Use-Cases for Stream Data Pipelines

Big Data on Kubernetes – [Day 4]

LUAN MORENO

CEO & Data Architect
Data Engineer & MVP

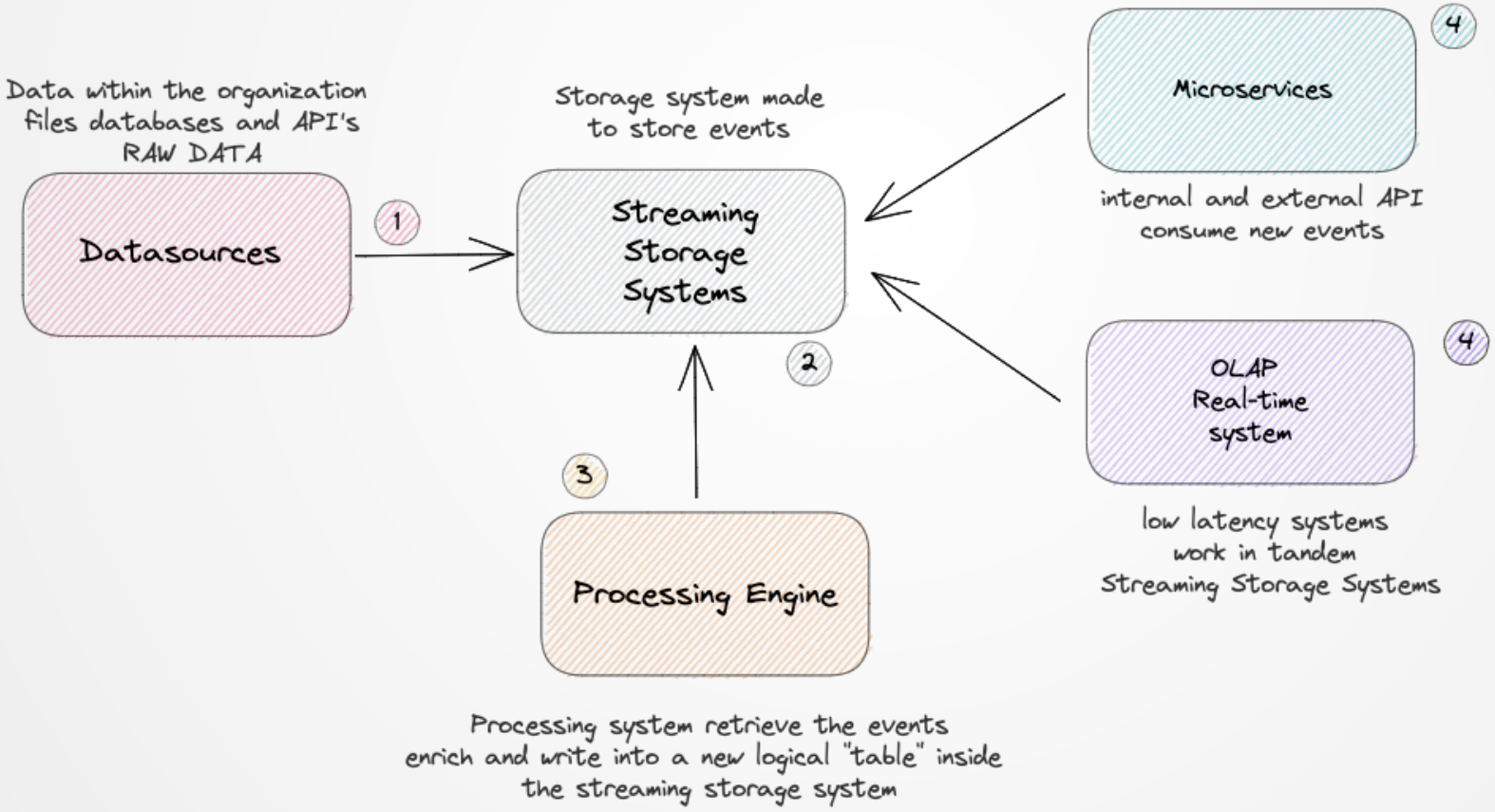
MATEUS OLIVEIRA

Big Data Architect
Data In-Motion Specialist

Stream Data Pipelines on Kubernetes

Symbiosis of Big Data and Kubernetes Infrastructure
Backbone System for Streaming Data Pipelines at Scale

