Open in app ↗ Sign up Sign In

Published in The PayPal Technology Blog

Jean-Georges Perrin Follow
Aug 3, 2022 · 12 min read · ▶ Listen

Save

The next generation of Data Platforms is the Data Mesh

Here’s why.

As enterprises become more agile, centralization appears more and more as a thing of the past world, a waterfall world. The same appears to be true with data platforms. Therefore, we are building a Data Mesh, this next generation of data platforms for PayPal Credit. This post details the evolution of data platforms, highlights their problems, and why we decided to build a Data Mesh. I will detail the four principles of the Data Mesh, how to get started, look at the architecture, and describe some of the challenges.

Evolution of data platforms

Before diving in the details of the Data Mesh, let’s review how the information industry came to this situation.

Sixty-five million years ago, dinosaurs… no, I will not go that far away in time. In 1971, Edgar Codd invented the third normal form, the key to relational databases. Soon after that, enterprises starting seeing the benefits of aggregating data, which opened the way to the creation of data warehouses.

With data warehouses came the need for more rigor in data management: you were creating a warehouse for data, so like in a logistics warehouse, aisles, shelves, and spaces must be clearly identified the data warehouse to

1.2K 10

accommodate incoming data and build ETL (extract, transform, and load) processes to fill the warehouse. Enterprises were now capable to perform analytics to a new dimension. Unfortunately, warehouses are not very flexible and with the increasing number of data sources, onboarding data became complex.

Let's imagine a retail company, Great Parts, with B2C and B2B activities. They have a few thousand stores across North America, a loyalty program, and they accept returns.

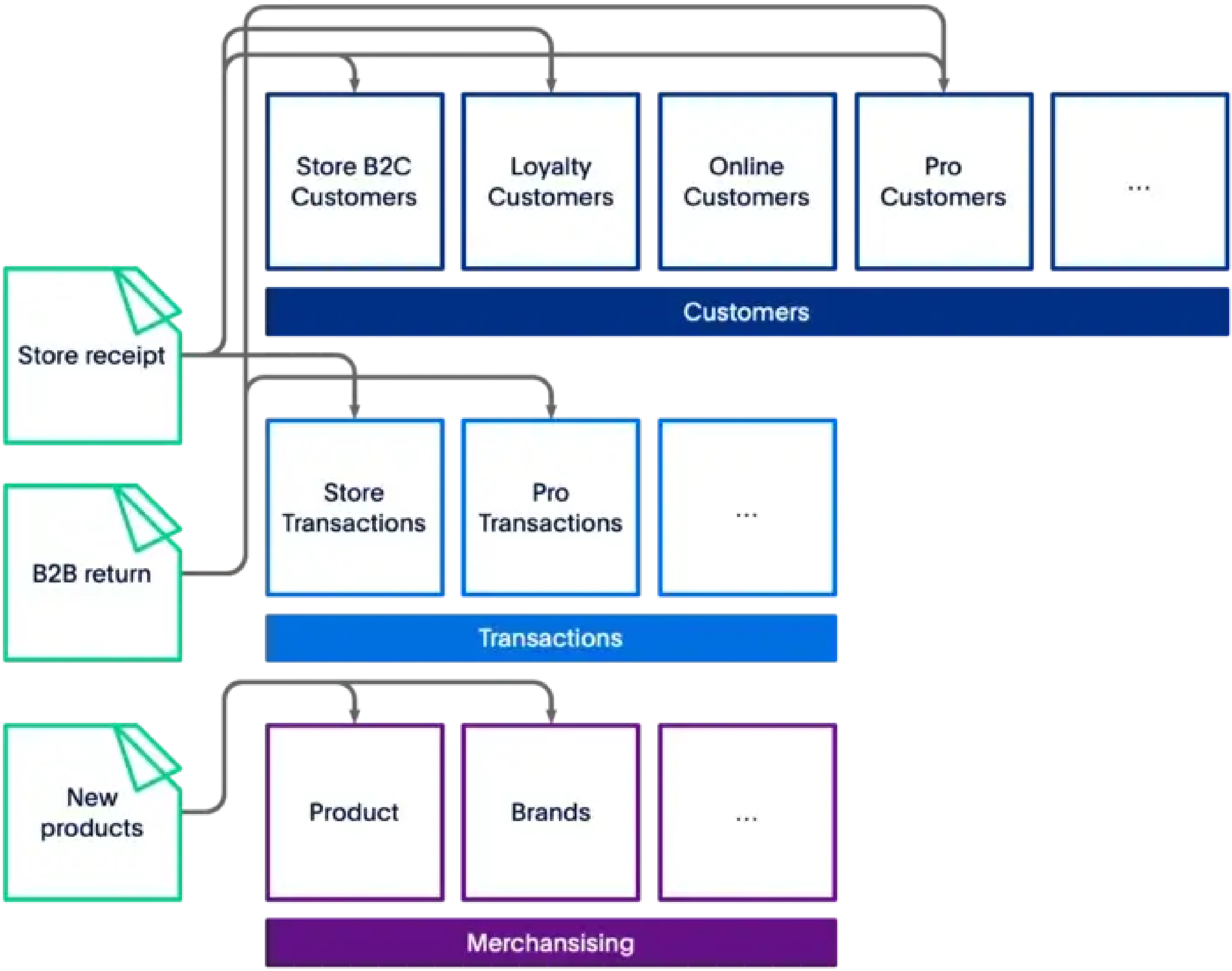

Figure 1 — Dataflows for new receipts, returns, and new products.

