Components and Functionality

Table of Contents

Introduction

Purpose of the Document

This wiki page serves as a comprehensive guide outlining the components and functionalities of the Data Ocean Architecture.

Following the structure proposed in the "Reference Architecture," this document will delve into how the architecture supports a scalable, organized, and secure system for handling a variety of data needs across the organization.

Target Audience

This document is intended for multiple audiences within the organization, including but not limited to:

- **Data Engineers**: For understanding the workflow and where they contribute to the architecture.
- **Data Scientists**: To comprehend how to access and interact with the data for analytical purposes.
- **Business Analysts**: For knowing how the data flows and where they can extract the information they need for reports and dashboards.
- **Data Architects**: Who are responsible for the overall structure and integrity of the data environment.
- **Technical Business Users**: To gain insights into what data is available, how to access it, and under what conditions.
- **Data Governance Teams**: For ensuring that the organization of data aligns with company policies and standards.

2. Data Flow

The Data Ocean Architecture is a sophisticated and comprehensive framework that facilitates efficient data processing and information dissemination within the organization.

This chapter delves into the intricate data flow within the Data Ocean Architecture, drawing insights from the Reference Architecture.

2.1. Overview

The architecture is composed of multiple interconnected blocks, each serving a distinct purpose in the data processing pipeline.

The Data Ocean Reference Architecture is designed to facilitate the smooth movement of data across various layers and components. The data flow within the Data Ocean Architecture traverses through a series of logically connected blocks, each contributing to the transformation of raw data into valuable insights. These blocks are strategically designed to ensure modularity, data integrity, and scalability.

This movement is guided by a well-structured flow that ensures data availability, quality, and accessibility throughout the data pipeline, ensuring timely information dissemination, serving diverse business needs within the organization.

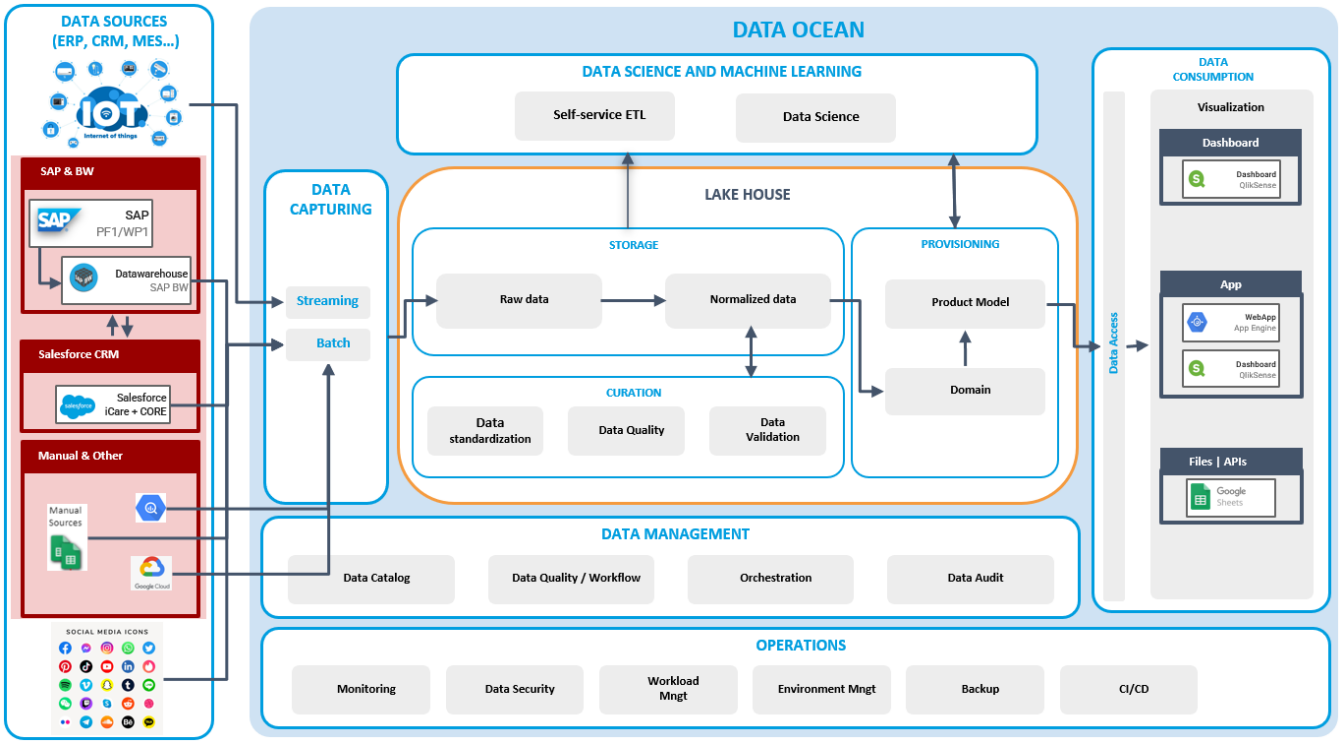
The Data Flow within the Data Ocean Reference Architecture is structured to ensure efficient data processing, emphasizing the importance of improving how data is extracted, transformed, and loaded (ETL), using technology effectively and to its fullest potential. At the same time, it highlights the crucial need for maintaining control over data, making sure data persists reliably, and ensuring data remains accessible throughout the entire process.

2.1.1. General Requirements: ETL Process Reliability and Data Consistency

1. **Reliable Repeatability**: ETL processes must prioritize achieving a high level of reliability and repeatability. This focus ensures consistent and dependable data handling across all stages of the process.
2. **Data Integrity and Consistency**: The ETL process is obligated to uphold data integrity and consistency throughout its execution. At no point should the process allow data to become inconsistent, ensuring the highest quality of data remains maintained.
3. **Transactional Execution**: ETL processes must operate in a transactional manner. This means that the process should either succeed entirely, resulting in complete data transformation and loading, or it should fail completely, leaving no partial or erroneous data in its wake.
4. **Criticality for End-User Data**: The significance of adhering to transactional execution becomes even more pronounced when handling data intended for end-user consumption. Allowing for partial or inconsistent data in such scenarios is unacceptable, as it could lead to misinformation or operational disruptions.
5. **Uncompromised Business Objects**: The ETL process must safeguard business objects from the risk of empty or inconsistent states. The execution of ETL activities should never result in a business object being devoid of meaningful data or falling into a state of inconsistency.
6. **End-User Facing Data**: ETL activities involving data exposed to end users must adhere to the highest standards of reliability and data consistency. This ensures that the data presented to end users is accurate, trustworthy, and aligned with business objectives.