1. Data source, APIs write now not data but events, can ingest data from databases and other data stores
2. All events are stored in Streaming Storage System [the heart of streaming pipeline]
3. Engine processor needs to have the ability to extract and transform events [match made in haven]
4. Now microservices can access direct data (events), and we see a new rise low latency system to query events

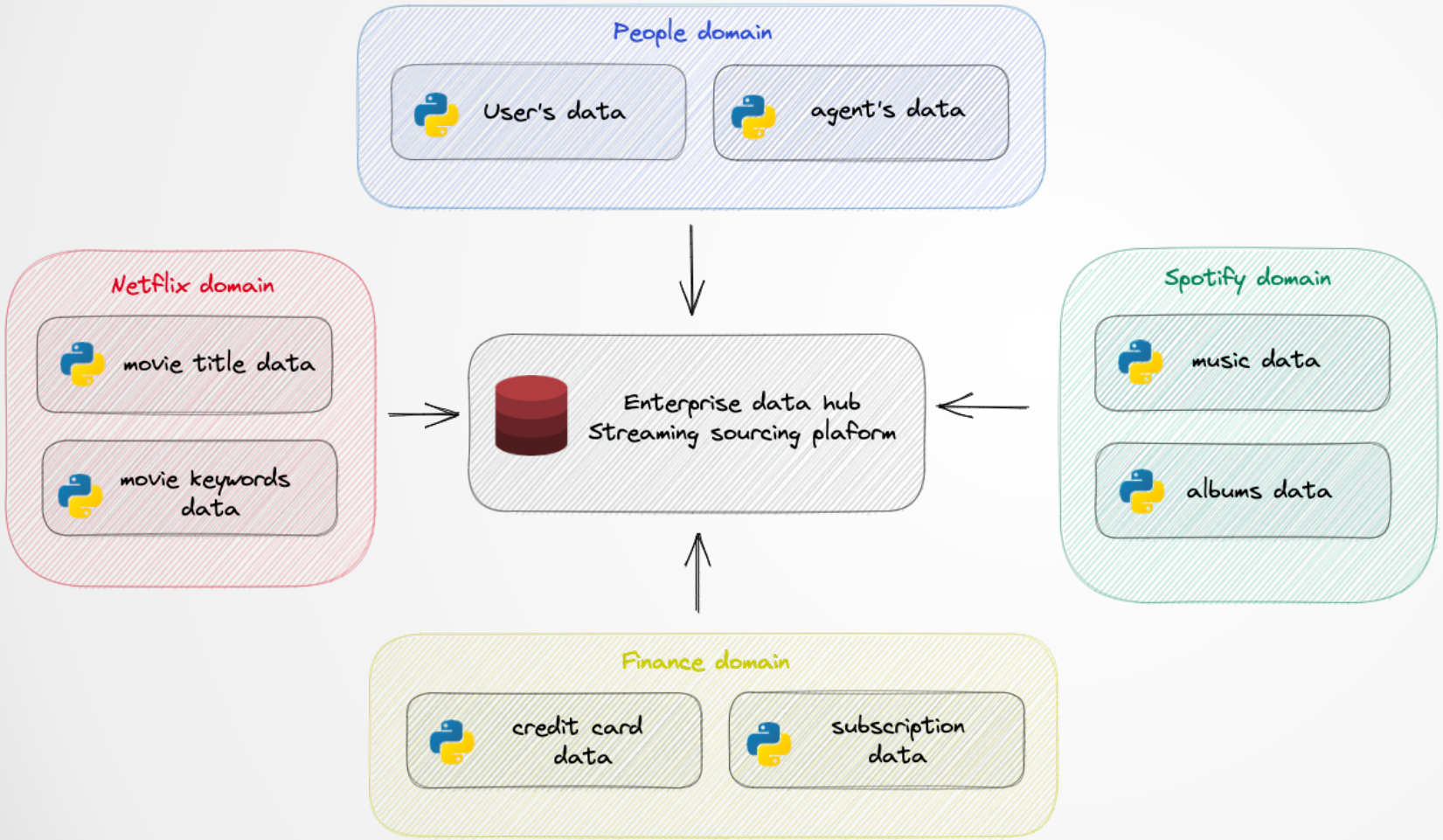


Use-Cases

- Ingesting Real-Time Data from Microservices into Apache Kafka
- Working with KafkaConnect Cluster on Kubernetes for Source and Sink Pipelines
- ETL in Real-Time using KSQLDB
- Data Enrichment in Near Real-Time using Apache Spark
- Using a OLAP System for Analytical Queries
- Orchestrating and Managing a Real-Time Pipelines with Lenses

