Published in Towards Data Science

Eric Broda Follow
Jan 18, 2022 · 7 min read · ✦ · ▶ Listen

Save

Data Mesh Architecture Patterns

The enterprise data mesh is revolutionizing how enterprises are managing their data. What are the foundational Data Mesh patterns?

Photo b… …splash

Data Mesh Patterns

The enterprise data mesh is emerging as a unique and compelling way to manage data within an enterprise. It brings "product thinking" to enterprise data management while enabling new levels agility and data governance in the enterprise. And it creates a "self-serve" capability with near real-time synchronization of data thereby creating the foundation for real-time digital enterprises.

But, alas, there is no single product that brings you the Data Mesh. Rather, an Enterprise's Data Mesh is composed of many commonly available components (see next section for a Data Mesh architecture recap).

Still, the key to success is understanding how these components interact. In this article, I will describe these interactions using Architecture Patterns.

Data Mesh Architecture Recap

An Enterprise Data Mesh is composed of many components (lots more detail available here, here, and here). Data Products, the primary building block within a Data Mesh, contain operational, analytic, and/or engagement data which is synchronized across the organiation using an Enterprise's Data Mesh. APIs are used to access data within a Data Product. To support federated governance, each Data Product contains an audit log that records data changes and a catalog of data it manages.

An Enterprise's Data Mesh has many Data Products. Data Products subscribe to each other's data such that when one Data Product changes its data, this change is communicated to other Data Products using Change Data Capture and an Event Streaming Backbone.

Lastly, an Enterprise Data Catalog — a synchronized aggregation of all Data Product catalogs and data changes– is used to make it easy for any user or developer to find, consume, and govern any data across the enterprise, while also providing the foundation for understanding data lineage across the enterprise.