By adhering to these requirements, the ETL processes can maintain a high degree of reliability, transactional consistency, and data integrity across various stages, ultimately contributing to accurate decision-making and seamless business operations.

For more specific and complete rules, please see Data Engineer Guidelines.



2.2. Key Components of Data Flow

The data flow within the Reference Architecture involves several key components, as described in the Reference Architecture, including:

- **Data Sources**: Both structured and unstructured data from various domains are extracted for processing.
- **Data Capturing**: Data is captured through batch and streaming processes.
- **Lake House Architecture**: Data undergoes storage, curation, and provisioning.
- **Data Science and Machine Learning**: Analytical processes are conducted on the data.
    - **Self-service ETL**: The architecture accommodates the capabilities; however, it does not explicitly endorse them due to their unique characteristics, which are closely intertwined with established best practices in Software Development and Architecture.
- **Data Management**: Data undergoes cataloging, validation, in an orchestrated way.
- **Operations**: Data security, workload management, environment management, and monitoring are applied.
- **Data Consumers**: Data is accessed by BI tools and portals.

Data Sources

This block encompasses all external data sources, such as databases, APIs, web scraping, and files from both internal and external entities. The architecture emphasizes and strongly recommends the direct access to these sources whenever possible, to maintain control and governance over data quality and security.

About Files

The company ought to make efforts to reduce its reliance on files. Numerous departments exhibit an excessive dependency on files, leading to an overwhelming proliferation of files throughout the organization.

Ingesting files can be harmful if they are not completely managed by a job within the context of a specific known extraction process.

Some files come from unknown or difficult-to-identify sources, which can lead to compromised data quality, inconsistent data formats, and unanticipated changes in file structures and layouts, adding to the difficulty of efficiently troubleshooting issues. These inconsistencies might cause errors throughout the data extraction, transformation, and loading stages, resulting in incorrect data processing, data loss, or incomplete data sets.
Also, inaccuracies that occur can have a domino effect, influencing downstream analytics, reporting, and decision-making processes. In the end, this can have an influence on data accuracy and reliability, as well as create security and compliance issues.

It is important to note that this strategy will weaken the integration process by adding instability and risks that can result in failures and disruptions in the ingestion pipeline. At the very least, concreate actions should taken for mitigating these issues and ensuring the resilience and robustness of the solutions.

The goal should be to have robust pipelines and processes.

Data Extraction

Data is extracted from diverse sources using ETL processes.

The data extraction process involves retrieving data from various sources as outlined in the Reference Architecture. It's crucial to adhere to specific time constraints to access internal business applications, ensuring minimal impact on business systems. These extraction jobs should prioritize simplicity and speed, operating without dependencies beyond agreed-upon timing and source system limitations.

The preferred approach is to extract all available data (full table), unless certain constraints like resource limitations or time considerations arise (if feasible, full data loads are preferred to maintain data integrity). While techniques like Change Data Capture (CDC) can be considered for obtaining deltas, triggers or control columns in source tables are generally not deemed secure.

Data Validation and Extraction Strategy:

If full data extraction is feasible and any potential impact is identified (such as dealing with large files), options for calculating a Delta are explored. This could involve comparing previous and current files before loading or utilizing platform functionalities post-loading. In the absence of restrictions or adverse impacts, the full data file is loaded onto the platform. Maintaining low data latency is essential, even though real-time or streaming use cases are not currently in play, although the Data-ocean solution should support these in the future.

Please refer back to the detailed recommendations provided in the section on Extracting Data from Known Sources for more comprehensive guidance on this matter.

Data Capturing and Ingestion

The extracted data is directed into Cloud Storage, which serves as a designated landing zone. This zone is composed of controlled buckets that effectively prevent duplication and guarantee the secure storage of error-free data.

The integration process ensures that exceptions or errors are properly managed.

At the moment, the architecture's primary emphasis is on managing data in groups at scheduled intervals, which we refer to as batch processing. This approach involves handling data in chunks at specific times. However, it's important to note that the architecture is built with flexibility in mind. This means that it's well-prepared to support streaming data as well, which involves dealing with data in a continuous flow as it arrives. While the architecture currently leans more towards batch processing, it's designed to smoothly accommodate streaming capabilities whenever they become relevant or necessary.

Streaming data, characterized by its continuous, real-time nature, holds immense value in scenarios where timely insights and rapid actions based on dynamic data changes are critical. Applications spanning real-time monitoring, Internet of Things (IoT) devices, social media sentiment analysis, financial transactions, and more can greatly benefit from the immediate processing of streaming data.

The modular nature of the Data Ocean Architecture ensures that streaming data pipelines can be integrated without causing disruption to existing batch processing flows. This adaptability showcases the architecture's forward-looking approach, positioning it to seamlessly embrace emerging technologies and evolving data processing requirements.

While not the primary objective, this layer does offer the option of adopting a standard Data Lakehouse approach as a potential solution.

RAW Block

Afterward, the data is loaded into the RAW layer. This layer keeps the data in its original format, which means it can handle different types of data, including structured, unstructured, and semi-structured data.

For a more comprehensive understanding, it is recommended to refer to Chapter 4, with a specific focus on the section that pertains to the subject of data organization.

Curated Data Layer

Data transitioned from the RAW layer to the Curated Data Layer, operating in tandem with the curation block, undergoing essential processes like normalization, validation, and integration. This block serves as a vital checkpoint, ensuring data accuracy, conformity to standards, and alignment with technical prerequisites such as integrity and quality.

In specific scenarios, Self-Service ETL and Data Science Tools have access to this layer for particular use cases.

For a more comprehensive understanding, it is recommended to refer to Chapter 4, with a specific focus on the section that pertains to the subject of data organization.

Provisioning

This block focuses on provisioning detailed data models in an environment tailored for specific use cases.

Self-Service ETL, Data Science Tools, and in special circumstances, technical users can access and interact with this layer.

Typically, these processes should operate at the Data Product (DP) level. However, if the outcomes are relevant to the entire enterprise and should be accessible across the organization, they could be made available at the Domain level.

Domain Data Models

The Domain Data Models layer represents the enterprise-level data models, oriented to Domain.

For a more comprehensive understanding, it is recommended to refer to Chapter 4, with a specific focus on the section that pertains to the subject of data organization.

Data Products (DP)

Data flows to the Data Products block, meaning that Data Products are crafted from Domain data, providing tailored insights for specific business/user needs.

Each Data Product operates independently, ensuring flexibility and minimizing dependencies.

