

What is a data mesh?

Article • 12/10/2024

Data mesh is an architectural pattern for implementing enterprise data platforms in large and complex organizations. Data mesh helps scale analytics adoption beyond a single platform and a single implementation team.

Background

The demand for analytics isn't a recent development. Organizations consistently needed to evaluate business performance and utilized computers for this purpose since their inception. Around the 1980s, organizations started to build data warehousing solutions by using databases specifically for decision support. These data warehousing solutions served organizations well for a long time.

However, as business changes and generates more diverse data, data warehousing solutions that use relational databases might not always be the best solution. In the 2000s, big data became a common term. Businesses adopted new solutions that allow analysis of large volumes of diverse data that could be generated with great velocity. These solutions include technology, like data lakes, and scale-out solutions that analyze large quantities of data.

In recent years, many organizations successfully use modern architectural and analytical patterns that combine data warehousing technologies and more recent big data technologies.

However, some organizations encounter issues when deploying analytical solutions that use analytical patterns. These solutions are commonly still implemented as monolithic solutions, where a single team is the platform provider and the team is doing data integration. Smaller organizations and organizations that have a high degree of centralization from a team setup perspective can use a single team. However, a larger organization using only a single team often creates a bottleneck. This bottleneck causes a huge backlog, which results in parts of an organization waiting for data integration services and analytical solutions.

This pattern becomes more common as organizations adopt modern data science solutions. Many modern data science solutions require more data than traditional business intelligence solutions did in the past.

The recent switch to using microservices as an application development pattern is another driver of long backlogs around data integration, because it increases the

number of data sources.

A single team handling all data ingestion on a single platform in a large organization can also be problematic. One team rarely has experts for every data source. Most organizations are decentralized and distributed from a business perspective. Different business units and departments handle different parts of the business operation, so data experts are typically spread out across various sectors.

A pattern called data mesh was introduced to solve these problems. Data mesh's goal is to let distributed teams work with and share information in a decentralized and agile manner.

Data mesh is a technical pattern that also requires organizational change. The benefits of a data mesh approach are achieved by implementing multi-disciplinary teams that publish and consume data products.

The following concepts are foundational for understanding data mesh architecture:

- Data domains
- Data products
- Self-serve platforms
- Federated governance

Data domains

Data domains are the foundation of data mesh. The concept of data domains comes from Domain Driven Development (DDD), a paradigm often used in software development to model complex software solutions. In data mesh, a data domain is a way to define boundaries around your enterprise data. Domains can vary depending on your organization, and in some cases, you can define domains around your organization. In other cases, you might choose to model data domains based on your business processes or source systems.

There are three aspects to data domains:

- Your chosen boundaries render themselves to long term ownership. They exist over a long period of time and have identified owners.
- Your domains should match reality, not just theoretical concepts.
- Your domains need to have atomic integrity. If areas have no relationship with each other, don't combine them in a domain together.

For more information about data domains and how you should define them, see [Data domains](#).

Data products

Data products are another important component of data mesh. Data products aim to take product thinking to the world of data. In order for your data product to be successful, it needs to provide a long term business value to the intended users. In data mesh, a data product involves data, code assets, metadata, and related policies. Data products can be delivered as an API, report, table, or dataset in a data lake.

A successful data product must be:

- **Usable:** Your product must have users outside of the immediate data domain.
- **Valuable:** Your product must maintain value over time. If it doesn't have long-term value, it can't succeed.
- **Feasible:** Your product must be feasible. If you can't actually build it, the product can't be a success. Your product must be feasible from both a data availability and a technical standpoint.

The code assets of a data product include code that generates it and code that delivers it. The code assets also include pipelines used to create the product and the product's final report.

For more information about data products, see [Cloud-scale analytics data products in Azure](#).

For specific guidance on using data mesh, see [What is a data product?](#).

Self-serve platforms

A core of data mesh is having a platform that allows the data domains to build their data products on their own. Data domains need to define data products by using the tools and processes that are relevant for users without having a strong dependency on a central platform or a central platform team. In a data mesh, you have autonomous teams developing and managing autonomous products.

While using decentralization and alignment with business users that understand your data, remember the generalists who also work on your platform. Because you have generalists, you can't have specialized tools that require specialist knowledge to operate as the core foundation of your mesh-based platform.

You can successfully implement your self-serve platform by adopting the practices outlined in [Design considerations for self-serve data platforms](#).

Federated governance

When you adopt a self-serve distributed data platform, you must place an increased emphasis on governance. Lack of governance leads to silos and data duplication across your data domains. Federate your governance, as people who understand the governance need exist within your domain aligned teams and among data owners.

To create your federated governance, implement automated policies around both platform and data needs. Use a high degree of automation for testing and monitoring. Adopt a code-first implementation strategy to handle standards, policies, data products, and platform deployment as code.

For more information on implementing federated governance aspects, see [Data governance overview](#).

Summary

Data mesh can be an effective way to implement enterprise data platforms, but it isn't the best solution for all organizations. Data mesh requires autonomous teams that can work independently. Data mesh works best in large and complex organizations that have independent business units and need to scale their analytics adoption beyond a single platform and implementation team.

When using data mesh, take special care when implementing your governance so you don't create silos. Always keep product thinking for data at the core of your implementation to ensure success.

Next steps

Data domains

Feedback

Was this page helpful?

 Yes

 No