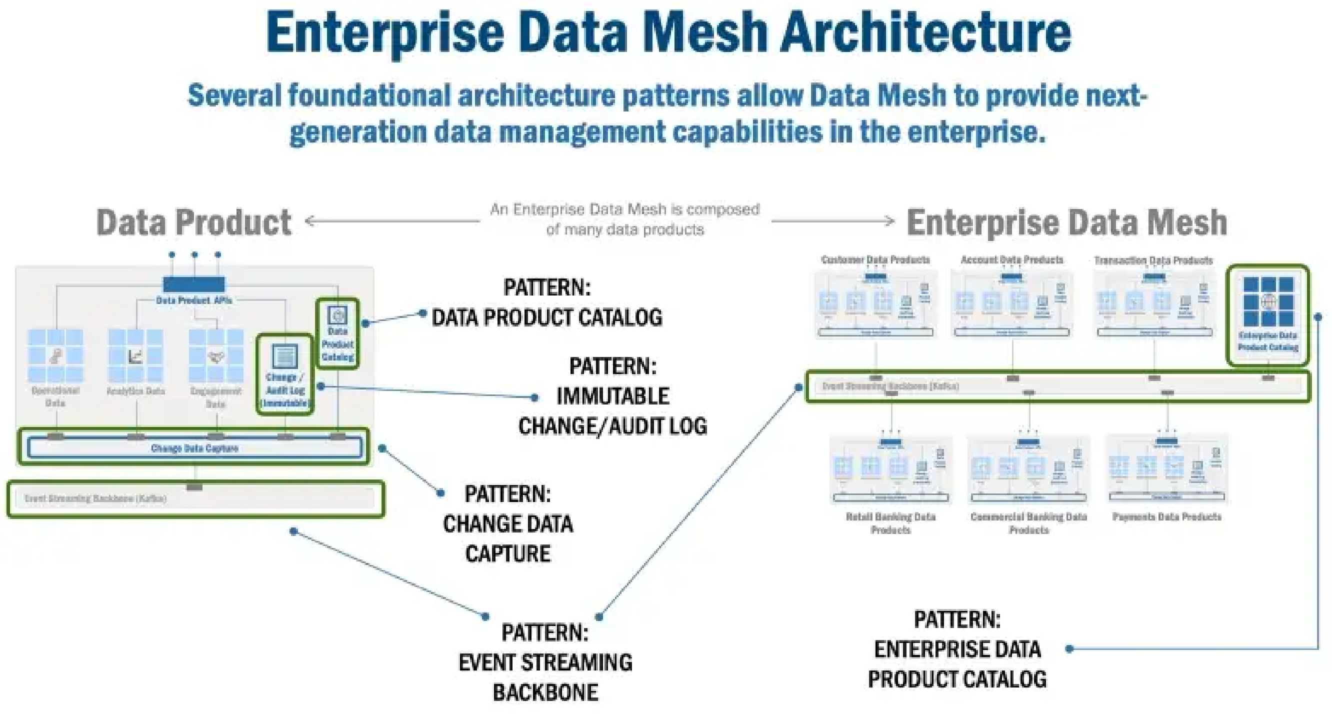

Enterprise Data Mesh Architecture
Several foundational architecture patterns allow Data Mesh to provide next-generation data management capabilities in the enterprise.
Data Product
An Enterprise Data Mesh is composed of many data products
Enterprise Data Mesh
Data Product APIs
Operational Data
Analytics Data
Engagement Data
Change / Audit Log (Immutable)
Data Product Catalog
Change Data Capture
Event Streaming Backbone (Kafka)
PATTERN: DATA PRODUCT CATALOG
PATTERN: IMMUTABLE CHANGE/AUDIT LOG
PATTERN: CHANGE DATA CAPTURE
PATTERN: EVENT STREAMING BACKBONE
Customer Data Products
Account Data Products
Transaction Data Products
Enterprise Data Product Catalog
Event Streaming Backbone (Kafka)
Retail Banking Data Products
Commercial Banking Data Products
Payments Data Products
PATTERN: ENTERPRISE DATA PRODUCT CATALOG

Figure 1, Enterprise Data Mesh Architecture

We will describe the following Architecture Patterns in this article:

- Change Data Capture
- Event Streaming Backbone
- Data Product Catalog
- Enterprise Data Product Catalog
- Immutable Change / Audit Log

