Whitepaper

Self-Serve Data Platform Foundational Framework

Trademarks and Permissions

and other Futurewei® trademarks are trademarks of Futurewei® Technologies. Huawei trademarks are trademarks of Huawei Technologies Co., Ltd. All other trademarks and trade names mentioned in this document are the property of their respective holders.

Notice

The purchased products, services, and features are stipulated by the contract made between Huawei and the customer. All or part of the products, services, and features described in this document may not be within the purchase scope or the usage scope. Unless otherwise specified in the contract, all statements, information, and recommendations in this document are provided "AS IS" without warranties, guarantees, or representations of any kind, either express or implied.

The information in this document is subject to change without notice. Every effort has been made in the preparation of this document to ensure accuracy of the contents, but all statements, information, and recommendations in this document do not constitute a warranty of any kind, express or implied.

FUTUREWEI® TECHNOLOGIES, INC.

Boston Research Center

Address: 111 Speen Street, Suite 114
Framingham, MA 01701
United States of America

Website: http://www.futurewei.com/

Table Of Contents

Executive Summary 4
Introduction 4
Data Mesh Brief 5
Topic Of Focus 6
Capabilities Of Self-Serve Data Platform 8
Conclusion 12
References 13

Executive Summary

- Decentralized data architecture offers numerous advantages over monolithic centralized data architecture, especially as data volumes continue to grow exponentially.
- The Self-Serve Platform, a fundamental principle of Data Mesh, reduces barriers for domain teams to take ownership, develop, and share data products, while enabling governance teams to monitor and ensure interoperability, compliance, and data product quality.
- The Self-Serve Platform comprises three layers:
    - Data Infrastructure Provisioning Plane
    - Data Product Developer Experience Plane
    - Data Mesh Supervision Plane
- Details of design decisions necessary to construct a robust and scalable framework for the Self-Serve Platform.

Introduction

We're currently experiencing an era inundated with data, recognized as the golden age of data. Data tracking agencies predict substantial growth in data generation over the next few years, albeit with some margin of error. However, there's a unanimous agreement that data generation is accelerating at an unprecedented pace.

With this surge in data, managing it becomes increasingly challenging. There are three primary approaches to data architecture: Centralized, Decentralized, and Hybrid. Recently, there's been a notable focus on decentralized data architecture, and for good reason.

Decentralized data architecture offers scalability benefits by distributing data processing and storage across multiple nodes. This approach allows for seamless expansion as data volume increases, reducing bottlenecks and enhancing system performance. It enables organizations to efficiently handle growing data volumes without sacrificing agility or performance.

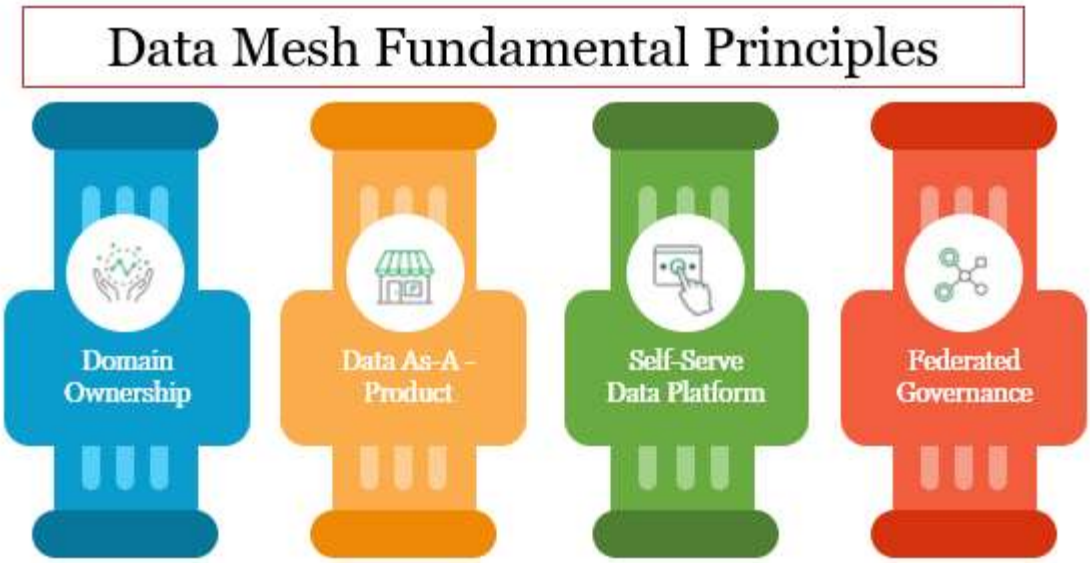
This paper delves into the infrastructure platform associated with one such decentralized approach known as Data Mesh. It would serve as a reference for more advanced or futuristic self-serving platforms.

