LTIMindtree

Whitepaper

Data Mesh

A Decentralized Data Architecture for Business Agility & Scale

Author:

Vijaya Phanindra

Principal - Data Engineering, LTIMindtree

Table of Contents

01. Overview 03

02. How agile and scalable are you with your current data platforms? 03

03. What is a data mesh and what are its principles? 06

04. The benefits of data mesh 07

05. The challenges 08

06. Data domains 09

Sample Use Case 09

07. Operationalizing data mesh on GCP 11

Organization of nodes in the mesh 11

Reference Architecture on GCP 12

Self-service infrastructure 13

Central Governance Node 14

Domain Nodes 16

Data Sharing 18

Roles and Responsibilities 18

08. Summary 20

09. References 21

10. About the Author 22

Overview

In many organizations, Data & Analytics departments are leading data monetization efforts and are transforming from a cost center to a revenue center. Organizations in the recent past have moved from just data warehouse platforms to data warehouses plus data lake patterns to manage the volume, velocity, and variety of the data. However, organizations face constraints such as centralized data teams with limited skill sets to support the data growth and activate AI/ML innovations. In this context, organizations are looking for data architecture patterns that can provide the required scale and agility necessary to support the development of data products and achieve data monetization.

How agile and scalable are you with your current data platforms?

In traditional data warehouse/data lake implementations, the centralized IT department developed ETL pipelines to extract data from the source systems and consolidate it in a monolithic centralized data platform. A semantic consumption model is created on the centralized data platform for businesses to consume the datasets. The demand for data has grown tremendously as the organizations started treating data as an asset, mandating the data availability across business groups, breaking silos, and embedding data-driven insights in every part of business decision-making. Centralized IT departments struggle to meet data demands, provide insights, activate AI/ML innovations, and enable data monetization at the speed required by the business. Moreover, central IT teams have little or no domain-specific knowledge of business datasets. The bottom line is that centralized data teams with monolithic data platforms do not scale well to the growing needs of data consumers in the organization.

How often do we hear the following statements from business teams?

- We often struggle to get a clear view of
  - What kind of data sources are captured and available?
  - Who owns the data and how to access it?

- It takes six months to integrate a new data source into our analytical system, and it takes three months to change/add a single column in our data pipeline.

- We have an appetite for more data but are unable to satisfy it.

- We need to change our pipelines every time the source schema changes; they are incredibly inflexible and rigid.

- We believe we are duplicating data sources throughout our analytical system, and we think there is a better way of managing this.

- Data attributes from the same source system transformed and stored in multiple analytical systems have different values, making reconciliation difficult and time-consuming.

- We're having trouble sharing datasets securely across different functional areas within our company and sharing data with third parties is out of the question.

- Our data engineering team face challenges in responding to business requests, adding new data sources, changing data pipelines, and developing new data consumers. We often struggle with capacity constraints and are unable to scale.

Who **owns** the data?

How fast new **data producers** can be **integrated?**

How **flexible** are data pipelines for change?

Are we **duplicating data**?

What data is **captured?** What is the **source?** Who can **access** and use it?

Do we have **e2e data observability**?

How easy and flexible to **share** datasets internally & externally to **third parties?**

How fast your data engineering team can **respond** to data consumers?

What is a data mesh and what are its principles?

Recent developments in the data landscape coupled with the availability of less expensive data processing technologies have opened the door to previously unimagined use cases. Organizations, on the other hand, are constrained to adapt due to monolithic and centralized data architectural patterns.

Data mesh(1) brings in the next paradigm shift in data architecture to address the scalability challenges faced by organizations and brings attention to the missing domain thinking. The four foundation principles are:

Data Mesh Principle	Remarks
01. Decentralized domain ownership	Domain-specific teams take full responsibility for their domain datasets (analytical or operational datasets) and develop explicit interfaces that adhere to centralized standards. This shift in responsibilities enables businesses to scale and respond fast to the data landscape and environment changes.
02. Data as product	Treat data as a first-class citizen, not as a byproduct of the business processes. Approach data as product and value-driven and deliver outcomes instead of capabilities(2).
03. Self-serve data infrastructure	Leverage Infrastructure as code practices to enable rapid agility in standing up infra and scale it up and out based on best practices and best-of-breed technologies. A centralized data infrastructure platform will abstract out the complexities and encourages reusability across data domains.
04. Federated computational governance	Automated central governance model to ensure interoperability and standards across different data domains.

