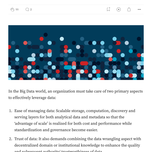
Introducing data mesh 2.0: A new era of data governance

Introduction: Why is data mesh relevant?

Capital One Tech · Following
Published in Capital One Tech · 10 min read · Jan 29



In the Big Data world, an organization must take care of two primary aspects to effectively leverage data:

1. Ease of managing data: Scalable storage, computation, discovery and serving layers for both analytical data and metadata so that the 'advantage of scale' is realized for both cost and performance while standardization and governance become easier.
2. Trust of data: It also demands combining the data wrangling aspect with decentralized domain or institutional knowledge to enhance the quality and subsequent authority/trustworthiness of data.

The primary purpose of wrangling with analytical data is to be able to create new insights that inform important business decisions. And it only happens when high-quality data is easily available to be consumed by the relevant consumers, both humans and machines. The greater the quality and rate of consumption, the faster the chance of revenue growth.

The evolving need for data mesh

Data lakes provide organizations with a cheap storage platform to store large volumes of pedigree data that kickstarted an era of a series of distributed data processing and analytics tools to operate over this data. Over time, they became data swamps — a dumping ground of data for various domain/LOBs with unclear vision for consumption needs and lack of ownership and restriction around duplication.

This eventually led to major issues with:

- Lack of data quality and trustworthiness (authoritative vs non-authoritative source of truth)
- Poor metadata management (registration and searchability) and discoverability
- Lack of governance and standardization (poor accuracy of both data and metadata)

And the paradigm of data mesh was introduced to solve this new set of problems in the data lake world.

What is a data mesh?

Data mesh is an approach to move beyond a monolithic data lake to a distributed data ecosystem with decentralized data processing and governance. It suggests four principles to achieve the promise of scale, while delivering quality and integrity guarantees needed to make data usable.

The data mesh suggests that each business domain is responsible for hosting, preparing and serving its data to its own domain and larger audience. This allows flexible and autonomous domain teams build and manage their own data products, promoting data ownership and accountability.

The data mesh paradigm is based on four principles

Domain ownership

Domain ownership talks about decentralization and distribution of responsibility to people who are closest to the data to support continuous change and scalability by making the business domain as a bounded context for data ownership.

Data as a product

This principle attempts to reduce the friction and cost of discovering, understanding, trusting and ultimately using quality data. Domain data product owners must have a deep understanding of who the data users are, how they use the data and what methods they are comfortable with consuming the data. Data product, consisting of code, data & metadata and infrastructure, is the architectural quantum of data mesh architecture.

Self-serve data platform

Self serve data infrastructure as a platform enables the domain teams to easily own their data products by creating a high-level abstraction of infrastructure that removes the complexity and friction of provisioning and managing the lifecycle of data products.

So, a self serve platform must have tooling that supports a domain data product developer's workflow of creating, maintaining and running data products with low operational knowledge than existing data technologies assume. However, it's not easy considering the diversity of today's data platform technologies to serve data. For example the domain team might be deploying its services on Docker containers and the delivery platform team might be responsible for deploying the containers to Kubernetes. Or a data product might be running its pipeline code on Spark jobs on a Hadoop/Yarn cluster.

Federated computational governance

Data mesh follows a distributed system architecture where a collection of independent data products exists side by side but with no independent life cycle and is built and deployed by likely independent teams.

However, to get value in the forms of higher order datasets, insights or machine intelligence there is a need for these independent data products to interoperate, to be able to correlate them, create unions, find intersections or perform other graphs or set operations on them at scale.

So, data mesh implementation requires a governance model that embraces decentralization and domain self-sovereignty while creating and adhering to a set of global rules (rules applied to all data products and their interfaces) for successful interoperability and an automated execution of governance decisions by the platform — a federated computational governance.

Key elements of data mesh principles

In summary, as per data mesh principles:

- Data product is the architectural quantum of sharing, owning, manufacturing, serving and governing analytical data.
- Data product is a composition of all components needed — code, data & metadata and infrastructure — all within the bounded context of a domain.
- So, each domain, besides defining and governing its data products, also must maintain its own infrastructure to produce and serve these data products while adhering to a set of global governance rules to enable interoperability of the data products.

A detailed discussion of the principles and architecture can be found here.

Data mesh challenges

While data mesh solves the ownership and governance aspects of analytical data by introducing a bounded domain context of data products, the same principles create new challenges:

- Since each domain manages its own data and data products, the advantage of processing large volumes of data at scale is lost, resulting in higher computational and run-the-engine costs for all domains within an enterprise.
- It introduces arbitrary uniqueness of technology solutions as multiple domains within the organization try to solve the same data wrangling problems independently; this also significantly increases the time to implement a mesh.
- Data mesh requires a high degree of technical maturity, as it depends on domain teams having the necessary skills to manage their data products independently. This in turn creates additional demand of specialized resources in an already specialized field of technology (e.g., new each domain needs separate Spark and DevOps experts to build their data infrastructure provisioning plans).
- Data mesh relies on domain teams taking ownership of their data products and adhering to governance standards and interoperability for successful interoperability. This requires strong collaboration and communication, as well as the establishment of organization-wide data governance standards for all domains. However, the biggest challenge in governance is not creating rules, rather enforcing adherence to those rules. In a data mesh world, adherence to a common set of governance rules is left to a domain's disposal, even the most basic set of governance rules are not enforced by common tooling thus risking interoperability at the enterprise level even if a small percentage of domains fail to adhere to the basic governance standards.
- A decentralized approach like the data mesh can lead to inconsistencies in data quality practices across different teams, which may impact the overall data quality within the organization.