Data Mesh Brief

Plenty of documentation exists in various forms such as books, websites, and articles regarding Data Mesh. Therefore, in this section, we will provide a concise overview of the fundamental principles of Data Mesh.

Domain Ownership: This principle asserts that domain teams are accountable for their respective data. It advocates structuring analytical data around domains, mirroring team boundaries that align with the system's bounded context. By embracing a domain-driven distributed architecture, ownership of analytical and operational data shifts from the central data team to the domain teams

Data-As-A-Product: The principle embodies a product-oriented mindset towards analytical data. It emphasizes that data has consumers beyond its originating domain. Accordingly, the domain team must ensure the provision of high-quality data to meet the needs of other domains. Essentially, domain data should be treated akin to any other public API.



Self-Serve Data Platform: Objective is to embrace a platform-oriented approach to data infrastructure. Led by a dedicated data platform team, it furnishes domain-agnostic functionalities, tools, and systems essential for constructing, executing, and sustaining interoperable data products across all domains. Through this platform, the data platform team empowers domain teams to effortlessly utilize and develop data products.

Federated Governance: This principle ensures the interoperability of all data products by advocating standardization throughout the data mesh, led by the governance group. Its primary aim is to establish a data ecosystem that adheres to organizational guidelines and industry regulations, promoting consistency and compliance across the board.

Topic Of Focus

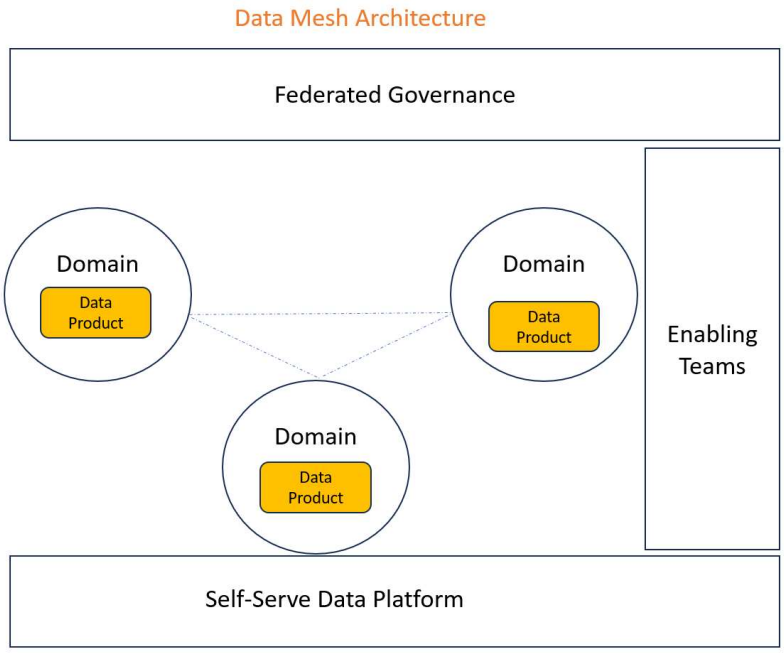
Self-Serve Data Platform.