It is important to note that distributed architecture requires organizations to depart from centralized, monolithic analytical systems and structures. Data mesh enables autonomous self-organizing teams to embrace agile methodology. The shift is more than a technology-driven architectural change and requires organizational structural changes to achieve the desired level of scalability. As a result, appropriate change management is necessary before introducing data as product thinking.

The benefits of data mesh

It is important to note that distributed architecture requires organizations to depart from centralized, monolithic analytical systems and structures. Data mesh enables autonomous self-organizing teams to embrace agile methodology. The shift is more than a technology-driven architectural change and requires organizational structural changes to achieve the desired level of scalability. As a result, appropriate change management is necessary before introducing data as product thinking.

> ***Data mesh pattern in simple terms is about scaling data teams and platforms, with the motive to build data products at a rapid rate.***

The challenges

As with any new architectural pattern changes, the following are some of the challenges to be addressed.

01

Difficult and time-intensive in creating and setting up self-serve centralized data infrastructure.

02

Domain teams with insufficient skills, each node should be equipped with application, data, domain to DevOps skills.

03

Unable to decide on the technology and time lost in debating and making a decision.

04

Managing the balance between centralization and autonomy to domain teams.

05

Interoperability across domain nodes, unreachable nodes results in data silos and data duplication.

06

After-thought implementation of FAIR (Findable, Accessible, Interoperable & Reusable) data principles.

Data Domains

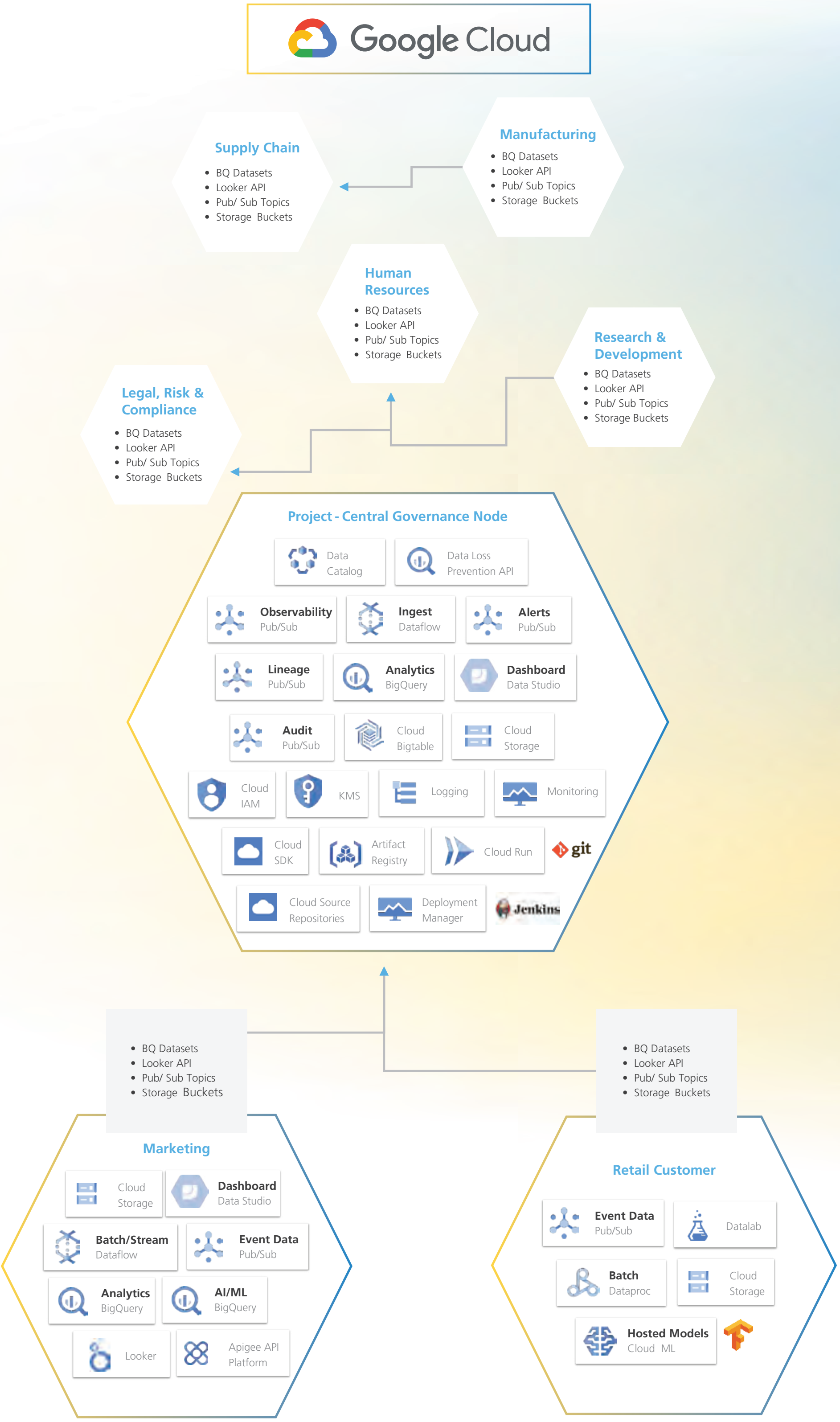
The first exercise to be done before embarking on journey is to define data domains. An exercise is to be carried out in collaboration with business in arriving at the domains. The domain can be as broad as CRM, Supply Chain, Finance, Marketing, Operations, IT and HR and as granular as sub-domains Employee, Compensation, Benefits, Learning, Talent under HR domain.

