What Is Data Mesh?

Simon Crosby · Following
10 min read · Jul 13

1 · 1 · 1

Intro

One thing that always impresses me any time I start working with a new customer is how often, when discussion starts driving around data, a while majority of stakeholders starts speaking of **database** tables. It is even more impressing to consider that this approach is often driven by **business** users. I heard countless times **business** owners saying that they need a given **database** table, or that they need to run a query against a given **database** against them. What happens that the focus is on the **technical details**, rather than the desired outcomes in terms of **information** and that the solution should provide.

From a certain point of view, when we design the data architecture of a solution, we are often begging the state of the art approach for software design. It was common, in the old fashioned waterfall model approach, to have the same business owner driving software development team that he would send a function, a web application, a web service or a database. Basically, it was not telling what he **needed**, but **what the team should be** doing to realize the artifacts he **thought** he needed. There were several issues with this approach, and I would like to focus on three ones:

- Business owners are experts in their business domain, and they may not be aware of state of the art technical solutions. The vision of the required artifacts can be far from the optimal solution.
- Focusing too much on tasks, it is clear that the project will lose control on what can add value to the solution, resulting in wasting effort in add useless functionalities.

Agile methodologies helped in fixing these behaviorists the focus of a project is on business outcomes rather than tasks and time management. It's feature brings value **for the business owner.** It will be implemented, and the business owner will **decide** if it is ready to be released, if it satisfies the rule of the product owner, that he sums quickly on business requirements and provide the needed information to the team to **make decisions**. With agile methodologies everyone is focusing on his strengths, and it is relevant only what is **adding value to the solution** rather than the emphasis to up-front planning.

As we are going to see, the Data Mesh pattern introduces a new way of thinking data architects, named Data Product, whose goal is to describe the information set that should be provided by a system to business terms, trying to put the focus on the **what** rather than the **how**. Moreover, the pattern enhances a **decentralized organizational model** that separates data ownership from the technical implementation, and it helps to **move** ownership from the infrastructure management function, responsible to hold technological blueprints, usually realized by IT teams. This model simplifies the management of **distributed data** and enhances **collaboration** and reuse, and grants the adoption of state of the art technology to manage end-to-end flows.

In the next paragraphs we are going to see:

- The never-ending struggles between the centralized approach to try to keep everything under strict control and the creation of decentralized silos to reflect the flexibility needed by business users.
- Why the mesh approach can solve it clearly segregating the role of the IT function, responsible to provide blueprints and standards and the business function, responsible to hold data products using those blueprints.

The Data Warehouse

From an architectural standpoint, I believe that anyone interested in the definition of a data management system, is familiar with the classical **data warehouse architecture**:

Figure 1 — Data Warehouse Architecture

At the beginning of the data flow, we have **data sources** which typically are enterprise applications, providing structured data that it' stored into a **Data Warehouse**, usually hosted by a DWH vendor, on premises infrastructure, and DWH experts. Starting from data, Usually, the DWH does not contain a replica of the source data, it only stores the result of the ETL process at times, to simplify debugging and optimization solving. The last period of analysis is stored in the DWH ingestion tables (it can be one hour of information, one day, one month, etc. according to the business needs).

In some cases, especially when data is complex, as additional layer, the Data Mart, is introduced, with the goal of getting together data coming from different DWHs or to reshape the data defined by DWH for an easier consumption, for example, for a given set of reports, data can be reorganized in a data mart (e.g. defining a star schema) so that it is more compliant with the reporting framework technology.

In a common, for large international companies, to manage several physical Data Warehouses:

- Different countries can have different regulations and so they need to shape the same information differently.
- Some countries may require that data is not stored outside of the country's boundaries; with this scenario, only local warehouses are allowed.
- Different departments can shape the same information differently, generally speaking, the way salesperson describe a customer can be very different from the way accounting or financial departments do.
- Different departments can have different data velocity requirements: in transportation market, for example, the accounting team can be satisfied with daily or weekly data updates, while operations need much faster updates (even more than within the same hour).

So, even if the DWH's premises is ideally meant to hold a unique data store for the whole organization, there are several technical scenarios that can't comply with this goal. To keep a balance, IT departments try to keep DWH number at a minimum since DWH storage is often expensive, and expensive software licenses are needed to run the database engine, so limiting the infrastructure size helps to limiting the cost.

In any case, the classical DWH approach with data management brings these issues:

- It is often hard to find master data: for example, there can be dozens of different software products within a large organization that can manage customer information, each one which its own keys and attributes and it may not be that simple to understand what information to take and which to discard.
- Once data is stored in the DWH tables, it is hard to understand the source of the information: which are the data sources of the Table of here they collaborate to produce the final record?
- Given the high integration with source systems, adding more data sources can be complex and costly and can require long analysis and implementation processes.

It may seem that having multiple Data warehouses, each one well designed for specific areas, could be a better approach than having a single shared and gigantic structure. I believe this is not far from the truth (and, by the way, I've experienced this approach in many organizations). Unfortunately, this approach brings the following issues to organizations in the overall view, since it is likely that more database licenses will be required.

