**nteroperability, and **R**eusability. [8]

The introduction of analytical data as a product adds to the list of existing responsibilities of cross-functional domain teams [9] and expands their roles to:

- Data product developer: The role responsible for developing, serving, and maintaining the domain's data products as long as the data products live and are being used. They will be working alongside their fellow application developers in the domain.
- Data product owner: The role accountable for the success of a domain's data products in delivering value, satisfying and growing the data users, and maintaining the life cycle of the data products. He assure continuity of ownership of data and accountability of success metrics such as data quality, decreased lead time of data consumption, and in general data user satisfaction through net promoter score.

Define these roles for each domain and allocate one or multiple people to the roles depending on the complexity of the domain and the number of its data products. Moreover, to make a data as a product, we need to satisfy these conditions:

- Reframe the Nomenclature: Data mesh suggests reframing receiving upstream data from ingestion to consumption. The minor distinction is that the upstream data has already been cleansed, processed, and is ready for consumption.
- Think of Data as a Product, not a mere asset.
- Establish a Trust-But-Verify Data Culture: The data as a product principle entails a variety of actions that contribute to a culture in which data users can trust the veracity of the data and focus on proving its suitability for their use cases. Data-as-a-product practices strive to create a new culture, moving away from presumption of guilt.
- Join Data and Compute as One Logical Unit: Coexistence of data and code is not a novel concept. The expansion of operational systems has resulted in a model in which each service handles its own code and data, as well as schema definition and upgrades. The link between the code and its data distinguishes an operational system.

### 2.1.3 Principle of the Self-Serve Data Platform

Data mesh's principle of a self-serve platform comes to the rescue to lower the cognitive load that the other two principles impose on the existing domain engineering teams: own your analytical data and share it as a product.

It shares common capabilities with the existing data platforms: providing access to polyglot storage, data processing engines, query engines, streaming, etc. However, it differentiates from the existing platforms in its users: autonomous domain teams made up primarily of generalist technologists. It manages a higher-level construct of a data product encapsulating data, metadata, code, and policy as one unit.

Its purpose is to give domain teams superpowers, by hiding low-level complexity behind simpler abstractions and removing friction from their journeys in achieving their outcome of exchanging data products as a unit of value. And ultimately it frees up the teams to innovate with data. To scale out data sharing, beyond a single deployment environment or organizational unit or company, it favors decentralized solutions that are interoperable.

### 2.1.4 Principle of Federated Computational Governance

Some of the common questions about the feasibility of data mesh revolve around governance concerns, now in a decentralized manner. How can we make sure individual data products comply with a set of common policies that make them secure, compliant, interoperable, and trustworthy? Particularly, how can we have any guarantees when each domain owns and controls its own data products and there is no longer a central team getting its arms around the data? What happens to the centralized governance team?

The answer to these questions lies within the data mesh governance model, called federated computational governance. Data mesh, like its predecessors the lake and warehouse, meets a similar set of governance objectives. But it differs in its operating model and how these objectives are met.

The data mesh governance model consists of three complementary pillars. First and foremost, it requires systems thinking, looking at the mesh as an ecosystem of interconnected data product and platform systems, and their independent and yet connected teams. Then try to find the leverage points and feedback loops to control the behavior of the mesh as a whole toward its objective, creating value through sharing data products at scale.

Second, apply a federated operating model. From the social and organizational per-spective, create a federated team of individual domains and platform representatives. Create incentives that are aligned with both domains' data product success as well as the success of the wider ecosystem. Let domains have autonomy and responsibility for the majority of policies that are in the sphere of their influence and control, while leaving cross-functional and a small set of policies to be defined globally.

Finally, from a practical and implementation perspective, data mesh governance heavily relies on embedding the governance policies into each data product in an automated and computational fashion. This of course heavily relies on the elements of the underlying data platform, to make it really easy to do the right thing. To bring these three pillars together, Figure 2.3 shows an example of this model.