Note that this document is concerned with operationalizing data mesh architectural patterns on Google Cloud Platform with a self-service infrastructure accelerator, a core tenant of data mesh principles, and arriving at a set of data domains is out of scope.

Sample Use Case

Consider a sample use case for a manufacturing company in the auto industry with the business units, Manufacturing, Supply Chain, Research & Development, Legal, Risk & Compliance, Human Resources, IT, Marketing, Retail Customer and Product. A central GCP project hosts key information business and technical metadata, data lineage, observability metrics. Business unit nodes will have the ability to publish essential metadata information to the central project node. Access to publish information is enabled at the time of node creation.

The nodes can choose to publish the data products in a predefined standard format, via BQ Datasets, Standard APIs, Storage bucket drop boxes with file extracts, or pub/subtopic and events. All nodes are permitted to search metadata and available data products from other nodes via the central node data catalog interface. Each node subscribes to another node in one of the standard interface mechanisms allowed. For instance, the marketing department subscribes to the customer BQ datasets for marketing campaigns and customer segmentation.



Google Cloud

Supply Chain

- BQ Datasets
- Looker API
- Pub/ Sub Topics
- Storage Buckets

Manufacturing

- BQ Datasets
- Looker API
- Pub/ Sub Topics
- Storage Buckets

Human Resources

- BQ Datasets
- Looker API
- Pub/ Sub Topics
- Storage Buckets

Research & Development

- BQ Datasets
- Looker API
- Pub/ Sub Topics
- Storage Buckets

Legal, Risk & Compliance

- BQ Datasets
- Looker API
- Pub/ Sub Topics
- Storage Buckets

Project - Central Governance Node

- Data Catalog
- Data Loss Prevention API
- Observability Pub/Sub
- Ingest Dataflow
- Alerts Pub/Sub
- Lineage Pub/Sub
- Analytics BigQuery
- Dashboard Data Studio
- Audit Pub/Sub
- Cloud Bigtable
- Cloud Storage
- Cloud IAM
- KMS
- Logging
- Monitoring
- Cloud SDK
- Artifact Registry
- Cloud Run
- git
- Cloud Source Repositories
- Deployment Manager
- Jenkins

- BQ Datasets
- Looker API
- Pub/ Sub Topics
- Storage Buckets

Marketing

- Cloud Storage
- Dashboard Data Studio
- Batch/Stream Dataflow
- Event Data Pub/Sub
- Analytics BigQuery
- AI/ML BigQuery
- Looker
- Apigee API Platform

- BQ Datasets
- Looker API
- Pub/ Sub Topics
- Storage Buckets

Retail Customer

- Event Data Pub/Sub
- Datalab
- Batch Dataproc
- Cloud Storage
- Hosted Models Cloud ML