Now imagine Great Parts decides to expand their loyalty program to B2B; you will have to build an ETL process between the B2B returns and your customer space. If you are planning on adding a new data source like clickstream from your mobile application, your web applications, and your B2B sites. It will be increasingly complicated, and you will have to manage the ETL spaghetti.

As often in our industry, Great Parts decided to completely shift from the data warehouse to a data lake. The pendulum just shifted drastically. In a data lake, you collect all the data you want and store it. Wherever. You can see where this is going.

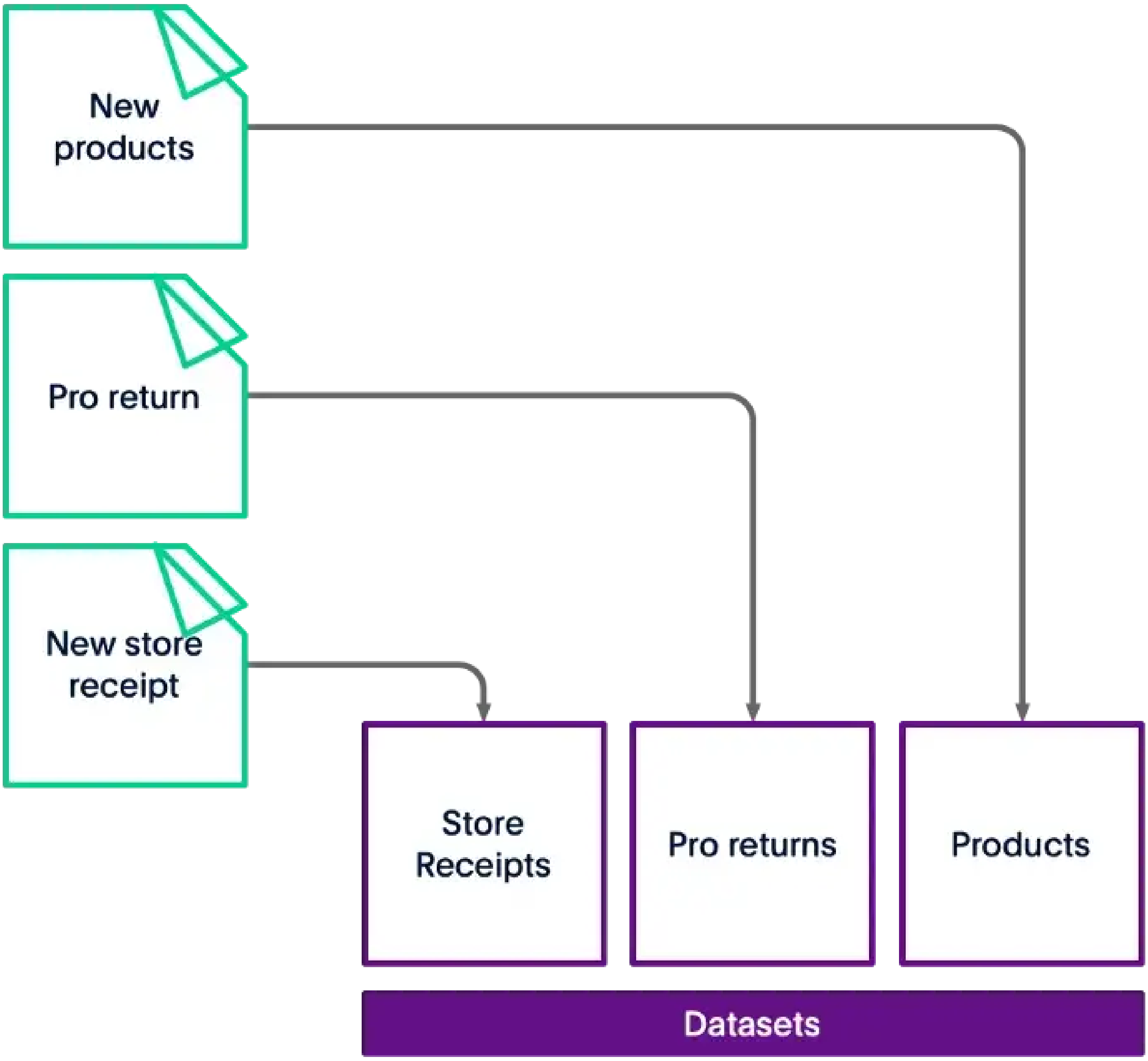

Figure 2 — Ingesting data in your data lake is much easier than with a data warehouse.