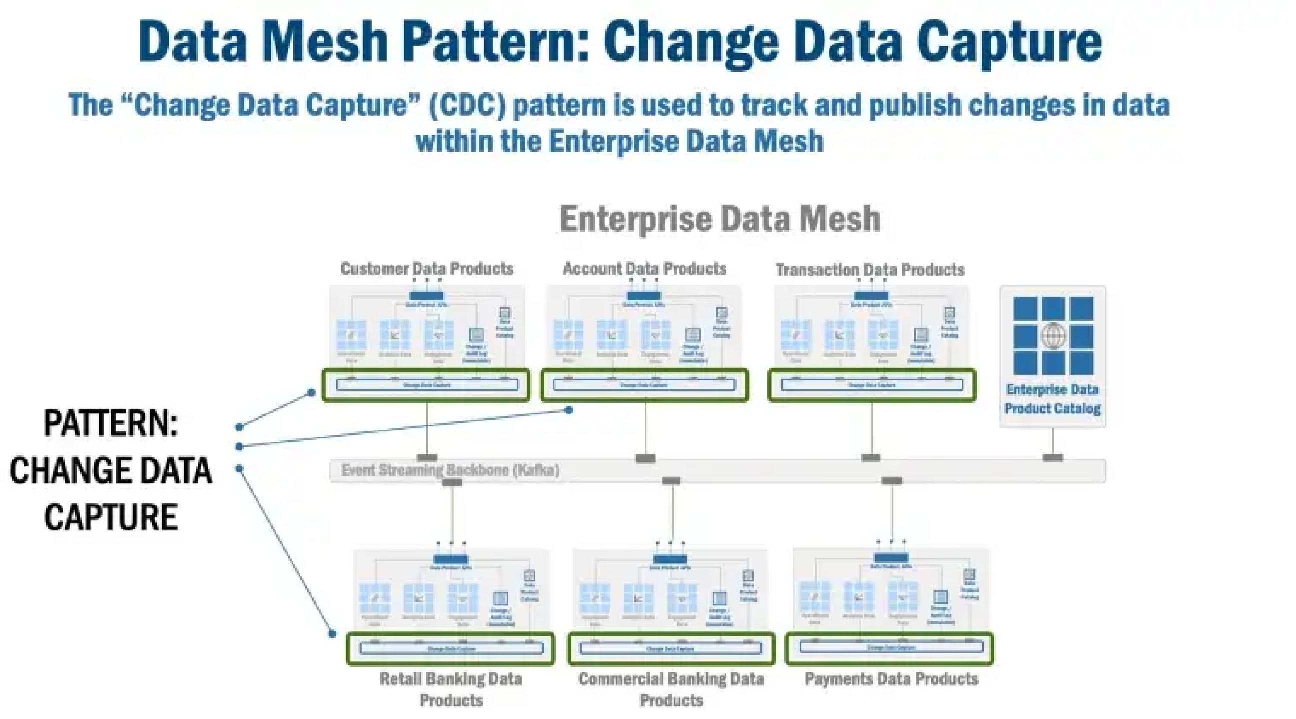
Data Mesh Pattern: Change Data Capture

Today, it is difficult to deliver data safely, reliably, and consistently between service and application boundaries. There are two approaches to address this challenge. First, data can be updated synchronously across multiple databases using protocols such as “two-phase commit” (2PC), but this approach is usually complex and costly and is usually reserved for situations where it is absolutely crucial to keep multiple data sources in sync.

A second approach is to update a primary database immediately while a secondary database is updated at a future time (but out of the scope of a transaction). Problems arise when the time span between when the primary and secondary databases are updated is longer than desired.

Change Data Capture (CDC) is a foundational component used by an Enterprise Data Mesh to address this challenge. CDC works by capturing and publishing entries in a database's transaction log but most importantly it does this *unobtrusively* and outside of the original transaction. This means that CDC transparently captures changes in operational (or analytic) data without impact to the originating application or transaction flow.

(Note: There is a lot more detail here for those looking for a more detail on how CDC works in an Enterprise)



Data Mesh Pattern: Change Data Capture

The "Change Data Capture" (CDC) pattern is used to track and publish changes in data within the Enterprise Data Mesh

Enterprise Data Mesh
Customer Data Products
Account Data Products
Transaction Data Products
Enterprise Data Product Catalog
PATTERN: CHANGE DATA CAPTURE
Event Streaming Backbone (Kafka)
Retail Banking Data Products
Commercial Banking Data Products
Payments Data Products

Figure 2, Data Mesh Pattern: Change Data Capture

But what does CDC do with the captured "event". In the Enterprise Data Mesh, it publishes the event to an Event Streaming Backbone (the next pattern) for distribution across the enterprise.

Data Mesh Pattern: Event Streaming Backbone

An **Event Streaming Backbone** distributes events in an Enterprise Data Mesh. Events commonly originate from applications, APIs, and, in our case, also from CDC. However, of particular importance is that any published event can be safely, reliably, and in *near real-time* consumed by any other subscribing entity.

