

Introductions

Friday, February 17, 2023 8:04 PM

Main CLDT link: [Cloud and Data Led Transformation Community of Practice - Home \(sharepoint.com\)](#)

Working copy link: **URL: Optum - Data Mesh v1.1**

Working copy link: <https://uhgazure.sharepoint.com/:p/r/teams/CloudDataServiceOfferingDevelopment/Shared%20Documents/Cloud%20Services%20Offering/Data%20Mesh/Optum%20-%20Data%20Mesh%20v1.1.pptx?d=wa91e21a8be7a446f84f02b7c77f31a05&csf=1&web=1&e=Dkzfs2>

[Data Mesh Architecture \(datamesh-architecture.com\)](#)

Data Mesh by Martin Fowler & [Zhamak Dehghani](#)

Tuesday, December 27, 2022 2:07 PM

Youtube Demo: [Data Mesh Paradigm Shift in Data Platform Architecture](#)



[How to Move Beyond a Monolithic Data Lake to a Distributed Data Mesh \(martinfowler.com\)](#)

How to Move Beyond a Monolithic Data Lake to a Distributed Data Mesh

Many enterprises are investing in their next generation data lake, with the hope of democratizing data at scale to provide business insights and ultimately make automated intelligent decisions. Data platforms based on the data lake architecture have common failure modes that lead to unfulfilled promises at scale. To address these failure modes we need to shift from the centralized paradigm of a lake, or its predecessor data warehouse. We need to shift to a paradigm that draws from modern distributed architecture: considering domains as the first class concern, applying platform thinking to create self-serve data infrastructure, and treating data as a product.

20 May 2019



[Zhamak Dehghani](#)

Zhamak is a principal technology consultant at Thoughtworks with a focus on distributed systems architecture and digital platform strategy at Enterprise. She is a member of Thoughtworks Technology Advisory Board and contributes to the creation of Thoughtworks Technology Radar.

[ENTERPRISE ARCHITECTURE](#)

[DATA ANALYTICS](#)

CONTENTS

- [The current enterprise data platform architecture](#)
- [Architectural failure modes](#)
- [Centralized and monolithic](#)
- [Coupled pipeline decomposition](#)
- [Siloed and hyper-specialized ownership](#)
- [The next enterprise data platform architecture](#)
- [Data and distributed domain driven architecture convergence](#)
- [Domain oriented data decomposition and ownership](#)
- [Source oriented domain data](#)
- [Consumer oriented and shared domain data](#)
- [Distributed pipelines as domain internal implementation](#)
- [Data and product thinking convergence](#)
- [Domain data as a product](#)
- [Discoverable](#)
- [Addressable](#)
- [Trustworthy and truthful](#)
- [Self-describing semantics and syntax](#)
- [Inter-operable and governed by global standards](#)
- [Secure and governed by a global access control](#)
- [Domain data cross-functional teams](#)
- [Data and self-serve platform design convergence](#)
- [The paradigm shift towards a data mesh](#)

Becoming a data-driven organization remains one of the top strategic goals of many companies I work with. My clients are well aware of the benefits of becoming [intelligently empowered](#): providing the best customer experience based on data and hyper-personalization; reducing operational costs and time through data-driven optimizations; and giving employees super powers with trend analysis and business intelligence. They have been investing heavily in building enablers such as data and intelligence platforms. Despite [increasing effort and investment in building such enabling platforms](#), the organizations find the results middling.



For more on Data Mesh, Zhamak went on to write a full book that covers more details on strategy, implementation, and organizational design.

I agree that organizations face a multi-faceted complexity in transforming to become data-driven; migrating from decades of legacy systems, resistance of legacy culture to rely on data, and ever competing business priorities. However what I would like to share with you is an architectural perspective that underpins the failure of many data platform initiatives. I demonstrate how we can adapt and apply the learnings of the past decade in building distributed architectures at scale, to the domain of data; and I will introduce a new enterprise data architecture that I call **data mesh**.

My ask before reading on is to momentarily suspend the deep assumptions and biases that the current paradigm of traditional data platform architecture has established; Be open to the possibility of moving beyond the monolithic and centralized data lakes to an intentionally distributed data mesh architecture; Embrace the reality of *ever present, ubiquitous and distributed* nature of data.

The current enterprise data platform architecture

It is *centralized, monolithic and domain agnostic aka data lake*.

Almost every client I work with is either planning or building their 3rd generation data and intelligence platform, while admitting the failures of the past generations:

- **The first generation:** proprietary [enterprise data warehouse](#) and [business intelligence](#) platforms; solutions with large price tags that have left companies with equally large amounts of technical debt; Technical debt in thousands of unmaintainable ETL jobs, tables and reports that only a small group of specialized people understand, resulting in an under-realized positive impact on the business.
- **The second generation:** big data ecosystem with a [data lake](#) as a silver bullet; complex big data ecosystem and long running batch jobs operated by a central team of hyper-specialized data engineers have created [data lake monsters](#) that at best has enabled pockets of R&D analytics; over promised and under realized.
- **The third and current generation data platforms** are more or less similar to the previous generation, with a modern twist towards (a) streaming for real-time data availability with architectures such as [Kappa](#), (b) unifying the batch and stream processing for data transformation with frameworks such as [Apache Beam](#), as well as (c) fully embracing [cloud based managed services](#) for storage, data pipeline execution engines and machine learning platforms. It is evident that the third generation data platform is addressing some of the gaps of the previous generations such as *real-time data analytics*, as well as *reducing the cost of managing big data infrastructure*. However it suffers from many of the underlying characteristics that led to the failures of the previous generations.

Architectural failure modes

To unpack the underlying limitations that all generations of data platforms carry, let's look at their architecture and their characteristics. In this writeup I use the domain of internet media streaming business such as Spotify, SoundCloud, Apple iTunes, etc. as the example to clarify some of the concepts.

Centralized and monolithic

At 30,000 feet the data platform architecture looks like Figure 1 below; a centralized piece of architecture whose goal is to:

- *Ingest* data from all corners of the enterprise, ranging from operational and transactional systems and domains that run the business, or external data providers that augment the knowledge of the enterprise. For example in a media streaming business, data platform is responsible for ingesting large variety of data: the 'media players performance', how their 'users interact with the players', 'songs they play', 'artists they follow', as well as 'labels and artists' that the business has onboarded, the 'financial transactions' with the artists, and external market research data such as 'customer demographic' information.
- *Cleanse, enrich, and transform* the source data into trustworthy data that can address the needs of a diverse set of consumers. In our example, one of the transformations turns the click streams of user interaction to meaningful sessions enriched with details of the user. This attempts to reconstruct the journey and behavior of the user into aggregate views.
- *Serve* the datasets to a variety of consumers with a diverse set of needs. This ranges from analytical consumption to exploring the data looking for insights, machine learning based decision making, to business intelligence reports that summarize the performance of the business. In our media streaming example, the platform can serve near real-time error and quality information about the media players around the globe through distributed log interfaces such as Kafka or serve the static aggregate views of a particular artist's records being played to drive financial payments calculation to the artists and labels.



Figure 1: The 30,000 ft view of the monolithic data platform

It's an accepted convention that the monolithic data platform hosts and owns the data that logically belong to different domains, e.g. 'play events', 'sales KPIs', 'artists', 'albums', 'labels', 'audio', 'podcasts', 'music events', etc.; data from a large number of disparate domains.

While over the last decade we have successfully applied [domain driven design and bounded context](#) to our operational systems, we have largely disregarded the domain concepts in a data platform. We have moved away from [domain oriented data ownership](#) to a centralized [domain agnostic data ownership](#). We pride ourselves on creating the biggest monolith of them all, the big data platform.

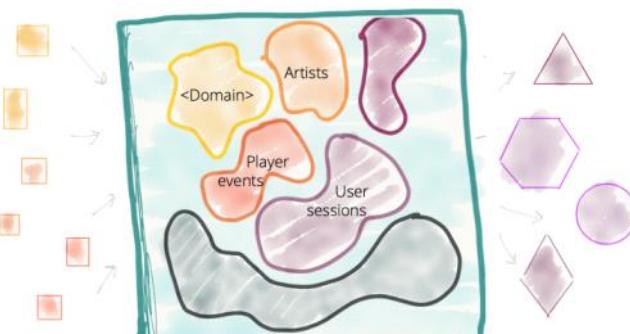


Figure 2: Centralized data platform with no clear data domain boundaries and ownership of domain oriented data

While this centralized model can work for organizations that have a simpler domain with smaller number of diverse consumption cases, it fails for enterprises with rich domains, a large number of sources and a diverse set of consumers.

There are two pressure points on the architecture and the organizational structure of a centralized data platform that often lead to its failure:

- **Ubiquitous data and source proliferation:** As more data becomes ubiquitously available, the ability to consume it all and harmonize it in one place under the control of one platform diminishes. Imagine just in the domain of 'customer information', there are an increasing number of sources inside and outside of the boundaries of the organization that provide information about the existing and potential customers. The assumption that we need to ingest and store the data in one place to get value from diverse set of sources is going to constrain our ability to respond to proliferation of data sources. I recognize the need for data users such as data scientists and analysts to process a diverse set of datasets with low overhead, as well as the need to separate the operational systems data usage from the data that is consumed for analytical purposes. But I propose that the existing centralized solution is not the optimal answer for large enterprises with rich domains and continuously added new sources.

- **Organizations' innovation agenda and consumer proliferation:** Organizations' need for rapid experimentation introduces a larger number of use cases for consumption of the data from the platform. This implies an ever growing number of transformations on the data - aggregates, projections and slices that can satisfy the [test and learn cycle of innovation](#). The long response time to satisfy the data consumer needs has historically been a point of organizational friction and remains to be so in the modern data platform architecture.

While I don't want to give my solution away just yet, I need to clarify that I'm not advocating for a fragmented, siloed domain-oriented data often hidden in the bowels of operational systems; siloed domain data that is hard to discover, make sense of and consume. I am not advocating for multiple fragmented data warehouses that are the results of years of accumulated tech debt. This is a concern that [leaders in the industry have voiced](#). But I argue that the response to these accidental silos of unreachable data is not creating a centralized data platform, with a centralized team who owns and curates the data from all domains. It does not organizationally scale as we have learned and demonstrated above.

Coupled pipeline decomposition

The second failure mode of a traditional data platform architecture is related to how we decompose the architecture. At 10,000 feet zooming into the centralized data platform, what we find is an architectural decomposition around the mechanical functions of *ingestion*, *cleansing*, *aggregation*, *serving*, etc. Architects and technical leaders in organizations decompose an architecture in response to the growth of the platform. As described in the previous section, the need for on-boarding new sources, or responding to new consumers requires the platform to grow. Architects need to find a way to scale the system by breaking it down to its **architectural quanta**. An architectural quantum, as described in [Building Evolutionary Architectures](#), is an independently deployable component with high functional cohesion, which includes all the structural elements required for the system to function properly. The motivation behind breaking a system down into its architectural quantum is to create independent teams who can each build and operate an architectural quantum. Parallelize work across these teams to reach higher operational scalability and velocity.

Given the influence of previous generations of data platforms' architecture, architects decompose the data platform to a *pipeline of data processing stages*. A pipeline that at a very high level implements a functional cohesion around the technical implementation of processing data; i.e. capabilities of *ingestion*, *preparation*, *aggregation*, *serving*, etc.

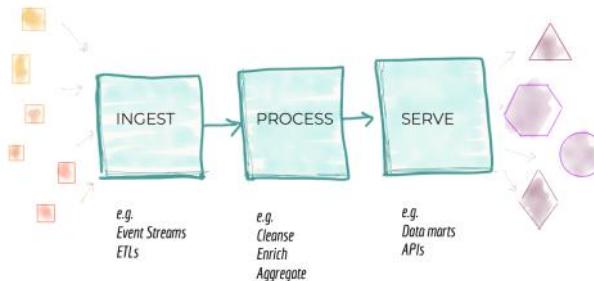


Figure 3: Architectural decomposition of data platform

Though this model provides some level of scale, by assigning teams to different stages of the pipeline, it has an inherent limitation that slows the delivery of features. It has high coupling between the stages of the pipeline to deliver an independent feature or value. It's decomposed *orthogonally to the axis of change*. Let's look at our media streaming example. Internet media streaming platforms have a strong domain construct around the type of media that they offer. They often start their services with 'songs' and 'albums', and then extend to 'music events', 'podcasts', 'radio shows', 'movies', etc. Enabling a single new feature, such as visibility to the 'podcasts play rate', requires a change in all components of the pipeline. Teams must introduce new ingestion services, new cleansing and preparation as well as aggregates for viewing podcast play rates. This requires synchronization across implementation of different components and release management across teams. Many data platforms provide generic and configuration-based ingestion services that can cope with extensions such as adding new sources easily or modifying the existing sources to minimize the overhead of introducing new sources. However this does not remove an end to end dependency management of introducing new datasets from the consumer point of view. Though on paper, the pipeline architecture might appear as if we have achieved an architectural quantum of a pipeline stage, in practice the whole pipeline i.e. the monolithic platform, is the smallest unit that must change to cater for a new functionality: unlocking a new dataset and making it available for new or existing consumption. This limits our ability to achieve higher velocity and scale in response to new consumers or sources of the data.

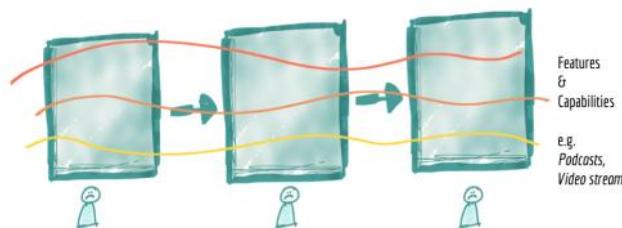


Figure 4: Architecture decomposition is *orthogonal to the axis of change* when introducing or enhancing features, leading to coupling and slower delivery

Siloed and hyper-specialized ownership

The third failure mode of today's data platforms is related to how we structure the teams who build and own the platform. When we zoom close enough to observe the life of the people who build and operate a data platform, what we find is a group of hyper-specialized data engineers siloed from the operational units of the organization; where the data originates or where it is used and put into actions and decision making. The data platform engineers are not only siloed organizationally but also separated and grouped into a team based on their technical expertise of big data tooling, often absent of business and domain knowledge.

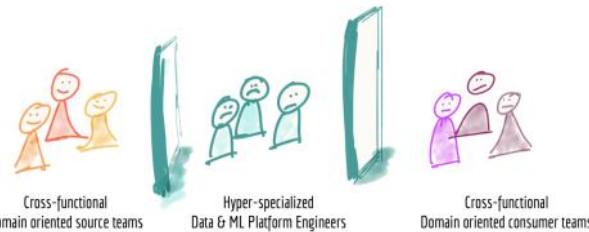


Figure 5: Siloed hyper-specialized data platform team

I personally don't envy the life of a data platform engineer. They need to consume data from teams who have no incentive in providing meaningful, truthful and correct data. They have very little understanding of the source domains that generate the data and lack the domain expertise in their teams. They need to provide data for a diverse set of needs, operational or analytical, without a clear understanding of the application of the data and access to the consuming domain's experts.

In the media streaming domain, for example, on the source end we have cross-functional 'media player' teams that provide signals around how users interact with a particular feature they provide e.g. 'play song events', 'purchase events', 'play audio quality', etc.; and on the other end sit the consumer cross-functional teams such as 'song recommendation' team, 'sales team' reporting sales KPIs, 'artists payment team' who calculate and pay artists based on play events, and so on. Sadly, in the middle sits the data platform team that through sheer effort provides suitable data for all sources and consumptions.

In reality what we find are disconnected source teams, frustrated consumers fighting for a spot on top of the data platform team backlog and an over stretched data platform team.

We have created an architecture and organization structure that does not scale and does not deliver the promised value of creating a data-driven organization.

The next enterprise data platform architecture

It embraces the *ubiquitous data* with a *distributed Data Mesh*.

So what is the answer to the failure modes and characteristics we discussed above? In my opinion a **paradigm shift** is necessary. A paradigm shift at the intersection of techniques that have been instrumental in building modern distributed architecture at scale; Techniques that the tech industry at large has adopted at an accelerated rate and that have created successful outcomes.

I suggest that the next enterprise data platform architecture is in the convergence of *Distributed Domain Driven Architecture*, *Self-serve Platform Design*, and *Product Thinking* with *Data*.

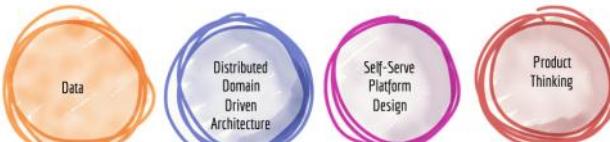


Figure 6: Convergence: the paradigm shift for building the next data platforms

Though this might sound like a lot of buzzwords in one sentence, each of these techniques have had a specific and incredibly positive impact in modernizing the technical foundations of operational systems. Let's deep dive into how we can apply each of these disciplines to the world of Data to escape the current paradigm, carried over from years of legacy data warehousing architecture.

Data and distributed domain driven architecture convergence

Domain oriented data decomposition and ownership

Eric Evans's book [Domain-Driven Design](#) has deeply influenced modern architectural thinking, and consequently the organizational modeling. It has influenced the microservices architecture by decomposing the systems into distributed services built around business domain capabilities. It has fundamentally changed how the teams form, so that a team can independently and autonomously own a domain capability.

Though we have adopted domain oriented decomposition and ownership when implementing operational capabilities, curiously we have disregarded the notion of business domains when it comes to data. The closest application of DDD in data platform architecture is for source operational systems to emit their business [Domain Events](#) and for monolithic data platform to ingest them. However beyond the point of ingestion the concept of domains and the ownership of the domain data by different teams is lost.

Domain Bounded Context is a wonderfully powerful tool to design the ownership of the datasets. Ben Stopford's [Data Dichotomy](#) article unpacks the concept of sharing of domain datasets through streams.

In order to decentralize the monolithic data platform, we need to reverse how we think about data, its locality and ownership. Instead of *flowing* the data from domains into a centrally owned data lake or platform, domains need to *host and serve* their domain datasets in an easily consumable way.

In our example, instead of imagining data flowing from media players into some sort of centralized place for a centralized team to receive, why not imagine a player domain owning and serving their datasets for access by any team for any purpose downstream. The physical location where the datasets actually reside and how they flow, is a technical implementation of the 'player domain'. The physical storage could certainly be a centralized infrastructure such as Amazon S3 buckets but player datasets content and ownership remains with the domain generating them. Similarly in our example, the 'recommendations' domain creates datasets in a format that is suitable for its application, such as a graph database, while consuming the player datasets. If there are other domains such as 'new artist discovery domain' which find the 'recommendation domain' graph dataset useful, they can choose to pull and access that.

This implies that we may duplicate data in different domains as we transform them into a shape that is suitable for that particular domain, e.g. a time series play event to related artists graph.

This requires shifting our thinking from a *push and ingest*, traditionally through ETLs and more recently through event streams, to *serving and pull* model across all domains.

The *architectural quantum* in a domain oriented data platform, is a *domain* and not the pipeline stage.