Now, it gets a little tricky when you try to consume your data — again. Storing is easy, however reading is complex. You can access the data by creating small data warehouses (databases or data marts) for your analytics loads, but you're back to the ETL spaghetti. It is the same dilemma when it comes to operational processing through micro-services.

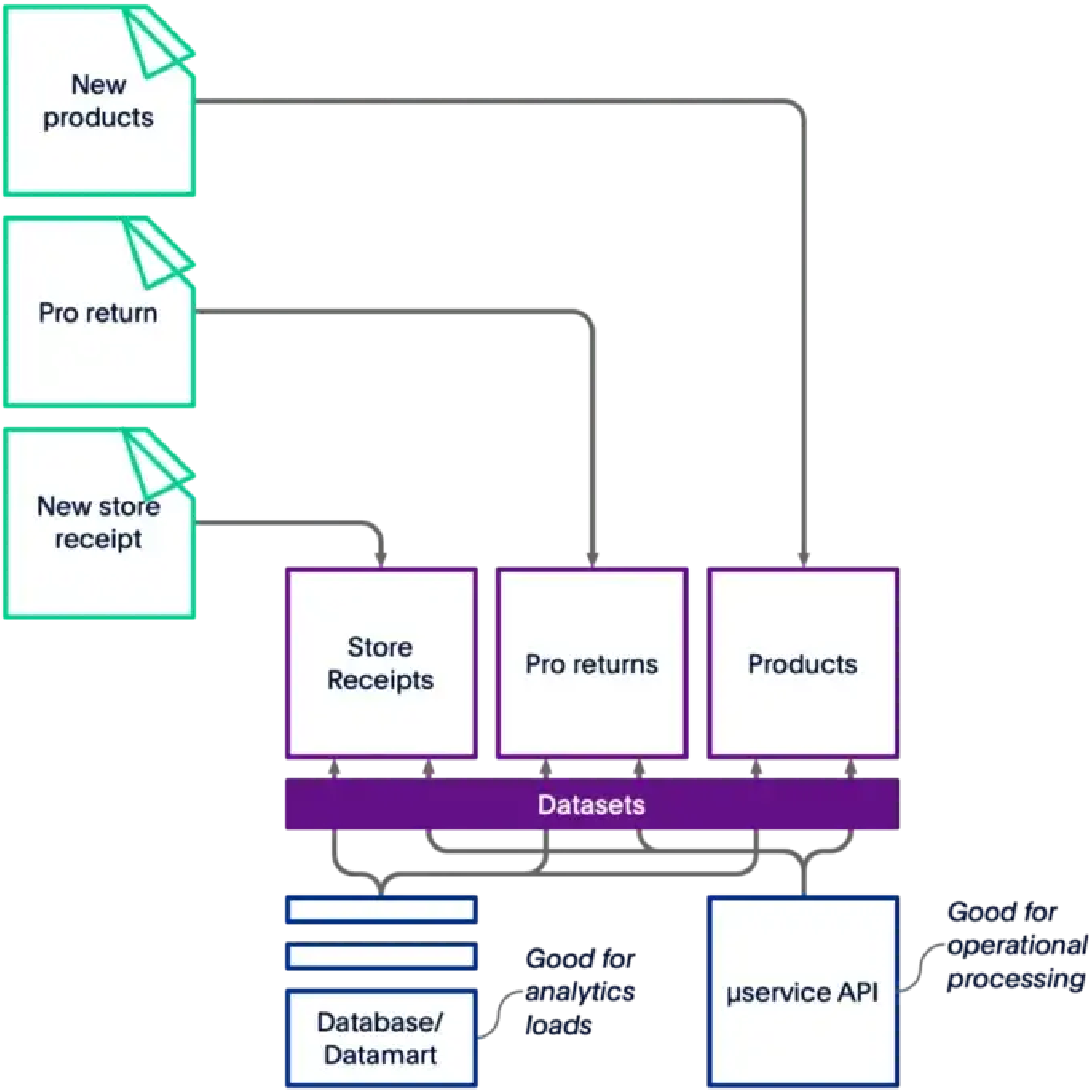

Figure 3 — Getting value from your data lake may be a bit tricky.

More recent architectures, like the data lakehouse, are trying to combine the best of the lake and warehouse, but they still lack the data quality, governance, and self-service features to ensure compliance with the enterprise and regulatory standards.

Opportunities

Unfortunately for some technologists, projects are not happening for the sake of technology; they are driven by opportunities and challenges. PayPal is not different. Let's look at the opportunities that presented themselves as PayPal's leadership considered a new data platform.

PayPal pioneered in self-service analytics, offering business analysts and data scientists access to our data warehouse very early compared to many companies.

The success of this initiative, combined with PayPal's willingness to move to the cloud, drove a need for a different type of data platform.

In addition to the self-service, data scientists' needs have evolved with more data discovery capabilities. As with many companies, data sources have increased, whether internally, through acquisitions, or even from external sources such as data provider.

As the business has become more and more complex, a major driver was increased compliance and auditability, and the challenge became about marrying big data, self-service discovery, experimentation, compliance, and governance, while providing a clear path from data experimentation to production.