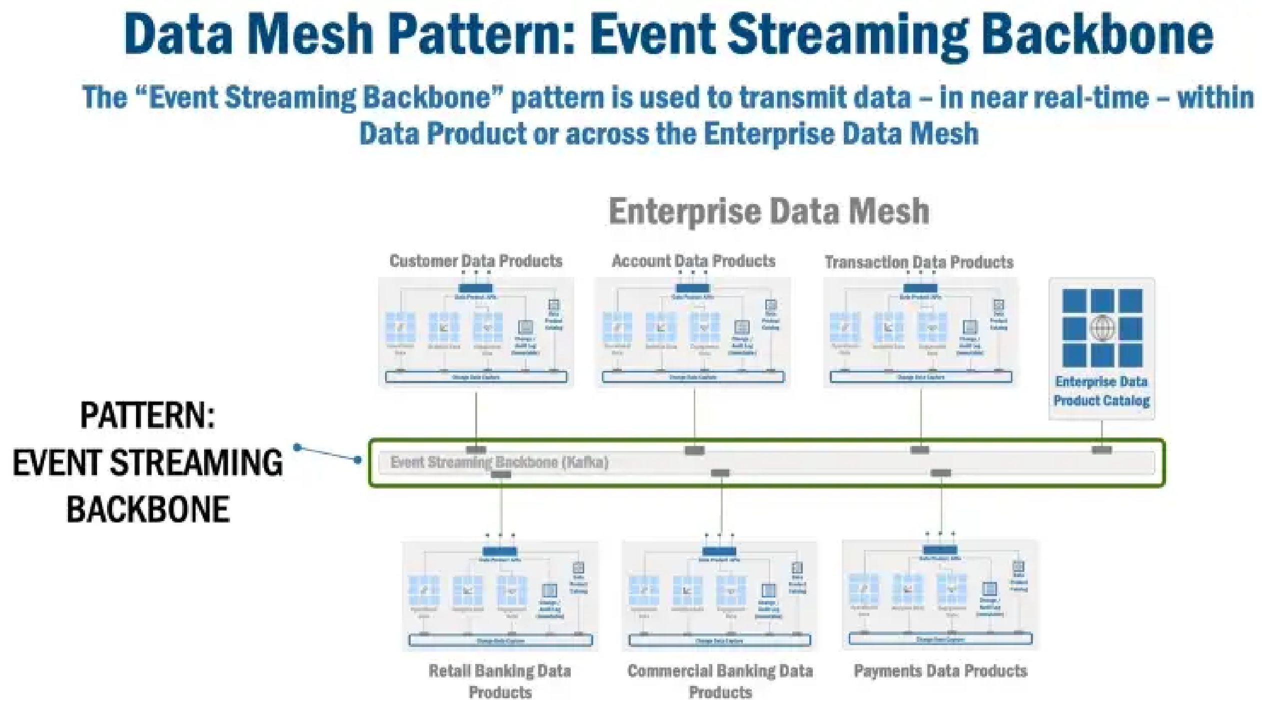

Data Mesh Pattern: Event Streaming Backbone
The "Event Streaming Backbone" pattern is used to transmit data – in near real-time – within Data Product or across the Enterprise Data Mesh
Enterprise Data Mesh
Customer Data Products
Account Data Products
Transaction Data Products
Enterprise Data Product Catalog
PATTERN: EVENT STREAMING BACKBONE
Event Streaming Backbone (Kafka)
Retail Banking Data Products
Commercial Banking Data Products
Payments Data Products

Figure 3, Data Mesh Pattern: Event Streaming Backbone

There are several core managed entities in an Event Streaming Backbone:

- **Events**, defined by JSON Schemas, are distributed within the Enterprise Data Mesh.
- **Topics** are used to queue and distribute events across the enterprise; The Enterprise Data Mesh uses well-known topics that act like queues by allowing many entities to publish and consume events.
- **Producers** publish events to topics; Producers in an Enterprise Data Mesh may be APIs, applications, or CDC.
- **Consumers** consume events from topics. Consumers in an Enterprise Data Mesh may be any entity or application that subscribe to a topic and are notified when an event is available for processing.
- **Event Stream Processors** process events either on a per-event or aggregating on a time-window basis enabling for very sophisticated and powerful analysis techniques in an Enterprise Data Mesh.
- **Brokers** manage the above components to ensure safe and reliable event communications across the entire Enterprise Data Mesh.

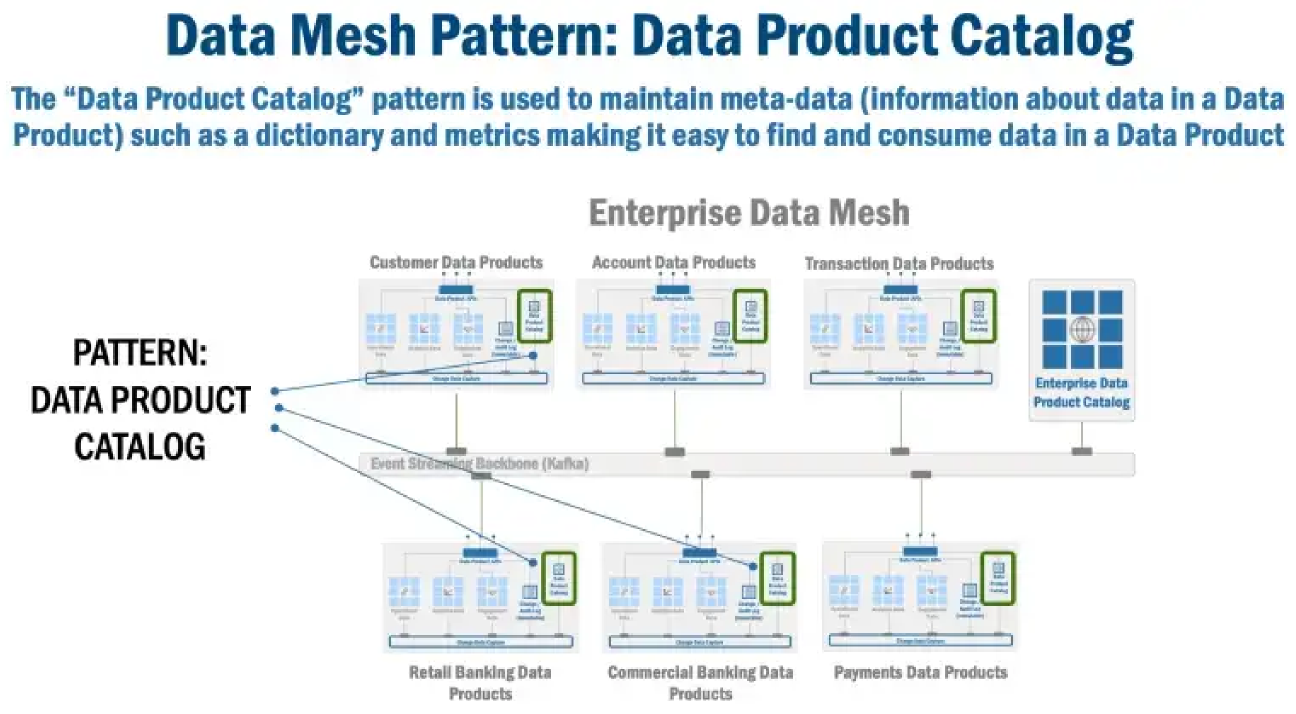
Data Mesh Pattern: Data Product Catalogs