Ingesting Real-Time Data from Microservices into Apache Kafka

Business Use-Case

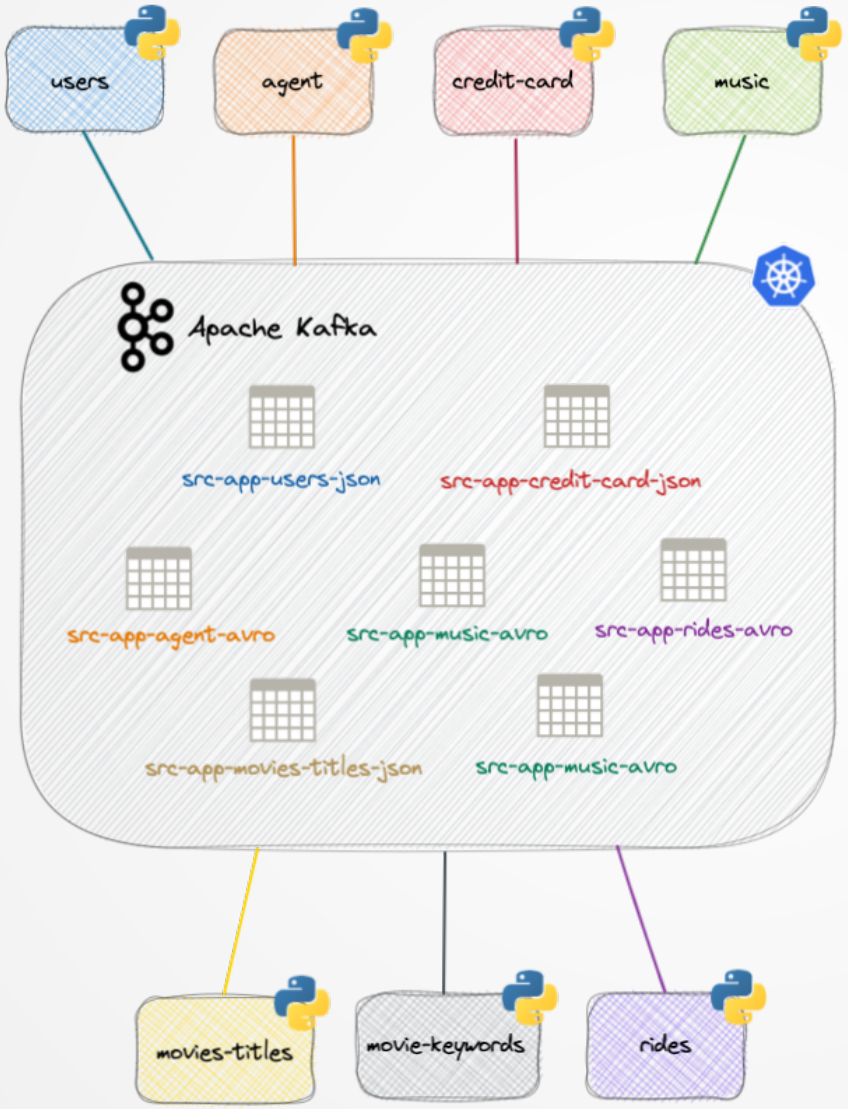


Business Use-Case Scenario

- Variety of Microservices
- Necessity of Domain of Data (Data Mesh)
- Scale Resources to Guarantee the Growth of Business
- Complexity Environment [Event Sourcing]