Our team at PayPal, GCSC IA (Global Credit risk, Seller risk, and Collections, Intelligence Automation) settled on the Data Mesh paradigm as it was the best suited for our customer needs.

The four principles of the Data Mesh

In May 2019, a brilliant engineer, Zhamak Dehghani, published a paper highlighting the basis of the Data Mesh. In her paper, Dehghani sets the ground for four principles, which, over the last couple of years refined into the Data Mesh's four core principles. I like to compare those principles to how the agile manifesto disrupted the waterfall-based lifecycle in software engineering. Data Mesh is bringing to data engineering many of the concepts you may have been familiar with agile software engineering.

Figure 4 — The Three Musketeers' motto was "One for all, all for one." Illustration by Maurice Leloir for the Calmann-Lévy edition, Paris, 1894. Source: Wikipedia.

Let's discover together those four principles.

1. Principle of Domain Ownership

The term "domain" has been so overused in the last decades that its meaning is almost gibberish. Nevertheless, let's try to tame the domain and ownership in this context.

A domain is a specific area of business you are focusing on. If you are in the healthcare industry, it can be a hospital or a specific department such as radiology. Identifying the domain sets the boundaries and helps you falling into scope-creep situations (as in, let's also include the hospital cafeteria in the project).

If you are familiar with domain-driven development, this principle will come naturally to you.

It is common sense: don't try to boil the ocean. Find the people who know a domain best, and associate them with a data architect. The decentralized team has a precious domain expertise: they know more about the data sources, data producers, rules, history, and evolution of systems than a centralized team that switches from domain to domain. Adding the data architect in the mix will bring the security, rules, and global governance in order to stay compliant with the enterprise policies.

2. Principle of Data as a Product

In software engineering, agile replaced the project by the product. It was only a question of time before data became one as well. Let's see what a data product can bring.

Focusing on a data product will enable you to switch from a project planning perspective to a customer-centric approach. Daunting? No, just DAUNTIVS, a data product must be:

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

In software architecture, the smallest deployable element is called a quantum. When applied to data architecture, the data quantum is the smallest deployable element bringing value. The data quantum is not related to quantum computing.

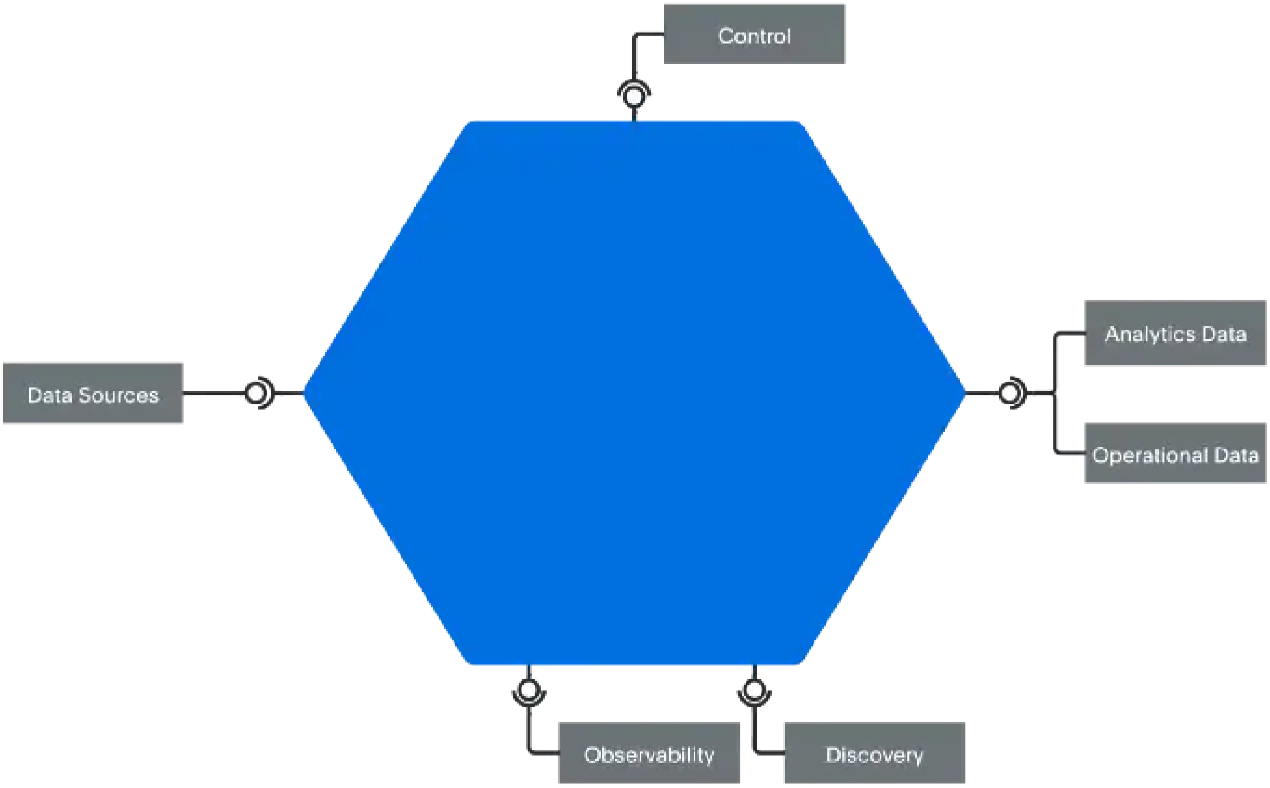

Figure 5 — The data quantum takes the shape of a hexagon, highlighting its multiple endpoints allowing access to data, metadata, observability, and control.