Self-Service ETL, Data Science Tools, BI tools though presentation views, and technical business users can access and interact with this layer for actionable insights.

For a more comprehensive understanding, it is recommended to refer to Chapter 4, with a specific focus on the section that pertains to the subject of data organization.

Data Management

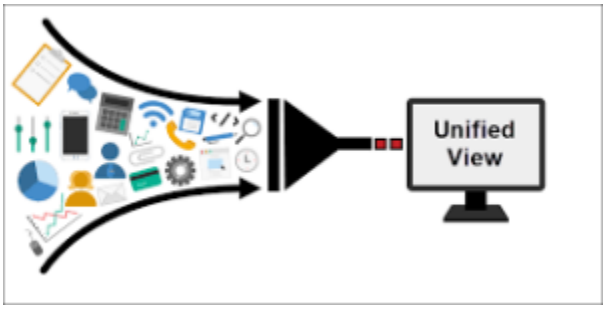
Data Management is out of scope of this document

Operations

Operations is out of scope of this document

3. Data Integration: Principles and Best Practices

3.1. What is Data Integration?



Data integration refers to the process of combining data from disparate sources into a unified and coherent view. It involves harmonizing data formats, structures, and semantics to ensure consistent and accurate information is available for analysis, reporting, and decision-making.

This chapter provides an overview of Data integration from a general and best practices standpoint. For a more comprehensive understanding, please consult the section on Data Engineer Guidelines.

3.2. Goals of Data Integration

The primary goals of data integration in the Data Ocean architecture include:

- Ensuring data consistency and accuracy across different sources.
- Facilitating seamless data movement between various layers.
- Providing a unified view of data for efficient analysis.
- Reducing data redundancy and ensuring a single source of truth.

3.3. Benefits of Effective Data Integration

Effective data integration offers several benefits:

- Enhanced data quality and reliability.
- Improved decision-making based on accurate insights.
- Increased operational efficiency through streamlined data processes.
- Faster access to integrated data, saving time and effort.

3.4. Best Practices for Data Integration

To achieve successful data integration within the Data Ocean architecture, consider these best practices:

- Define clear data integration goals and objectives.

- Establish data governance to ensure data consistency and quality.
- Utilize standardized data formats and semantics.
- Implement data transformation processes to harmonize data.
- Leverage automation tools and platforms for efficient integration.
- Continuously monitor and validate integrated data for accuracy.

3.5. Data Integration Challenges and Solutions

Challenges in data integration may include data silos, schema differences, and varying data velocities. Solutions include:

- Creating a data integration strategy and roadmap.
- Implementing data virtualization to access data without physically moving it.
- Adopting data cataloging and metadata management for better data discovery.

3.6. ETL Process Restructuring and Autonomy

Several key principles guide the restructuring of ETL processes to achieve greater autonomy and efficiency:

- Extraction processes are designed to be independent of other pipeline processes and feed into the RAW and subsequent layers.
- Transformation processes are decoupled from Extraction and Loading, enabling execution at any time of day for data preparation.
- Loading processes for final EDW data models or Data Products are restructured to be independent of Transformation processes.
- Extraction processes can be autonomously executed and shared across multiple downstream chains.
- Transformation processes can operate independently of Loading processes, allowing heavy data preparation whenever needed, with data available in the EDW as required.
- Load and data refresh occur by domain and are independent processes.
- The ETL processes are restructured to be independent of other data pipeline processes, promoting autonomy, reusability, and process efficiency.

3.6.1. Idempotent ETL

The aim is to design ETL processes to be idempotent, ensuring that if the same operation is executed multiple times, the outcome remains consistent and doesn't produce unintended side effects.

3.7. ETL Management and Monitoring

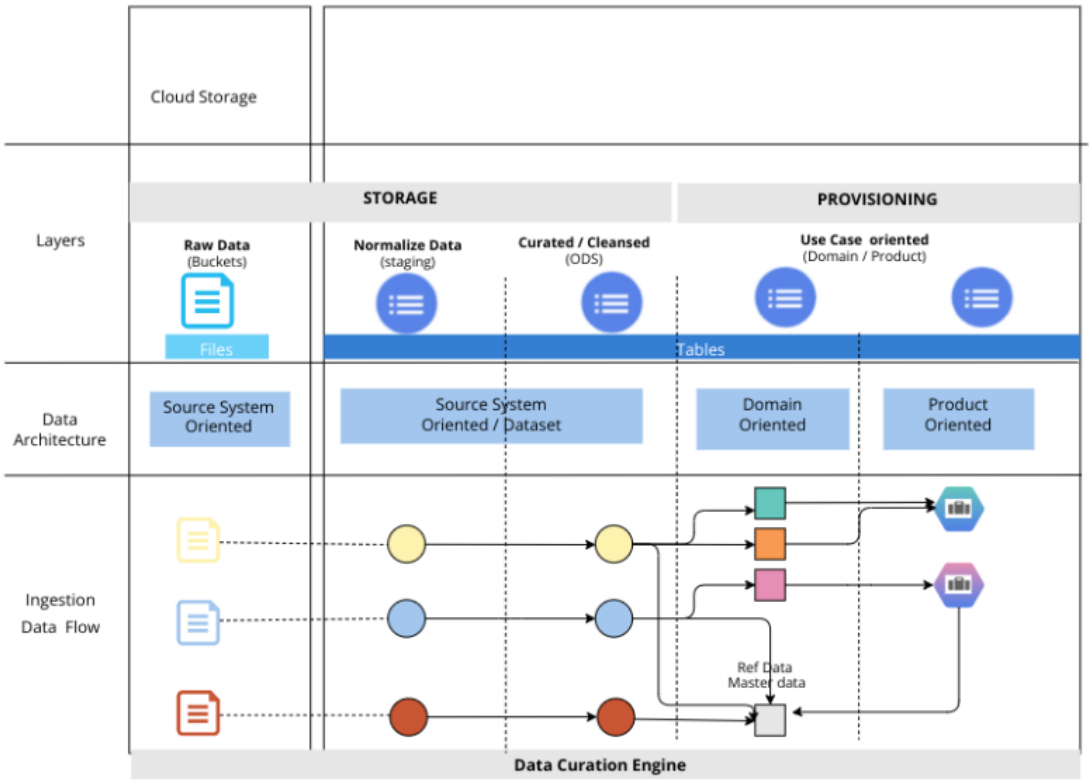
The restructured ETL processes operate under orchestration, generating audit information. This information can be used to obtain job status, supervised under operations, in a monitored and secure environment.