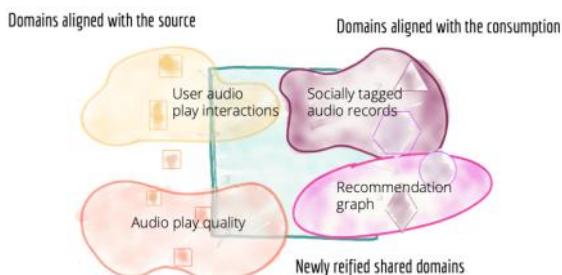


Figure 7: Decomposing the architecture and teams owning the data based on domains - source, consumer, and newly created shared domains

Source oriented domain data

Some domains naturally align with the source, where the data originates. The *source domain datasets* represent the *facts and reality of the business*. The *source domain datasets* capture the data that is mapped very closely to what the operational systems of their origin, *systems of reality*, generate. In our example facts of

the business such as 'how the users are interacting with the services', or 'the process of onboarding labels' lead to creation of domain datasets such as 'user click streams', 'audio play quality stream' and 'onboarded labels'. These facts are best known and generated by the operational systems that sit at the point of origin. For example the media player system knows best about the 'user click streams'.

In a mature and ideal situation, an operational system and its team or organizational unit, are not only responsible for providing business capabilities but also responsible for providing the *truths of their business domain* as source domain datasets. At enterprise scale there is never a one to one mapping between a domain concept and a source system. There are often many systems that can serve parts of the data that belongs to a domain, some legacy and some easy to change. Hence there might be many *source aligned datasets* aka *reality datasets* that ultimately need to be aggregated to a cohesive domain aligned dataset. The business facts are best presented as business [Domain Events](#), can be stored and served as distributed logs of time-stamped events for any authorized consumer to access.

In addition to timed events, source data domains should also provide easily consumable historical snapshots of the source domain datasets, aggregated over a time interval that closely reflects the interval of change for their domain. For example in an 'onboarded labels' source domain, which shows the labels of the artists that provide music to the streaming business, aggregating the onboarded labels on a monthly basis is a reasonable view to provide in addition to the events generated through the process of onboarding labels.

Note that the source aligned domain datasets must be separated from the internal source systems' datasets. The nature of the domain datasets is very different from the internal data that the operational systems use to do their job. They have a much larger volume, represent immutable timed facts, and change less frequently than their systems. For this reason the actual underlying storage must be suitable for big data, and separate from the existing operational databases. Section [Data and self-serve platform design convergence](#) describes how to create big data storage and serving infrastructure.

Source domain datasets are the most foundational datasets and change less often, as the facts of business don't change that frequently. These domain datasets are expected to be permanently captured and made available, so that as the organization evolves its *data-driven* and *intelligence* services they can always go back to the business facts, and create new aggregations or projections.

Note that source domain datasets represent closely the *raw data* at the point of creation, and are not fitted or modeled for a particular consumer.

Consumer oriented and shared domain data

Some domains align closely with the consumption. The consumer domain datasets and the teams who own them, aim to satisfy a closely related group of use cases. For example the 'social recommendation domain' that focuses on providing recommendations based on users social connections to each other, create domain datasets that fit this specific need; perhaps through a 'graph representation of social network of users'. While this graph dataset is useful for recommendation use case, it might be also useful for a 'listeners notifications' domain, which provides data regarding different types of notifications that are sent to the listener, including what people in their social network are listening to. So it is possible that 'user social network' can become a shared and newly reified domain dataset for multiple consumers to use. The 'user social network' domain team focuses on providing an always curated and up-to-date view of the 'user social network'.

The consumer aligned domain datasets have a different nature in comparison to source domains datasets. They structurally go through more changes, and they transform the source domain events to aggregate views and structures that fit a particular access model, such as the graph example we saw above. A domain oriented data platform should be able to easily regenerate these consumer datasets from the source.

Distributed pipelines as domain internal implementation

While the datasets ownership is delegated from the central platform to the domains, the need for cleansing, preparing, aggregating and serving data remains, so does the usage of data pipeline. In this architecture, a data pipeline is simply an internal complexity and implementation of the data domain and is handled internally within the domain. As a result we will be seeing a distribution of the data pipelines stages into each domain.

For example the source domains need to include the cleansing, deduplicating, enriching of their domain events so that they can be consumed by other domains, without replication of cleansing. Each domain dataset must establish a *Service Level Objectives* for the quality of the data it provides: timeliness, error rates, etc.

For example our media player domain providing audio 'play clickstream' can include cleansing and standardizing data pipeline in their domain that provides a stream of de-duped near real-time 'play audio click events' that conform to the organization's standards of encoding events.

Equally, we will see that aggregation stages of a centralized pipeline move into implementation details of consuming domains.

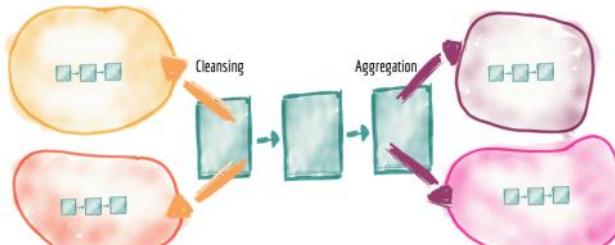


Figure 8: Distribute the pipelines into the domains as a second class concern and the domain's internal implementation detail

One might argue that this model might lead to duplicated effort in each domain to create their own data processing pipeline implementation, technology stack and tooling. I will address this concern shortly as we talk about the [Convergence of Data and Platform Thinking with Self-serve shared Data Infrastructure as a Platform](#).

Data and product thinking convergence

Distribution of the data ownership and data pipeline implementation into the hands of the business domains raise an important concern around accessibility, usability and harmonization of distributed datasets. This is where the learning in applying *product thinking* and ownership of data assets come handy.

Domain data as a product

Over the last decade operational domains have built [product thinking](#) into the capabilities they provide to the rest of the organization. Domain teams provide these capabilities as APIs to the rest of the developers in the organization, as building blocks of creating higher order value and functionality. The teams strive for creating the best developer experience for their domain APIs; including discoverable and understandable API documentation, API test sandboxes, and closely tracked quality and adoption KPIs.

For a distributed data platform to be successful, domain data teams must apply product thinking with similar rigor to the datasets that they provide; considering their data assets as their products and the rest of the organization's data scientists, ML and data engineers as their customers.

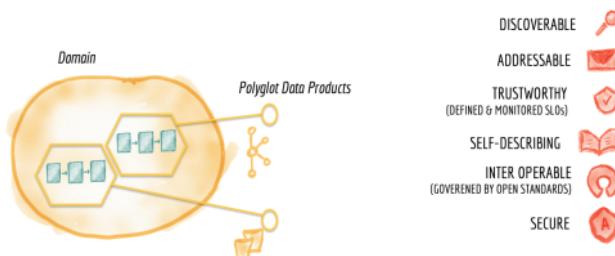


Figure 9: Characteristics of domain datasets as product

Consider our example, internet media streaming business. One of its critical domains is the 'play events', what songs have been played by whom, when and where. This key domain has different consumers in the organization; for example near real-time consumers that are interested in the experience of the user and

possibly errors so that in case of a degraded customer experience or an incoming customer support call can respond quickly to recover the error. There are also a few consumers that would prefer the historical snapshots of the daily, or monthly song play event aggregates.

In this case our 'played songs' domain provides two different datasets as its products to the rest of the organization; real-time play events exposed on event streams, and aggregated play events exposed as serialized files on an object store.

An important quality of any technical product, in this case domain data products, is to delight their consumers; in this case data engineers, ml engineers or data scientists. To provide the best user experience for consumers, the domain data products need to have the following basic qualities:

Discoverable

A data product must be easily discoverable. A common implementation is to have a registry, a data catalogue, of all available data products with their meta information such as their owners, source of origin, lineage, sample datasets, etc. This centralized discoverability service allows data consumers, engineers and scientists in an organization, to find a dataset of their interest easily. Each domain data product must register itself with this centralized data catalogue for easy discoverability.

Note the perspective shift here is from a single *platform extracting and owning* the data for its use, to each *domain providing its data as a product in a discoverable fashion*.

Addressable

A data product, once discovered, should have a unique address following a global convention that helps its users to programmatically access it. The organizations may adopt different naming conventions for their data, depending on the underlying storage and format of the data. Considering the ease of use as an objective, in a decentralized architecture, it is necessary for common conventions to be developed. Different domains might store and serve their datasets in different formats, events might be stored and accessed through streams such as Kafka topics, columnar datasets might use CSV files, or AWS S3 buckets of serialized [Parquet](#) files. A standard for addressability of datasets in a polyglot environment removes friction when finding and accessing information.

Trustworthy and truthful

No one will use a product that they can't trust. In the traditional data platforms it's acceptable to extract and onboard data that has errors, does not reflect the truth of the business and simply can't be trusted. This is where the majority of the efforts of centralized data pipelines are concentrated, cleansing data after ingestion. A fundamental shift requires the owners of the data products to provide an acceptable [Service Level Objective](#) around the truthfulness of the data, and how closely it reflects the reality of the events that have occurred or the high probability of the truthfulness of the insights that have been generated. Applying data cleansing and automated data integrity testing at the point of creation of the data product are some of the techniques to be utilized to provide an acceptable level of quality. Providing [data provenance and data lineage](#) as the metadata associated with each data product helps consumers gain further confidence in the data product and its suitability for their particular needs.

The target value or range of a data integrity (quality) indicator vary between domain data products. For example, 'play event' domain may provide two different data products, one near-real-time with lower level of accuracy, including missing or duplicate events, and one with longer delay and higher level of event's accuracy. Each data product defines and assure the target level of its integrity and truthfulness as a set of SLOs.

Self-describing semantics and syntax

Quality products require no consumer hand holding to be used: they can be independently discovered, understood and consumed. Building datasets as products with minimum friction for the data engineers and data scientists to use requires well described semantics and syntax of the data, ideally accompanied with sample datasets as exemplars. Data schemas are a starting point to provide self-serve data assets.

Inter-operable and governed by global standards

One of the main concerns in a distributed domain data architecture, is the ability to correlate data across domains and stitch them together in wonderful, insightful ways; join, filter, aggregate, etc. The key for an effective correlation of data across domains is following certain standards and harmonization rules. Such standardizations should belong to a global governance, to enable interoperability between polyglot domain datasets. Common concerns of such standardization efforts are field type formatting, identifying [polysemes](#) across different domains, datasets address conventions, common metadata fields, event formats such as [CloudEvents](#), etc.

For example in the media streaming business, an 'artist' might appear in different domains and have different attributes and identifiers in each domain. The 'play eventstream' domain may recognize the artist differently to 'artists payment' domain that takes care of invoices and payments. However to be able to correlate the data about an artist across different domain data products we need to agree on how we identify an artist as a polyseme. One approach is to consider 'artist' with a federated entity and a unique global federated entity identifier for the 'artist', similarly to how [federated identities](#) are managed.

Interoperability and standardization of communications, governed globally, is one of the foundational pillars for building distributed systems.

Secure and governed by a global access control

Accessing product datasets securely is a must, whether the architecture is centralized or not. In the world of decentralized domain oriented data products, the access control is applied at a finer granularity, for each domain data product. Similarly to operational domains the access control policies can be defined centrally but applied at the time of access to each individual dataset product. Using the [Enterprise Identity Management system \(SSO\)](#) and [Role Based Access Control](#) policy definition is a convenient way to implement product datasets access control.

Section [Data and self-service platform design convergence](#) describes the shared infrastructure that enables the above capabilities for each data product easily and automatically.

Domain data cross-functional teams

Domains that provide data as products; need to be augmented with new skill sets: (a) the *data product owner* and (b) *data engineers*.

A data product owner makes decisions around the vision and the roadmap for the data products, concerns herself with satisfaction of her consumers and continuously measures and improves the quality and richness of the data her domain owns and produces. She is responsible for the lifecycle of the domain datasets, when to change, revise and retire data and schemas. She strikes a balance between the competing needs of the domain data consumers.

Data product owners must define success criteria and business-aligned Key Performance Indicators (KPIs) for their data products. For example, the lead time for consumers of a data product to discover and use the data product successfully, is measurable success criteria.

In order to build and operate the internal data pipelines of the domains, teams must include data engineers. A wonderful side effect of such cross-functional team is cross pollination of different skills. My current industry observation is that some data engineers, while competent in using the tools of their trade, lack software engineering standard practices, such as continuous delivery and automated testing, when it comes to building data assets. Similarly software engineers who are building operational systems often have no experience utilizing data engineering tool sets. Removing the skillset silos will lead to creation of a larger and deeper pool of data engineering skill sets available to the organization. We have observed the same cross-skill pollination with the DevOps movement, and the birth of new types of engineers such as [SREs](#).

Data must be treated a foundational piece of any software ecosystem, hence software engineers and *software generalists* must add the experience and knowledge of data product development to their tool belt. Similarly infrastructure engineers need to add knowledge and experience of managing a data infrastructure. Organizations must provide career development pathways from a *generalist* to a *data engineer*. The lack of data engineering skills has led to the *local optimization* of forming centralized data engineering teams as described in section [Siloed and hyper-specialized ownership](#).

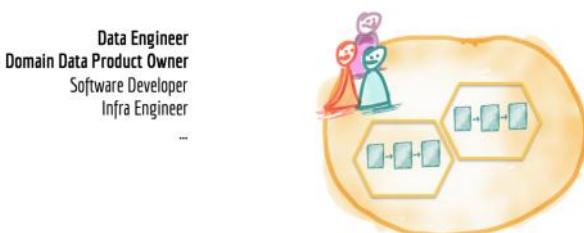


Figure 10: Cross functional domain data teams with explicit data product ownership
Data and self-serve platform design convergence

One of the main concerns of distributing the ownership of data to the domains is the duplicated effort and skills required to operate the data pipelines technology stack and infrastructure in each domain. Luckily, building common infrastructure as a platform is a well understood and solved problem; though admittedly the tooling and techniques are not as mature in the data ecosystem.

Harvesting and extracting domain agnostic infrastructure capabilities into a data infrastructure platform solves the need for duplicating the effort of setting up data pipeline engines, storage, and streaming infrastructure. A data infrastructure team can own and provide the necessary technology that the domains need to capture, process, store and serve their data products.

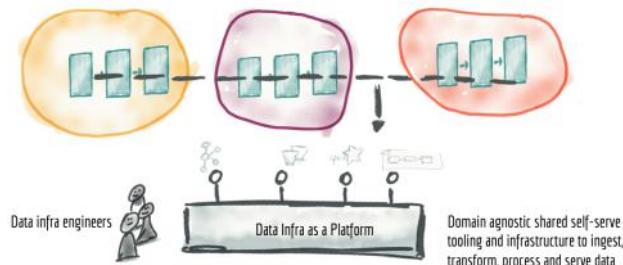


Figure 11: Extracting and harvesting domain agnostic data pipeline infrastructure and tooling into a separate data infrastructure as a platform

The key to building the *data infrastructure as a platform* is (a) to not include any domain specific concepts or business logic, keeping it domain agnostic, and (b) make sure the platform hides all the underlying complexity and provides the data infrastructure components in a self-service manner. There is a long list of capabilities that a self-service data infrastructure as a platform provides to its users, a domain's data engineers. Here are a few of them:

- Scalable polyglot big data storage
- Encryption for data at rest and in motion
- Data product versioning
- Data product schema
- Data product de-identification
- Unified data access control and logging
- Data pipeline implementation and orchestration
- Data product discovery, catalog registration and publishing
- Data governance and standardization
- Data product lineage
- Data product monitoring/alerting/log
- Data product quality metrics (collection and sharing)
- In memory data caching
- Federated identity management
- Compute and data locality

A success criteria for self-service data infrastructure is lowering the 'lead time to create a new data product' on the infrastructure. This leads to automation, required for implementing the capabilities of a 'data product' as covered in section [Domain data as a product](#). For example, automating data ingestion through configurations and scripts, data product creation scripts to put scaffolding in place, auto-registering a data product with the catalog, etc.

Using cloud infrastructure as a substrate reduces the operational costs and effort required to provide on-demand access to the data infrastructure, however it doesn't completely remove the higher abstractions that need to be put in place in the context of the business. Regardless of the cloud provider there is a rich and ever growing set of data infrastructure services that are available to the data infra team.

The paradigm shift towards a data mesh

It's been a long read. Let's bring it all together. We looked at some of the underlying characteristics of the current data platforms: *centralized, monolithic*, with *highly coupled pipeline* architecture, operated by silos of *hyper-specialized data engineers*. We introduced the building blocks of a ubiquitous **data mesh** as a platform; *distributed data products* oriented around domains and owned by *independent cross-functional teams* who have embedded data engineers and data product owners, using common *data infrastructure* as a platform to host, prep and serve their data assets.

The data mesh platform is an intentionally designed distributed data architecture, under centralized governance and standardization for interoperability, enabled by a shared and harmonized self-service data infrastructure. I hope it is clear that it is far from a landscape of fragmented silos of inaccessible data.

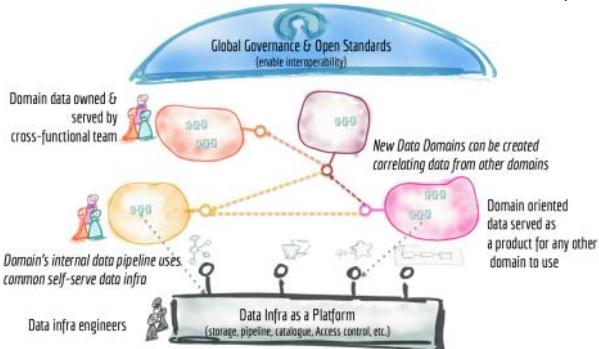


Figure 12: Data mesh architecture from 30,000 foot view

You might ask where does the [data lake](#) or data warehouse fit in this architecture? They are simply nodes on the mesh. It's very likely that we won't need a data lake, because the distributed logs and storage that hold the original data are available for exploration from different addressable *immutable datasets* as products. However, in cases where we do need to make changes to the original format of the data for further exploration, such as labeling, the domain with such need might create its own lake or data hub.

Accordingly, the data lake is no longer the centerpiece of the overall architecture. We will continue to apply some of the principles of data lake, such as making immutable data available for explorations and analytical usage, to the *source oriented domain data products*. We will continue to use the data lake tooling, however either for internal implementation of data products or as part of the shared data infrastructure.

This, in fact, takes us back to where it all began: [James Dixon in 2010](#) intended a data lake to be used for a single domain, multiple data domains would instead form a 'water garden'.

The main shift is to treat *domain data product* as a first class concern, and data lake tooling and pipeline as a second class concern - an implementation detail. This inverts the current mental model from a centralized data lake to an ecosystem of data products that play nicely together, a *data mesh*.

The same principle applies to the data warehouse for business reporting and visualization. It's simply a node on the mesh, and possibly on the consumer oriented edge of the mesh.

I admit that though I see the data mesh practices being applied in pockets at my clients, enterprise scale adoption still has a long way to go. I don't believe

technology is the limitation here, all the tooling that we use today can accommodate distribution and ownership by multiple teams. Particularly the shift towards *unification of batch and streaming* and tools such as [Apache Beam](#) or [Google Cloud Dataflow](#), easily allow processing addressable polyglot datasets. Data catalog platforms such as [Google Cloud Data Catalog](#) provide central discoverability, access control and governance of distributed domain datasets. A wide variety of [cloud data storage](#) options enables domain data products to choose fit for purpose polyglot storage.