Introducing data mesh 2.0

What if we borrow the principles of data mesh and implement them over a series of self serving horizontal data wrangling, serving and governance platforms managed by centralized teams?

From data mesh world:

- Embrace the idea of domain ownership of data products which increases trust of data.
- Onboard the data product as a logical bounded context which further enhances ownership and trust.
- Leverage the self-service principle to accommodate both common and additional governance needs of each domain's governance thus significantly reducing time to market.

Combine these with the principles of horizontal enterprise platforms

- Centralized data platforms for processing data — especially metadata management (governance and DQ rules baked into that), ingestion, curation, features calculation, data product creation and serving — to enjoy the advantages of economies and process at scale for lower overall cost and better governance.
- Standardization in design time and runtime processes and tools to significantly increase the interoperability of data products while reducing run-the-engine (RTE) cost.
- Horizontal platforms make lineage and alerting monitoring much easier thus further increasing trust on data. Using data intelligence to increase the quality of data and its trustworthiness via proactive and reactive notifications capabilities easily built once in the central platform and leveraged by many.
- Leverage Build by Data Leveraged by Many (BDLM) mindset
- Retain the advantages of a data lake: In the public cloud world, a data lake is nothing but a series of managed pedigree blobs of data residing on the cloud with an already mature governance structure to manage these folders as per their internal and external needs (finance, audit, compliance, data sharing with external entities etc.). All an organization needs is to organize these folders as per its need.

For data mesh 2.0 to work, horizontal enterprise platforms must have the following capabilities

Painless and well governed inter sourcing and co-development facilities so the domains can build their own unique (or reusable) capabilities within the platform:

- Capability to bring a domain's code and run it on the platform as long as it adheres to the governance controls set by the platform.
- Layered governance: for every aspect of data wrangling, the horizontal platform demands a basic set of governance controls while allowing for additional controls to be added by individual domain teams (e.g., during data movement, schema validation, sensitive data element identification, element level data quality checks and automated tokenization checks are must and provided by the platform by default). The domain teams may enforcedadd additional governance checks as needed within the platform (e.g., file level data publishing completion checks etc.).
- The horizontal platform enforces an enterprise data model for the cross-domain composite data products while the domains have the flexibility to add additional entities and attributes to these data products as per their need (without altering the data product keys).
- Domains are allowed to publish datasets outside of the data product network as per this internal and external needs (finance, audit, compliance, data sharing with external entities etc.). All an organization needs is to organize these folders as per its need.

Embracing the future: The promise of data mesh 2.0 and centralized platforms

The journey from decentralized data management to the innovative Data Mesh data mesh 2.0 represents a transformative leap in the world of data governance. By embracing principles like domain ownership, data products, self-service infrastructure and federated computational governance, organizations are achieving greater trust, quality and scalability in their data ecosystems.

As we look ahead, integrating these principles with centralized platforms signifies a promising future where data can be harnessed across all domains, setting the stage for a transparent, trusted and data-rich landscape.

Learn more about cloud computing at Capital One

Originally published at https://www.capitalone.com.

Authored by: Arjya Basu, Data Architect, Bank Architecture. Arjya is a data architect with more than two decades of experience in data and cloud. He is currently on the Bank Architecture team focusing on data architecture.

DISCLOSURE STATEMENT: © 2024 Capital One. Opinions are those of the individual author. Unless noted otherwise in this post, Capital One is not affiliated with, nor endorsed by, any of the companies mentioned. All trademarks and other intellectual property used or displayed are property of their respective owners. Capital One is not responsible for the content or privacy policies of any linked third party sites.

Cloud · Data Mesh · Data Governance · Data Lake · Data Mesh Architecture

Written by Capital One Tech
Following
0/K Followers · Editor for Capital One Tech
Find our founding, and be a partner in change the banking industry. Today, our innovations are making banking better for tens of millions of our customers.

More from Capital One Tech and Capital One Tech

Capital One Tech in Capital One Tech
10 microservices design patterns for better architecture
Consider using these popular design patterns to make your microservices design easier
10 min read · Jan 10
Andrew Brunham in Capital One Tech
2022 Open Source BPM Comparison
A compare and contrast of open source BPM platforms
10 min read · Nov 7, 2022
Jon Bratseth in Capital One Tech
Learning to Use Go Reflection
Find Go's Fitness on Tap
8 min read · Dec 13, 2017
Capital One Tech in Capital One Tech
Event Streaming: Tips for Event-Driven Apps
Tips for building event-driven apps that aspire to be observe, record and react to events in real...
7 min read · Dec 28, 2022
See all from Capital One Tech
See all from Capital One Tech

Recommended from Medium

Tracy Hanson in Kenyan Engineering
Data Dictionary: How I Learned to Stop Worrying and Love Repository...
Getting teams to embrace a central system for calculating metrics...
10 min read · Feb 1
Jacques V. Gaudard
Lakehouse Format Trends to Watch in 2026
Rise of Snowflake Lakehouse, challenges around Iceberg ordering, Hypermodular...
7 min read · Jan 31
Data Lake
Your Data Warehouse thinking is killing your AI ambitions
Stop trying to build better data movement
7 min read · Jan 29
Sebastian Herold in Rasoo Labs
Data Platform @ Rasoo (Part -2)—Cheap, efficient and scalable...
Data Platform @ Rasoo (Part -1)—Cheap, efficient and scalable analytics
7 min read · Jan 31
cdecosta
Why I stopped using DCO in my data modeling
Find 20 bits
5 min read · Jan 29
Yusuf Sarica in Teamwebinfo
Kubernetes for Data Engineering: An End-to-End Guide
Data engineering has evolved from running scripts on a single server to managing distributed systems across clusters. In...
10 min read · Jan 28
See more recommendations