4. Data Organization

Data Organization serves as a crucial element of the Reference Architecture, designed to align data processes across various business units effectively. Th is solution is structured around a layered approach, demarcating distinct stages of data processing, all orchestrated through the prism of transversal Domains or subjects.

The architecture's structure adheres to a Domain-Driven approach that aligns well with Solvay Atom's transformation goals and fosters data culture and accountability.

This chapter outlines the core components, the layered architecture, and the nine specific business Domains.



4.1. Architectural Layers

The reference architecture delineates a coherent framework composed of five distinctive layers. The Data Ocean architecture includes several layers to manage the complexity and demands of a data-driven organization. Each layer signifies a significant phase in the data journey, spanning from data ingestion to the creation of actionable insights:

4.1.1. Raw Data Layer (Ingestion Layer)

Also known as the Landing Area, this layer is the initial repository where data is ingested swiftly and efficiently. Data retains its native format at this stage, with no transformations applied, which ensures that the data can serve as a point-in-time archive.

The layer is hierarchically organized based on subject areas, data sources, and time of ingestion.

Access to this layer is restricted to prevent unauthorized or incorrect usage.

4.1.2. Normalized Data Layer (Staging - STG)

This layer serves as an intermediary to enhance performance in transferring data from the Raw layer to the Curated layer. The data is loaded in near raw format, primarily used for layout validation, basic data checks and control, housing data that is not yet ready for direct consumption.

Complex files, such as XML or JSON, are not processed in this layer.

To simplify data access, a relational layer is in place, where, for simple file formats, such as CSV files, a direct one-to-one conversion to table format is established, with additional control information and metadata. For more complex files, they are just included into a table format with specialized column data types, supplemented by the same additional control information and metadata.

4.1.3. Curated / Cleansed Data Layer (ODS)

Also referred to as the Conformed Layer, Curated Data is transformed into consumable datasets, structured for specific purposes and possible partitioning to a more granular level. Cleansing, validation, and normalization may be applied to enhance data quality. Preliminary transformations often take place at this stage to ensure high-quality, consistent data.

Within the Data Curation layer, the data undergoes a series of cleansing and transformation activities. These processes encompass vital steps including normalization, validation, and integration. The primary objectives are to ensure data accuracy, adherence to standards, and alignment with technical prerequisites such as data integrity and quality.

The comprehensive treatment of data involves several tasks. These tasks comprise standardizing dates to a consistent format, aligning string values for uniformity, validating and generating keys for efficient linking, and resolving complex structures, such as those found in JSON or XML hierarchies.

Despite the current absence of Data Quality verification within this stage, as it lies beyond the current scope, the intention remains to integrate these improvements into the Data Curation layer in the foreseeable future. It is worth noting, however, that an independent project is already actively addressing this particular aspect.

For more in-depth information, please refer to the section on Data Curation.

It's noteworthy that in cases where simple file formats like CSV are utilized, they maintain a direct one-to-one correspondence with their respective source data. However, when dealing with intricate file formats, a one-to-many relationship can emerge between the source and their table formats, due to the complexities involved.

Where possible, the curated data follows the same column naming convention as the source data.

In specific scenarios, for particular use cases, operational reporting could be created, when a more agile and expedient solution is required, or when it simplifies complexity and reduces the strain on operational system resources. Importantly, this alternative approach is designed to minimize any potential impact on operational systems.

4.1.4. Use-Case Oriented Layers (Domain/Product)

These specialized layers apply additional business logic to the data. They source data from the Cleansed layer and are enforced with any needed business logic or security measures. Data models to address business analytical requirements are also created here.

4.1.4.1. Use-Case Oriented: Domain (DM)

This layer presents a central enterprise-level repository, structured as per specific Domains or subjects. Each Domain represents a key area of business relevance. It acts as an Enterprise Data Warehouse (EDW), housing **subject-oriented, integrated, time-variant, and nonvolatile collection of data that serves as a single version of truth for the organization**, adhere to a Data-oriented philosophy, reflecting the true and real relationships among operational entities, mapping business entities closely to operational systems, remaining independent of particular user needs and independent of changing business requirements.

These models, will be structured based on Dimensional Modeling principles in a snowflake flavor. This design doesn't prioritize performance, usage simplicity or cater to specific user/business prerequisites, instead, it accentuates data-oriented attributes and facilitate cross-domain analysis. This approach enhances the resilience of represented entities and the resulting Data Model, while concurrently reducing costs and efforts associated with the development and maintenance of Data Processes (ETL) and Data Pipelines.

For more in-depth information, please refer to the section on Data Modeling at Domain level.

4.1.4.2. Use-Case Oriented: Data Product (DP)

These are the final insights derived from the architecture. Data Products are subject-specific, optimized for performance and tailored to cater to particular user or business needs.

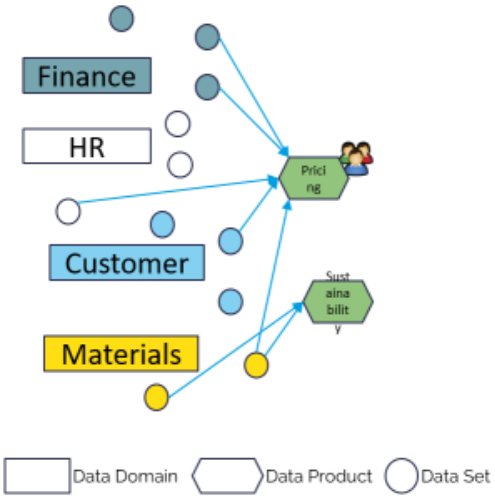
This layer facilitates informed decision-making and empowers users with actionable insights.

The data model should be in aligned with the needs of Data Visualization Teams and adhere to any particular constraints associated with the BI tools employed. It is advisable for the model to be straightforward, easily comprehensible, and designed for performance and specific user/business requirements.

Each Data Product operates independently, ensuring flexibility and minimizing dependencies.

Data is served though presentation views to the BI tools, providing tailored insights for specific business/user needs.

For more in-depth information, please refer to the section on Data Modeling at DP level.