The needs are real and tools are ready. It is up to the engineers and leaders in organizations to realize that the existing paradigm of *big data* and *one true big data platform* or data lake, is only going to repeat the failures of the past, just using new cloud based tools.

This paradigm shift requires a new set of governing principles accompanied with a **new language**:

- *serving over ingesting*
- *discovering and using over extracting and loading*
- *Publishing events as streams over flowing data around via centralized pipelines*
- *Ecosystem of data products over centralized data platform*

Let's breakdown the big data monolith into a harmonized, collaborative and distributed ecosystem of data mesh.

Communication

Tuesday, December 27, 2022 9:07 PM

RE: Data Mesh



Malhotra, Dinesh

To ● Cheema, Dave

Retention Policy UHGIinbox (90 days)

Expires 3/28/2023



Hi Dave – thank you very much for stepping in and taking a lead on this important topic

As you formally step in this role to lead a new thought-leader content, I am sharing my expectations and some guidance tips to help you be a successful thought leader:

1. You are the primary owner for this thought leadership work product – and you need to have a perspective on this topic to get the dialogue started
 - a. As the primary leader, you will have to do most of the research to facilitate the dialogue
 - b. Be prepared to facilitate the meeting – everyone would be counting on you for that
 - c. After this thought-leader content is delivered, you will be the formal SME on this topic in PTS to be pulled in for various opportunities as needed
2. While this is a technology topic and the technology understanding would be critical, our PoV / white paper should also address organizational, people and process aspects for companies that may contemplate adopting Data Mesh:
 - a. For the meeting on Jan 10 – I would suggest creating a draft outline in the form of Table of Contents for a white paper
 - i. Put a brief description (3-5 bullets / sentences) against each Table of Contents as to what you think would be worth incorporating / addressing
 - ii. Question you identified for the meeting could help tee up this Table of Contents – may require some critical thinking
3. Outside the meeting, you will be the primary owner and driver for the white paper to bring it to life
 - a. Based on the meeting on Jan 10, identify who else from the team could help contribute for some specific aspects of this PoV; engage them and partner with them to build the content
 - b. You may have collaborators to contribute – you would still own the successful completion of this white paper
 - c. I will be happy to serve as a reviewer and provide feedback and input as appropriate
4. After the initial meeting, put together a timeline with targeted milestones – typically, these PoVs can take anywhere between 3-6 weeks depending upon the complexity of the topic
 - a. Initial draft ready date
 - b. Initial draft QA date
 - c. Draft submission to marketing team for their review / feedback
 - d. Draft finalization date

Thank you for taking the leadership on this important practice building initiative in the new performance year. I appreciate it.

Best,

Dinesh Malhotra

M: +1 773-398-7713

RE: Mesh or not to Mesh



Malhotra, Dinesh

To ● Cheema, Dave

Cc ○ Shin, John S; ○ Nand, Durga; ○ Anantha Ramakrishnan, Srinivasan

Retention Policy UHGIinbox (90 days)

Expires 3/27/2023

Thank you Dave for your leadership.

- Can you please put together an initial outline to help react and confirm the scope of this PoV?
- How soon can we get on a calendar to review the outline to confirm the scope?

Please send the invite to this group. Thank you!!!

Best,

Dinesh Malhotra

M: +1 773-398-7713

From: Cheema, Dave <dave.cheema@optum.com>

Sent: Tuesday, December 27, 2022 11:57 AM

To: Malhotra, Dinesh <dinesh.malhotra@optum.com>; Anantha Ramakrishnan, Srinivasan <srinivasan_anantharamakrishnan@optum.com>

Cc: Shin, John S <john.shin@optum.com>; Nand, Durga <durga.nand@optum.com>

Subject: RE: Mesh or not to Mesh

Yes sir. I will do my best. Thank you.

Regards,

Dave Cheema

(952) 205-0802

From: Malhotra, Dinesh <dinesh.malhotra@optum.com>

Sent: Tuesday, December 27, 2022 11:54 AM

To: Anantha Ramakrishnan, Srinivasan <srinivasan_anantharamakrishnan@optum.com>; Cheema, Dave <dave.cheema@optum.com>

Cc: Shin, John S <john.shin@optum.com>; Nand, Durga <durga.nand@optum.com>

Subject: RE: Mesh or not to Mesh

Importance: High

Agree - these both are in relatively nascent stages – with good degree of overlap. One difference often called out for Data Fabric is inclusion of ‘data virtualization’ as a technology enabler – Data Mesh doesn’t advocate that.

I recommend you, John and Cheema should collaborate to build a PoV – when to Mesh and when not to Mesh, when to go for Fabric etc.

- This could be a PoV that factors in the organization size, organizational agility, data value potential, data usage scope, data maturity etc. to assess whether juice is worth the squeeze – and what would drive the ROI for making such an investment
- We shouldn’t try to make this PoV an education material on Mesh and Fabric – but a slide delineating key characteristics of the two would be good to have
- As we build the initial strawman – we should quickly assess if there is a ‘market play’ for our team to position some key services we can sell

Dave – is this something you can lead and give us a jump start on?

Best,

Dinesh Malhotra

M: +1 773-398-7713

From: Anantha Ramakrishnan, Srinivasan <srinivasan_anantharamakrishnan@optum.com>

Sent: Tuesday, December 27, 2022 11:44 AM

To: Malhotra, Dinesh <dinesh.malhotra@optum.com>

Cc: Shin, John S <john.shin@optum.com>; Nand, Durga <durga.nand@optum.com>

Subject: RE: Mesh or not to Mesh

Thank you Dinesh, Sure will do the read of this opinion.

I am we all are aware of a competing concept, data fabric and there were many debates to qualify which one is better and when etc.

Regards

Ananth

Anantha Ramakrishnan

M: +1 551 358 4609

From: Malhotra, Dinesh <dinesh.malhotra@optum.com>

Sent: Monday, December 26, 2022 6:00 PM

To: Anantha Ramakrishnan, Srinivasan <srinivasan_anantharamakrishnan@optum.com>

Cc: Shin, John S <john.shin@optum.com>; Nand, Durga <durga.nand@optum.com>

Subject: Mesh or not to Mesh

Hi Anantha - take a read of this article on Data Mesh as a concept.

<https://martinfowler.com/articles/data-monolith-to-mesh.html>

Slightly long reading material - should you prefer, you can also watch Zhamak Dehgani's YouTube video to get the drift

<https://youtu.be/52MCFe4v0UU>

Data Mesh is not a panacea and can't be advocated for every client situation.

I do feel it offers some concepts and guiding principles that are time-tested and can allow us to infuse some critical thinking in our deliverable for Insurance Solution and also for our dialogue with Geisenger.

Please take a read and I would like to get your thoughts on what elements can be aligned with our recommendations for DIG project and thoughts it generates for Geisenger client meeting.

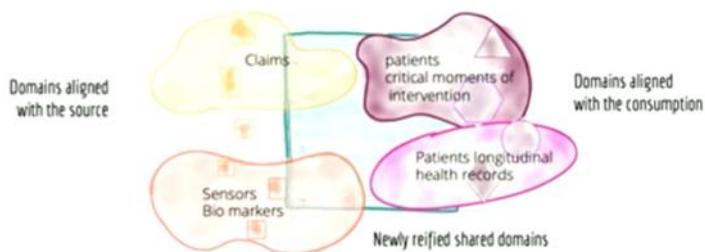
Miscellaneous

Tuesday, December 27, 2022 4:24 PM

DATA MESH PRINCIPLES



DOMAIN ORIENTED DATA DECOMPOSITION & OWNERSHIP



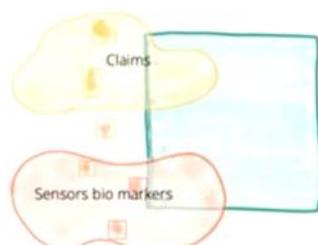
SOURCE ORIENTED (NATIVE) DOMAIN DATA

Facts & reality of business

Immutable timed events /
Historical snapshots

Change less frequently

Permanently captured



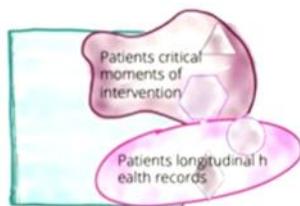
CONSUMER ORIENTED DOMAIN DATA

Fit for consumer purpose

Aggregation / Projections / Transformed

Change more often

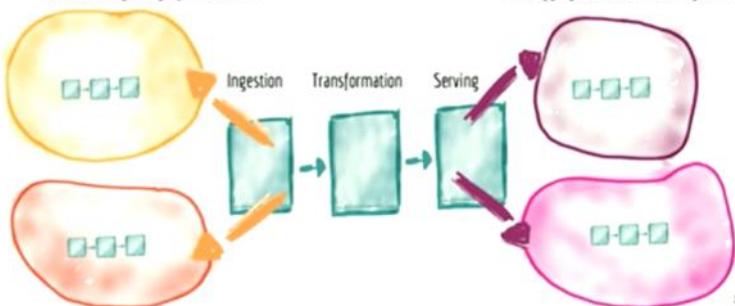
Can be recreated



DISTRIBUTED PIPELINES IN DOMAINS

More cleansing, integrity checks here

More aggregations, ML modelling here



Domains are the first class concern
Top Level partitions

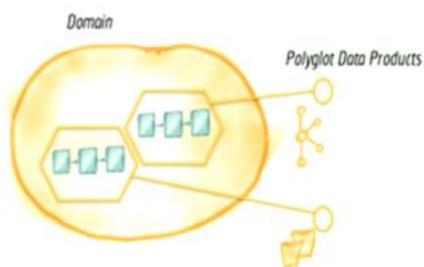
Data pipelines are second class concern
Implementation details

Architectural Quantum shifts from a
pipeline to a domain (datasets)

Domain datasets are immutable (time series)



DOMAIN DATA AS A PRODUCT



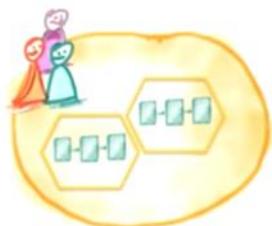
DOMAIN DATA AS A PRODUCT

Aka Data Products



CROSS-FUNCTIONAL DOMAIN TEAMS

Data Engineer
Domain Data Product Owner
Software Developer
Infra Engineer
--



Domain Datasets as a product

Discoverable
Inter-operable
Explicit Quality Objectives
Secure
Shared

Data consumers as customers

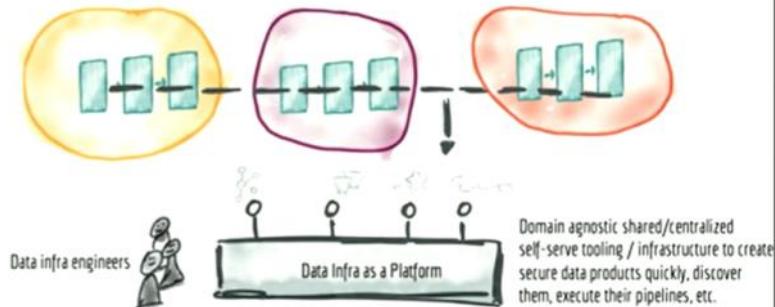
Data Product Owner role

Cross-functional team ownership

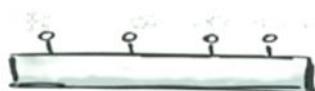
Success criteria: Decreased lead time to discover and consume a data product

Product Thinking

DATA INFR AS A PLATFORM



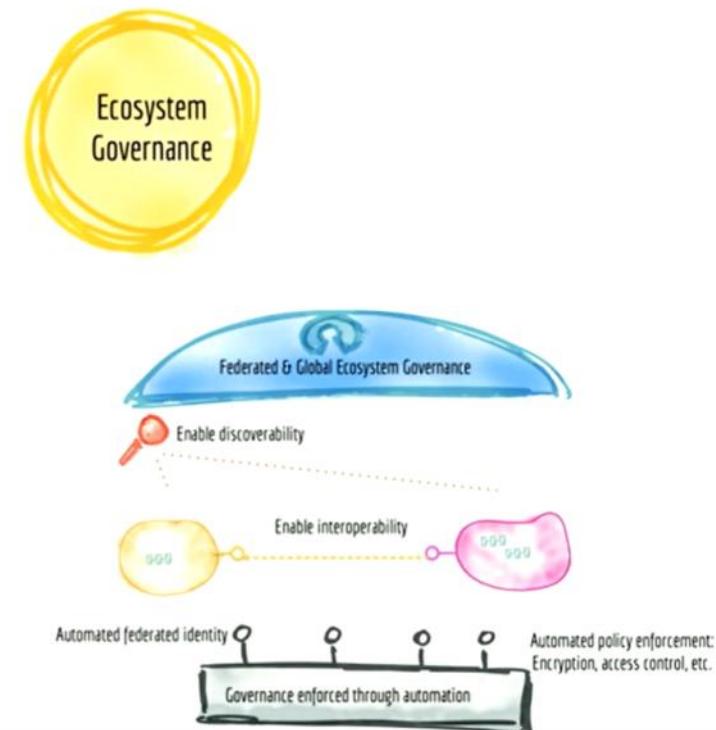
Scalable polyglot storage on demand
Encryption for data at rest and in motion
Unified data access control
Data product discoverability
Data product SLO / metrics collection & sharing
Data pipeline orchestration / templates
Data Product CI/CD pipeline
Automate ecosystem governance
Guidelines
Data product scaffolding

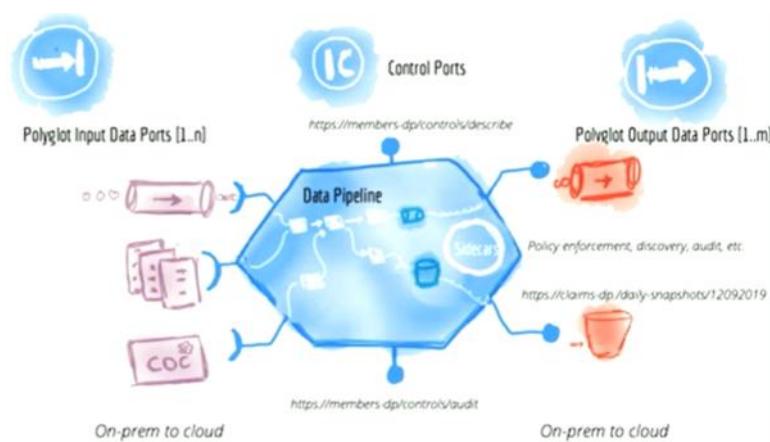
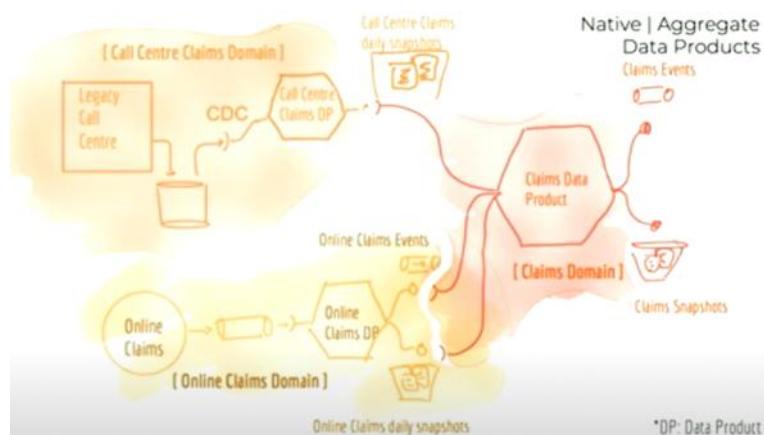
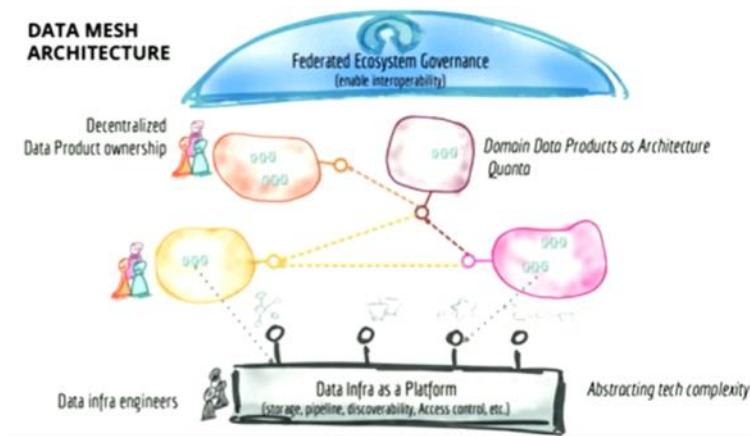


Data infrastructure & Tooling (DataOps)
Shared, Self-serve, as a Platform
Domain agnostic
Owned by data infra and tooling team
At incubation it's opinionated
Ideally built on Cloud data services (despite lack of maturity)



Success criteria: reduced lead time to create new secure & discoverable data products





DATA DISCOVERABILITY A FIRST CLASS CONCERN

```
curl -X GET http://(epiURL)/api/v1/dataproducts/(dataProductGUID)
...
{
  "dataProductGUID": "ed457526-8d7e-4f58-905a-18aadd6eab",
  "createDate": "2019-06-07T19:11:59.490Z",
  "owner": "Data Product A",
  "displayname": "Data Product A",
  "sourceCodeLocation": "https://github.com/dataneshexample/data_product_A.git",
  "description": "Contains datasets related to Data Product A with semantically same data represented in various formats",
  "readableDataProductKey": "data_product_A",
  "requiredSecurityGroup": "You can reach out to the owner to request access until this is fully implemented"
}
```



WHERE IS THE LAKE?
WHERE IS THE WAREHOUSE?



ADOPT A NEW LANGUAGE

FROM	TO
Ingesting	Serving
Extracting & Loading	Discovering & Consuming
Flowing data through centralized Pipelines	Publishing output data ports
Centralized Data Lake Warehouse Platform	Ecosystem of Data Products

CLOSING THE DIVIDE



“
*By doubting we are led to question,
 by questioning we arrive at the truth.*”

- Pierre Abélard

How to Move Beyond a Monolithic Data Lake to a Distributed Data Mesh

Many enterprises are investing in their next generation data lake, with the hope of democratizing data at scale to provide business insights and ultimately make automated intelligent decisions. Data platforms based on the data lake architecture have common failure modes that lead to unfilled promises or waste. To address these failure modes we need to shift from the centralized paradigm of a lake or its associated data warehouse. We need to shift to a paradigm that draws from modern distributed architectures, understanding domains as the first class citizens, applying platform thinking to create self-service data infrastructure and treating data as a product.

22 May, 2019



Zhamak Dehghani

CONTENTS

- The current enterprise data platform architecture
- Architectural failure modes
- Centralized data architecture
- Coupled pipeline decomposition
- Shared and hyper-specialized parenting
- The next enterprise data platform architecture
- Domain driven data architecture and architecture convergence
- Domain oriented data decomposition and parenting
- Source oriented domain data
- Centralized pipelines as shared internal data
- Distributed pipelines as domain internal implementation
- Data and product thinking convergence
- Domain data as a product
- Summary
- Additional resources

<https://martinfowler.com/articles/data-monolith-to-mesh.html>

Data Mesh

Dinesh Malhotra, John shin, durga nand, Anantha Ramakrishnan, Srinivasan

Data Mesh evolution: data warehouse, Data Lake, Data Mesh

How is it different from the Data pipelines and data lakes

Key ingredients of Data Mesh Architecture

- Data, Distributed Domain Driven Architecture, Self-serve platform
- Design, Product thinking

Ben Stopford's Data Dichotomy article unpacks the concept of sharing of domain datasets through streams.

instead of imagining data flowing from media players into some sort of centralized place for a centralized team to receive, why not imagine a player domain owning and serving their datasets for access by any team for any purpose downstream. The physical location where the datasets actually reside and how they flow, is a technical implementation of the 'player domain'.

