



Introduction

Much has been said and written about Data Mesh as it is an exciting concept that is still maturing and that has generated a lot of enthusiasm with data practitioners. The term Data Mesh was first coined by Zhamak Dhegani in 2019 in the blog post [How to Move Beyond a Monolithic Data Lake to a Distributed Data Mesh](#), refined later on a second blog post, [Data Mesh Principles and Logical Architecture](#) and afterwards Zhamak released the [Data Mesh book](#), and excellent read that is recommended for any data practitioner.

Here on this post I assume you are familiar with the Data Mesh concept and theory; if not, the sources above are the golden sources to familiarise yourself with it.

How and why did we get here?

The Data Mesh concept relies on the fact that previous architectural styles and approaches have failed because they leverage a centralised approach, where IT centralises technology, skills, ownership, that created a bottleneck in delivering value derived from data to consumers.



The image above depicts the different data architecture approaches that historically have been used:

1. **Data Warehouse Era** - Initially when Data Warehouses were designed the ultimate goal was to support a demand of business requirements that was perfectly met by the supply of data the company acquired and generated.
2. **Data Warehouse Era** - Reality of this era was that the match between demand and supply of data was never met as demand outpaced the supply of data and the speed of which the central IT could deliver releases and features to business users.
3. **Data Lake Era** - Circa 2010 Data Lakes became quite popular as they promised to democratize data, taking advantage of the "data deluge" that increasingly companies were facing and allowing data consumers to take advantage of that data with increased agility compared to what they could get from the Data Warehouse. On most cases Data Lakes often complemented the Data Warehouse rather than replace it.
4. **Self Service Era** - Despite the promise of the data democratisation of the Data Lake, due to the centralisation of technology, skills, ownership, that was adopted even for Data Lakes, and also due to Data Lakes being more difficult to manage and use, the majority of the Lines of Business inside the organisations turned to self service tools and technologies that promised the agility and delivery of value the business demanded. Wide adoption of self service technology was often without the IT team input and typically it created ungoverned data silos, often creating severe data quality issues.
5. **Data Mesh Era** - We are currently on the Data Mesh era, where data practitioners look into this new concept as a way to once again, have a balance of business demand and supply of data, removing the bottlenecks of a centralised IT by decentralising technology, skills and ownership.

Data Mesh vs Data Warehouses/Lakes/Lakehouses

The Data Mesh promise is that, 1) leveraging domain ownership, 2) treat data as products, 3) leverage a self serve data platform and 4) enforce a federated computational governance, optimal exploration and use of data can be achieved, something previous architectural approaches have failed to deliver.

My opinion and empirical observation from years of doing systems architecture is that often architectures and systems might fail to deliver on their promises but seldom technology is the only or the most influential culprit. Data Warehouses, Data Lakes and Data Lakehouses if architected on a pragmatical way, leveraging best practices, leveraging the most appropriate technology for the use case and if deployed and operated by a team that has the correct skill set will be successful and can even be part of a successful Data Mesh approach and architecture.

To my eyes the question is not deploying a Data Mesh or a DW/Lake/Lakehouse. It is rather deploying a Data Mesh and a DW/Lake/Lakehouse as often the backbone of the Data Mesh is a DW/Lake/Lakehouse architecture that supports the several data domains.

Data Platform Decentralisation

Decentralisation of domain ownership doesn't necessarily mean a complete decentralisation of the data infrastructure. Keep in mind that modern cloud data infrastructure and platforms are designed with modularity in mind and they are the complete opposite of a monolith and as such offer the prefect technology capabilities to address a Data Mesh that needs a complete decentralised data platform or one that consolidates the different domains on a smaller set of cloud services providing flexibility, isolation, agility and cost efficiency.

Don't get me wrong, decentralisation of the data platform is a consideration when deploying a Data Mesh, but that design decision needs to be assessed as it involves a trade off between agility, cost, operational effort, governance, performance, security and compliance.

Careful thought need to be done so that deploying a Data Mesh for a particular case solves challenges instead of creating unnecessary challenges.

Decentralisation of the data platform itself should be done as needed and if needed and should have a rationale behind it and associated business and/or technical benefits. Below are some examples, not exhaustive, for which decentralisation of the platform might make sense:

1. Large organization that operates in different geographies and segregates the data platform by geography leveraging a different tenant for each geography
2. Data domains that curate and provide data products that are of high volume (millions to billions of records), that have stringent SLAs and that if not isolated on a different instance (of a cloud DW or a Spark Cluster as examples) would dramatically affect the other domains to serve their data products
3. Organisation that wants to track cloud operating costs for each domain, that want to make the domains accountable for their costs and hence segregate per domain all cloud services and resources that have associated cost

Data Products

To me the most interesting concept that Data Mesh brings is enticing everyone to think about Data Products curated within a bounded context and a define data domain that addresses that bounded context. This is a powerful concept that can be applied to Data Meshes as well as to modern cloud Data

Warehouses, Lakes and Lakehouses. I would even go further and say that this should be the standard approach to producing and delivering value with data regardless of the architecture approach and deployment that is chosen as it creates accountability, increases quality and improves business outcomes as a consequence.

Federated Computational Governance

Governing a Data Mesh and enforcing access management and data security are crucial to deploy and operate a Data Mesh. Decentralising ownership if not accompanied with proper security guard rails can lead to 1) data exfiltration and potential data breaches or 2) creation of locked down data silos. Both will hamper seriously any Data Mesh initiative and hence governance should be top of mind when designing a Data Mesh architecture.

Doing Data Mesh at scale and having several domains producing data products, having dozens, hundreds or even thousands of data consumers means that automation is crucial to 1) grant access to data products, 2) to track data products usage and 3) to audit access to those data products.

Cloud is built with fine grained security and access control for all APIs and since in cloud everything is an API for which invocation can be automated, data mesh and cloud are the perfect combination.

Practical Data Mesh Deployment Example

On this section I will focus on describing a Data Mesh deployment in [OCI \(Oracle Cloud Infrastructure\)](#) leveraging a practical approach.

Architectural Styles

Let me state the obvious, Data Mesh is an approach, not a technology and also not an architecture. It advocates

Personas

Domains and Segregation

This is a test; let's see how it renders...

Implementation Example

Use Case

Cloud Topology

Summary