To create, deploy, run, and manage a data product, substantial infrastructure provisioning is necessary. The specialized skills required for this task would be challenging to duplicate across different domains. Crucially, teams can only autonomously own their data products if they have access to a streamlined infrastructure abstraction. This abstraction eliminates the complexities

and obstacles in provisioning and managing data product lifecycles. Thus, a new principle emerges: **Self-serve data as a platform**, facilitating domain autonomy.

One major concern when distributing data ownership to different domains is the redundant work and specialized skills required to manage the technology stack and infrastructure for data pipelines in each domain.

Consolidating domain-neutral infrastructure capabilities into a centralized data infrastructure platform resolves the need for replicating efforts in setting up data pipeline engines, storage, and streaming infrastructure across domains. This platform, managed by a dedicated data infrastructure team, supplies the necessary technology for domains to handle data capture, processing, storage, and retrieval efficiently.



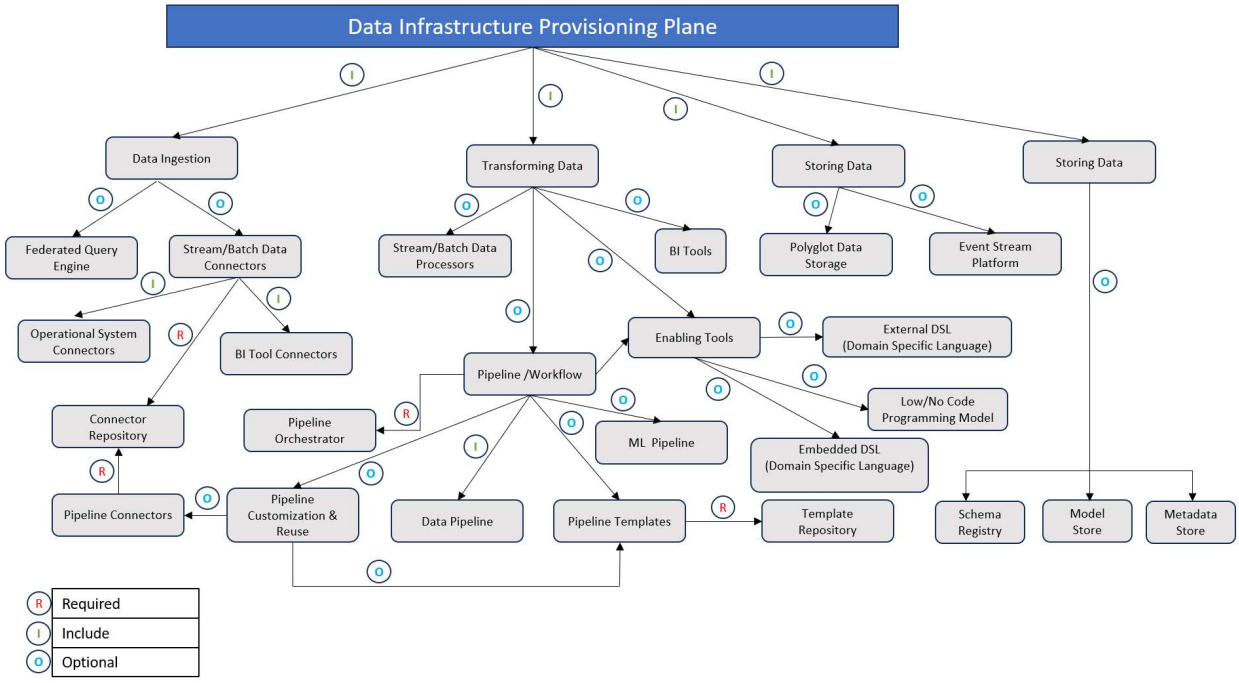
The key principle in establishing the data infrastructure platform is two fold: (a) ensuring it remains free from domain-specific concepts or business logic, maintaining domain neutrality, and (b) ensuring the platform abstracts away complexity, offering data infrastructure components through self-service mechanisms. This approach simplifies operations while providing the essential tools for effective data management across diverse domains.