Data Mesh

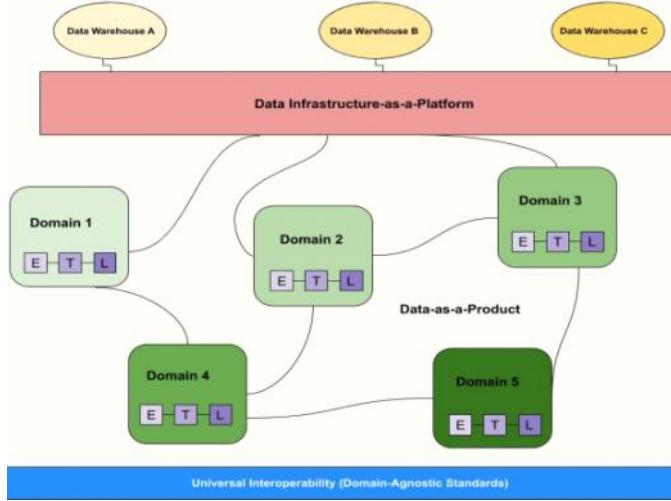
Zhamak Dehghani, founder of the data mesh, defines the data mesh as “**a socio-technical shift—a new approach in how we collect, manage, and share analytical data.**” The data mesh is **NOT A TECHNICAL SOLUTION** or even subset of technologies – it’s an **organizational paradigm for how we manage and operationalize data**, made up of several different technologies, whether open source or SaaS.

Data mesh **makes your data discoverable, widely accessible, secure, and interoperable** — giving you better decision-making power and faster time to value.

According to Google Trends, in 2021, “data mesh” overcame the “[data lakehouse](#)” that had, until now, been fairly popular in the industry.

What is a data mesh?

An organizational construct that embraces the ubiquity of data in the enterprise by leveraging a domain-oriented, self-serve design. It is based domain-driven design - a flexible, scalable software development paradigm that matches the structure and language of your code with its corresponding business domain. A data mesh supports distributed, domain-specific data consumers and views “data-as-a-product,” with each domain handling their own data applications. The universal interoperability layer connects these domains and their associated data using the same syntax and data governance.

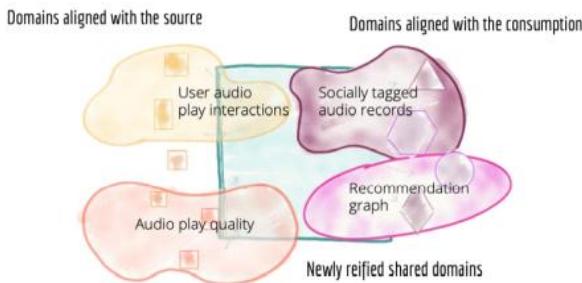


A data mesh is composed of three separate components: 1. **data sources**; 2. **data infrastructure** (storage, pipeline, catalog, etc.); 3. **domain-oriented data products** managed by functional owners. Underlying the **data mesh architecture** is a layer of **universal interoperability**, reflecting **domain-agnostic standards**, as well as **observability and governance**.

The 3 main components of a data mesh

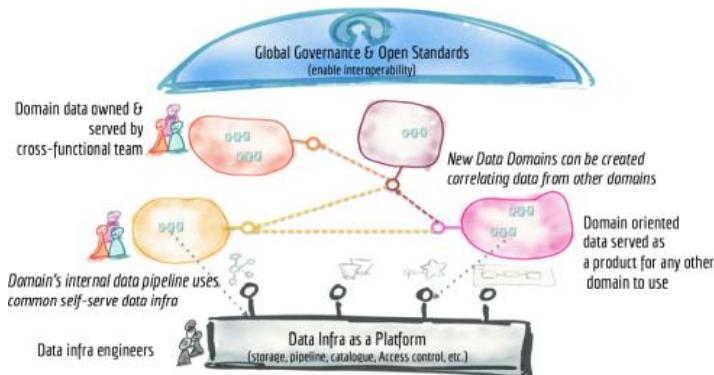
1. Domain-oriented data owners and pipelines

- a. Data meshes **federate data ownership among domain data owners** who are held accountable for providing their data as products, while also **facilitating communication between distributed data** across different locations. While the **data infrastructure is responsible for providing each domain with the solutions with which to process it**, domains are tasked with managing ingestion, cleaning, and aggregation to the data to generate assets that can be used by business intelligence applications.



- b. **Each domain is responsible for owning their data and associated data products**, but a set of capabilities applied to all domains that stores, catalogs, and maintains access controls for the raw data. Once data has been served to and transformed by a given domain, the domain owners can then leverage the data for their analytics needs.

- 2. **Self-service functionality** Data meshes leverage principles of domain-oriented design to deliver a self-service data platform that allows users to abstract the technical complexity and focus on their individual data use cases. To address the duplication of efforts and skills needed to maintain data pipelines and infrastructure in each domain concern, the data mesh gleans and extracts domain-agnostic data infrastructure capabilities into a central platform that handles the data pipeline engines, storage, and streaming infrastructure. Meanwhile, **each domain is responsible for leveraging these components to run custom ETL pipelines**, giving them the support necessary to easily serve their data as well as the autonomy required to truly own the process.



3. Interoperability and standardization of communications

- Underlying each domain is a **universal set of data standards** that helps facilitate collaboration between domains (and data products) when necessary. It's inevitable that some data (both raw sources and cleaned, transformed, and served data sets) will be valuable to more than one domain. **To enable cross-domain collaboration, the data mesh must standardize on formatting, governance, discoverability, and metadata fields**, among other data features. Moreover, much like an individual microservice, **each data domain must define and agree on SLAs and quality measures that they will "guarantee" to its consumers**.

When a data mesh makes sense

The data lake provided with real-time data availability and stream processing, with the goal of ingesting, enriching, transforming, and serving data from a centralized data platform. This type of architecture falls short in a few ways:

- Centralized pipelines gives teams less control over increasing volumes of data
- Domain-oriented data architectures, like data meshes, give teams the best of both worlds: a centralized database (or a distributed data lake) with domains (or business areas)

responsible for handling their own pipelines.

Should you team implement a data mesh?

1. **Quantity of data sources** - How many data sources does your company have?
2. **Team size** - How many data analysts, data engineers, product managers are on the data team?
3. **Number of data domains** - How many functional teams rely on your data sources, how many products do you have, how many data driven features are being built? And the total.
4. **Data engineering bottlenecks** - How frequently data engineering team a bottleneck to implement new data products (on a scale of 1 - 10; 1 being Never and 10 being always)?
5. **Data governance** - How much priority is the data governance (on a scale of 1 - 10; 1 not a concern and 10 being keeps up at night)?

Answer each of the questions above with a number and add them all together for a total - your data mesh score. If data mesh score is:

1. 1 - 15: Don't need a data mesh
2. 15 - 30: Incorporate some data mesh practices and concepts so that a later migration might be easier
3. 30 or more: A data mesh will support any ongoing/future initiatives to democratize data and provide self-service analytics

3 common data mesh challenges to implement a data mesh

1. **Achieving high quality data** - Data quality is top of mind. One of the biggest challenges organizations face when implementing a data mesh is achieving and maintaining high-quality data across various systems. A good data mesh will mandate scalable and self-serve observability into your data, giving your team a full picture of your data health.
2. **Organization-wide adoption of the data mesh** - Each domain must define and agree on SLAs and quality measures that they will "guarantee" to its customers.
3. **Showing value early on while planning for the long term** - Increase self-serve access to data. The data should be readily available to them at a moment's notice.

Challenges

- Data Mesh is a longer term investment
- Buy-in and commitment from the broader business
- Business priorities can shift quickly
- Established structures and ways of working
- Change generates friction
- Hybrid technology landscape

Enablers

- Transformation programme
- Cloud adoption and self-service
- Squad delivery model
- Small expert team: can think, do and respond to change quickly and autonomously
- Attack the problem from different angles:
 - Find manageable use cases to validate the approach and create traction
 - Build new capabilities e.g. Data Discovery
 - Make fundamental data sources consumable
 - Share and educate

How to trust data across your data mesh

A data mesh actually mandates scalable, self-serve observability into your data. Domains cannot truly own their data if they don't have observability. Such self-serve capabilities inherent to any good data mesh include:

- Encryption for data at rest and in motion
- Data product versioning
- Data product schema
- Data product discovery, catalog registration, and publishing
- Data governance and standardization
- Data production lineage
- Data product monitoring, alerting, and logging
- Data product quality metrics

When packaged together, these functionalities and standardizations provide a robust layer of observability. The data mesh paradigm also prescribes having a standardized, scalable way for individual domains to handle these various tenants of observability, allowing teams to answer these questions and many more:

- Is my data fresh?
- Is my data broken?
- How do I track schema changes?
- What are the upstream and downstream dependencies of my pipelines?

References

Articles

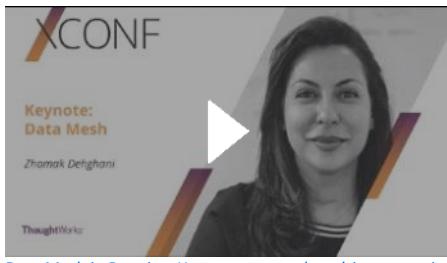
- [How to Move Beyond a Monolithic Data Lake to a Distributed Data Mesh \(martinfowler.com\)](#)
- [Data Mesh Principles and Logical Architecture \(martinfowler.com\)](#)
- [Data Mesh Applied. Subtitle: Moving step-by-step from mono... | by Sven Balnojan | Towards Data Science](#)
- [Intuit's Data Mesh Strategy. Intuit's mission is 'Power Prosperity... | by Tristan Baker | Intuit Engineering | Medium](#)
- [How JPMorgan Chase built a data mesh architecture to drive significant value to enhance their enterprise data platform | AWS Big Data Blog \(amazon.com\)](#)

Videos

- [Data Mesh at HelloFresh - A Work in Progress; Data Mesh Learning Meetup #003](#)



- [Keynote - Data Mesh by Zhamak Dehghani](#)



- [Data Mesh in Practice: How to set up a data driven organization - Max Schultze, Zalando](#)



- [Data Mesh Paradigm Shift in Data Platform Architecture](#)



- [How JPMorgan Chase built a data mesh architecture to drive significant value to enhance their enterprise data platform | AWS Big Data Blog \(amazon.com\)](#)
- [Data Mesh at CMC Markets: Past, Present and Future; Data Mesh Learning Meetup #008 \(Fixed Audio\)](#)



DECOMPOSE DATA AROUND DOMAINS

Distribute the ownership

Domains aligned with the origin of data

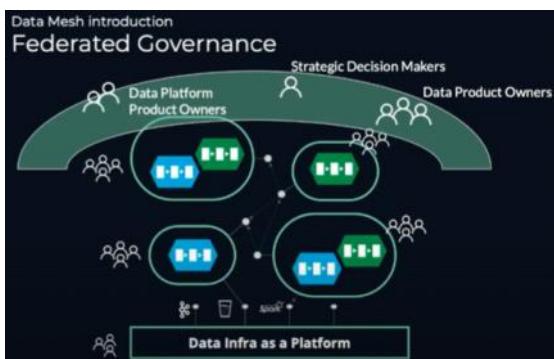
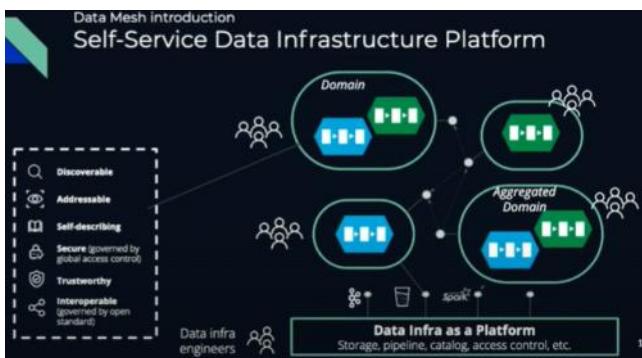
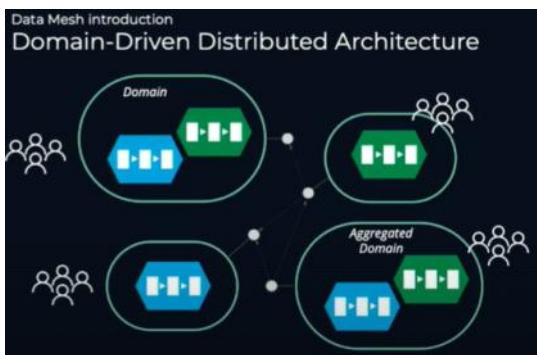
Facts & reality of business
Immutable trend events
Historical snapshots
Change less frequently
Permanently captured



Domains aligned with the consumption

Fit for consumer purpose
Aggregations / Projections / Transformed
Change more often
Can be recreated





Data Mesh is...

- Product Thinking for data
- Domain-Driven Design applied to distributed data
- Platform Thinking for data infrastructure
- Federated Governance

Data Mesh is not...

- A detailed framework or solution to be implemented

Scaling the Mesh Centralization Challenges

- Falling into the trap of taking centralized responsibility about data
- No explicit ownership
- No data quality guarantees
- Becoming the bottleneck



Scaling the Mesh Providing a Data Infrastructure Platform

- Data Agnostic
- Self Service
- Standardization of Tooling



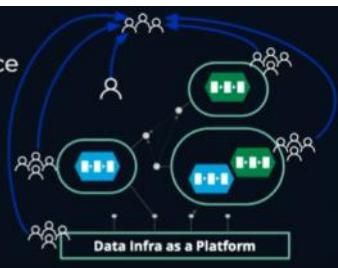
Organizational Alignment Decentralization Challenges

- Unaligned Data Products
- Missing Interoperability
- Data Silos



Organizational Alignment Federated Governance

- Bringing representatives of all groups to the table
- Localization of domain internal concerns
- Standardization of global processes



The Data Mesh Journey Summary

Getting Started

- Foster empathy
- Create incentives
- Start small

Scaling the Mesh

- Self Service
- Data Agnostic
- Infrastructure Platform

Organizational Alignment

- Bring local and global incentives together
- Enable decentral decision making
- Interoperability through global standardization

CMC Markets Data & Transformation

① Data is at the core of transformation

- New cloud-based products
- Valuable data is on premise

② Existing data is **decentralised/siloed**

⇒ Coupling of “**knowledge**” and “**work**” in a single team

- Find where the data you need might be
- Then wait for work to be done for you
- Results in queuing, additional work and other inefficiencies

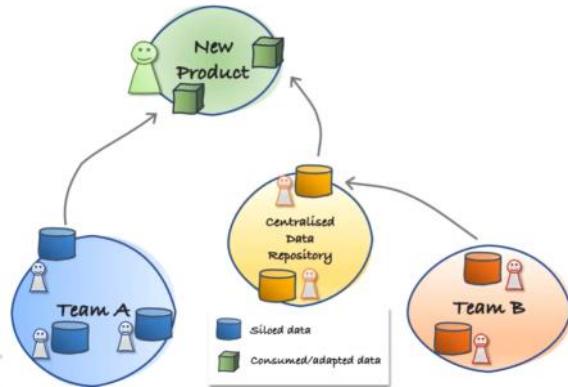
③ Limited conventions, standards and norms shared across teams



Context

◎ Common questions

- How do you find the data you need?
- How can you understand and trust the data?
- How do you make data available for new products?
- How can you allow the business to innovate at scale?



What Data Goes Inside the Data Mesh?

- ◎ What data? **All data!**
 - "Data Neutrality"
- ◎ Data is neither **analytical** nor **operational**, use cases are
 - Analytical and operational planes describe systems (people + software), not the data itself
 - Not a useful criterion to decide the scope of the Data Mesh

What Data Goes Inside the Data Mesh?

- ◎ Key ideas
 - **Data on the inside vs Data on the outside**
(Pat Helland, 2005)
 - **Fundamental data sources**

What Data Goes Inside the Data Mesh?



◎ Data on the inside vs Data on the outside

- *Data internal to domains vs Data shared with other domains*
- We borrowed and adapted Pat Helland's concept
- Identify data of common interest ⇒ Data on the Outside
- Data in the Data Mesh needs to be discoverable, consumable, managed, etc. It's hard work!
- Allows to maintain teams autonomy by clarifying the boundaries of the data domains and the interface between them

Refs:

- Pat Helland - 2005 <http://cidrdb.org/cidr2005/papers/P12.pdf>
- Adrian Colyer, 2016 <https://blog.acolyer.org/2016/09/13/data-on-the-outside-versus-data-on-the-inside/>
- Pat Helland - 2020 <https://queue.acm.org/detail.cfm?id=3415014>

What Data Goes Inside the Data Mesh?