Operationalizing data mesh on GCP

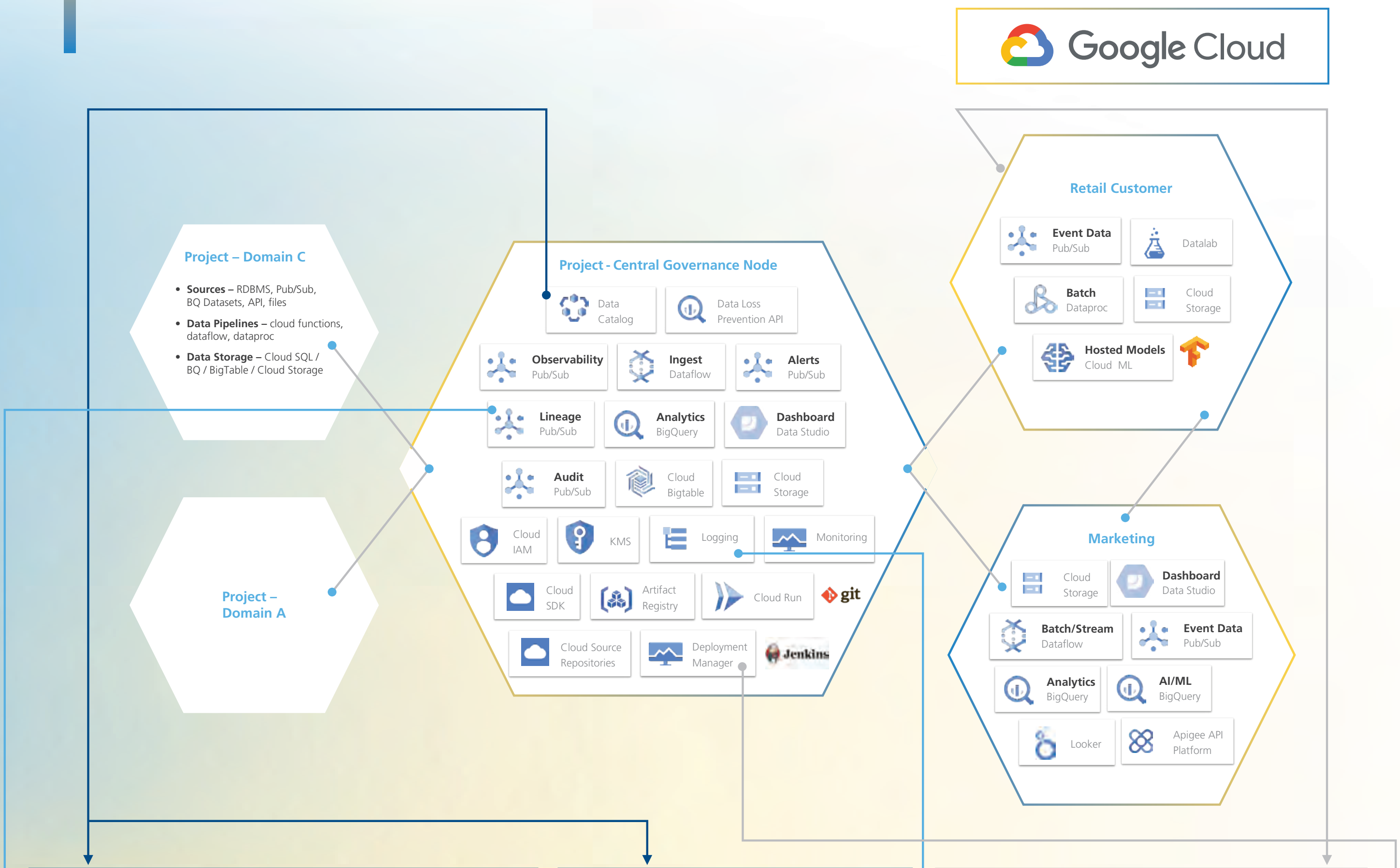
Organization of nodes in the mesh

GCP resource hierarchy consists of Organization, Folders, Projects, and resources. Projects in GCP are basic blocks for using Google Cloud resources. Henceforth, in the document node and GCP project is used synonymously to mean the same. Projects that make a clear distinction provide separation of concerns in terms of resources, security, and billing. Each project has a billing account attached to it. Charges billed to the attached account as per consumption of the services.

We are assuming that the projects and services are within a particular region and zone. It is possible to have a data mesh spanning multiple regions. However, there are disadvantages such as data transfer costs, latency, and cross-country data transfer regulations. It is better to have a dedicated mesh per region (one per environment) preferably than one global mesh with nodes spanning multiple regions.