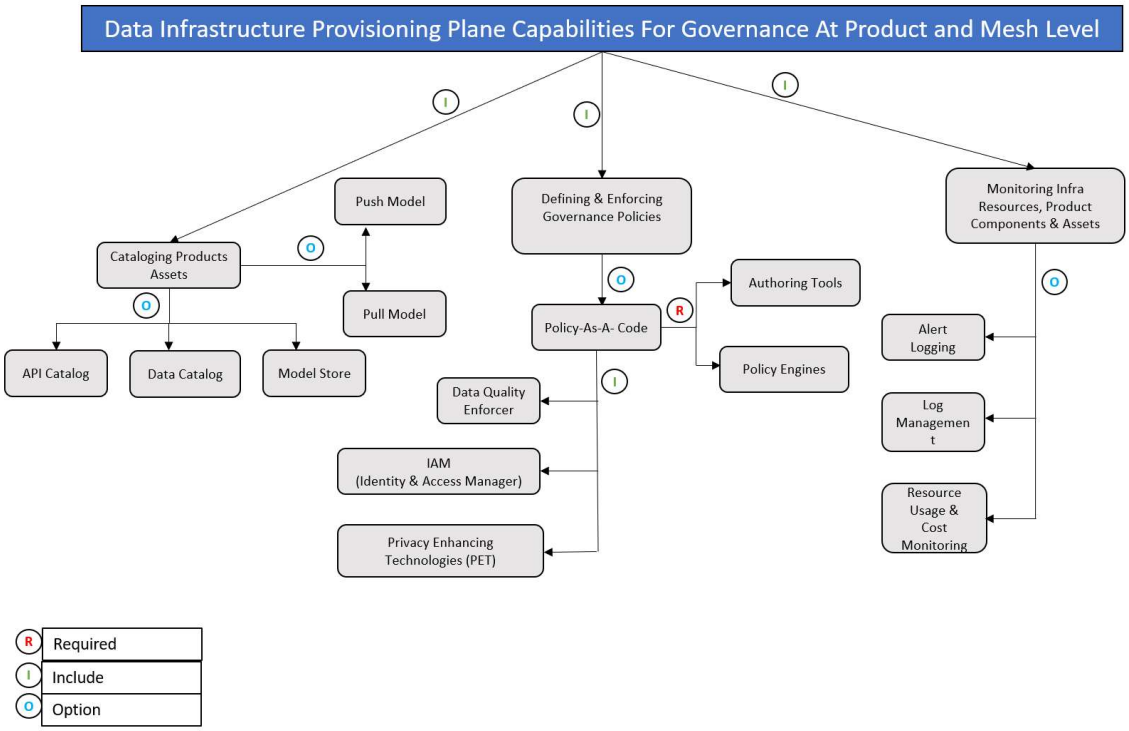
One key measure of success for self-serve data infrastructure is reducing the time it takes to create a new data product on the system. This reduction in lead time is crucial because it encourages automation, which is necessary for implementing the functionalities of a "**data product**," as discussed in the section about treating domain data as a product.