Information Distribution Across the Organization

Let assume to have a company, let's call it "Cosmos" which have, for simplicity, a three-level organization:

- L1: Enterprise
- L2: Division
- L3: Department

The enterprise is divided into Divisions, and they are split into Departments. IT teams represents one of the Divisions. For data management purposes, it provides a centralized Data warehouse and some local data stores are available to serve specific departments' divisions. To produce Enterprise reporting, a data mart incorporating the central Data warehouse and other business critical local databases has been developed. Any time the central DWH needs to be updated, an IT project is started with an analysis stream to understand which information is needed and which is available. Some databases are often developed by a project sponsored by business departments's divisions and realized by external suppliers. In these projects, IT has a governance role, checking the compliance to company guidelines and ensuring the quality of produced artifacts. After the go-live project artifacts are incorporated in IT operation processes.

With scenario in mind, let us ask: *what is it likely to happen any time a department needs to analyze a new set of information?*

In my experience, the response will have to face two options:

1. Add a new set of tables to the centralized data warehouse starting an analysis project to identify needed data sources, possible overlaps, and possible integration issues.
2. Start a new project with a supplier with the help of IT for the governance to build a new local data warehouse to manage the new data.

Option 2 will generally be faster than option 1, but often would imply a duplicate of data and, in the long term, it will create a data silo that will be available to an existing solution, but must will be sacrificed for the sake of speed. When there is enough time, Option 1 is likely to be selected. Iterating this approach repeatedly, the result will look like the picture below.

Figure 2 — Data Warehouse across organization

Zooming out and looking at the picture from a comparatively perspective, we would see several data stores (data warehouse, databases, data lakes, etc.) which holds information relevant at specific organizational levels: some can store information useful for specific departments, others for specific divisions, a few can hold information relevant across divisions or even for the whole organization.

Given this, which is the enterprise level information? How can we build it? It is likely that, according to the kind of analysis we would be running, a different set of data stores would be selected, data would be transformed and harmonized to have a consistent view at enterprise level. It is not released the technical form of this aspect level. It could be a data mart, a virtual view, a set of materialized views, a set of operational views, but, it represents the *union of information spread across the company* that has value for the whole organization.

Distributed data

The Cosmos example obviously does not perfectly match the structure of any company in the world, but it is good enough to represent a company, at least for the IT organization, that I have been in the past 15 years. There is a quite I like very much and that could be here:

Embrace reality and deal with it.

It is a fact, within complex organizations, that information tends to be distributed rather than centralized and that, in many cases, business prefer to have ownership of data that they need to perform their daily activities. Ownership is a matter of the scope of this writing is decide if this is right or wrong, but it happens. Often, there is a struggle between IT departments, that would like to centralize everything for the sake of governance, and business users who want to do centralize everything for the sake of speed and flexibility.

Data mesh helps to find a balance between these two attitudes, distributing roles and responsibilities across IT and business functions so to find a match between IT governance and business flexibility. With this approach we wrote that IT provides blueprints, standards, and governance, while business strategy is to step through them are some technical expertise to create data (the adoption), describing it in business terms so that it can be understandable by anyone in the organization. Let's see in the context. To be more specific:

- Each data artifact should have a clear owner, responsible for its maintenance, and business users, who want to use this to understand its purpose, it must be clear who his needs to reach out, the **Data Owner**.
- Each data artifact should expose from which other data artifacts it is taking data, and differently it must expose its **lineage**.

From an IT perspective, IT organization is responsible for two tasks:

1. provides a set of policies and standards that are strictly required to build artifacts. It is not relevant which one has a team is using, but it must comply with the standards. And differently, IT function provides the infrastructure to build contents, it defines **architectural blueprints**.
2. It monitors the produced artifacts, creating a data catalogue of all the available contents within the organization, taking a census on lines (lineage) and tracking the complete Data Lineage of every artifact. IT functions will use these data to enforce **artifacts reuse** across the organization.

As discussed in the Cosmos example, IT can be involved together with external suppliers in the realization of artifacts, but anyone of them will be realized as a local project within a centralized infrastructure.

Data Mesh Architecture

Let us to a step back and define some key terms that will be some useful later, consider these two facts:

- Every application, at least in the context of data management, produces and consumes data: it can be in the application field, in files, databases, data streams, etc.
- Every time an application needs to share its data with another application, we need to implement a data integration layer: usually this integration layer is based on ETL or ELT pipelines, but it could be any kind of transformation to make data coming from application A understandable for application B.

With these two facts in minds, we can synthesize the data transfer to application B to application A identifying application A as a **Data Producer** and application B as a **Data Consumer**.

Figure 3 — Data Producer — Data Consumer

- *Data Provider:* it represents the application who is generating the desired data; data is expected to be known and owned by owner (usually the application owner).
- *Data Consumer:* it represents the application which is consuming data for its own purposes within a specific context (it can be a company organizational unit, a project team, etc.). A consumer can be a producer as well.

NOTE: data integration is a complex topic and I assume that several questions are growing in your mind:

- Is data integration implying to copy data from provider to consumer?
- What is it going to happen to applications A, if we find out that data in application A is wrong or corrupted?