You're probably wondering, "Hey, how is that different from my data lake with a couple of data governance tools?" The answer is that size matters: instead of an entire enterprise-level lake, you focus on a single domain. It's definitely more "byte size" and chewable.

Thanks to its smaller size and scope, implementation is faster and the value from data is reinjected in the company a lot faster.

3. Principle of the Self-Serve Data Platform

When I was a kid, in France, I loved going to the local supermarket with my parents as it had a cafeteria where I could put on a tray all the food I wanted. The self-service empowered me to make (bad) food choices. But what does it mean when it comes to a data-platform?

Since its inception in 2001, Agile has proven to be a working methodology. Agile software engineering empowered software engineers. The way to empower data scientists is to give them access to data.

Data scientists and analysts spend (too much) time in their data discovery phase. In many situations, they find a piece of data in a random column in a table somewhere and take a bet on the fact that this is what they need. Sometimes it works, sometimes your PB&J toast does not fall on the jelly side.

Empowering the data scientists means that you must give them access to not only a basic catalog of fields, but precise definitions, active and passive metadata, feedback loops, and much more. They are your customers, you want to be this 5-star Yelp cafeteria, not this crappy 1-star shack.

4. Principle of Federated Computational Governance

Every word of this principle has a very important meaning. Let me try to convey to you their crucial interpretation.

Information technology has become so ubiquitous in our day-to-day life. States and governments have developed laws to manage how personal data is handled and used. Famous examples include Europe's GDPR (2016), California's CCPA (2018), and France's National Commission on Informatics and Liberty (1978). Of course, those constraints are not the only push towards governance in enterprises; companies like PayPal often have data governance rules and protections that may go beyond what the law requires.

But why a push towards **computational governance** and not just data governance? Because data governance is simply too limitative. Even when you include metadata in your governance (and of course you do), you are still missing the entire eco-system of computational resources linked to your systems. In a modern cloud-based world, you must account for many more assets. It made sense to extend from data to computational governance.

Your data governance team creates policies applicable to the entire organization, which the domain team will follow to achieve enterprise-level consistency and compliance. However, the domain team owns the local governance at the quantum level, maximizing the team's expertise.

Four principles

Like Alexandre Dumas' Three Musketeers, who were four, the four principles of the Data Mesh are intertwined.

Each principle influences each other, and as you design and build your data mesh, you cannot look at one principle in isolation: you need to progress on the four fronts at the same time. It is easier than it seems as you will see how PayPal is building such a data mesh.