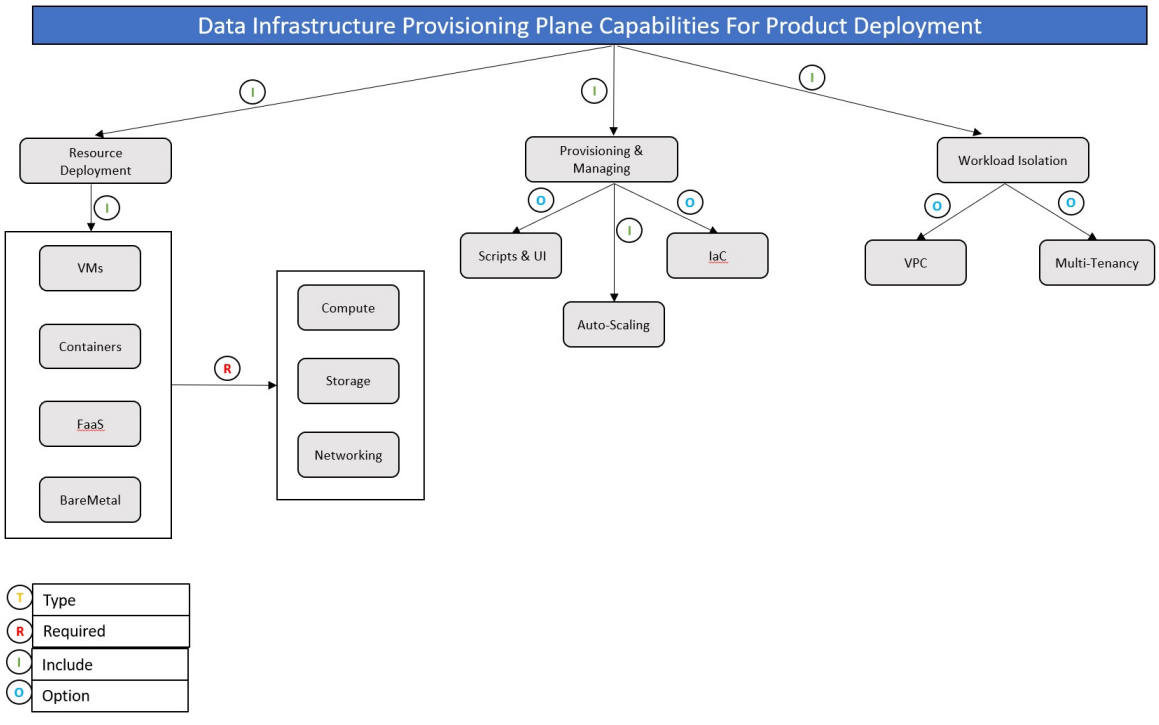
Capabilities Of Self-Serve Data Platform

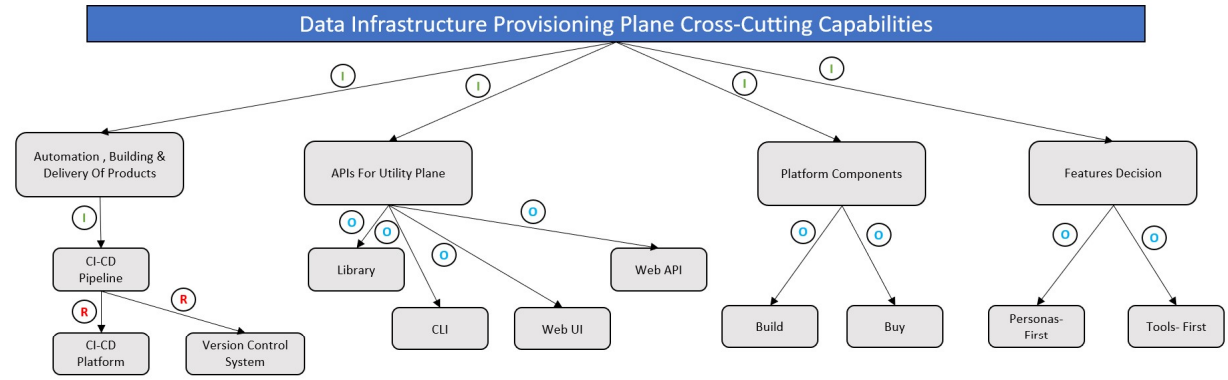
The self-serve platform capabilities are categorized into multiple planes. It's important to note that a plane represents a distinct level of existence, integrated yet separate. This concept is akin to physical and consciousness planes, or control and data planes in networking. Importantly, a plane does not necessarily imply a strict hierarchical access model, but rather delineates different dimensions of functionality.