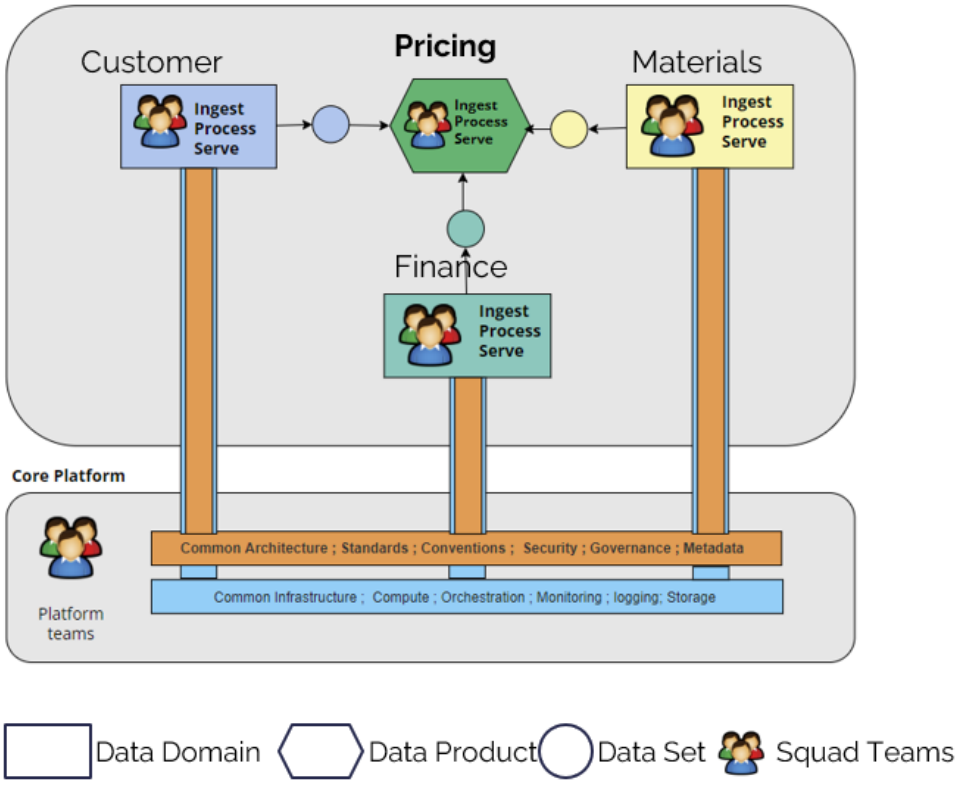
4.1.4.3. Use-Case Oriented Sandbox

This optional layer is proposed for advanced analysts and data scientists to conduct experiments.

This optional layer is suggested for advanced analysts and data scientists to run experiments. It's a space where they can conduct tests to find patterns, correlations, or validate machine learning models.

It can also serve the purpose of conducting complex analyses for technical Business Analysts who are restricted from directly altering Data Ocean Schemas - no one except the project initiatives within the scope of the Data Ocean implementation is permitted to create, modify, or write on Data Ocean Schemas.

4.2. Domain-Driven Architecture

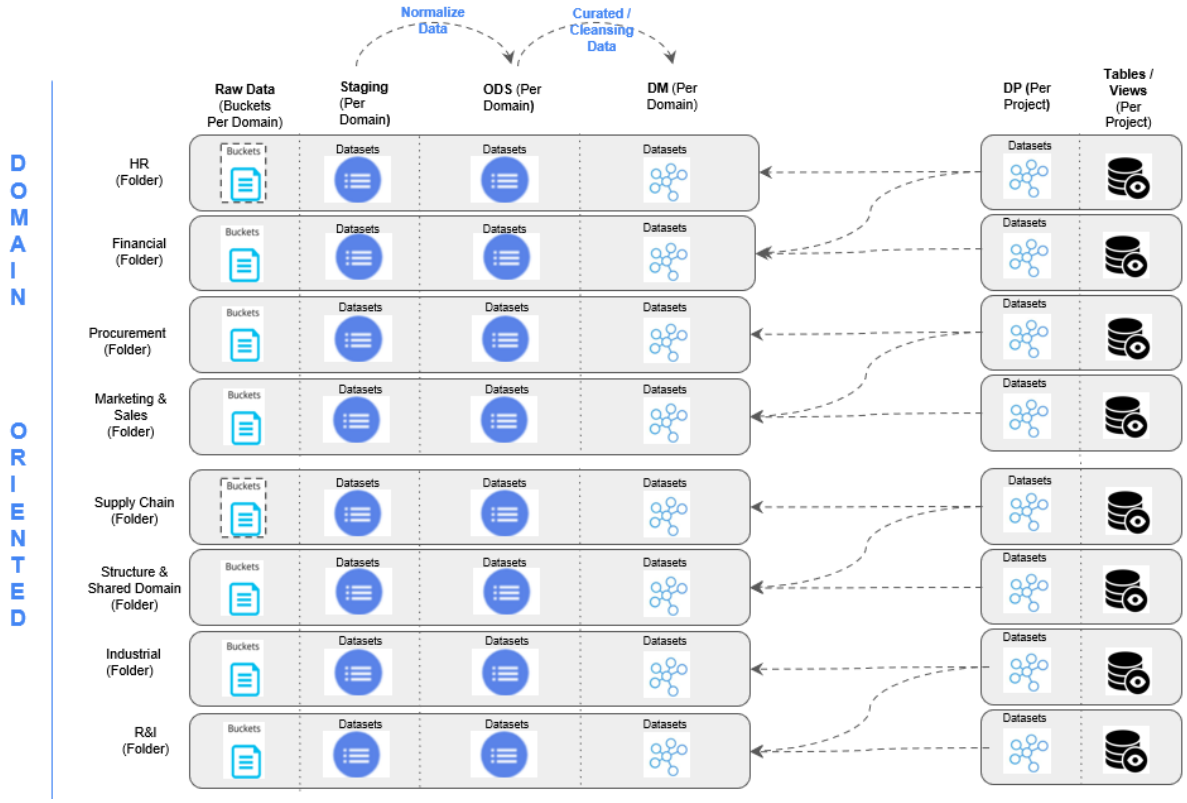


Domain-Driven Architecture is an approach that focuses on the core domains within the organization to design and build scalable, maintainable, and governed data sets.

The data organization within the architecture is designed around nine business domains:

4.2.1. Business Domains

1. **HR (Human Resources)**: Focuses on employee data, covering aspects like recruitment, payroll, and performance metrics.
2. **Procurement**: Centralizes data related to vendor management, contracts, and procurement cycles.
3. **Finance**: Manages financial records, including budgets, income, expenses, and other fiscal reports.
4. **Marketing & Sales**: Addresses customer interaction data, sales metrics, and market analysis.
5. **Supply Chain**: Deals with logistical data concerning supply chain management, inventories, and distribution.
6. **Structure & Shared Domain**: Contains data shared across various business units and aspects related to the organizational structure.
7. **Industrial**: Houses data related to manufacturing, equipment health, and quality control.
8. **(Research & Innovation)**: Maintains data on R&D projects, patents, and scientific research.
9. **Sustainability**: Oversees environmental responsibilities, sustainable business practices, and social governance.