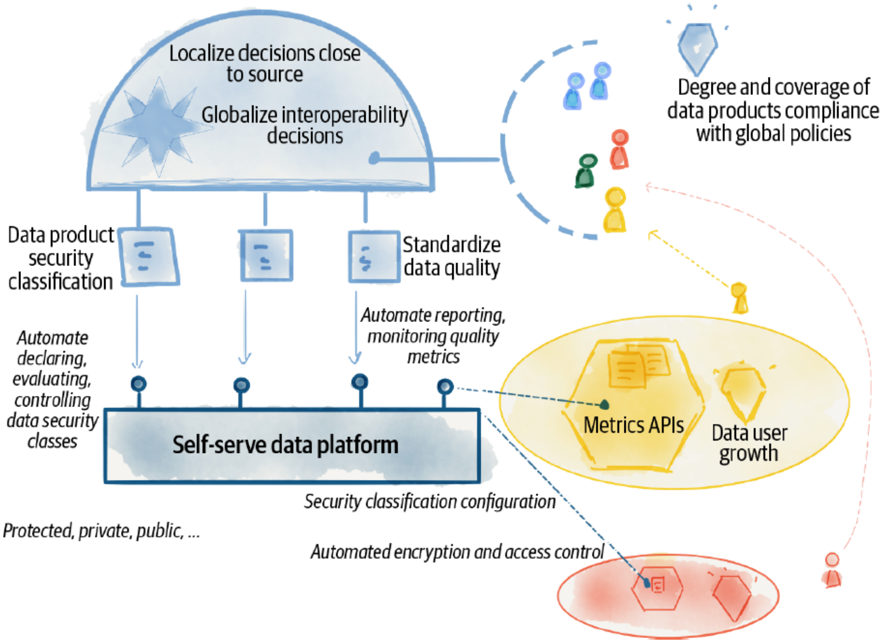
Localize decisions close to source
Globalize interoperability decisions
Degree and coverage of data products compliance with global policies
Data product security classification
Standardize data quality
Automate reporting, monitoring quality metrics
Automate declaring, evaluating, controlling data security classes
Self-serve data platform
Metrics APIs
Data user growth
Security classification configuration
Protected, private, public, ...
Automated encryption and access control

Figure 2.3: Example of data mesh governance operating model

Data mesh governance seeks to improve the existing approach to data governance at the intersection of a decentralized value system meeting automation and computation.

Data mesh governance is the method by which we define and apply what is right and how to do the right thing in an ecosystem of data product teams. The definition of what is right and how to do the right thing has occupied the great minds for centuries, from Aristotle in Ancient Greece to the German Idealism era to modern political systems. As human beings we have wrestled with the definition and regulation of what is right somewhere in between common good and utilitarianism—approaches that achieve maximum benefit and happiness of the community—and individualism, which is prioritizing an individual's happiness and liberty and doing what is right for them.

While data mesh is not a field of philosophy, it wrestles with a similar dilemma: how to do the right thing while maintaining the autonomy, freedom, and individualism of different domains and also achieving the greater good through consistency and standardization across all data products. It defines an approach to data governance that attempts to continuously find the dynamic equilibrium between localization of decisions so that the domains can go fast versus globalization and centralization of decisions for everyone to go far.

I do think catalysts such as computational governance and dual-system incentive structures can lead to domain behaviors that ultimately result in the greater good of the mesh ecosystem. However, the mesh will not reach an optimal state unless the majority of domains become intelligently augmented—with embedded ML-based systems in each of their products and systems. The continuous need for trustworthy and useful data across multiple domains to train ML-based solutions will be the ultimate motivator for the adoption of data mesh governance and doing the right thing.

## 2.2 Why we need Data Mesh?

Summarize recap of chap 678

## 2.3 Data Mesh Architecture Design

### 2.3.1 The Logical Architecture

### 2.3.2 The Multiplane Data Platform Architecture

# Chapter 3

# DATA PRODUCT DESIGN & IMPLEMENTATION

# Chapter 4

# DATA MESH IN USE

## 4.1 DATA MESH IN COMBINATION WITH DATA LAKEHOUSE

## 4.2 FRAMEWORKS AND TECHNOLOGIES FOR DATA MESH

## 4.3 CASE STUDY AND DEMO

# References

[1] Z. Dehghani, “Prologue: Imagine data mesh,” in *Data Mesh: Delivering Data-Driven Value at Scale*. O’Reilly Media, 2022, pp. xxv–xxxviii.

[2] C. Jochen, V. Larysa, and S. Harrer. (2023) Data mesh from an engineering perspective. [Online]. Available: https://www.datamesh-architecture.com/

[3] I. A. Machado, C. Costa, and M. Y. Santos, “Data mesh: Concepts and principles of a paradigm shift in data architectures,” *Procedia Computer Science*, vol. 196, pp. 263–271, 2022.

[4] D. Nicolas. (2023) Will the buzz around data mesh really help solve my data challenges? [Online]. Available: https://kpmg.com/be/en/home/insights/2023/03/lh-the-impact-of-data-mesh-on-organizational-data.html

[5] Z. Dehghani, “Data mesh in a nutshell,” in *Data Mesh: Delivering Data-Driven Value at Scale*. O’Reilly Media, 2022, pp. 3–14.

[6] E. Evans, *Domain-Driven Design: Tackling Complexity in the Heart of Software*. Addison-Wesley Professional, 2003.

[7] Z. Dehghani, “Principle of domain ownership,” in *Data Mesh: Delivering Data-Driven Value at Scale*. O’Reilly Media, 2022, pp. 15–28.

[8] M. D. Wilkinson *et al.*, “The fair guiding principles for scientific data management and stewardship,” *Scientific data*, vol. 3, no. 1, pp. 1–9, 2016.

[9] Z. Dehghani, “Principle of data as a product,” in *Data Mesh: Delivering Data-Driven Value at Scale*. O’Reilly Media, 2022, pp. 29–46.