Multiple environments (Dev, QA, UAT, and PROD) for the same domain node will have different GCP projects configured to environment-specific central node/project. A domain has complete control over its project, with owner, admin, and editor roles. The security controls available for project-to-project communication are Cloud IAM Authentication and Authorization, User Identities and Service Identities, and VPC Service controls to restrict traffic across projects for services with public endpoints. By default, every node will have a default set of access privileges for the central project during instantiation.

Reference Architecture on GCP



Data Catalog

> Automatic technical metadata collection
> Business metadata via predefined tag templates
> Default all datasets across domains published
> Domains can restrict based on sensitivity
> DLP to categorize data for PII

> Federate Governance Board
> Data Stewards
> IAM Roles & Service Principals
> Security / Network Controls

> Well defined interfaces for interoperability
> APIs / BQ Authorized Views / Flat Files / Messaging Systems

Observability / Lineage / Audit

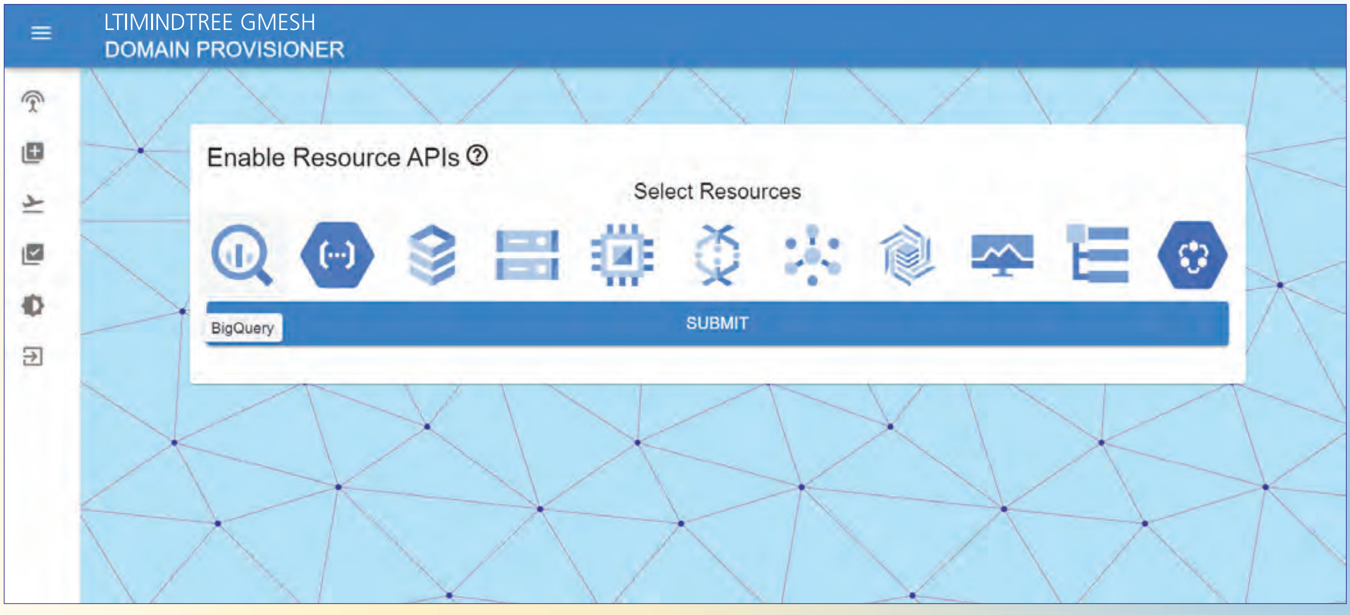
> Each domain sends observability/lineage/audit messages to global Topic respectively
> Messages are processed by dataflow ingested to BQ
> Real time alerts via pub/sub topic subscription on observability / lineage / audit

> Centralized Operations
> Logging across multiple projects
> Monitoring and alerting across projects

> Self-service Infrastructure Portal
> Cloud Deployment Manager / API for provisioning domain node in the mesh
> Default opinionated stack for domain
> Default templates for business metadata
> Default access to Catalog/linage/central Pub/Sub

Self-service infrastructure

One of the core tenants of data mesh is to provide a self-service infrastructure provisioning system for domain teams. LTIMindtree's GMESH provides the blueprints for provisioning opinionated infrastructure as outlined in previous sections. Using GMESH and a self-service UI, a domain can deploy complete nodes or specific services to their domain/project along with the default service accounts/IAM Roles and groups. These permissions and roles help link with central node infrastructure. A domain will have admin/owner rights to its project, and the project owner/admin will be able to provision resources directly. The linkage to central node/project services requires manual wiring. Using the self-service UI accelerator such as LTIs GMESH to provision the resources wiring to the central governance node is automated adhering to security controls and standard policies.