◎ Find fundamental data sources

- Fundamental = **authoritative**, and **primary**
- Belongs to a central business domain
- With clear ownership
- e.g. tradable prices, trades (vs P&L)

Our Approach



◎ What we do

- Data-centric and heuristic approach - not "ivory tower"
- Emerged from attempting to map all data flows within the business
- We are building on top of a strong platform baseline

◎ What we don't do

- We are not starting from technology
- We are not consolidating or linking up existing data silos
- We are not building point-to-point data integrations

Cloud Infrastructure



◎ Cloud enablement and shared services

- Collaboration, experience and skills

◎ Data Infrastructure

- General infrastructure patterns and solutions
- Common way of building
- Reusable components
- Self-serve and real life

◎ Tools we use: AWS, Terraform, GitHub Actions, containers

◎ Hybrid platform model (cloud and on-prem)

Data Discovery

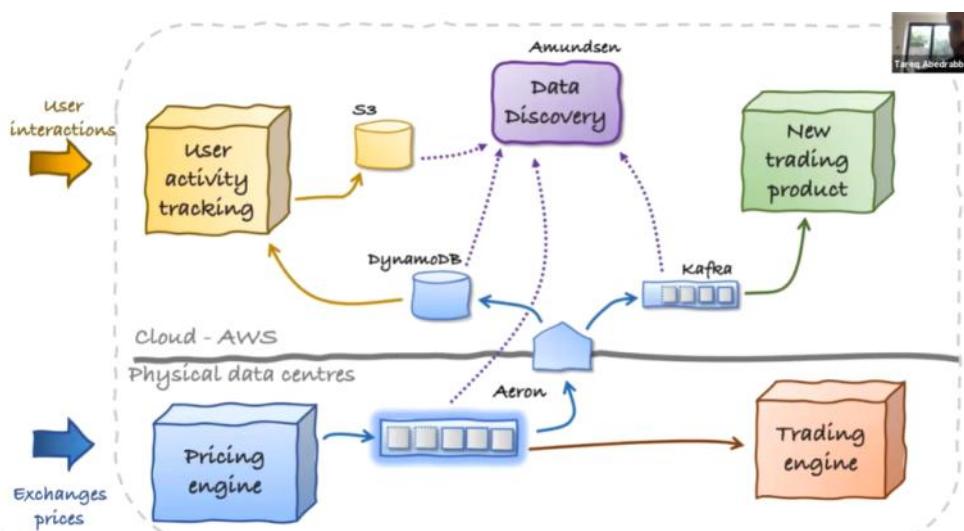
- ◎ Essential capability to “see through” the Data Mesh
- ◎ For data consumers
 - Starting point to find the data they need
 - Enhances self-service
 - Improves the interaction between data consumers and data owners
- ◎ For data owners
 - Clear entry point to onboard new data sources
 - Centralises metadata, while data remains decentralised
 - Data Discovery is a subset of Metadata Management
(data governance, protection, etc...)
- ◎ Not just a data catalogue

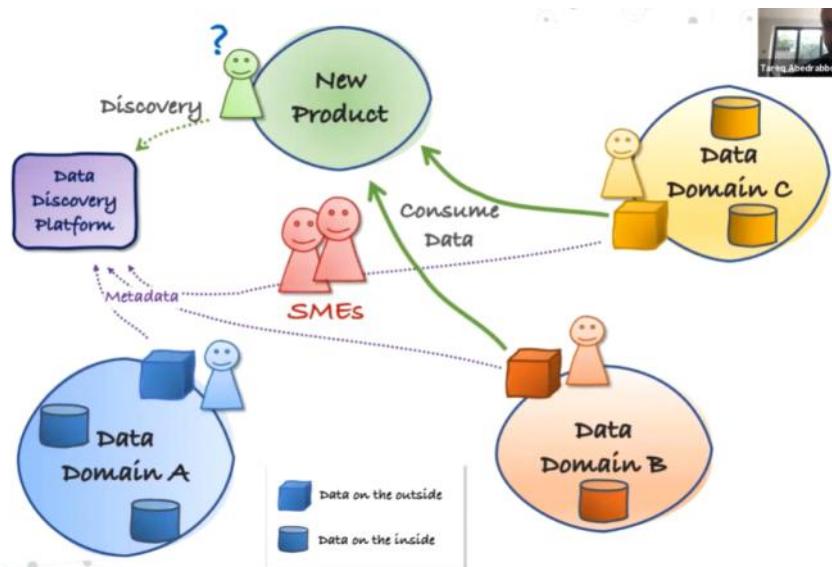
Data Discovery

- ◎ We are trialling Amundsen (amundsen.io)
 - Still emerging but has a great community and traction
 - Open source: extendable and customisable
 - Simple architecture, cloud-friendly
 - Backed by a graph database
 - Deployed on our cloud infrastructure

Data Natively Accessible

- ◎ Goal: data natively accessible on the cloud and on-prem
 - New products are built on the cloud
 - Many fundamental data sources are on-prem
- Example
 - ◎ Make **tradable prices** natively available on the cloud
 - **Low-latency data stream** generated from CMC based on market prices
 - Gap: available on-prem through low-latency messaging but not on cloud
 - ◎ Solution: Aeron-based pricing bridge
 - Low-latency and reliable messaging on-premise and on the cloud
 - High fidelity: preserving semantics to enable a variety of use cases
 - ◎ Multiple views, based on the same data, to serve different types of use cases
 - Latest prices, Timeseries, Event log (streaming)...





Data Mesh + Transformation

- ◎ Data Mesh needs a broad scope to be effective
 - Not just a change of technology
 - Multiple parts of the business are involved
 - Shift in mindset
- ◎ Data Mesh & Transformation must share the same ethos
 - Decentralised
 - Collaborative
 - Allows scale and autonomy
- ◎ DDD foundation
 - ⇒ business needs to reorganise around *business domains*

Challenges

- ◎ Data Mesh is a longer term investment
- ◎ Buy-in and commitment from the broader business
- ◎ Business priorities can shift quickly
- ◎ Established structures and ways of working
- ◎ Change generates friction
- ◎ Hybrid technology landscape

Enablers

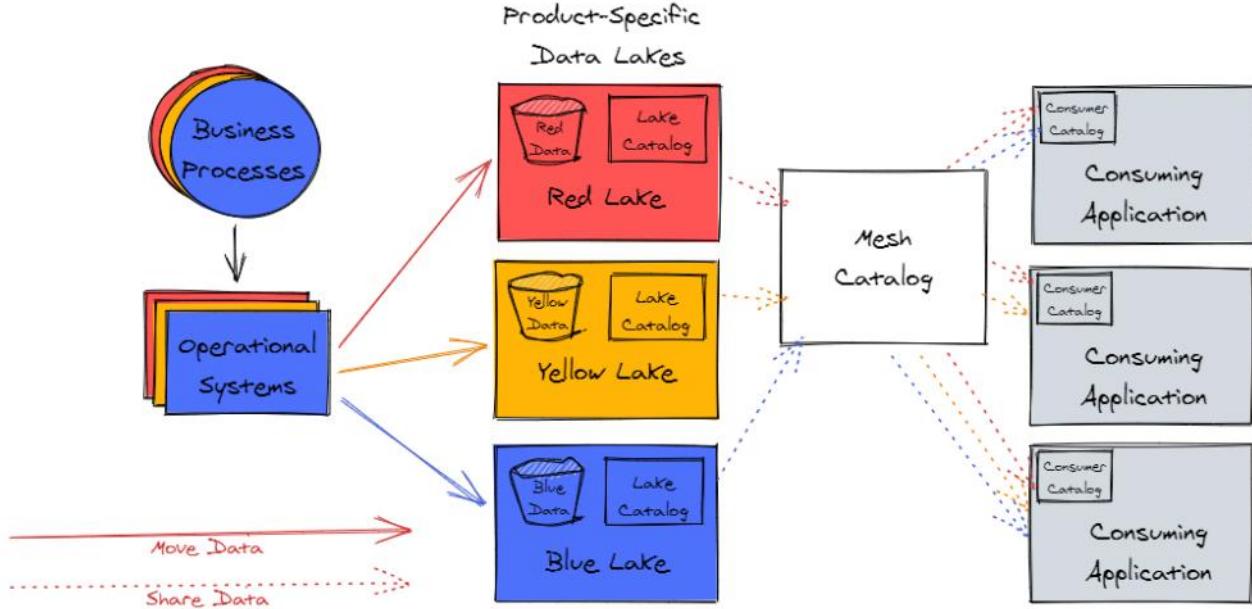
- ◎ Transformation programme
- ◎ Cloud adoption and self-service
- ◎ Squad delivery model
- ◎ Small expert team: can think, do and respond to change quickly and autonomously
- ◎ Attack the problem from different angles:
 - Find manageable use cases to validate the approach and create traction
 - Build new capabilities e.g. Data Discovery
 - Make fundamental data sources consumable
 - Share and educate

Future

- ◎ Gradual growing of the Data Mesh
- ◎ Onboarding more fundamental data sources
- ◎ Data discovery adoption
- ◎ The cloud becomes a viable environment for more data use cases

JP Morgan Chase bank

The data product-specific lakes that hold data, and the application domains that consume lake data, are interconnected to form the data mesh. A data mesh is a network of distributed data nodes linked together to ensure that data is secure, highly available, and easily discoverable. The following diagram illustrates this architecture.

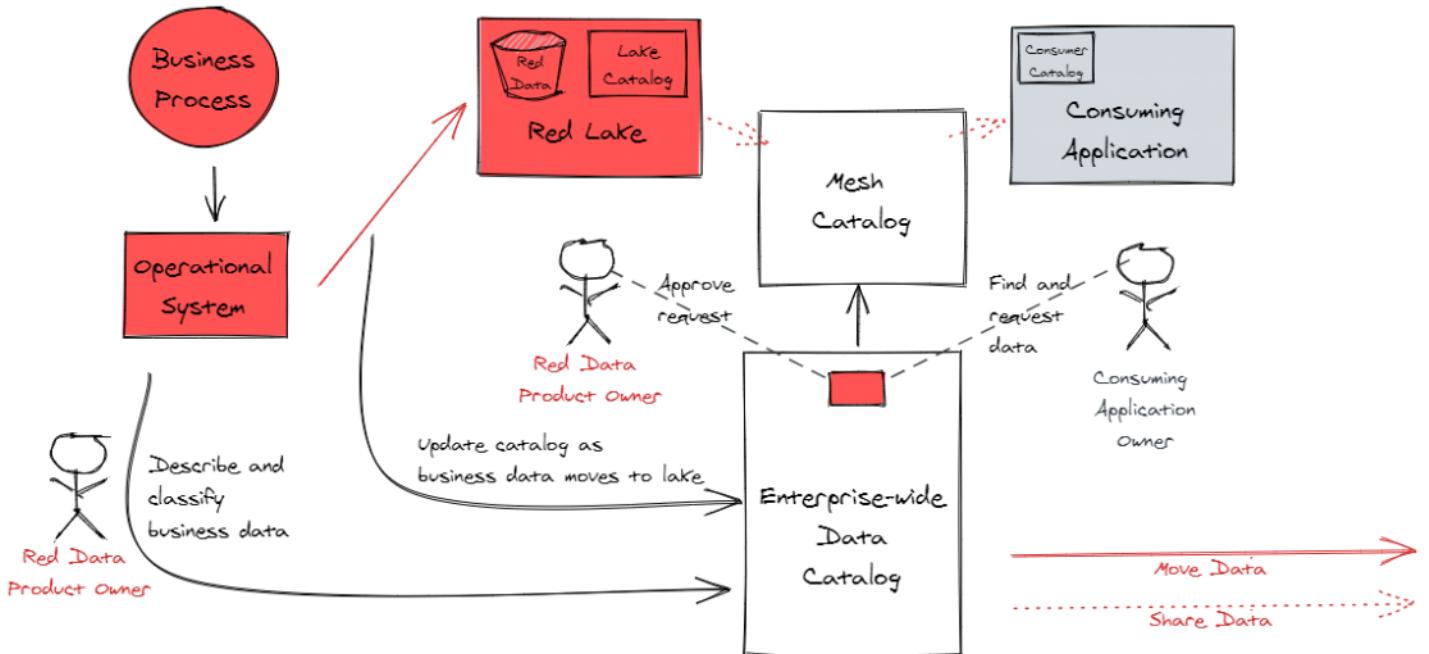


Empower the right people to make control decisions

Our data mesh architecture allows each data product lake to be managed by a team of data product owners who understand the data in their domain, and who can make risk-based decisions regarding the management of their data.

When a consumer application needs data from a product lake, the team that owns the consumer application locates the data they need in our enterprise-wide data catalog (see the following diagram). The entries in the catalog are maintained by the processes that move data to the lakes, so the catalog always reflects what data is currently in the lakes.

The catalog allows the consumption team to discover and request the data. Because each lake is curated by a team that understands the data in their domain and can help facilitate rapid, authoritative decisions by the right decision-makers, the consumption team's wait time is minimized.



How JPMorgan Chase built a data mesh architecture to drive significant value to enhance their enterprise data platform

Data doesn't diminish in value as it's used —the more value you can extract via enterprise-wide visibility, real-time analytics, and more accurate AI and machine learning (ML) predictions.

Organizations that are good at sharing data internally as legally permissible can realize more value from their data resources than organizations that aren't.

But like any resource, data risks must be managed, particularly in regulated industries. Controls help to mitigate such risks, so organizations that have strong controls around their data are exposed to less risk than those that don't.

This presents a paradox: data that is permitted to be freely shareable across the enterprise has the potential to add tremendous value for stakeholders, but the more freely shareable the data is, the greater the possible risk to the organization. To unlock the value of our data, we must solve this paradox. We must make data easy to share across the organization, while maintaining appropriate control over it.

JPMorgan Chase Bank, N.A. (JPMC) is taking a two-pronged approach to addressing this paradox. Firstly, by defining data products, which are curated by people who understand the data and its management requirements, permissible uses, and limitations. And secondly, by implementing a data mesh architecture, which allows us to align our data technology to those data products.

This combined approach achieves the following:

- Empowers data product owners to make management and use decisions for their data
- Enforces those decisions by sharing data, rather than copying it
- Provides clear visibility of where data is being shared across the enterprise

Let's first look at what the data mesh is, and then at how the data mesh architecture supports our data product strategy, and how both enable our businesses.

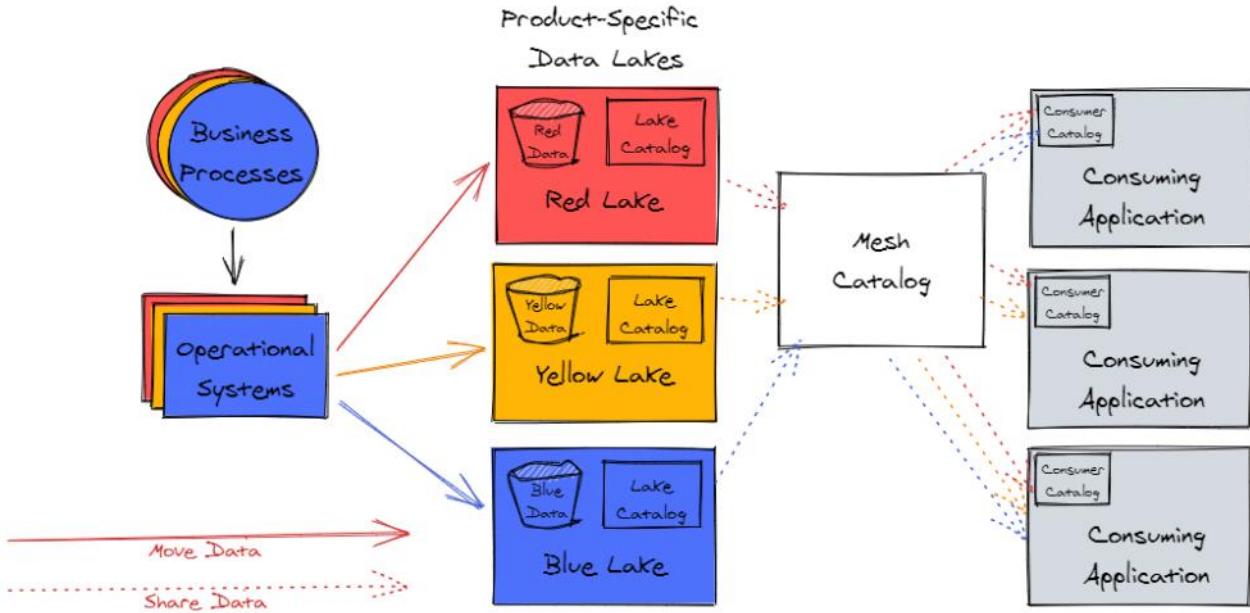
Aligning our data architecture to our data product strategy

JPMC is comprised of multiple lines of business (LOBs) and corporate functions (CFs) that span the organization. To enable data consumers across JPMC's LOBs and CFs to more easily find and obtain the data they need, while providing the necessary control around the use of that data, we're adopting a data product strategy.

Data products are groups of related data from the systems that support our business operations. They are broad but cohesive collections of related data. We store the data for each data product in its own product-specific data lake, and provide physical separation between each data product lake. Each lake has its own cloud-based storage layer, and we catalog and schematize the data in each lake using cloud services. One can use cloud-based storage and data integration services such as [Amazon Simple Storage Service](#) (Amazon S3) and data integration services like [AWS Glue](#) to provide these capabilities.

The services that consume data are hosted in consumer application domains. These consumer applications are physically separated from each other and from the data lakes. When a data consumer needs data from one or more of the data lakes, we use cloud services to make the lake data visible to the data consumers, and provide other cloud services to query the data directly from the lakes. One could use services such as the AWS Glue Data Catalog to make data visible, [AWS Lake Formation](#) to securely share data, and [Amazon Athena](#) to interactively query the data.

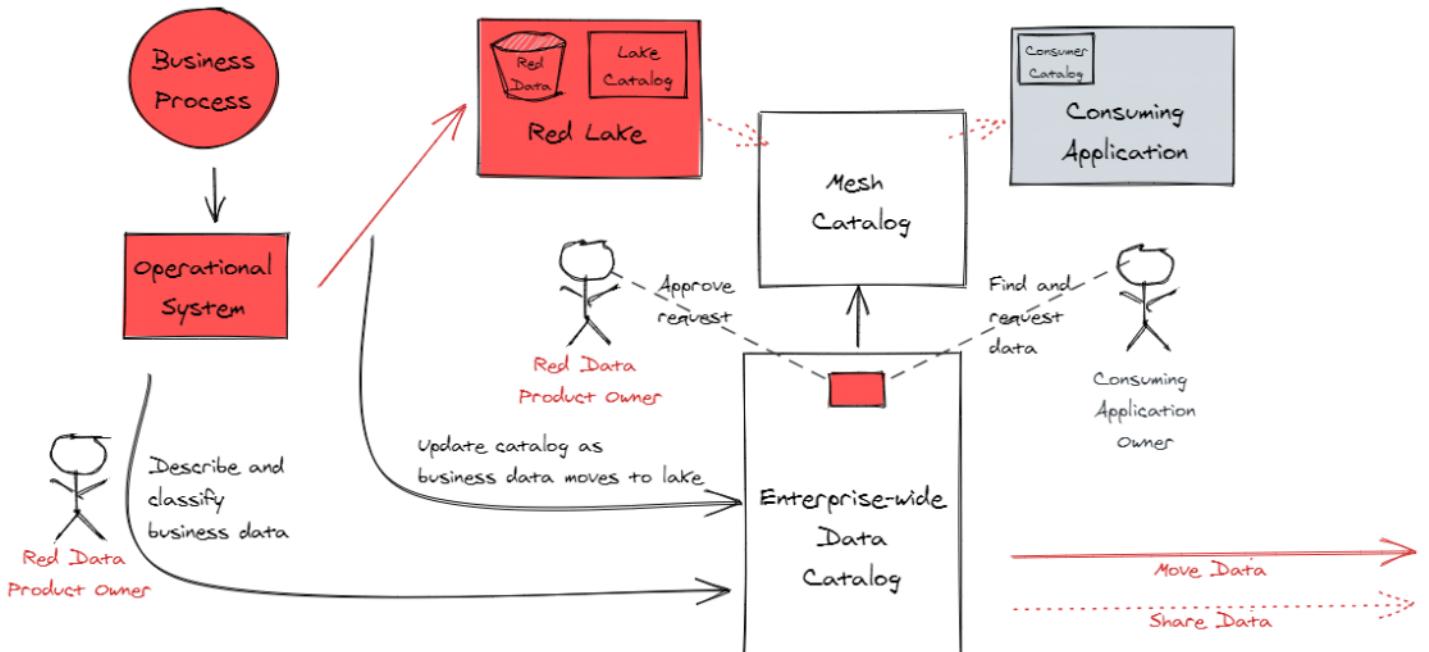
The data product-specific lakes that hold data, and the application domains that consume lake data, are interconnected to form the data mesh. A data mesh is a network of distributed data nodes linked together to ensure that data is secure, highly available, and easily discoverable. The following diagram illustrates this architecture.



Empower the right people to make control decisions

Our data mesh architecture allows each data product lake to be managed by a team of data product owners who understand the data in their domain, and who can make risk-based decisions regarding the management of their data.

