

Enterprise Data Architect Case Study: Data Platform Transformation for AI

Role: Enterprise Data Architect

Time Allotment: 60 minutes (30-minute Presentation / 30-minute Strategic Discussion)

Goal: Assess the candidate's ability to provide high-level architectural strategy, justify technical choices based on **Cost, Scale, and Observability (CSO)**, and lead the crucial transition from legacy Analytics to **Data for AI**.

The Challenge (The Situation & Mandate)

Our multi-tenant data platform is currently blocked from achieving AI capabilities due to legacy architectural debt. Your mandate is to provide strategic and technical suggestions to our team on how to execute the full transformation.

Problem	Explanation (The Pain)	Strategic Impact (CSO)
Variable Schema & Embedded Data	Our platform is low-code/no-code , meaning the schema is customer-specific for the same application. Data is highly nested, and critical Master Data is embedded .	Cost & Scale: Leads to massive duplication, requires complex, inefficient processing logic, and drives high cloud costs.
Monolithic Lock-in & Efficiency	The system is proprietary and inefficient, operating in batch and preventing migration to cost-effective OSS Spark/Kubernetes .	Cost & Latency: Blocks near real-time data flow, preventing cloud cost optimization.
Low Latency Block	We cannot consistently process data at the required 5-10 minute latency needed for real-time AI features and conversational agents.	Observability & AI: Hinders proactive monitoring and makes predictive use cases impossible.

The Presentation Mandate (30-Minute Focus: Strategy and Suggestion)

The candidate must present their architectural strategy and technical decisions to address the following four key challenges, using relevant examples from their past experience.

1. The Foundational Fix: Data Layering, Decoupling, and Cost

- **Architectural Suggestion:** Propose a high-level **multi-layered architectural design** (e.g., Bronze, Silver, Gold) for the new data platform.
- **Separation Strategy (Variable Schema Handling):** Detail the process for **separating embedded Master Data** from transactional data within your layers. Explain how this process can be **parameterized** to handle the **customer-specific schema variations** (low-code problem) while ensuring **lossless data fidelity**.
- **Cost Justification:** Explain how this separation strategy immediately enables **significant cloud cost savings** by eliminating embedded data redundancy.

- **Modeling:** What conceptual **data modeling pattern** (e.g., Data Vault, Star Schema) would you apply to the Semantic (Gold) Layer to organize this complex, multi-collection data efficiently for **both analytics and AI feature generation?**

2. Achieving Real-Time Performance and Scalability (CSO)

- **Approach:** Outline your strategy to achieve **sub-30-minute latency** and handle volumes up to **5 million records per day**.
- **PoC Prioritization:** Which **two critical PoCs** would you suggest our team prioritize (e.g., Flink vs. high-frequency batch, OSS Spark) to de-risk both the **real-time latency goal** and the **Kubernetes transition** simultaneously?
- **Scalability:** How does your proposed solution ensure **horizontal scalability** and high availability when dealing with highly dependent data flowing between multiple applications?

3. The Open-Source Transition, Observability, and Metadata Knowledge Base

- **Risk Mitigation:** Detail the most critical **two PoCs** required to validate the migration path to **OSS Spark/Kubernetes**. How do you mitigate the risk of vendor lock-in and ensure the core processing logic runs seamlessly in the new environment?
- **Observability (O):** Detail the design and purpose of the new **Manifest Pattern** (tracking/logging mechanism). How does this pattern enhance observability and provide the operations team with granular control (i.e., pausing or resuming customer data flow) that the old system lacked?
- **Metadata Knowledge Base:** Design the strategy for a centralized **Data Entity Knowledge Base** (Data Catalog/Knowledge Graph). How will this system manage **metadata, schema, join conditions, business definitions (non-technical names), and descriptions** for a **multi-tenant, multi-application** environment? How is this critical to enabling conversational AI features?

4. Strategic Pivot: Data for AI and Agent Services

- **Architectural Bridge:** Explain how the low-latency, normalized foundation enables the next phase, serving data agents (e.g., Conversational Reporting).
- **Recommendation:** Draw a high-level conceptual flow showing the role of a **Feature Store** in your architecture. Justify why this specialized data layer is necessary to power **Conversational Reporting Agents** and provide **real-time predictions**.

Key Discussion Points (For the Interviewer)

- **CSO Trade-offs:** "In your design, where is the trade-off between **Data Quality Checks** and **Real-Time Latency**? How do you ensure high quality without sacrificing the sub-30-minute goal?"
- **Team & Leadership:** "If you were leading this effort, how would you structure the initial PoC team's tasks to leverage the existing team's knowledge while pushing them toward **OSS/Kubernetes best practices**?"
- **Business Quantification:** "Beyond the architectural benefit, what is your estimated *minimum percentage of cloud cost savings* achievable by eliminating the embedded master data redundancy using your proposed solution?"