4.2.2. Additional Domains

- **Common Domain**: Stores technical information and metadata.
- **Default Domain**: A catch-all domain that accommodates data that doesn't fit into other specified domains.

4.2.3. Domains and Classification

Domains serve as a classification framework established by Data Governance to encapsulate crucial business aspects. With nine business domains, two additional domains accommodate technical and miscellaneous data. Each Domain corresponds to specific business units and can also be likened to Domain-Driven Design (DDD) Domains.

- **Holistic Data Coverage**:
  - Domains encompass data pertinent to their respective subjects.
  - The technical Domain caters to metadata and contextual information, while the miscellaneous Domain holds data that does not fit within specific Domains and is universally relevant.

4.2.4. Domain Responsibilities

Each domain has responsibilities related to its function, ranging from data collection to decision-making, implementation, and assessment of policies and practices.

- Domains are responsible for maintaining quality datasets.
- Each domain must ensure their data meets specified standards such as being discoverable, addressable, and trustworthy.
- Each Domain has an associated Data Architect - Each Domain is overseen by an appointed Data Architect who holds technical responsibility for the associated datasets. This role involves defining Data Models specific to the Domain and providing assistance for ongoing and new initiatives.

4.2.5. Roles Within Domains

- **Data Product Owner**: Responsible for consumer satisfaction, quality of the domain datasets, and overall data lifecycle management.
- **Data Team**: Focused on platform enhancements, monitoring, automation, and alerting.

4.2.6. Value and Benefits

- Centers data acquisition, processing, and serving with domain experts
- Decreases common data pain points like data cleansing and orientation
- Supports the emergence of a data product focus

4.2.7. Capabilities and Infrastructure

The architecture is equipped with:

- Scalable, secure, and governed storage
- Encryption standards
- Metadata management
- Data pipeline orchestration
- Unified Access Control
- Monitoring, alerting, and logging
- Self-service capabilities

4.2.8. Governance and Team Structure

- Aims to reduce duplication of effort across domains
- Provides essential shared services and tools
- Focuses on delivering value while adhering to security and governance protocols

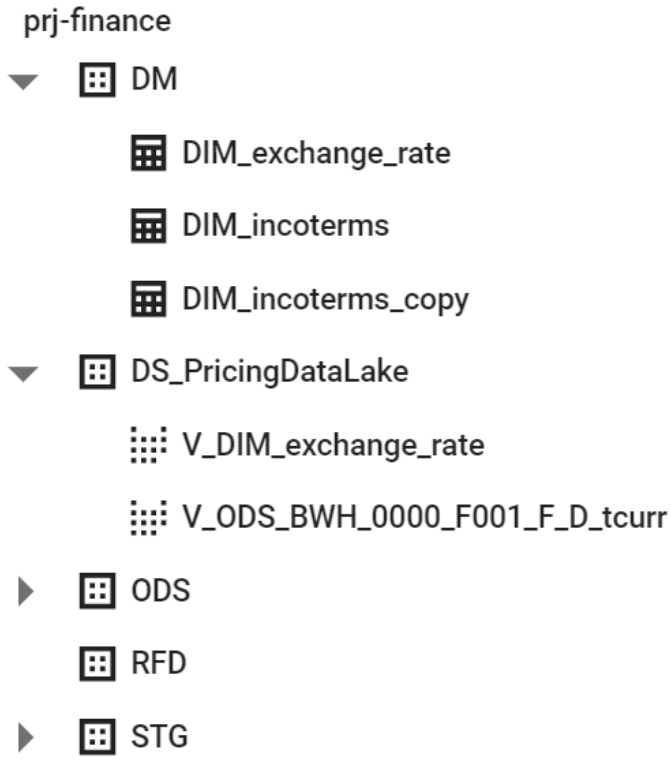
4.2.9. Domain-Centric Approach

Within each Domain, the architecture embraces a standardized structure, implemented through distinct schemas:

1. **Staging/RAW (STG)**: Raw Data ingested from various sources is stored here in its original form. This layer is crucial for quick validation and control, housing data that is not yet ready for direct consumption.
2. **Curated (ODS)**: Data transitions to the Curated Layer, where it's prepared for consumption. Cleansing, validation, and normalization may be applied to enhance data quality.
3. **Domain Models/EDW (DM)**: The Enterprise Data Warehouse (EDW) embodies this layer, featuring comprehensive data models that adhere to a data-oriented philosophy. These models, structured based on Dimensional Modeling principles, facilitate cross-domain analysis.
4. **Working Data Layer (WDL)**: This schema accommodates temporary tables and objects that support specific analytical or operational processes.
5. **Data Presentation Layer (DPL)**: Intended as an abstraction layer, the DPL is manifested through views that separate data usage from its physical representation. These views ensure security, access control, and a consistent data consumption experience. DPL introduces abstraction, enabling users to interact with data via views that conceal underlying complexities. By decoupling data usage from its representation, this layer enhances user experience and security, allowing controlled operations within designated environments, namely point the views to a alternative location, while solving an issue.
6. **DS_<DP_Identification>**: These schemas ensure isolation and control for Data Products (DPs). They are designed for specific Data Products, granting them a secure and controlled environment to operate within.
7. **Reference Data (RFD)**: This schema holds common reference data, fostering consistency and shared understanding across the organization. Vie ws are employed to facilitate access.
8. **DS_APP_<Application Code>**: This schema accommodates configuration information tailored to specific web applications, ensuring a smooth consumption experience.

For further information and naming convention, please refer to Data Architect - Naming Convention.