When a consumer application needs data from a product lake, the team that owns the consumer application locates the data they need in our enterprise-wide data catalog (see the following diagram). The entries in the catalog are maintained by the processes that move data to the lakes, so the catalog always reflects what data is currently in the lakes. The catalog allows the consumption team to discover and request the data. Because each lake is curated by a team that understands the data in their domain and can help facilitate rapid, authoritative decisions by the right decision-makers, the consumption team's wait time is minimized.

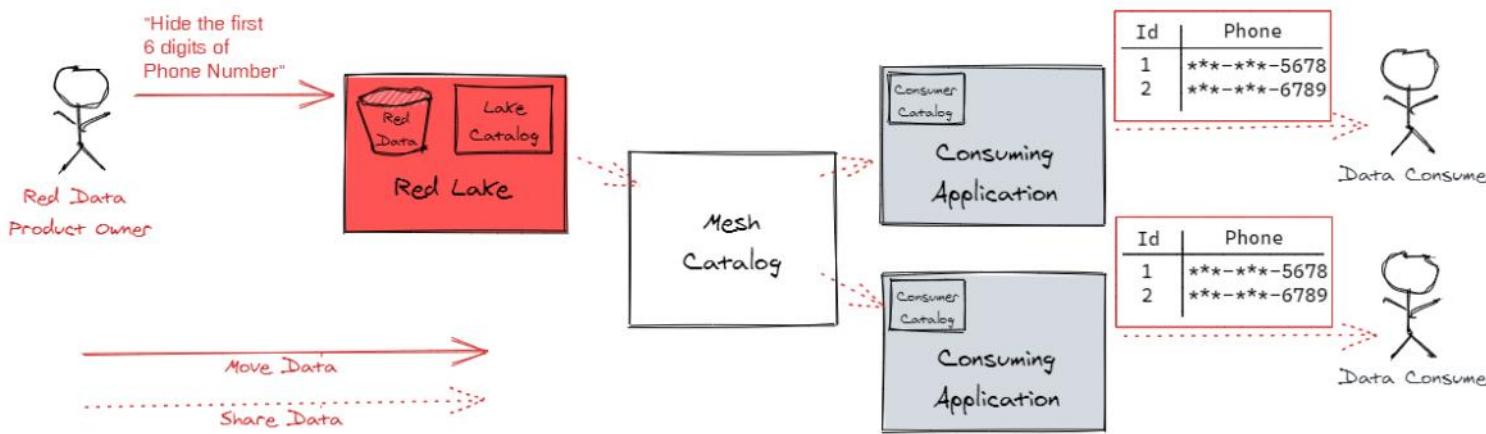
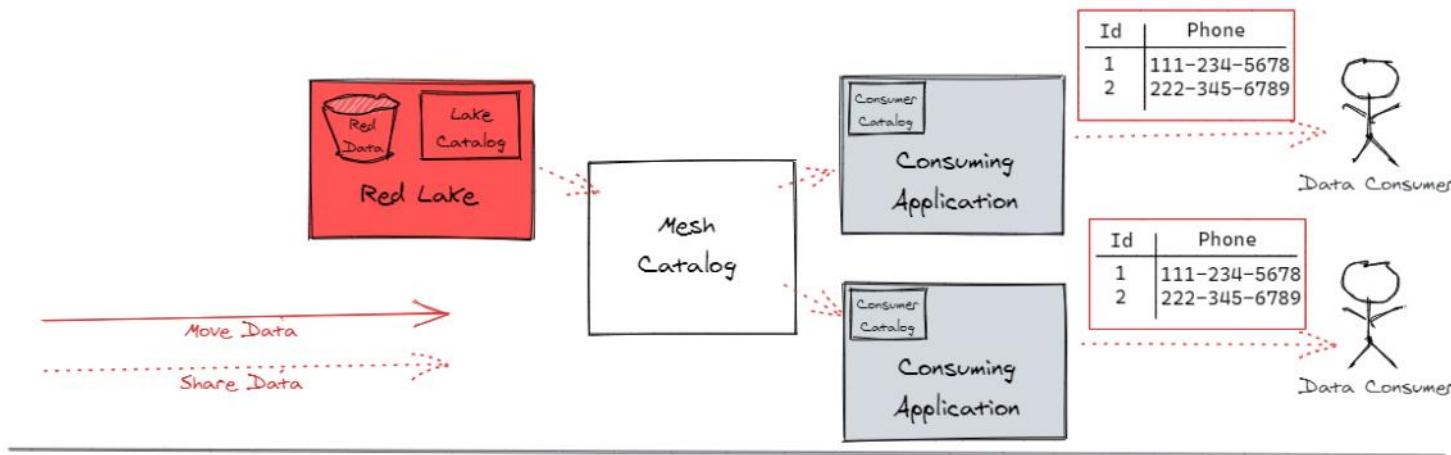


Enforce control decisions through in-place consumption

The data mesh allows us to share data from the product lakes, rather than copying it to the consumer applications that use it. In addition to keeping the storage bill down, sharing minimizes discrepancies in the data between the system that produced the data and the system that consumes it. That helps ensure that the data being consumed for analytics, AI/ML, and reporting is up to date and accurate.

Additionally, because the data doesn't physically leave the lake, it's easier to enforce the decisions that the data product owners make about their data. For example, if the data product owners decide to tokenize certain types of data in their lake, data consumers can only access the tokenized values. There are no copies of the untokenized data outside of the lake to create a control gap.

In-place consumption requires more sophisticated access control mechanisms than those needed to control access to copied data, however. When data is consumed in place, we need to restrict visibility at a very granular level—to specific columns, records, and even to individual values (see the following diagram). For example, when a system from one of our LOBs queries a pool of firm-wide reference data shared through a lake, that system may only be granted permission to access the reference data that pertains to that line of business.

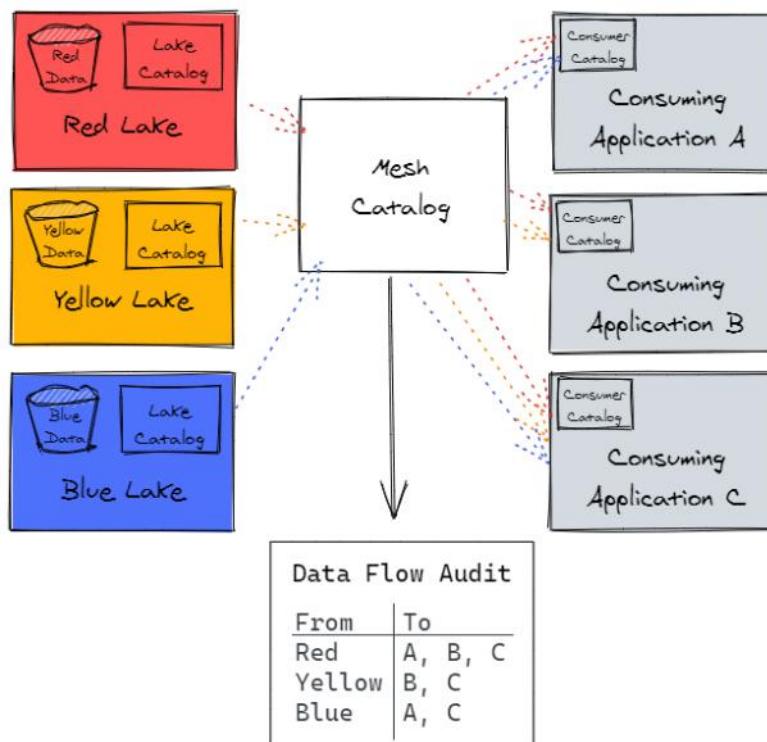


Provide cross-enterprise visibility of data consumption

Historically, data exchanges between systems were either system-to-system or via message queues. Because we didn't have a central, automated repository of all data flows, data product owners couldn't easily see when their data was flowing between systems.

Our data mesh architecture addresses the visibility challenge by using a cloud-based mesh catalog to facilitate data visibility between the lakes and the data consumers. One could use AWS Glue Data Catalog or a similar cloud-based data cataloging service to enable this.

This catalog doesn't hold any data, but it does have visibility of which lakes are sharing data with which data consumers. This offers a single point of visibility into the data flows across the enterprise, and gives the data product owners confidence that they know where their data is being used (see the following diagram).



Data mesh in action

Here's an example to illustrate how the data mesh architecture enables our business.

In the past, teams producing firm-wide reports extracted and joined data from multiple systems in multiple data domains to produce reports.

Through the data mesh architecture, the data product owners for those data domains make their data available in lakes. The enterprise data catalog allows reporting teams to find and request the lake-based data to be made available in their reporting application. The mesh catalog allows auditing the data flows from the lakes to the reporting application, so it's clear where the data in the reports originates.

Conclusion

JPMC's data mesh architecture aligns our data technology solutions to our data product strategy. By providing a blueprint for instantiating data lakes that implements the mesh architecture in a standardized way using a defined set of cloud services, we enable data sharing across the enterprise while giving data owners the control and visibility they need to manage their data effectively.

+++++

Don't get distracted by the hype

If your ultimate goal is to drive "business value" from data, how does the data mesh concept help you? Focus on using the data mesh as a way to get to your end goal. There are two key concepts to consider:

The data mesh isn't the beginning

"the overall operational data corpus is supposed to be scattered. It got that way through optimization, not incompetence." The data is supposed to live in a distributed state. It will be on-premises, it will be in the cloud, it will be in multiple clouds. You need to ask: "Has the inventory taken of all the data needed? Is it known where it all lays?"

The data mesh isn't the end

Data mesh needs a data foundation to operate successfully. If your goal is to generate value from the data, how do you materialize the results of your data mesh? This is where data products come into play.

We know that value from data comes from its usage and its application. If every company is a "data company," its currency is the "data products".

+++++
+++++

Data Mesh Use Cases

In the data mesh implementation, the central IT still exists to build a self-service data platform, but it does not own the data. For instance, a marketing company with the central IT team responsible for delivering an enabling technology, is still responsible for overarching governance and security for connected systems but individual functional teams have responsibility for the data itself

IT and DevOps

Data mesh offers a modern development approach to data analytics and software teams. It reduces data latency by providing instant access to query data from proximate geographies without access limitations.

Sales and Marketing

The distributed data enables sales and marketing teams to curate a 360-degree perspective of consumer behaviors and profiles from various systems and platforms to create more targeted campaigns, increase lead scoring accuracy, and project customer lifetime values (CLV), churn, and other essential performance metrics.

AI and Machine Learning Training

Data mesh enables development and intelligence teams to create virtual data warehouses and data catalogs from different sources to feed machine learning (ML) and artificial intelligence (AI) models to help them learn, without having to consolidate data in a central location.

Loss Prevention

Data mesh implementation in the financial sector creates faster time-to-insight at lower operating costs and operational risks. Distributed data analytics compacts fraudulent behavior modeling to detect and prevent fraud in real-time. It allows international financial bodies to analyze data locally – within any particular country or region, to identify fraud threats without replicating data sets and transporting them to their central database.

Global Business

A decentralized data platform makes it easy to comply with worldwide data governance rules to provide global analytics across multiple regions with end-to-end data sovereignty and data residency compliance.

+++++
+++++

What problems does data mesh solve?

Data meshes provide a solution to the **shortcomings of data lakes** by allowing greater autonomy and flexibility for data owners, facilitating greater data experimentation and innovation while lessening the burden on data teams to field the needs of every data consumer through a single pipeline.

+++++

Why you are not ready to adopt data mesh

1. You are not operating at a scale where decentralization makes sense
2. You do not have a strong business-case for how adopting Data Mesh will deliver business value for individual business units
3. You treat Data Mesh as a technical solution with a fixed target rather than an operating model that continuously evolves over time
4. Your organizational culture does not empower bottom-up decision-making
5. You do not have clearly established roles & responsibilities and incentive structure for distributed data teams
6. You do not have a critical mass of data talent
7. Your data teams have low engineering maturity
8. You expect to find off-the-shelf software to help you adopt Data Mesh
9. You do not have buy-in to "shift-left" security, privacy, and compliance
10. You do not consider Data Governance to be a core activity to be prioritized against other activities in every data team's backlog

Conclusion

Data Mesh is not a silver bullet for all your data management problems. It can be challenging — maybe even unrealistic — to implement for many organizations.

Regardless, I predict that many large organizations looking to scale impact from data and analytics will eventually adopt some version of Data Mesh; however not every element of Data Mesh will be implemented, or at least not immediately. For example, building a reusable and self-serviceable Data Platform as a Product is already emerging as a best practice for cloud adoption. This requires implementing the platform using infrastructure-as-code and CI/CD with embedded and continuous controls, setting the foundations for Federated Computational Governance. Once organizations are confident that they can operate the Cloud Data Platform safely and at-scale, Decentralized Data Ownership can be adopted widely. This might in turn prompt adjustment over Data Domain boundaries and even the org chart to allow distributed data teams to effectively develop Data as Product.

Instead of trying to implement every element of Data Mesh, I would recommend first focusing on how you can empower your data teams to deliver value faster and more frequently for your customers, then work backwards to identify and adopt specific elements of Data Mesh that will help you to achieve this goal.

+++++

Source oriented domain data

The *source domain datasets* represent the *facts and reality of the business*. They capture the data that is mapped closely to what the operational systems of their origin, *systems of reality*, generate. At enterprise scale there is never a one to one mapping between a domain concept and a source system. There are often many systems that can serve parts of the data that belongs to a domain, some legacy and some easy to change. Hence there might be many *source aligned datasets* aka *reality datasets* that ultimately need to be aggregated to a cohesive domain aligned dataset.

source data domains should also provide easily consumable historical snapshots of the source domain datasets. Source domain datasets are the most foundational datasets and change less often, as the facts of business don't change that frequently. These domain datasets are expected to be permanently captured and made available, so that as the organization evolves its *data-driven* and *intelligence* services they can always go back to the business facts, and create new aggregations or projections.

The source domain datasets represent closely the *raw data* and are not fitted or modeled for a particular consumer.

Consumer oriented and shared domain data

The consumer domain datasets and the teams who own them, aim to satisfy a closely related group of use cases. For example the 'social recommendation domain' that focuses on providing recommendations based on users social connections to each other. They structurally go through more changes, and they transform the source domain events to aggregate views and structures that fit a particular access model. A domain oriented data platform should be able to easily regenerate these consumer datasets from the source.

Distributed pipelines as domain internal implementation

While the datasets ownership is delegated from the central platform to the domains, the need for cleansing, preparing, aggregating and serving data remains, so does the usage of data pipeline. In this architecture, a data pipeline is simply an internal complexity and implementation handled internally within the domain. Each domain dataset must establish a *Service Level Objectives* for the quality of the data it provides: timeliness, error rates, etc.

Data and product thinking convergence

To address the concerns such as, accessibility, usability and harmonization of distributed datasets applying *product thinking* and ownership of data assets is useful.

Domain data as a product

Over the last decade operational domains have built *product thinking* into the capabilities they provide to the rest of the organization. Domain teams provide these capabilities as APIs to the rest of the developers in the organization, as building blocks of creating higher order value and functionality. The teams strive for creating the best developer experience for their domain APIs; including discoverable and understandable API documentation, API test sandboxes, and closely tracked quality and adoption KPIs.

For a distributed data platform to be successful, domain data teams must apply product thinking with similar rigor to the datasets that they provide; considering their data assets as their products and the rest of the organization's data scientists, ML and data engineers as their customers.

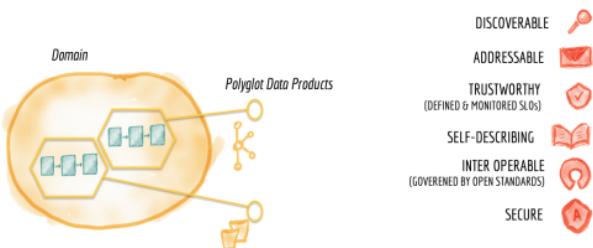


Figure 9: Characteristics of domain datasets as product

Discoverable

A data product must be easily discoverable. A common implementation is to have a registry, a data catalogue, of all available data products with their meta information such as their owners, source of origin, lineage, sample datasets, etc. This centralized discoverability service allows data consumers, engineers and scientists in an organization, to find a dataset of their interest easily. Each domain data product must register itself with this centralized data catalogue for easy discoverability.

Addressable

A data product should have a unique address following a global convention that helps its users to programmatically access it. Different domains might store and serve their datasets in different formats. A standard for addressability of datasets in a polyglot environment removes friction when finding and accessing information.

Trustworthy and truthful

A fundamental shift requires the owners of the data products to provide an acceptable *Service Level Objective* around the truthfulness of the data, and how closely it reflects the reality of the events that have occurred or the high probability of the truthfulness of the insights that have been generated. Applying data cleansing and automated data integrity testing at the point of creation of the data product are some of the techniques to be utilized to provide an acceptable level of quality. Providing *data provenance* and *data lineage* as the metadata associated with each data product helps consumers gain further confidence in the data product and its suitability for their particular needs.

Self-describing semantics and syntax

Quality products require no consumer hand holding to be used: they can be independently discovered, understood and consumed. Provide well described semantics and syntax of the data. Data schemas are a starting point to provide self-serve data assets.

Inter-operable and governed by global standards

To correlate data across domains and stitch them together in wonderful, insightful ways; join, filter, aggregate, etc. is challenging. The key for an effective correlation of data across domains is following certain standards and harmonization rules. Such standardizations should belong to a global governance, to enable interoperability between polyglot domain datasets. Common concerns of such standardization efforts are field type formatting, identifying *polysemes* across different domains, datasets address conventions, common metadata fields, event formats such as *CloudEvents*, etc.

Interoperability and standardization of communications, governed globally, is one of the foundational pillars for building distributed systems.

Secure and governed by a global access control

Accessing product datasets securely is a must, whether the architecture is centralized or not. In the decentralized domain oriented data products, the access control is applied at a finer granularity, for each domain data product. Define policies centrally but apply at the time of access to each individual dataset product. Using the *Enterprise Identity Management system (SSO)* and *Role Based Access Control* policy definition is a convenient way to implement product datasets access control.

Domain data cross-functional teams

Domains that provide data as products; need to be augmented with new skill sets: (a) the *data product owner* and (b) *data engineers*. Data product owners must define success criteria and business-aligned Key Performance Indicators (KPIs) for their data products. In order to build and operate the internal data pipelines of the domains, teams must include data engineers. A wonderful side effect of such cross-functional team is cross pollination of different skills. Removing the skillset silos will lead to creation of a larger and deeper pool of data engineering skill sets available to the organization.

Data must be treated a foundational piece of any software ecosystem, hence software engineers and *software generalists* must add the experience and knowledge of data product development to their tool belt. Similarly infrastructure engineers need to add knowledge and experience of managing a data infrastructure.



Figure 10: Cross functional domain data teams with explicit data product ownership

Data and self-serve platform design convergence

One of the main concerns of distributing the ownership of data to the domains is the duplicated effort and skills required to operate the data pipelines technology stack and infrastructure in each domain. Luckily, building common infrastructure as a platform is a well understood and solved problem; though admittedly the tooling and techniques are not as mature in the data ecosystem. Harvesting and extracting domain agnostic infrastructure capabilities into a data infrastructure platform solves the need for duplicating the effort of setting up data pipeline engines, storage, and streaming infrastructure.