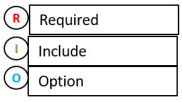
- **Data Infrastructure Provisioning Plane**: Facilitates the provisioning of essential infrastructure necessary for running the components within a data product and its interconnected network of products. This encompasses the setup of distributed file storage, storage accounts, access control management systems, orchestration for internal code execution within data products, and deployment of distributed query engines across a network of interconnected data products. Typically, this interface is utilized directly by advanced data product developers or other data platform planes. It represents a foundational aspect of data infrastructure lifecycle management, operating at a relatively granular level.



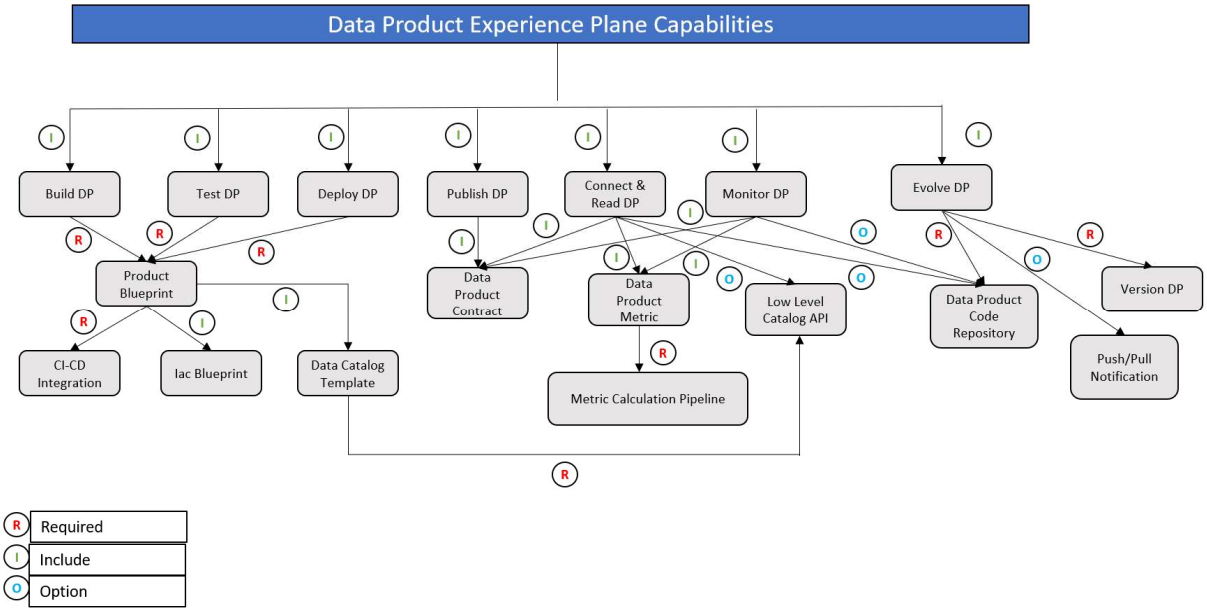

Data Infrastructure Provisioning Plane Capabilities For Governance At Product and Mesh Level
I
I
I
Cataloging Products Assets
O
Push Model
Pull Model
O
O
API Catalog
Data Catalog
Model Store
Defining & Enforcing Governance Policies
O
Policy-As-A- Code
R
Authoring Tools
Policy Engines
I
Data Quality Enforcer
IAM (Identity & Access Manager)
Privacy Enhancing Technologies (PET)
Monitoring Infra Resources, Product Components & Assets
O
Alert Logging
Log Managemen t
Resource Usage & Cost Monitoring
R Required
I Include
O Option


Data Infrastructure Provisioning Plane Capabilities For Product Deployment
I
I
I
Resource Deployment
I
VMs
Containers
FaaS
BareMetal
R
Compute
Storage
Networking
Provisioning & Managing
O
O
O
Scripts & UI
I
IaC
Auto-Scaling
Workload Isolation
O
O
VPC
Multi-Tenancy
T Type
R Required
I Include
O Option