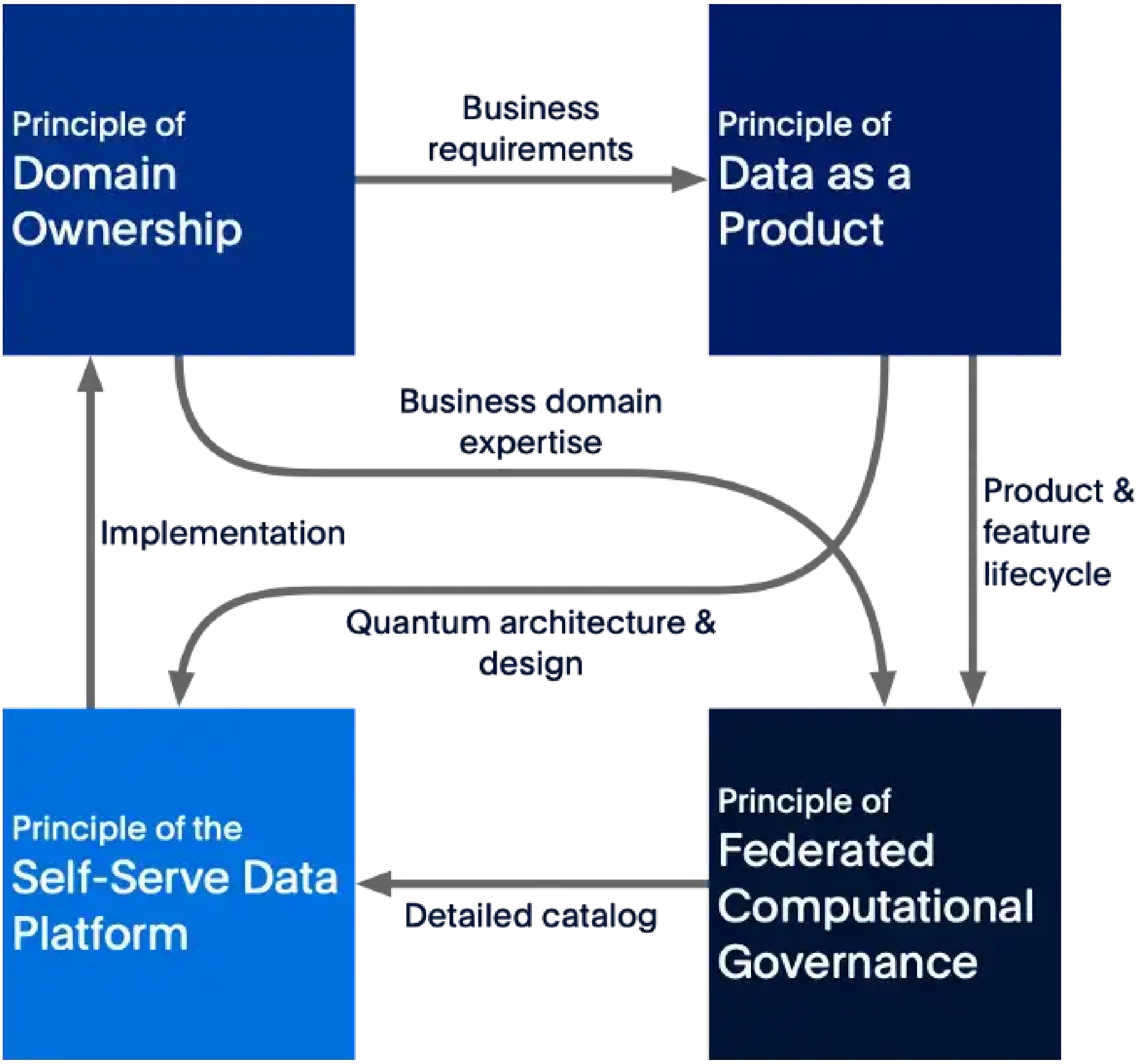
Principle of Domain Ownership
Business requirements
Principle of Data as a Product
Business domain expertise
Implementation
Product & feature lifecycle
Quantum architecture & design
Principle of the Self-Serve Data Platform
Detailed catalog
Principle of Federated Computational Governance

Figure 6 — An attempt at mapping the influences of one principle towards the others.

Building our first Data Quantum

Now that you have read about the motivation, opportunities, and governing principles, it seems about time to build your first data quantum, or, more precisely, architect it before you implement it.

Before building an entire data mesh, you will need to focus on each data quantum. The data mesh is a composition of data quanta.