Central Governance Node

A central node/project governs and manages services common to all the domains in the mesh.

The personas accessing the central node are:

- Data Product Owners
- Data Governance Stewards
- Data Engineers
- Infrastructure Engineers

The governance node manages the following services common to all domains:

- Data Catalog
- Data Lineage
- Data Sensitivity
- Auditing
- Logging
- Monitoring and Alerting
- Security

Data Catalog

A central catalog(3) is used in a data mesh to enable and implement the FAIR principles (Findable, Accessible, Interoperable, and Reusable) of data. Data Catalog in GCP provides infrastructure and services for capturing technical metadata (automatically) and business metadata (manually using tag templates). GCP Data Catalog can integrate on cloud and on-prem data assets.

The primary personas using catalog are:

- Business Users
- Data Engineers
- Data Stewards

In GCP for a domain-specific node/project, the data Catalog service automatically catalogs the following assets.

- BigQuery datasets, tables, views,
- BigQuery external tables in Cloud Storage, Cloud Bigtable, or Google Sheets
- Cloud Pub/Sub message topics
- Databases and tables in Dataproc Hive meta store.

Using IAM permissions and service accounts domain chooses the datasets and GCP services that can automatically publish metadata to the central data catalog. For instance, a domain with RAW, STAGING, and CONSUME layers in BigQuery can publish the metadata across the three layers to the central node or publish only the CONSUME layer datasets keeping the flexibility and control within the domain.

With GCP data catalog all users who have access to data will by default have access to metadata. In general, the recommendation is to open the catalog to the entire organization with metadata from all domains by default and only restrict if there are any data-sensitive requirements.

Data Lineage

Data mesh requires a centralized standard data lineage system(4).

The three primary personas using the lineage system are

- Business users for data validation, business rules, and data discovery.
- Data engineers for data observability, data quality detection, troubleshooting, and remediation.
- Data stewards for ensuring that each node in the mesh complies with the central standards and practices.

The central governance board consultation with the domain team defines the tag templates required for lineage information.

Every node in the mesh should adhere to and mandatorily populate the central data catalog using the data catalog API. All the nodes during the instantiation will have required access to publish lineage information. The tag templates will continue to evolve with more use cases onboarded to the data mesh, so having a template version is critical.

Note: GCP Cloud Dataplex service(5) captures most of the information mentioned along with AI/ML intelligence capabilities that avoids multiple services. Dataplex is a new product with a preview release at the time of this whitepaper going for publication.

Domain Nodes

Each domain-specific node is associated with a project in GCP, which will enable easy management of billing, tracking, usage, APIs, security, and access permissions. In addition, projects will have users and service identities to manage across different domains. Domain teams own and manage domain specific GCP projects in the mesh. The central infrastructure team provisions the resources inside the node based on the type of the node.

Domain Consumers

Each of the domains chooses to support one of the following interfaces for downstream consumers.

01 BigQuery Datasets

BigQuery supports authorized views. Authorized views make data sharing secure and easy. Authorized view lets share query results with a set of users/groups. IAM groups provide access to datasets across projects. The storage costs are incurred by the domain sharing the datasets and the query costs by the domain accessing the datasets.

Note: GCP Cloud Dataplex service(5) captures most of the information mentioned along with AI/ML intelligence capabilities that avoids multiple services. Dataplex is a new product with a preview release at the time of this whitepaper going for publication

02 Serverless RESTful APIs

There are several ways to develop APIs on GCP, and this whitepaper will describe two of them.

- **Looker APIs with Apigee endpoints**
  LookerML semantic model on top of BigQuery datasets and Looker APIs on top of them. Looker APIs are further attached to apigee endpoints for data APIs over BigQuery (or CloudSQL datasets)
- **Traditional API development**
  For traditional API development, GCP supports a variety of solutions. In this paper, we consider serverless API development with Cloud Functions, Apigee, and CloudSQL as database

03 File Drop boxes

There are scenarios where each domain has to make available scheduled data extracts in cloud storage either for internal consumer nodes / downstream systems or external vendors. The following scenarios need to be supported.