Data, they say, is the new gold and mining it will bring immense insight and wealth. But in most enterprises today data is strewn across many groups in an organization. Sales owns customer data, distribution owns supply chain, and finance owns transactions and accounts.

Unfortunately, this makes it very difficult to find data, and once found, it is even more difficult to bring it together to make holistic business decisions. The result: slow, costly, and uninformed decision making.

A **Data Product Catalog** (DPC) contains information about data ("meta-data") for a Data Product. The DPC provides the information that makes it easy for any authorized person or application to find, view, and consume Data Products in the Enterprise Data Mesh. A DPC offers several benefits:

- **Easy to Manage**, by enabling local ownership and accountability.
- **Easy to Change and Grow**, by allowing decision making to be localized and faster.
- **Easy to Find, View, and Consume data**, by making by making it easy for any (authorized) entity to find, view, and consume data (i.e., "self-serve").



Data Mesh Pattern: Data Product Catalog

The "Data Product Catalog" pattern is used to maintain meta-data (information about data in a Data Product) such as a dictionary and metrics making it easy to find and consume data in a Data Product

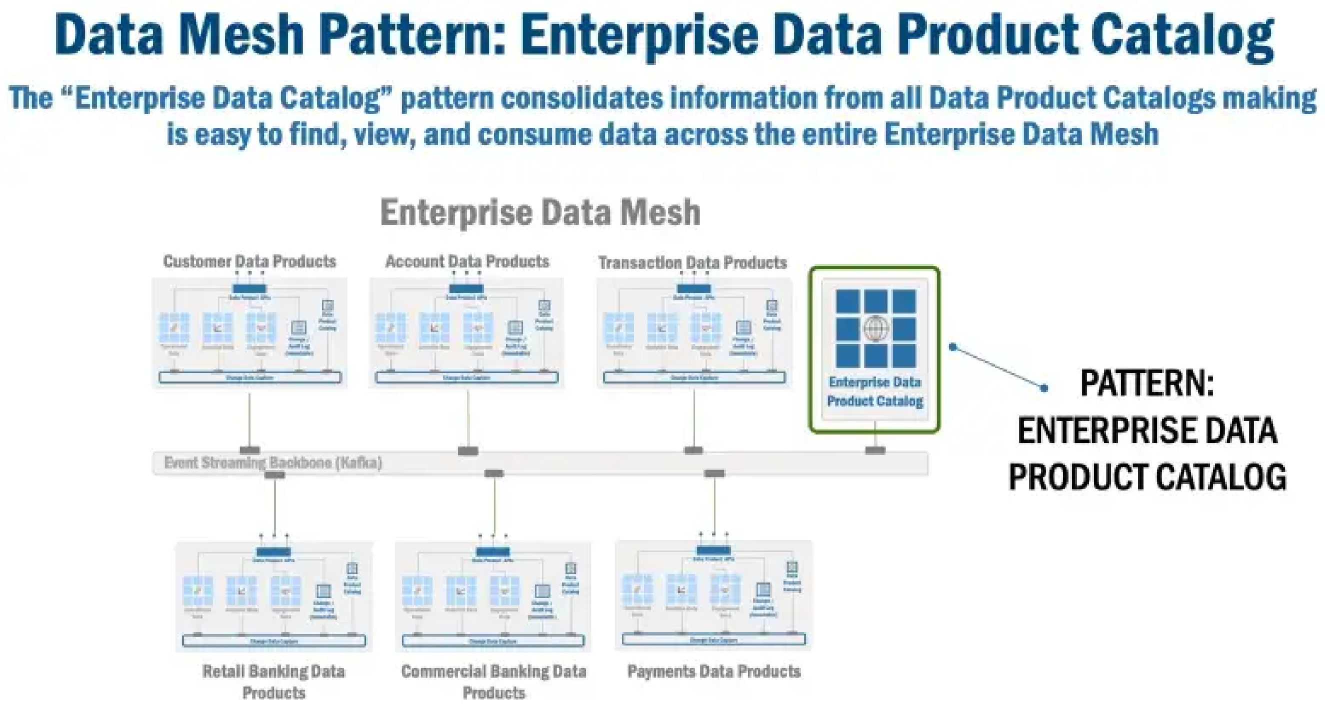
Enterprise Data Mesh
Customer Data Products
Account Data Products
Transaction Data Products
Enterprise Data Product Catalog
PATTERN: DATA PRODUCT CATALOG
Event Streaming Backbone (Kafka)
Retail Banking Data Products
Commercial Banking Data Products
Payments Data Products