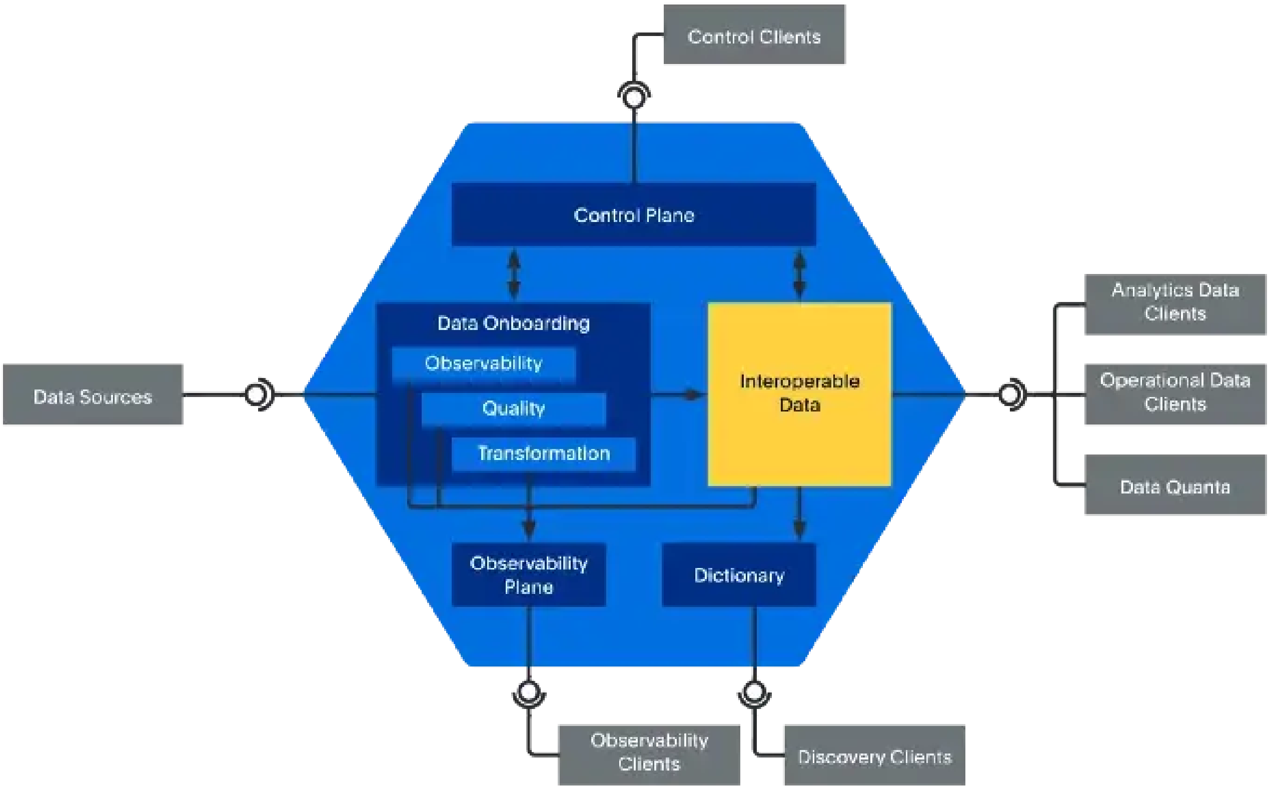

Figure 7 — Unwrapping the data quantum, what's inside?

You can divide the data quantum in five subcomponents:
• The dictionary,
• The observability plane,
• The control plane,
• The data onboarding, and
• The interoperable data.

The dictionary interface is the precious sesame to your passive metadata. Your data quantum users can connect, without authentication, to the dictionary. Their data discovery is then extremely simplified as they can browse the dictionary in a very interactive way without a need for specific permissions, with additional description, and access to data lineage. When they find what they need, they can easily check that they have access or request access to the data.

The observability plane brings an interface between the built-in observability of the data quantum and REST clients. This allows a data scientist to gauge the quality of the data within the data quantum and decide if the data quantum will match their SLO (service-level objectives) expectations.

The control plane offers access to a REST API where you can control the onboarding and the data store(s). If you want to create a new version of your dataset in the data quantum, then, there is an API call for that. Do you need to control which data quality rules should be applied in your data onboarding? There is an API call for that. This interface is mainly oriented for data engineers managing the data quanta.

As you can imagine, the three sets of APIs are similar for each data quantum: there is no need to learn a new API for each data quantum. To simplify your usage, you can wrap your REST APIs in a Python API accessible via a notebook.

The data onboarding component is your old data pipeline on steroids. In many (if not all) pre-Data Mesh data engineering projects, the focus was on the data pipeline. The Data Mesh puts the pipeline back in its place. The pipeline is important but is an element of the data onboarding, such as observability or the application data quality rules. Adding all those functions in this component secures the classic, often failing, fragile ETL process. Yup, the days where the pipeline is the quarterback of the team are behind us.

Last, but seriously not the least, the interoperable model is your critical data in a consumable way. I could have represented this component as the classic cylinder you can see in older architecture diagrams, but remember that the data exposed by a data quantum is not always relational.

The promise of the data quantum is to separate the application from the data. This has an impact on the data modeling inside the data quantum.

Welcome to the Mesh

So far, you have learned a lot about the data quantum (plural: data quanta). Hopefully, you see the value of the data quantum, but what additional value would a data mesh bring over a crowd of data quanta?

A set of data contracts governs each data quantum: the primary data contract defines the relationship between the data quantum and its users. It also describes the interoperable model and SLA (service-level agreement) details. This consumer-oriented data contract can also be called output or user data contract.

Figure 8 illustrates the role of the data contract. A data quantum may have several data contracts as the input and offer a data contract for its consumer.

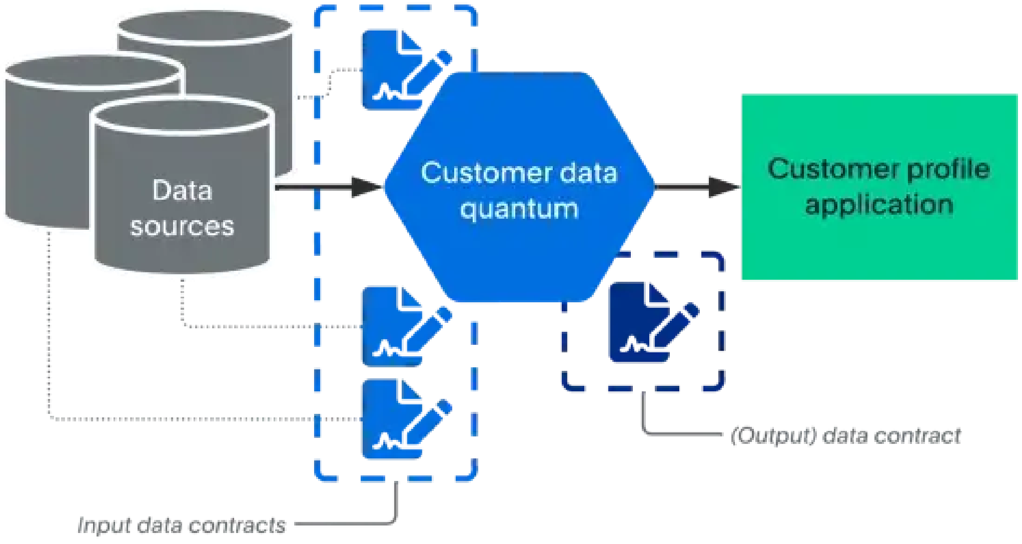

Figure 8 — The importance of the data contract: it defines the relationship with the producers and the consumers.

