Phase 3: Stream Processing POC

Group - Pi

Github Repository

https://github.com/1captain0/stream-processing-workshop.git

Files and topologies created

Question 1: Which artist events had the highest and lowest ticket sales compared to their capacity?

Source: Events, Tickets Artists

File: src/main/java/org/improving/workshop/samples/ArtistEventsTracker.java.java

Question 2: Leverage customer streaming milestones to trigger discounted ticket alerts for an artist's next event.

Source: Streams, Events, Customer, Email

File: src/main/java/org/improving/workshop/samples/DiscountMilestoneEmail.java

Question 3: How does the streaming behavior of customers influence their likelihood of purchasing tickets for an artist's events ?

Source: Events, Tickets, Streams

File: src/main/java/org/improving/workshop/samples/StreamingBehaviourTracker.java

Retrospective Results

What went well?

Incremental, Test-Driven Development:
Built the topology step-by-step using tests, which made the development process reliable and easier to debug.

Serde Handling:
Understood and applied correct SerDes for each operation (joins, groups, etc.), avoiding common deserialization issues.

Topology Design:
Designed modular and logical pipelines using KStreams and KTables appropriately based on data needs.

Understanding the Impact of Kafka Streams in Business:
Successfully handled event-time filtering, milestone logic, and data enrichment from multiple streams and how it can be leveraged to answer business questions.

What didn’t go well ?

Negative & Edge Case Testing:
Limited coverage for scenarios like missing customer data, past events, or malformed input.

Serde Configuration Management:
Initial confusion around Serde mismatches caused delays—could benefit from standardized configuration.

Collaboration and Peer Review:
We should have held early team syncs to review other team members topology designs. That way, we could have incorporated their best practices from the start

Actions that can be taken

Introduce Parameterized & Negative Test Cases:
Expand test suite to cover edge cases and ensure resilience in production.

Document Design Decisions & Test Strategy:
Maintain a short design and testing guide for onboarding and continuity

Collaboration and Peer Review:
Schedule a recurring team review meeting at regular intervals to share and align on best practices.