Figure 4, Data Mesh Pattern: Data Product Catalog

Data Mesh Pattern: Enterprise Data Product Catalog

Enterprise Data Product Catalog (EDPC) is a repository that aggregates meta-data from all local Data Product Catalogs (DPCs). The Enterprise Data Catalog is used to store information and statistics — meta-data — about *all* data maintained in an Enterprise Data Mesh making it easy to find, view, consume, and govern data:

- **Data Scientists** use the EDPC to find the location of data in the enterprise that they can use to train their models.
- **Business Users** use the EDPC to find information required for business decisions.
- **Developers** use the EDPC to understand data structures required by their applications.
- **Governance Professionals** uses EDPC to understand and monitor data across the enterprise, enabling federated computational governance within the Enterprise Data Mesh.



Data Mesh Pattern: Enterprise Data Product Catalog

The "Enterprise Data Catalog" pattern consolidates information from all Data Product Catalogs making is easy to find, view, and consume data across the entire Enterprise Data Mesh

Enterprise Data Mesh
Customer Data Products
Account Data Products
Transaction Data Products
Enterprise Data Product Catalog
PATTERN: ENTERPRISE DATA PRODUCT CATALOG
Event Streaming Backbone (Kafka)
Retail Banking Data Products
Commercial Banking Data Products
Payments Data Products

