WI22A26S - DS Data Management Fundamentals

IT architecture design draft: A large scale e-commerce architecture

by

Schildt, Ole and Meyer, Philipp

Duale Hochschule Baden-Württemberg (DHBW)
Heilbronn

Course of Studies:	Business Informatics – Data Science
Course Code:	WI22A2
Supervisor:	Prof. Dr. Giacomo Welsch
Submission Date:	10th of April, 2025

Contents

1 Design Concept - Data Mesh 1

2 Architectural Idea 1

2.1 Data Sources and Integration 1

2.2 Data Processing and Storage 1

2.3 Data Quality, Governance, Compliance and Security 2

2.4 Data Analysis and Visualization 2

3 Operational Scenarios 3

1 Design Concept - Data Mesh

The Data Mesh concept addresses the limitations of centralized data solutions where a central data team becomes a bottleneck, hindering timely data-driven decisions. This occurs because the team struggles with data pipeline issues and lacks sufficient domain knowledge, while domain teams, despite owning relevant operational data, rely on the central team for analytical insights. Data Mesh resolves this by shifting analytical data ownership to domain teams, aligning it with business areas or data domains. Traditionally, a central data team manages data pipelines and analysis. In a Data Mesh, their role transitions to providing a self-serve data infrastructure platform that empowers domain teams, who now own their data, to conduct cross-domain analysis (Christ et al., 2025; Dehghani, 2022).

2 Architectural Idea

2.1 Data Sources and Integration

Data originates from diverse source systems, including the webshop, ERP, CRM (e.g., Salesforce), Google Analytics, and payment processors. In addition, data is generated by the microservices landscape, where the decentralized architecture of microservices provides greater agility and scalability than monolithic systems in e-commerce, enabling independent deployment and updates (Auer et al., 2021; Francesco et al., 2017).

Change Data Capture (CDC) serves as the mechanism for source systems, including these microservices, to ingest this data into Kafka, a distributed event streaming platform. The publisher-consumer pattern governs communication within Kafka: source systems (publishers) write data to Kafka topics, and downstream systems (consumers) read data from these topics. Within Kafka, data is stored as ordered, immutable sequences of events or records.

Confluent, alongside Kafka, validates data schemas, ensuring data quality and consistency. The Delta Lake Connector facilitates a unidirectional transfer of schema-validated data from the middleware into the data lakehouse.

2.2 Data Processing and Storage

Delta Lake, implemented on Google Cloud Storage (GCS), serves as the foundation for the data lakehouse. The selection of Delta Lake is driven by Databricks being the primary operational platform for data processing with Python, Spark, Scala and Jupyter Notebooks.

To provide accessible data to the Data Domains, a Medallion Architecture is employed. The Central Data Team handles the primary ETL orchestration for Bronze and Silver layers. In contrast, Gold layer population is largely managed by the Data Domains through their own ETL processes. Beyond this, Data Products and AI Artifacts are also generated, typically using ELT operations.

2.3 Data Quality, Governance, Compliance and Security

Data Contracts serve as a core principle, establishing clear expectations and requirements for data product development. These self-serviced and accessible contracts define producer obligations, allowing internal implementation flexibility and backward-compatible extensions to incentivize data sharing and enable service level monitoring (with contract cancellation as an option). This approach directly contributes to data quality by ensuring correctness through contractual requirements, promoting consistency via defined schemas, and addressing completeness by mandating required data fields. Service Level Agreements(SLA) within contracts also address actuality by setting expectations for data freshness, while contracts can also enforce uniqueness constraints (Harrer & Christ, 2025).

Underlying this, Unity Catalog with Metastore, Lineage, Sharing, and Audit Logging provides a unified data governance layer. Specifically, the metastore provides a central repository for metadata, enabling data discovery and controlled sharing, while lineage tracks data origins and transformations, ensuring transparency and trust. Access Management is managed through Databricks Workspaces, integrating Unity Catalog and IAM with Google SSO, with automated reassessment reminders to maintain security.

2.4 Data Analysis and Visualization

For data analysis and visualization, the architecture employs a dual approach between the Data Domains and Self-Service Business Intelligence (SSBI). Streamlit is available for Data Domains to create custom dashboards, while BigQuery in conjunction with Google Looker supports SSBI for use by non-data teams, such as Purchase and Controlling.

3 Operational Scenarios

1. **Recommendations:** Google Analytics data flows through Kafka and Confluent into the bronze layer of the data lakehouse. The Recommendations Data Domain then queries this data, along with other relevant data, via the Unity Catalog, and trains a recommendation model within their dedicated Databricks Workspace. This model is subsequently published in the AI Registry for real-time inference using data from the data lake.

2. **Forecast:** ERP sales data is ingested into the bronze layer of the data lakehouse via Kafka. The Finance Forecasting Data Domain queries this data through the Unity Catalog to produce a sales forecasting dashboard (a data product) using Streamlit. Additionally, the forecasting model is stored in the AI Registry for broader organizational use.

3. **Warehouse Planning:** The Supply Chain Domain leverages the forecasting model from the AI Registry, combined with other silver and gold layer data accessed via the Unity Catalog, to perform prescriptive analytics. This analysis supports decisions on optimal new warehouse locations.

4. **Controlling SSBI Dashboard:** Data from payment processors is loaded into the bronze layer via Kafka. The central Data Team creates prepared factual and dimensional tables in the silver layer. The Controlling department then uses SSBI tools to create their own gold layer tables and dashboards, enabling insights for debt collection procedures.

5. **GenAI Competitor Analysis:** To maximize the flexibility of the Data Mesh architecture, a Data Domain operates autonomously. They scrape annual business reports from competitors and employ an agentic Graph Retrieval Augmented Generation (RAG) chat system to support information gathering for Controlling.

References

Auer, F., Lenarduzzi, V., Felderer, M., & Taibi, D. (2021). From monolithic systems to microservices: An assessment framework. *Information and Software Technology*, *137*, 106600. https://doi.org/10.1016/j.infsof.2021.106600

Christ, J., Visengeriyeva, L., & Harrer, S. (2025). Data mesh: Data mesh from an engineering perspective [Accessed: 2025-04-10].

Dehghani, Z. (2022). *Data mesh: Delivering data-driven value at scale* (1st). O' Reilly Media.

Francesco, P., Malavolta, I., & Lago, P. (2017, April). Research on architecting microservices: Trends, focus, and potential for industrial adoption. https://doi.org/10.1109/ICSA.2017.24

Harrer, D. S., & Christ, J. (2025). Datamesh-governance.com [Accessed: 2025-04-09]. https://github.com/datamesh-governance/datamesh-governance.com/tree/main