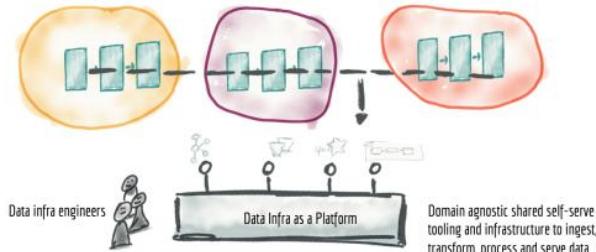


Figure 11: Extracting and harvesting domain agnostic data pipeline infrastructure and tooling into a separate data infrastructure as a platform

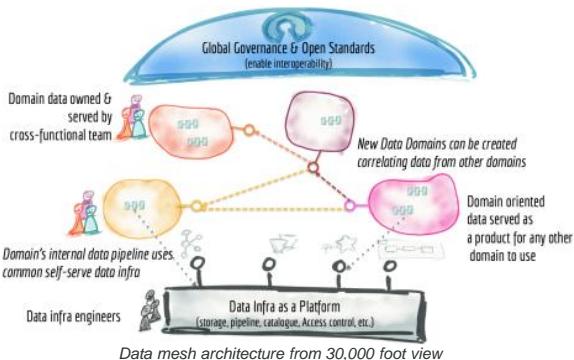
The key to building the **data infrastructure as a platform** is (a) to not include any domain specific concepts or business logic, keeping it domain agnostic; (b) make sure the platform hides all the underlying complexity and provides the data infrastructure components in a self-service manner.

A success criteria for self-serve data infrastructure is lowering the 'lead time to create a new data product' on the infrastructure requires automation, necessary for implementing the capabilities of a 'data product' as covered in section [Domain data as a product](#).

The paradigm shift towards a data mesh

The building blocks of a ubiquitous **data mesh** as a platform; *distributed data products* oriented around domains and owned by *independent cross-functional teams* who have embedded data engineers and data product owners, using common *data infrastructure* as a platform to host, prep and serve their data assets.

The data mesh platform is an intentionally designed **distributed data architecture, under centralized governance and standardization** for interoperability, enabled by a shared and harmonized self-serve data infrastructure.



The [data lake](#) or data warehouse are simply nodes on the mesh. In cases where we do need to make changes to the original format of the data for further exploration, such as labeling, the domain with such need might create its own lake or data hub. The data lake is no longer the centerpiece of the overall architecture. The data lake tooling will only be used for internal implementation of data products or as part of the shared data infrastructure.

The main shift is to treat *domain data product* as a first class concern, and data lake tooling and pipeline as implementation details. The data warehouse for business reporting and visualization is a node on the mesh.

Data catalog platforms such as [Google Cloud Data Catalog](#) provide central discoverability, access control and governance of distributed domain datasets. A wide variety of [cloud data storage](#) options enables domain data products to choose fit for purpose polyglot storage.

+++++
+++++

To decentralize the monolithic data platform, we need to reverse how we think about data. Instead of *flowing* the data from domains into a centrally owned data lake or platform, domains need to *host and serve* their domain datasets in an easily consumable way.

The physical location where the datasets actually reside and how they flow, is a technical implementation; the physical storage could certainly be a centralized infrastructure, but datasets content and ownership remains with the domain generating them.

This requires shifting our thinking from a *push and ingest* to *serving and pull* model across all domains. The *architectural quantum* in a domain oriented data platform, is a *domain* and not the pipeline stage.

From <<https://martinfowler.com/articles/data-monolith-to-mesh.html#DomainOrientedDataDecompositionAndOwnership>>

The purpose of this meeting is to discuss the Data Mesh – the latest concept in data warehousing. We'd like to discuss the following:

1. What it is?
 2. How is different from the existing data warehousing concept and architectures?
 3. Who are its competitors?
 4. What would be the use cases for its usage?
 5. What are its nuances and complexities?
 6. Any challenges with its realization, e.g., concepts, processes, tools, technologies, training, barrier to entry, etc.
 7. What competitive advantages will it offer to the client? Any value adds?
-

[Data Mesh Architecture \(datamesh-architecture.com\)](http://datamesh-architecture.com)

[\(1\) Building a successful Data Mesh – More than just a technology initiative | LinkedIn](#)

Data Modernization: [D1_Data-modernization-and-the-cloud.pdf \(deloitte.com\)](#)

**Simply put, data modernization
means moving data from legacy
databases to modern databases.**

Meetings

Thursday, January 12, 2023 1:01 PM

Meeting with Dinesh 1/12/2023

- To have more forward looking conversations
- Where are the delivery, execution and business value gap?
- Draw some thoughts - create a discussion documents
- Context, gap, size of gap, business impact
- Business is not happy with the CareData team
- CareData missed some deadlines
- Think where else you could contribute to the growth of our business
- What relationships you can build
- How do you get the multiplier effect
- Reach out to some client contacts at Kaiser
- Get marketing help. Start getting recognized as a brand
- Businesses are not able to measure the value of their investment
- Health plans and Payviders is what we're targeting

Data Mesh

- Why is it important?
 - What is the business applicability
 - What is about?
 - How to be successful in that model
 - What should they do to be successful
-

Meeting 1/13/2023 with OAS, PTS

Subject: Data Mesh

what is the challenge with data?

What kind of volume is being handled?

Enables data modernization

Anantha (01/13/2023) - "Does data mesh architecture a method for modernization , and can that evolve into a cloud mesh (⌚ Getting fancy - Promise Friday evening is not the reason!!) ?.".

The point is multi cloud is a reality for large organization. Having said, the investment on data mesh architecture to modernize on-prem to cloud, be repurposed to integrate multi-cloud ? (cloud mesh)..?

Dinesh (01/14/2023) -

Love it - keep the creative ideas flowing. Would love us to question the current philosophy being promoted regarding Data Mesh and identify additional value hypothesis that would accelerate business value capture for the companies from their data assets.

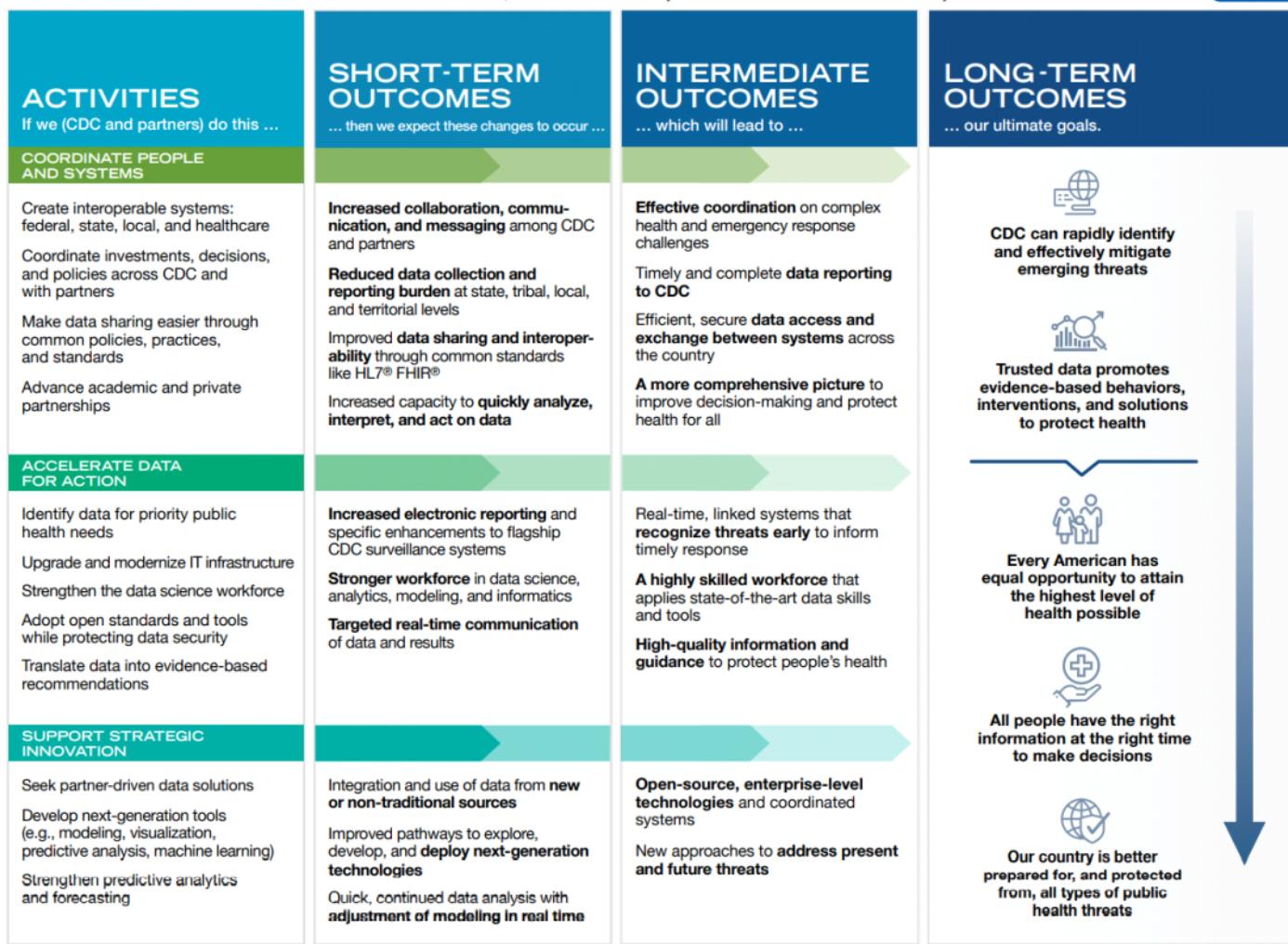
My understanding about Data Mesh:

- A Data Architecture paradigm shift
- Core objective is to improve the speed to insights
- Product centric view of data as an asset
- Domain-focused
- Self-service enabled
- Fit for purpose governance

Last year participated in Blues conference and IBM presented this topic - and their position too was architecture paradigm focused and 100% technology agnostic to improve the speed of data product development to accelerate business value capture

Haven't come across any technologies positioned as an enabler or accelerator for DM yet - not to say that SW vendors won't capitalize on it

CDC Data Modernization Initiative | A Roadmap of Activities and Expected Outcomes



318212 /

A Five-Point Data Modernization Roadmap to Consider

Spending trends make for headline-grabbing news as digital transformation efforts continue to grow. Gartner says [worldwide IT spending will grow](#) 3 percent this year and [forecasts cloud spending](#) to reach nearly \$500 billion. There's a larger objective behind the individual tech shifts: meeting customer demands. If customer needs hadn't evolved to require faster applications or to make data available within milliseconds, few companies would be making such large investments. Shoppers require a seamless omnichannel experience, travelers need to be able to see whether their flights have been delayed or canceled, and investors must know how their assets are performing at all times. Companies need a modern data infrastructure to make all those functions possible.

For many companies, there is pressure to create new revenue streams, which is why creating digital applications and services is a priority. Established companies trying to catch up to startups are the ones that can benefit most from data modernization to make those new apps and services come to life.

Over the past two years, companies have vastly accelerated their digital transformation efforts, but success hasn't been as widespread. According to [research from McKinsey](#), top performers report capturing an average of 50 percent of the full revenue benefits their transformations could have achieved, compared to a 31 percent average across all respondents in McKinsey's survey.

To get the best return on investment for data modernization, you'll want to clearly identify where you want to create new business value. Creating this blueprint is the most crucial step to digital business success, according to IDG's [2021 Digital Business Study](#). Here's a checklist of five components of data modernization, why they matter, and how you can benefit from improving them.

Aiding the Creation of New Digital Products

Service level concerns like availability and security are at the core of any organization's [data management ecosystem](#), but the stakes have been raised to also account for the new applications and services customers want. To satisfy that demand, you need to offer an experience with low latency, fast application response times, and near-real-time access to fresh data.

Financial services organizations are ahead in this game, [according to Deloitte](#), with 89 percent of respondents in its survey has started, if not fully implemented, a data modernization campaign. Customers don't expect to have to go to a bank anymore to deposit a check. They'd rather make the transaction through an app and see it immediately reflected in their accounts.

Data Responsiveness

For an application to provide a positive user experience, it must always be available. The only way an app can be accessible 24/7 is if its data is accessible 24/7. Some systems of record may not always be available, but you still need a way to have always-on access to the data so the apps can always be active.

If apps are unresponsive or don't have the most updated information available, you risk losing customers. In fact, 57 percent of respondents in FullStory's [Digital Experience Survey](#) report leaving if a site or app loads slowly and 60 percent have rage-clicked in frustration on an underperforming site or app.

Legacy IT infrastructure, with siloed systems and many connections between data and systems of record, creates latency issues. By improving data responsiveness, you can reduce user friction and allow apps to load faster for increased performance. In the world of customer satisfaction, every millisecond matters.

Speed of Development

The ability to serve data in real-time doesn't just benefit customers; it also allows developers to work faster and accelerate the time to market for your applications. In many competitive sectors, quick time to market is one of the most crucial business priorities.

If you can create a holistic view of all your data and associated technologies, developers will have more streamlined processes to build applications and access the necessary data to develop services.

Trends like middleware, API providers, and microservice architecture are all rapidly changing what application development looks like. If you can offer developers seamless access to the data they need, you'll help them navigate an evolving landscape, which can shorten development cycles.

Migrating Data to Modern Infrastructure

Many established companies have most of their data stored in legacy systems like mainframes and are gradually moving this data to the cloud. According to Flexera's [State of the Cloud Report 2022](#) study, 63 percent of companies report heavy usage of public clouds.

Cloud migration can make you more agile and plays into data modernization, which is the second biggest driver for cloud migration, [according to Deloitte research](#). Only security and [data protection](#) ranked higher in motivations for adopting cloud technologies.

For many companies today, it's not so much about how to leverage the cloud, but rather how to properly migrate and modernize their infrastructure so they can quickly deploy digital applications and optimize operations.

Integrating Data

One of the biggest challenges to tackle as you modernize your data is creating an infrastructure that gives you a holistic view and a way of integrating data in the most optimal way.

Many companies are turning to digital integration hubs, an architecture that decouples applications from systems of record via a real-time replication of the data into a high-performance data store. With this low-latency data fabric that's always available, responsive, and accurate, you can reduce your tech stack complexity and make your applications and services run smoother.

With spending increasing across the IT landscape, companies are under acute pressure to make sure they get these consequential decisions right. Consider this checklist as a step in your evaluation of where your company should be investing. Data modernization is a journey that will require thoughtful business decisions and a precise strategy to ensure that you're capturing the most value, and satisfying your customers' demands.

+++++

Develop Data Strategy Roadmap in 5 Steps

Have you developed your data strategy but haven't made much progress on reaching your key business objectives? Are you struggling to prioritize data initiatives or convince stakeholders to buy in? A data strategy roadmap may be the missing link in your strategy, and in this blog, we'll discuss what it is and why you need one. By the end, you'll have a clear idea on how to effectively build a data strategy roadmap.

Creating a data strategy is a huge first step—don't let the time and effort you put into it go to waste. You need a documented and actionable plan that provides you with a clear path forward and the specific details for how you will actually execute your data strategy. Without a data strategy roadmap in place, it can be difficult to allocate the necessary resources, establish policies and processes, build or improve data infrastructure, train staff on data literacy or new platforms, and even gain critical buy-in to start any other initiative.

A data strategy roadmap will ensure the success of your data strategy.

What is a Data Strategy Roadmap and Why Do You Need One?

A data strategy roadmap is a plan that outlines the implementation process for how an organization will effectively manage, analyze, and utilize data to achieve its business goals. It includes objectives, necessary resources, and a timeline.

A data strategy roadmap helps organizations align data initiatives with business objectives, track progress, measure success, and allocate resources effectively. It also helps to ensure that all necessary people and departments are involved—and it serves as a tool for meeting milestones and staying organized. By following a data strategy roadmap, organizations can use data to drive business growth and make decisions more effectively.

When Should You Build a Data Strategy Roadmap?

A data strategy roadmap should be developed whenever an organization is working on its [data strategy](#)—whether it's at the start of a data initiative, as part of a broader business strategy, or at any other time when the organization is looking to optimize its use of data.

There are several situations where building a data strategy roadmap may be particularly beneficial. For example:

- **When stuck in a progress rut:** A data strategy roadmap can fix any prioritization and support issues and help you make progress toward more valuable and sustainable objectives.
- **When starting a new data initiative:** A data strategy roadmap can help you plan and prioritize data initiatives and ensure that they are aligned with business goals and have the necessary resources and support.
- **When launching a new product or service:** A data strategy roadmap can help you understand customer needs and use data to inform the development and marketing of the new offering.
- **When entering a new market:** A data strategy roadmap can help you gather and analyze market data to inform expansion plans and identify potential opportunities and challenges.
- **When looking to improve operational efficiency:** A data strategy roadmap can help you identify areas where data can optimize processes and improve efficiency.
- **When seeking to drive innovation:** A data strategy roadmap can help you identify trends and patterns in data that can inform new ideas.

When to Build a Data Strategy

Roadmap:

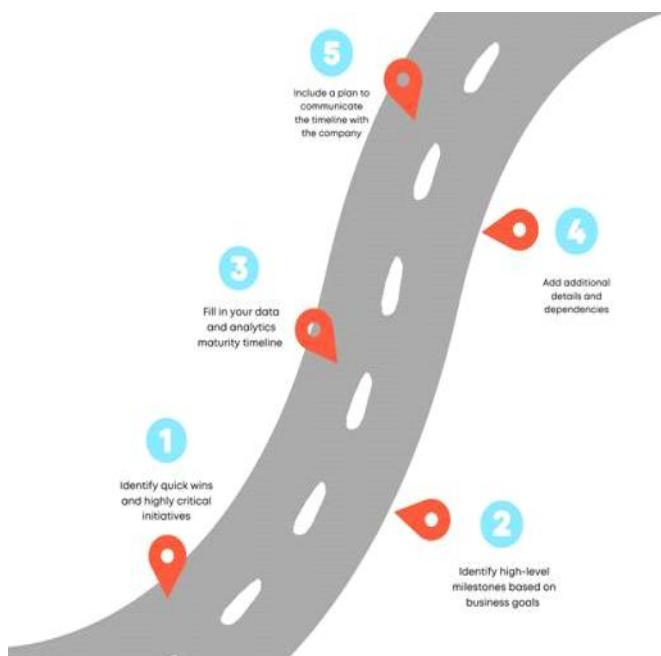


Build a data strategy roadmap whenever you are working on your data strategy—regardless of what initiates it.

Step 1: A [data strategy assessment](#) is a good starting point to building an effective data strategy roadmap.

The Five Steps to Building a Data Strategy Roadmap

To build a data strategy roadmap, identify quick wins and critical initiatives, set high-level milestones based on business goals, add initiatives and details to the timeline, and plan communication with the company to use data to drive business growth.



A data strategy roadmap should include quick wins and highly critical initiatives, high-level milestones, timeline with initiatives, additional details and dependencies, and a plan to communicate the timeline with the company.

Step 1: Identify quick wins and highly critical/urgent initiatives.

The first step in building a data strategy roadmap is to identify which data and analytics initiatives should be prioritized. This includes identifying low hanging fruit that can be easily implemented, as well as highly critical or urgent initiatives that must be addressed to meet business goals.