Of course, data replication can result in the proliferation of local copies of the same information, generating governance and integration complexity and making error flying dramatically complicated. It is outside of the scope of this writing to investigate those details and the approaches we have at our disposal. We indicate a full writing in this topic in the future.

Anyway, there is no escape from data integration. Regardless of the scope and the goals for application A and B (they can be from business applications, a business application and a data warehouse, etc.), we will need to integrate them, in our focus should be on making this integration as easy as possible. Let's define a set of rules that should be satisfied to make data produced ready to the integration. Let's define some domains key attributes of a data product:

- It should be available for broad consumption.
- It should be expressed in common language (no technical names).
- It should have a clear and well-known owner.
- It should be aligned to its content.
- It should not be confidential to specific needs of data consumers.
- It should be decoupled from the source application data management pattern.
- It should be expressed directly from the source and not obfuscated via other systems.
- It should remain compatible from the moment created.
- It should adhere to central interoperability standards.
- It can be built on top of other data products.

Each one of these requirements would require a deep discussion, and we will provide it in another writing.

The idea is to have an artifact provided by an identifiable owner capable of providing meaningful information, without the need to know or understand its source application technology or data structure. Said differently, a data product is a self-contained artifact that is governed to satisfy **business needs** leaving all the technicality behind.

We said that each data product lives within a context: let's name it **Data Domain**. A data domain represents a set of people organized around a common business purpose (they can be part of the same organizational unit, of the same project, etc.) that generate data products that impact the above requirements and the company quality standards.

Figure 4 — Data Domain — Data Product Owner(s)

With this in mind, and remembering that a data warehouse already consolidates data coming from source applications, we can rethink the data distribution across the company as the production of data products by different data domains:

Data Domain
Data Domain
Data Domain

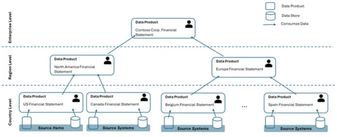

Figure 7 — Mesh Example

The result will be a network of Data Products:

- Each Data Product has a clear and defined Data Owner
- The relationship of data consumption will define the Data Lineage of each Data Product

With this approach we have defined a clean architecture with these benefits:

- Whenever possible, each Data Product relies on already existing Data Products, enhancing reuse, limiting the connectivity to source applications (ERPs, CRMs, etc..) and centralizing data transformation and cleansing logics.
- Each Data Product has a clear owner: whenever there are issues with the inputs provided by a Data Product, there is clear responsibility on who should investigate the issue.
- The adoption of a full lineage helps in identifying the end-to-end data chain, explicating the source a Data Product is using to build its own results. The full lineage also helps to understand the impacts of wrong or corrupted set of information in one of the items.

Conclusion

In conclusion, we saw that even if technical approaches like data warehouses tend to build central application solutions, in large organizations data together with systems tend to be **distributed across geographies and organizational units**. The Data Mesh architecture provides an approach to smoothly manage de-centralized analytics platforms and data stores, keeping a strict governance over policies and platforms.

At the same time, Data Mesh architecture, thanks to the concept of Data Product, encourages the definition of well-defined and well-described artifacts with a clear ownership that are ready for broad consumption and describe in natural language. This helps business users to focus on the information they need rather than the technical items required to build that information.

In the next writing we will elaborate on these topics and in particular:

- How can we organize Data Domains efficiently?
- Is there any technological pre-requisite to adopt the Data Mesh approach? In general, what is required from an organization to successfully implement the pattern?
- How can we handle the data integration pattern without creating countless copies of the same data? Which strategies can we adopt?

Data Mesh | Data Mesh Architecture | Data Architecture | Data Governance

4

A

Written by Alessio Cesana

13 Followers

Following

More from Alessio Cesana

Alessio Cesana

Efficient Data Domains Organization

#datamesh: a journey to better #data. First stop: #datadomains. How to create them efficiently?

6 min read · Nov 24

2

See all from Alessio Cesana

Recommended from Medium

janmeskens

Data Usability: How to Build Better Data Products?

Looking ahead to the future, the utilization of data is poised to become a prevailing norm,...

8 min read · Aug 9

485 2

Willem Koenders in ZS Associates

A simple reference architecture for data products

Data products and data mesh are the "talk of the town" in the data governance and...

7 min read · Jul 5

39 1

Lists

data science and AI

38 stories · 2 saves

Elyadata in Elyadata

Data Architecture: Bridging the Gap Between IT and Data Teams

This article is the first in a series of articles on data architecture, including both conceptual...

6 min read · Nov 22

79 1

Piethein Strengholt

Understanding Master Data Management's integration...

A successful Master Data Management (MDM) implementation is no simple task. It...

9 min read · 6 days ago

143 2

Christianlauer in CodeX

What is DataGPT and why should Data Engineers and Scientists...

Are Data Experts loosing their Jobs due to AI?

3 min read · Nov 10

741 10

Everton Gomede, PhD

Batch vs. Streaming Data Ingestion: Choosing the Right...

Introduction

5 min read · Aug 3

2

See more recommendations

Help Status About Careers Blog Privacy Terms Text to speech Teams