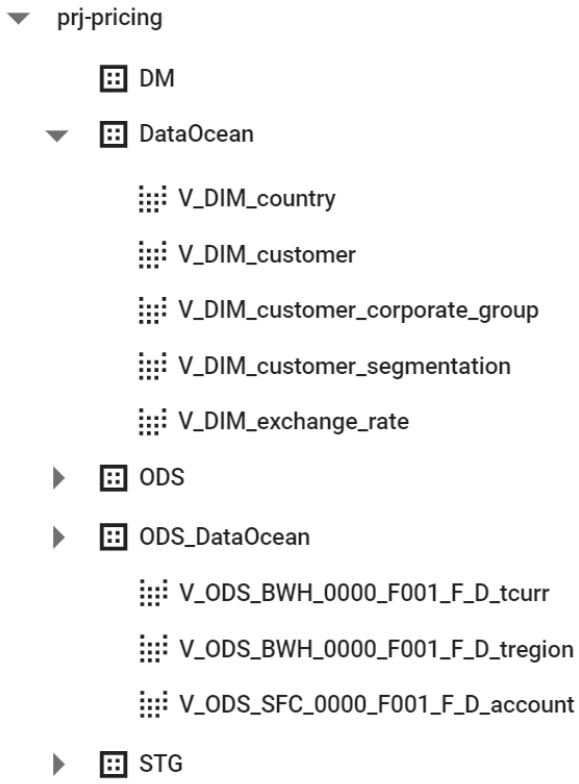
An example - Finance



Explanation:

- For the Domain Finance, in this example, the standard schemas are visible.
- Finance is exposing to the Pricing Data Lake two views
  - A Dimension called DIM_exchange_rate
  - a table from ODS called BWH_0000_F001_F_D_tcurr
- The Finance domain comprises three distinct data objects, for which it holds the responsibility.
- at the Domain level, using RFD doesn't make sense

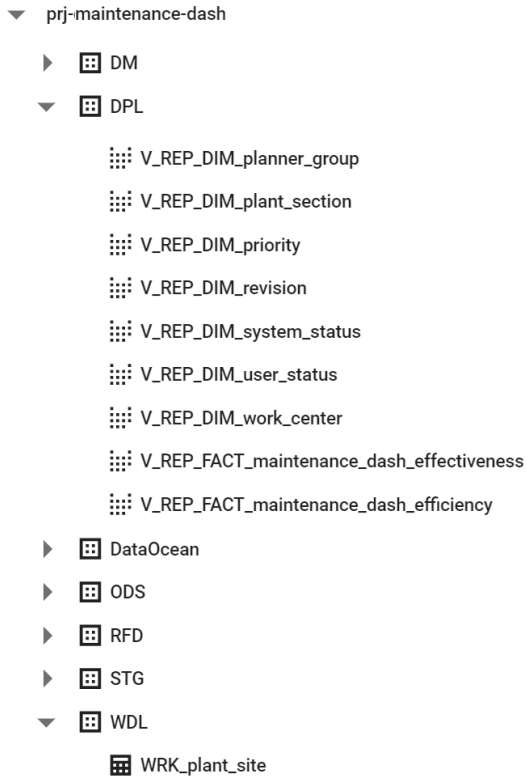
An example - Pricing data Lake



Explanation:

- For the Pricing Data Lake, in this example, the standard schemas are visible.
- Pricing has not established ownership of any specific Data Object.
- Pricing is using 5 Data Objects from the Data Ocean (at this level, the domain itself is not pertinent, as the responsibility for managing access and security does not rest with Pricing).
- Pricing is using 3 Data Objects from the ODS/Curated Layer of the Data Ocean (at this level, the domain itself is not pertinent, as the responsibility for managing access and security does not rest with Pricing).

An example - Data Product Maintenance Dashboard



Explanation:

- For the Data Product Maintenance Dashboard in this example, the standard schemas are visible.
- The Data Product Maintenance Dashboard is exposing to the BI tool 7 dimension and two fact tables
- The Data Product Maintenance Dashboard is using a temporary table is processing jobs
- The Data Product Maintenance Dashboard my have some references for DataOcean and for Reference Data
- The Data Product Maintenance Dashboard is a Data Product, but also can load external files directly, if they are specific for the product. As in a Domain, external data may be load in Cloud Storage and further processed (to STG, ODS and finally DM)

4.2.10. Data Access and Usage

The architecture offers carefully orchestrated access to data products:

- **Views for Data Access**: Data Products interact with data through views, granting controlled access to data in Domains. These views enable column masking, filtering, and structural concealment to ensure data security.These views provide a comprehensive data consumption experience.
- **Domain-Specific Schemas**: Security and isolation are upheld by associating each Domain with specific schemas, granting tailored access to technical users and authorized stakeholders., granting selective access based on user roles.
- **DS_DataOcean Schema**: Simplifying data access, this schema hosts views that grant a seamless interaction with data, negating the need to provide full table or view names.expedites data interaction, simplifying access by hosting views that circumvent the need for verbose table or view references.

4.2.11. Data Products (DP)

The Data Products (DP) in the Data Ocean architecture hold a central role in transforming raw data into actionable insights. DP is envisioned as an isolated entity with a dedicated focus on generating tailored solutions for specific business needs.

This isolation principle extends to ensure that no DP relies on another DP, or vice versa, thus establishing a modular and self-contained ecosystem.

- **Dependency Isolation**:
    - The DP's autonomy is paramount. Each DP can be quickly dropped, created or modified, not impacting other systems.
    - It can draw its insights from a simple query on top of existing Domain entities.
    - In certain instances, a DP may require sources distinct from the Domain. These sources should be exclusive to that particular DP, minimizing any potential impact on other DPs or Domains.
- **Promotion to Domain Level**:
    - Entities within a DP may evolve to become relevant to other DPs or Domains.
    - In such cases, these entities are "promoted" to the Domain level. However, this promotion's intricacies should be carefully managed at the chain scheduling level to ensure no disruption to existing processes and Domain load chain scheduling.
- **BI Tool Interaction**:

    - DP interacts with Business Intelligence (BI) tools through presentation views, ensuring an abstraction layer that decouples technical complexities from the user experience.
    - BI tools exclusively access presentation views using designated service accounts, preserving security and optimizing data usability.
- **Secure Access and Sandbox**:
    - Technical business users and analysts can access DP with specific user privileges, adhering to established security procedures.
    - Any modifications or new object creations within DP are strictly prohibited. For experimental purposes, a sandbox is available, materialized through a dedicated schema with limited quotas and periodic data purging.

4.2.12. Special Case: Domain Data Lake (DL)

In cases demanding more complex and expansive solutions, a distinct form of DP emerges—the "Data Lake" (DL). This special DP addresses intricate business requirements and projects on a larger scale. Referred to as the "<area name> Data Lake," it follows a star schema data model to accommodate diverse data sources.

- **Isolated and Self-Contained**:
    - Like regular DPs, a DL is self-contained, avoiding dependencies on other DPs or Domains.
    - It retains its own unique sources, separate from the Domain.
- **Dependency Management**:
    - Similar to DPs, dependencies stemming from a DL to other DPs or Domains necessitate careful management within the chain scheduling framework.
- **DP Reliance on DL**:
    - DPs that rely on a DL derive their data exclusively from this DL, maintaining a clear boundary between the two.