- **File extracts for internal consumers**
  Each domain that shares the extracts has its own IAM group with required identities (users/service accounts). Internal consumers have restricted access, enforced with IAM groups across domain nodes. Only one bucket is created for all the users in that group.
- **File extracts for external vendors**
  Each of the domains needs to share extracts to entities/vendors external to the organization. In this case, the best practice is to create one bucket per vendor during the vendor onboarding process. Use IAM groups and roles to give specific bucket access to vendor users. This method assumes vendors are also on GCP and has access to Google Cloud Platform and users have google accounts.

- **One time / limited-time access for an object**
  GCP cloud storage allows signed URLs to give one-time or limited access to an object with or without a google account required for access to GCP.

03 Pub/Sub Topics

Cloud Pub/Sub service enables the exchange of real-time event data with other domains. Pub/Sub allows access control at the project level and the resource level. Cross project communication(7) is accomplished using IAM and service accounts.

Data Sharing

Secure data sharing is of paramount importance in a data mesh. The data-sharing service should help locate internal and external datasets, securely publish, or consume data and analytical models, and provide ways for charging back data asset consumers.

Roles and Responsibilities

Roles in Central Node

- **Data Engineers –** Implement standard data pipeline templates reused across domain nodes.

- **Cloud Infra Architect/Admin –** Responsible for implementing, managing, and administering cloud infra resources and security for the central domain and domain-specific nodes.

- **Cloud Architect/Engineer –** Responsible for implementing applications using cloud services including DevOps experience.

- **Data Stewards –** Responsible for the overall management of data and metadata across the data mesh and administer in accordance with organizations' policy and regulatory requirements.

- **Data quality analyst –** DQ is important for decentralized architecture such as Data Mesh, while data stewards are business-oriented, data quality analysts are more technical. DQ analysts' responsibility is to make sure the data across the nodes is to a level of the desired standard, build tools and dashboards for continuous monitoring, and take remediation action when required.

Roles in Domain Node

- **Data Engineers –** Responsible for implementing data pipelines for the domain in accordance with federated governance standards, best practices, and quality standards.

- **Domain Owner –** Domain Owner is the business owner of the domain and is responsible for the existence of such domain.

- **Data Product Owner –** Responsible for designing and implementing data-driven products.

- **Cloud Engineer (Full-Stack) –** Responsible for implementing applications using cloud services including DevOps experience.

- **Data quality engineer –** Responsible for using the global templates and standards and implementing domain appropriate DQ solutions and checks.

The engineering managers and scrum masters are a common requirement for the central node and the domain nodes.

Summary

Data Mesh is a paradigm shift in the way we think about centralized data teams and monolithic data platforms. It's important to note that the precursor to embarking on data mesh patterns is to produce enterprise domains. Choosing enterprise-level domains is a critical task that organizations must do before implementing data mesh architectural patterns.

The next crucial step in this process is to create a self-serve data infrastructure that abstracts out the functionality common to all data domains. Google cloud platform with its smart analytics solutions makes this daunting task simple and helps overcome technical challenges bringing all nodes together in the mesh. LTIMindtree's GMESH accelerator for data mesh architecture simplifies the process of enabling centralized infrastructure and reduces overall timelines with automated deployment pipelines.

References

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

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

https://hbr.org/2018/10/how-to-build-great-data-products

https://cloud.google.com/data-catalog/docs/concepts/overview

https://cloud.google.com/architecture/architecture-concept-data-lineage-systems-in-a-datawarehouse

https://cloud.google.com/blog/products/data-analytics/introducing-google-cloud-dataplex

https://cloud.google.com/blog/products/data-analytics/introducing-analytics-hub-for-dataanalytics- Exchanges

https://cloud.google.com/pubsub/docs/access-control#sample_use_case_cross-project_communication

About the Author

Vijaya Phanindra
Principal - Data Engineering
LTIMindtree

Vijaya leads the GCP data and analytics technical centre of excellence at LTIMindtree. He has over 17 years of industry experience in data analytics and has been working with our enterprise clients in financial services, technology, manufacturing, and pharma verticals. His core expertise is in solution architecting, consulting, and technical implementation of big data & analytics engagements. Vijaya holds a master's degree in computer science and has done GMITE management program from IIM-Bangalore.