Determining priorities allows you to:

- Clarify the goals and objectives and enable the team to focus on the most important aspects of the data strategy roadmap as well as allocate resources accordingly.
- Identify potential risks and challenges and put contingency plans in place to mitigate those risks.
- Set realistic expectations and communicate them clearly to stakeholders.

- Align the roadmap with the overall business strategy and support the achievement of the business' goals.

Step 2: Identify high-level milestones based on business goals or expected changes to the business.

The next step is to identify high-level milestones based on business goals or expected changes to the business. These milestones will help to determine the required pace for completing data initiatives that will support business goals within the desired timeframe.

Ask questions such as:

- What are my company's big picture goals or the "big rocks" and what timeframe are those goals set at?
- What important company events or seasonal factors need to be on the timeline?
- Are there any known critical changes to the company that should be included on the timeline such as a new product launch, planned acquisition, or a new ERP system go-live date?
- What level will our data and analytics maturity need to be at each of these milestones on the timeline?

Step 3: Fill in the timeline with initiatives.

With the high-level milestones in place, the next step is to fill in the timeline with the initiatives that will be required to build out the data architecture and other requirements in iterations that will incrementally increase data and analytics maturity while also providing value. In the first iteration, ask questions such as:

- Do we need to set up new tools or technologies? If so, do we have an implementation partner or do we need to hire one?
- Will we need to create new data pipelines? Do we have the in-house talent to do this?
- Are our definitions consistent across the organization? Do we have a data governance program in place to address this issue going forward?

Remember to account for all the steps required for the first iteration, even if they seem minor. Skipping a small task early on may result in having to go back and reengage someone who now has other priorities. It is important to consider how each tool, technology, or process fits into the broader data architecture and to approach implementation in a phased manner.

Step 4: Add additional details and dependencies.

To ensure that the data strategy roadmap is actionable and achievable, it is important to add additional details and dependencies. This includes:

- Identifying the people and resources that will be required to complete the initiatives, including any external partners or vendors that may be involved.
- Identifying any dependencies or interdependencies that may impact the timeline, such as the completion of certain initiatives that are required for others to move forward.
- Creating a resource plan that outlines the required roles and responsibilities of each team member, as well as the estimated time and budget required for each initiative. This helps to ensure that everyone is clear on their roles and responsibilities and that the team has the necessary resources to complete the initiatives as planned.
- Considering any constraints or limitations that may impact the timeline or budget, such as regulatory requirements, data privacy considerations, or technical limitations. These factors may require additional planning and resources to address, so it is important to factor them into the roadmap.

Step 5: Include a plan to communicate the timeline with the company.

Once the data strategy roadmap has been developed, it is important to communicate it to the company to build buy-in and ensure that everyone is aligned and working toward the same goals. To effectively communicate:

- Clearly and concisely communicating the vision and goals of the data strategy roadmap, highlighting how it will support the achievement of key business goals and align with the overall business strategy.
- Sharing the timeline and milestones with the company and clearly explaining the purpose and significance of each initiative.
- Regularly engaging with stakeholders to gather feedback and input on the data strategy roadmap, ensuring that it is realistic and addresses the needs and concerns of all relevant parties.
- Providing updates on the progress of the roadmap, addressing any issues or challenges that arise, and adjusting as needed to reflect changing priorities or circumstances.
- Ensuring that communication is transparent and open, encouraging collaboration and buy-in from all relevant parties.

Review, Revise, and Refocus Your Data Strategy Roadmap

A data strategy roadmap is not a static document, and it is important to review and revise it regularly to ensure that it remains relevant and aligned with business objectives. This may involve revising priorities, updating resources, and adjusting timelines as needed. Remember to involve key stakeholders in the review process, as they can provide valuable insights and perspectives on the effectiveness of the data strategy roadmap. By staying agile and adaptable,

organizations can effectively use data to drive business growth and decision-making.

+++++

Data Mesh ChatGPT

Friday, March 3, 2023 11:38 AM

<https://chat.openai.com/chat>

Data Mesh is a relatively new concept in the field of data architecture that aims to solve some of the common problems faced by organizations when working with large amounts of data. It is a decentralized approach to data management that emphasizes the importance of domain-driven design, autonomous teams, and federated data governance. In a traditional centralized approach to data management, all data is owned and managed by a centralized data team, and data flows from source systems to a centralized data warehouse. This approach often leads to data silos, data inconsistencies, and bottlenecks in data access and delivery. Data Mesh, on the other hand, advocates for a decentralized approach where data is owned and managed by domain-specific teams, with clear boundaries and interfaces for data ownership and sharing.

The key principles of Data Mesh include:

1. Domain-Driven Design: Data is organized around business domains, rather than technical functions, to ensure that data is aligned with business needs.
2. Autonomous Teams: Each domain team has full ownership and accountability for the data within their domain, and they are responsible for the entire data lifecycle, from collection to consumption.
3. Federated Data Governance: A federated governance model allows for each domain team to manage their own data in a way that meets the overall organizational policies and standards.
4. Data Products: Data is treated as a product, with clear interfaces, documentation, and service level agreements (SLAs) to ensure that data is consumable by other teams in a self-service manner.

Overall, Data Mesh aims to create a more flexible, scalable, and sustainable approach to data management, with a focus on agility, autonomy, and alignment with business needs.

Use cases:

Some of the common use cases for Data Mesh include:

1. Large-scale Data Integration: Organizations that need to integrate large volumes of data from disparate sources can benefit from a Data Mesh approach. By organizing data around business domains and enabling domain-specific teams to manage their own data, organizations can improve data quality, reduce duplication, and increase the speed of data integration.
2. Self-Service Analytics: Data Mesh allows domain teams to manage their own data and make it available to other teams in a self-service manner. This can help organizations to democratize data access and empower teams to make data-driven decisions without relying on a centralized data team.
3. Data Monetization: Data Mesh can enable organizations to monetize their data by treating data as a product and making it available to external customers or partners. By creating clear interfaces, documentation, and SLAs for data products, organizations can create new revenue streams and unlock the value of their data.
4. Real-time Data Processing: Data Mesh can enable real-time data processing by allowing domain teams to manage their own data streams and processing pipelines. By providing a standardized interface for data streams, organizations can improve data quality, reduce latency, and increase the speed of data processing.
5. Compliance and Security: Data Mesh can help organizations to meet compliance and security requirements by providing a federated governance model that allows domain teams to manage their own data in a way that meets organizational policies and standards. This can help to reduce the risk of data breaches and ensure that data is managed in a secure and compliant manner.

Overall, Data Mesh can be a valuable approach for organizations that need to manage large amounts of data in a flexible, scalable, and sustainable way.

When do you need Data Mesh?

Data Mesh is a relatively new approach to managing data in large and complex organizations. It is designed to address the challenges of scaling data infrastructure and analytics in organizations where data is widely distributed across multiple business units, teams, and systems. Here are some scenarios where Data Mesh may be appropriate:

1. Large organizations with complex data landscapes: Data Mesh can be useful in organizations with large and complex data landscapes, where data is siloed across multiple teams and systems. It can help organizations to break down data silos and create a more efficient and effective data infrastructure.
2. Organizations with a need for data democratization: Data Mesh can be useful in organizations where data needs to be accessible and usable by a wide range of stakeholders across different business units and teams. It can help to democratize data by making it easier for stakeholders to find, access, and use data in their work.
3. Organizations with a need for agility: Data Mesh can be useful in organizations where agility is key, and where the ability to quickly adapt to changing business needs and priorities is critical. By decentralizing data ownership and management, Data Mesh can help to create a more agile data infrastructure that can respond quickly to changing requirements.
4. Organizations with a need for scalability: Data Mesh can be useful in organizations that are rapidly growing and need to scale their data infrastructure and analytics capabilities to keep up with demand. By breaking down data silos and creating a more efficient and effective data infrastructure, Data Mesh can help organizations to scale their data operations more easily.
5. Organizations with a need for better data quality and governance: Data Mesh can be useful in organizations where data quality and governance are critical, and where data needs to be managed and controlled carefully. By decentralizing data ownership and management, Data Mesh can help to improve data quality and governance by ensuring that data is managed by the people who understand it best.

What problem does Data Mesh Solve?

The journey towards implementing Data Mesh involves several steps, including:

1. Understand Business Needs: The first step towards implementing Data Mesh is to understand the specific business needs and challenges that the organization is facing. This includes identifying the domains within the organization, the data sources that exist within each domain, and the challenges associated with managing and sharing data across domains.
2. Identify Domain Teams: The next step is to identify the domain teams that will be responsible for managing the data within each domain. Domain teams should be composed of subject matter experts who have a deep understanding of the data within their domain.
3. Define Domain Interfaces: Once the domain teams have been identified, the organization should define the interfaces between domains. This involves defining the data standards and protocols that will be used to exchange data between domains.
4. Establish Federated Governance: Data Mesh requires a federated governance model in which each domain team is responsible for managing their own data. This requires the development of a governance framework that outlines the roles and responsibilities of each domain team and establishes guidelines for data quality, security, and privacy.
5. Implement Data Infrastructure: Once the governance framework has been established, the organization can begin implementing the necessary data infrastructure to support Data Mesh. This may include the development of data platforms, data pipelines, and data products that are managed by domain teams.
6. Monitor and Evaluate: Finally, the organization should continuously monitor and evaluate the effectiveness of the Data Mesh implementation. This includes tracking key performance metrics such as data quality, data sharing, and data monetization, and making adjustments to the implementation as needed.

Overall, the journey towards implementing Data Mesh requires a significant investment in people, processes, and technology. However, by adopting this approach, organizations can create a more flexible, scalable, and adaptable approach to data management that can better accommodate changing business needs and support innovation.

How is Data Mesh different from Data Lake?

Some key differences between Data Mesh and Data Lake include:

1. Ownership and Governance: In Data Mesh, ownership and governance of data is decentralized and distributed across domain teams. In Data Lake, ownership and governance is typically centralized within a single team.
2. Data Structure: In Data Mesh, data is organized around business domains and each domain team can use the data structures that make the most sense for their domain. In Data Lake, data is typically organized in a centralized schema or data model.
3. Data Processing: In Data Mesh, each domain team is responsible for processing and preparing their own data. In Data Lake, a centralized team is typically responsible for processing and preparing the data for analysis.

- Scalability: Data Mesh is designed to be more scalable than Data Lake, as it can accommodate rapid growth and changing business needs by enabling domain teams to manage their own data.

Overall, while both Data Mesh and Data Lake are designed to improve data management and analysis, they are fundamentally different approaches that are suited to different organizational needs and priorities.

How to be successful in Data Mesh?

To be successful in implementing the Data Mesh model, organizations need to follow several key principles and best practices:

- Define Business Domains: The first step in implementing Data Mesh is to define the business domains within the organization. These domains should be aligned with the business processes, goals, and objectives of the organization.
 - Establish Clear Ownership: Each domain should have clear ownership and accountability for the data within that domain. Domain teams should be composed of subject matter experts who have a deep understanding of the data within their domain.
 - Implement Federated Governance: Data Mesh requires a federated governance model in which each domain team is responsible for managing their own data. This requires the development of a governance framework that outlines the roles and responsibilities of each domain team and establishes guidelines for data quality, security, and privacy.
 - Encourage Collaboration: Data Mesh encourages collaboration and communication across different domain teams. Organizations should encourage the sharing of knowledge, best practices, and data across domains to facilitate innovation and better decision-making.
 - Implement Modern Data Infrastructure: To support Data Mesh, organizations should implement modern data infrastructure that enables domain teams to manage and process their own data. This may include data platforms, data pipelines, and data products that are managed by domain teams.
 - Monitor and Measure Performance: To ensure the success of Data Mesh, organizations should monitor and measure the performance of the implementation over time. This includes tracking key metrics such as data quality, data sharing, and data monetization, and making adjustments to the implementation as needed.
- Overall, success in implementing the Data Mesh model requires a cultural shift towards greater collaboration, decentralization, and empowerment of domain teams. By following these best practices and principles, organizations can create a more flexible, scalable, and adaptable approach to data management that can better accommodate changing business needs and support innovation.

Benefits:

- Improved Data Quality: Data Mesh encourages domain teams to take ownership of their own data, which can help to improve data quality. By ensuring that data is aligned with business needs and managed in a consistent and standardized manner, organizations can reduce data duplication and inconsistencies.
 - Faster Time to Value: Data Mesh enables domain teams to manage their own data, which can increase the speed of data access and delivery. This can help organizations to accelerate their time to value and respond more quickly to changing business needs.
 - Greater Agility: Data Mesh provides a more flexible and adaptable approach to data management. By organizing data around business domains and enabling domain teams to manage their own data, organizations can respond more quickly to changes in business requirements and adapt their data architecture accordingly.
 - Reduced Complexity: Data Mesh can help to reduce the complexity of data management by providing clear boundaries and interfaces for data ownership and sharing. This can reduce the risk of data silos and bottlenecks in data access and delivery.
 - Empowered Teams: Data Mesh empowers domain teams to take ownership of their own data and make it available to other teams in a self-service manner. This can help to democratize data access and empower teams to make data-driven decisions without relying on a centralized data team.
 - Data Monetization: Data Mesh can enable organizations to monetize their data by creating data products and making them available to external customers or partners. This can help organizations to create new revenue streams and unlock the value of their data.
- Overall, Data Mesh can provide a more sustainable, scalable, and adaptable approach to data management that can help organizations to realize the full value of their data.

Drawbacks

some potential drawbacks that organizations should be aware of when considering this approach:

- Increased Complexity: While Data Mesh can reduce complexity in some areas, it can also introduce new complexities. Organizations must manage the relationships between domain teams, ensure that data is shared in a standardized way, and maintain a federated governance model across domains. This can add complexity to the data architecture and require significant coordination and communication.
 - Talent and Skillset Requirements: Data Mesh requires domain teams to have a high level of data expertise and technical skills to manage their own data. This can create challenges for organizations that do not have the necessary talent in-house or that struggle to attract and retain data professionals.
 - Lack of Centralized Control: Data Mesh relies on domain teams to manage their own data, which can lead to a lack of centralized control over the data. This can create challenges for organizations that need to ensure compliance and data security, as well as those that require a centralized view of the data.
 - Implementation Challenges: Implementing a Data Mesh architecture requires significant changes to the organization's data management processes and technology infrastructure. This can be a complex and time-consuming process that requires significant investment in people, processes, and technology.
 - Integration Challenges: Organizations that adopt a Data Mesh architecture may face integration challenges when trying to connect data from different domains. This can require significant effort to establish standardized interfaces and protocols for data exchange.
- Overall, Data Mesh can offer many benefits, but it also requires careful planning and execution to ensure that the benefits outweigh the potential drawbacks. Organizations should carefully consider their specific business needs, talent and skillset requirements, and technology infrastructure before adopting a Data Mesh architecture.

Adoption

some potential drawbacks that organizations should be aware of when considering this approach:

- Increased Complexity: While Data Mesh can reduce complexity in some areas, it can also introduce new complexities. Organizations must manage the relationships between domain teams, ensure that data is shared in a standardized way, and maintain a federated governance model across domains. This can add complexity to the data architecture and require significant coordination and communication.
 - Talent and Skillset Requirements: Data Mesh requires domain teams to have a high level of data expertise and technical skills to manage their own data. This can create challenges for organizations that do not have the necessary talent in-house or that struggle to attract and retain data professionals.
 - Lack of Centralized Control: Data Mesh relies on domain teams to manage their own data, which can lead to a lack of centralized control over the data. This can create challenges for organizations that need to ensure compliance and data security, as well as those that require a centralized view of the data.
 - Implementation Challenges: Implementing a Data Mesh architecture requires significant changes to the organization's data management processes and technology infrastructure. This can be a complex and time-consuming process that requires significant investment in people, processes, and technology.
 - Integration Challenges: Organizations that adopt a Data Mesh architecture may face integration challenges when trying to connect data from different domains. This can require significant effort to establish standardized interfaces and protocols for data exchange.
- Overall, Data Mesh can offer many benefits, but it also requires careful planning and execution to ensure that the benefits outweigh the potential drawbacks. Organizations should carefully consider their specific business needs, talent and skillset requirements, and technology infrastructure before adopting a Data Mesh architecture.

Challenges

Data Mesh can be a valuable approach for organizations that face the following challenges:

- Data Silos: Organizations that have multiple systems and data sources that are not well integrated or shareable may benefit from adopting a Data Mesh architecture. Data Mesh can help to break down data silos and enable better collaboration and data sharing across different teams and domains.
- Data Governance: Organizations that struggle with data governance and management may benefit from Data Mesh. By empowering domain teams to manage their own data, Data Mesh can create a more decentralized approach to data governance that can improve data quality and reduce the risk of data breaches.
- Data Complexity: Organizations that deal with complex and diverse data sources may benefit from adopting Data Mesh. By organizing data around business domains and enabling domain teams to manage their own data, Data Mesh can provide a more flexible and adaptable approach to data management that can better accommodate complex data structures.
- Scalability: Organizations that need to scale their data management capabilities rapidly may benefit from Data Mesh. By enabling domain teams to manage their own data, Data Mesh can

provide a more scalable approach to data management that can accommodate rapid growth and changing business needs.

31. Data Monetization: Organizations that want to monetize their data assets may benefit from adopting Data Mesh. By creating data products that are managed by domain teams, Data Mesh can enable organizations to create new revenue streams and unlock the value of their data.
Overall, Data Mesh can be a valuable approach for organizations that need a more flexible, scalable, and adaptable approach to data management. It is particularly well-suited to organizations that deal with complex data structures, struggle with data governance, or need to rapidly scale their data management capabilities. However, organizations should carefully evaluate their specific needs and consider the potential benefits and drawbacks of Data Mesh before adopting this approach.

Data Mesh Journey

The journey towards implementing Data Mesh involves several steps, including:

32. Understand Business Needs: The first step towards implementing Data Mesh is to understand the specific business needs and challenges that the organization is facing. This includes identifying the domains within the organization, the data sources that exist within each domain, and the challenges associated with managing and sharing data across domains.
 33. Identify Domain Teams: The next step is to identify the domain teams that will be responsible for managing the data within each domain. Domain teams should be composed of subject matter experts who have a deep understanding of the data within their domain.
 34. Define Domain Interfaces: Once the domain teams have been identified, the organization should define the interfaces between domains. This involves defining the data standards and protocols that will be used to exchange data between domains.
 35. Establish Federated Governance: Data Mesh requires a federated governance model in which each domain team is responsible for managing their own data. This requires the development of a governance framework that outlines the roles and responsibilities of each domain team and establishes guidelines for data quality, security, and privacy.
 36. Implement Data Infrastructure: Once the governance framework has been established, the organization can begin implementing the necessary data infrastructure to support Data Mesh. This may include the development of data platforms, data pipelines, and data products that are managed by domain teams.
 37. Monitor and Evaluate: Finally, the organization should continuously monitor and evaluate the effectiveness of the Data Mesh implementation. This includes tracking key performance metrics such as data quality, data sharing, and data monetization, and making adjustments to the implementation as needed.
- Overall, the journey towards implementing Data Mesh requires a significant investment in people, processes, and technology. However, by adopting this approach, organizations can create a more flexible, scalable, and adaptable approach to data management that can better accommodate changing business needs and support innovation.
- +++++