Figure 5, Data Mesh Pattern: Enterprise Data Product Catalog

Data Mesh Pattern: Immutable Change/Audit Log

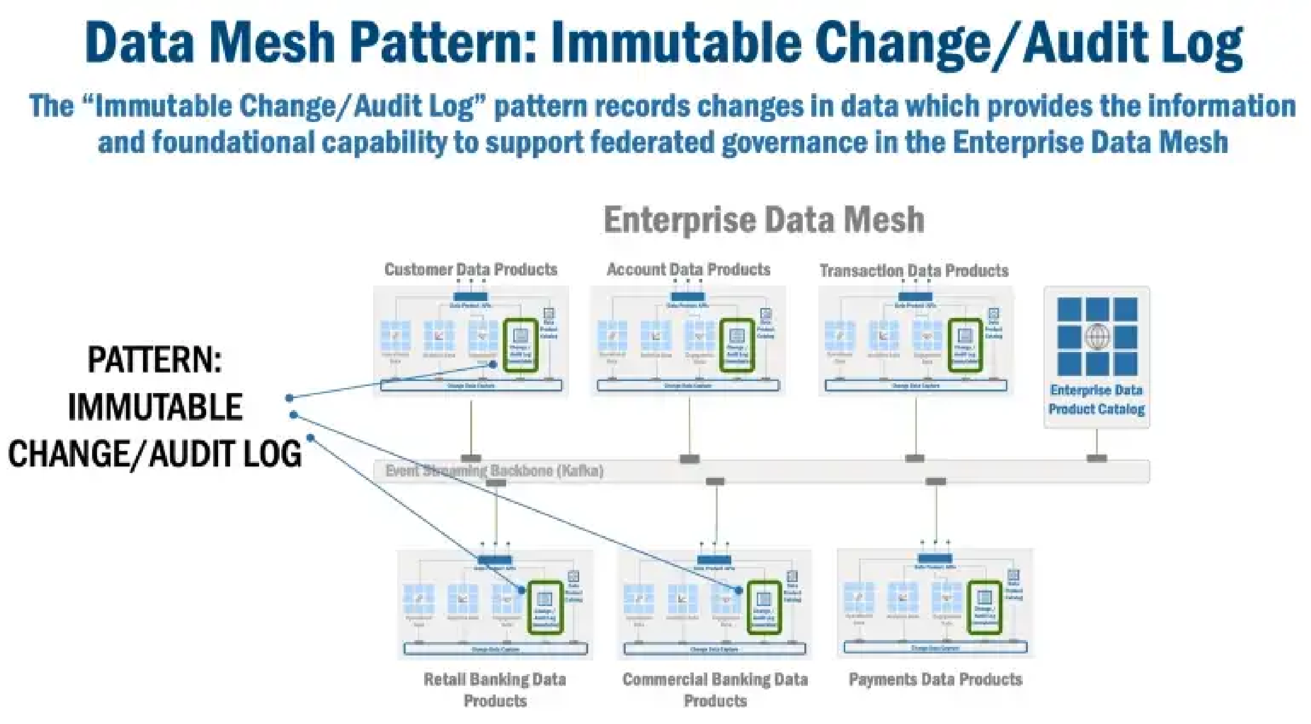
Understanding the lineage of data — defined as the aggregated list of changes data undergoes — is crucial for governance and regulatory purpose. Why is this important? Consider a common situation today: The emergence of AI/Machine

Learning is now a mandatory enterprise capability. Data Scientists use sophisticated models to enable and make critical business decisions.

However, in many enterprises, especially those in health care and finance, the actual viability of these models is predicated on the ability to meet regulators demands for reproducibility and traceability (more information is available here and here). Unfortunately, most enterprise do not have the capability to track data lineage in a fashion required by auditors or regulators.

An Enterprise Data Mesh's **Immutable Change/Audit Log** addresses this need by retaining a history data changes within the Enterprise Data Mesh for future audit and governance purposes. Local Data Product change/audit logs are updated automatically upon any data change in the data. These logs are then propagated to an Enterprise Data Product Catalog (EDPC) allowing for a consolidated history of data changes in the enterprise.

In other words, the EDPC contains the data lineage for all elements within the Enterprise Data Mesh. The EDPC uses this data to provide a searchable index of meta-data — which explicitly includes references to each Data Product's immutable change/audit log — allows data lineage to be easily found and confirmed.


Data Mesh Pattern: Immutable Change/Audit Log

The "Immutable Change/Audit Log" pattern records changes in data which provides the information and foundational capability to support federated governance in the Enterprise Data Mesh

Enterprise Data Mesh
Customer Data Products
Account Data Products
Transaction Data Products
Enterprise Data Product Catalog
PATTERN: IMMUTABLE CHANGE/AUDIT LOG
Event Streaming Backbone (Kafka)
Retail Banking Data Products
Commercial Banking Data Products
Payments Data Products

Figure 6, Data Mesh Pattern: Immutable Change/Audit Log

Concluding Thoughts

An Enterprise Data Mesh is becoming a foundational enabler of the real-time digital enterprise. Architecture patterns provide an established way to describe Data Mesh interactions. And while there is no readily available tool available, the first step to building your own organizations Data Mesh is to understand the foundational patterns that enable a Data Mesh.

Hopefully this article gives you the necessary insight to kickstart your own Enterprise Data Mesh!

***

All images in this document except where otherwise noted have been created by Eric Broda (the author of this article). All icons used in the images are stock PowerPoint icons and are free from copyrights.

Data Mesh Data Science Change Data Capture Event Streaming

Enterprise Architecture