5. Data Modeling and Security

The architecture employs meticulous data modeling techniques across its layers:

- **RAW Layer**:
    - In the RAW layer, tables retain RAW layout or file handler information, along with control metadata. Tables here are named consistently with source nomenclature.
- **Curated Layer**:
    - Curated tables mirror the operational layout or file, enriched with control metadata. Like the RAW layer, naming conventions adhere to source nomenclature.
- **EDW/Domain Layer**:
    - The Domain layer features a star-schema with a hint of snowflake design where feasible.
    - A focus on data orientation, true data representation, and dimensional modeling ensures that the model captures the essence of source data. Independence from specific business requirements is crucial, fostering adaptability and flexibility.
    - For more in-depth information, please refer to the section on Data Modeling at Domain level.
- **DP/Data Mart**:
    - DPs exhibit varied modeling approaches based on simplicity, performance, and usability.
    - These can include One-Big-Table, OLAP, or Simple Star-Schema designs.
    - Entities that prove valuable within a DP can potentially transition to the Domain level through a careful promotion process.
    - For more in-depth information, please refer to the section on Data Modeling at DP level.

6. Security and Access

The architecture aligns security measures with the specific needs of each domain and data product, ensuring that sensitive information remains secure and accessible only to authorized personnel.

The architecture maintains stringent security protocols and user access controls:

- **ETL and BI Tools**: ETL tools leverage a dedicated service account to access all layers. BI tools exclusively interact with DPs through the Data Presentation Layer (DPL) schema and views using designated service accounts.
- **Data Science and Self-Service ETL**: Data Science tools and Self-Service ETL platforms access RAW, Curated, Domain/EDW, or DP using service accounts. Temporary tables are managed within the Working Data Layer (WDL) schema, and sandbox requests facilitate experimentation.
- **Technical Users**: Technical users such as Data Engineers, Data Scientists, and Data Architects access Domains or DP schemas, contingent on their roles and development requirements.
- **Data Visualization**: Data Visualization experts access DP schemas in read-only mode, aligning with their visualization needs.
- **Business Users**: Business users can access data upon request in a read-only capacity. Any table creation requirements entail requesting a designated sandbox.

Key Principles of Data Security:

1. **Domain-Centric Security:** Security measures are tailored to the unique requirements of each domain. This approach enables granular control over data access, ensuring that only authorized users can interact with data within their respective domains.
2. **Least Privilege:** Users are granted the minimum level of access necessary to perform their tasks. This reduces the risk of accidental or intentional data breaches caused by excessive permissions.

3. **Segregation of Duties:** Access rights are separated to prevent conflicts of interest or unauthorized manipulation of data. Different roles, such as data stewards, analysts, engineers, and administrators, have distinct levels of access based on their responsibilities.
4. **Data Encryption:** Sensitive data is encrypted both at rest and in transit to prevent unauthorized access. This ensures that even if data is intercepted, it remains unintelligible to unauthorized parties.
5. **Authentication and Authorization:** Users are required to authenticate themselves through secure methods before accessing data. Authorization mechanisms ensure that users can only access the data they are authorized to view or manipulate.
6. **Monitoring and Auditing:** Robust monitoring tools are implemented to track data access and changes. Audit logs provide a trail of activities, aiding in identifying and mitigating security incidents.

Access Control in Data Ocean Architecture:

1. **Role-Based Access Control (RBAC):** Access to different layers, schemas, and data products is determined by roles assigned to users. Roles are defined based on job responsibilities, ensuring that users can access only the data relevant to their roles.
2. **Service Accounts:** Service accounts are used for automated processes, such as ETL jobs, data pipelines, and data science tasks. They are granted specific access rights and are separate from user accounts to minimize human intervention and potential security breaches.
3. **Views and Abstraction Layers:** Data access is provided through views that abstract underlying data structures. These views ensure that users can access data in a controlled and consistent manner, while maintaining data security and integrity.
4. **Data Presentation Layer (DPL):** BI tools and data visualization tools interact with the data through the DPL, which provides a layer of abstraction. This separation prevents users from directly accessing sensitive data and ensures that only authorized and relevant data is presented.

Access Control Workflow:

1. **User Access Request:** Users submit requests for data access based on their roles and responsibilities. Requests are reviewed by the appropriate data stewards and administrators.
2. **Role Assignment:** Based on the user's role, access rights are assigned. These rights define which domains, layers, and data products the user can access.
3. **Security Views and Abstraction:** Users access data through security views and abstraction layers, ensuring that they interact with the data according to their permissions and roles.
4. **Monitoring and Auditing:** All data interactions are monitored and audited. Any unusual activity triggers alerts, and audit logs help in identifying security incidents and potential breaches.

Benefits of Secure Data Access in Data Ocean Architecture:

1. **Data Protection:** Sensitive data is safeguarded from unauthorized access, reducing the risk of data breaches and leaks.
2. **Compliance:** Data access controls align with industry regulations and compliance standards, ensuring that data handling practices meet legal requirements.
3. **Data Integrity:** Security measures prevent unauthorized data modification, preserving data accuracy and integrity.
4. **Efficient Collaboration:** Secure access controls enable collaboration between business and technical teams without compromising data security.
5. **Trust:** Establishing robust security measures builds trust with stakeholders and customers, demonstrating a commitment to data privacy and confidentiality.

7. Conclusion

The Data Ocean Reference Architecture lays the foundation for a robust and comprehensive data management framework that aligns seamlessly with Solvay Atom's transformation goals. Through its layered approach and Domain-Driven Design philosophy, this architecture addresses the complex challenges of data organization, processing, and utilization. By defining distinct stages for data flow – from Raw Data to Curated Data to Use-Case Oriented Layers – the architecture ensures that data is ingested, processed, and transformed in a controlled and structured manner.

The architecture's emphasis on business Domains fosters a culture of data ownership, accountability, and collaboration across different functional areas. The nine identified business Domains, ranging from HR to Finance to Marketing, reflect the diverse nature of Solvay Atom's operations and provide a tailored approach to data management for each domain. The Domain-Driven Architecture promotes scalability, maintainability, and governed data sets, resulting in higher data quality, consistency, and reliability.

By incorporating security by design other and security measures throughout the architecture, data integrity and confidentiality are upheld. Users are granted specific access rights based on their roles, ensuring that sensitive information remains protected and only accessible to authorized personnel. This comprehensive security approach mitigates risks, ensures compliance with regulatory standards, and builds trust among stakeholders.

In conclusion, the Data Ocean Reference Architecture establishes a solid framework that enables Solvay Atom to harness the full potential of its data assets. By aligning data processes, emphasizing Domain ownership, and implementing stringent security measures, this architecture provides a strategic advantage in leveraging data for informed decision-making, innovation, and sustainable growth.