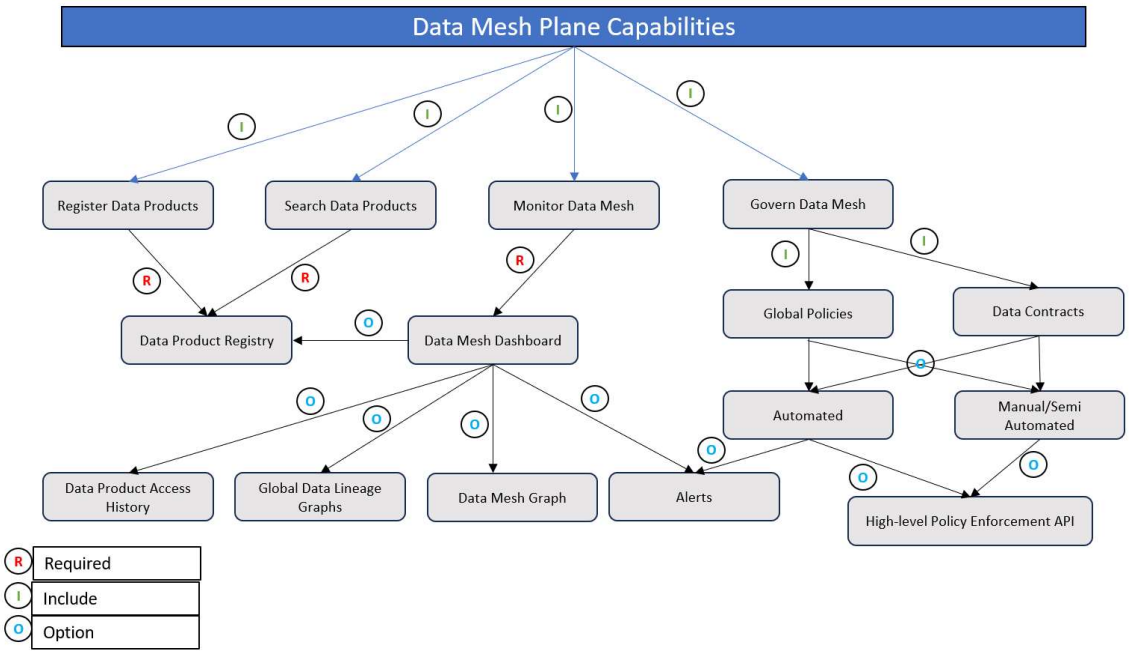




- **Data Product Developer Experience Plane**: Serving as the primary interface for the average data product developer, this plane streamlines the developer workflow by abstracting away complexities. Unlike the 'provisioning plane', it operates at a higher level of abstraction, simplifying the management of data product lifecycles. Through intuitive declarative interfaces, it automates the implementation of standardized cross-cutting concerns and global conventions, ensuring consistency across all data products and interfaces.



- **Data Mesh Supervision Plane:** Certain capabilities are most effectively delivered at the mesh level, where a network of interconnected data products operates globally. Although individual data products may possess the necessary capabilities, it's more efficient to offer them at the mesh level. For instance, discovering data products tailored to a specific use case is facilitated through search or browsing within the data product network. Similarly, correlating multiple data products to generate advanced insights is best achieved through executing semantic queries across the entire mesh.



Conclusion

I advocate for aligning architectural design decisions with the three planes of the Self-Serve Data Platform, which can aid in identifying crucial design and implementation challenges within such a platform. As decentralized data architecture gains traction and becomes more prevalent, this approach can contribute to refining and enhancing the self-serve data platform, maximizing its potential to add value to organizations through optimal data utilization.

References

https://martinfowler.com/articles/data-mesh-principles.html

https://www.datamesh-architecture.com/