When the data quanta have meshed, like in Figure 9, the resulting data quantum inherits the data contracts from the source data quanta. This mechanism simplifies interoperability, increases data quality, and decreases time to market.

Our team is tackling this challenge as you read this article.

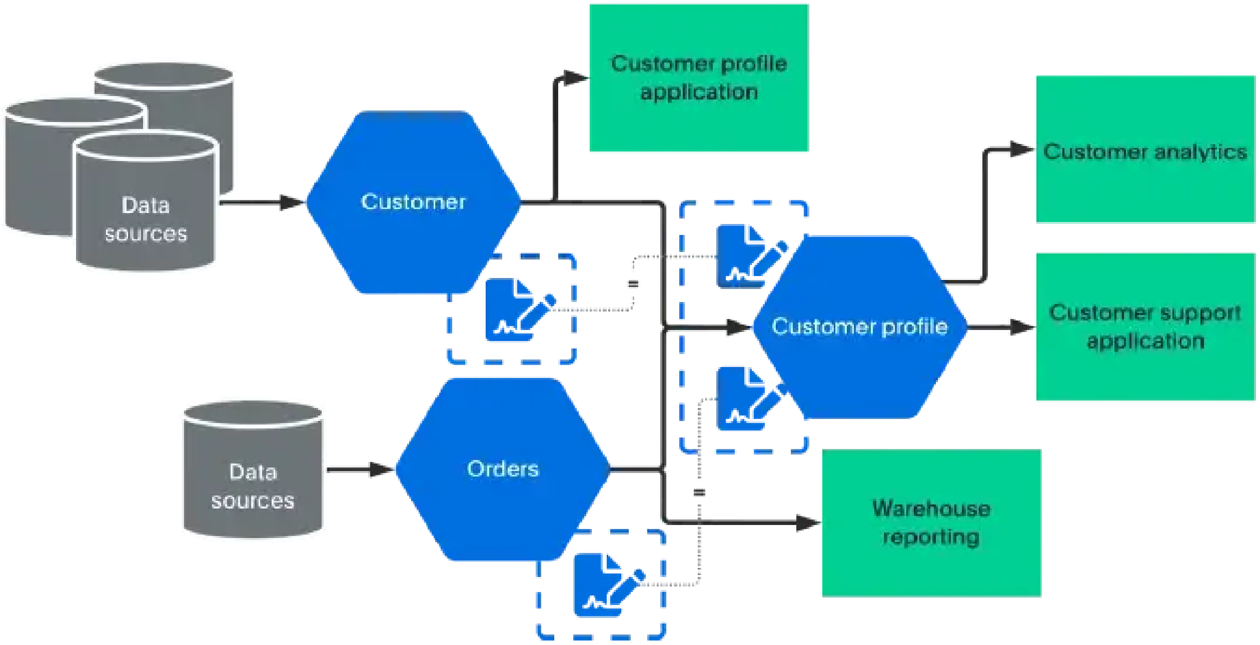

Figure 9 — Meshing the data quanta increases the delivered value by the data mesh, without affecting data freshness.

Challenges

The road to building a data quantum is paved with good ideas, but the devil is clearly in the implementation.

Nevertheless, here are a few advice to get started.

As with any disruptive technologies and methodologies, be prepared to guide your users through this transition. Many data engineers live by the sacrosanct data pipeline and reducing their idol to a mere component in a mesh can be traumatizing.

Prepare your leadership for time to stand up a new platform: they should not expect results two weeks after you start (or even three…).

As with all product development, identify clearly who your users are and what tools they currently use. You may need to transition or extend their tooling, and this may create friction and resistance.

And the truth is there is no "Data Mesh product" out there. There might be bricks, elements, or components that can be assembled to help you build your mesh (Spark remains a fantastic engine to perform your data transformation at scale — and more). However, there is nothing like an OTS (off the shelf) platform, whether commercial or open source.

The lack of software vendors in the field fosters innovation but equally chaos. The next few months will tell us whether we made the right choices in terms of user experience, technology choices, and implementations.

Jean-Georges "jgp" Perrin is a technology leader focusing on building innovative and modern data platforms at PayPal, president of AIDAUG, and author of Spark in Action, 2nd edition (Manning).

Extra resources

There are two excellent books

- The reference book: **Data Mesh: Delivering Data-Driven Value at Scale** (O'Reilly) by Zhamak Dehghani.

- A more practical book, that takes some liberties from Dehghani's reference and offers a slightly different viewpoint: **Data Mesh in Action** (Manning Publications) by Jacek Majchrzak, Sven Balnojan, and Marian Siwiak.

Videos

- **Introduction to Data Mesh** by Zhamak Dehghani.
- Yours, humbly featuring **The next generation of Data Platforms is the Data Mesh — DataFriday 2x07**.

Website

- Definitely pay a visit to Scott Hirleman's Data Mesh Learning, where you will find links to the newsletter and fast-growing Slack community.

Data Mesh Data Platforms Data Quantum Data Governance