- Without Traditional Database, Data Hub Proposal [Streaming Storage]

Ingesting Real-Time Data from Microservices into Apache Kafka

Stack



Technology Stack

- Microservices Written in Python
- Enterprise Data Hub – Apache Kafka
- Append-Only Log Structure
- Event-Driven Approach

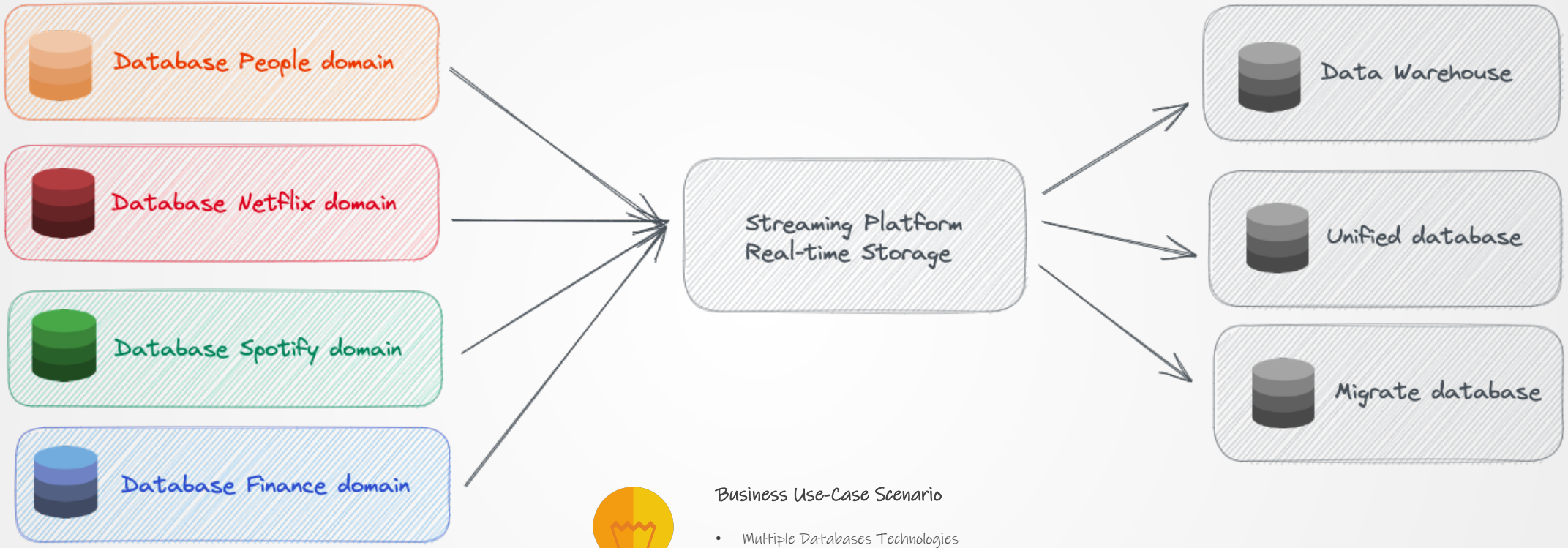
Advantages

- Usage of Microservices Pattern
- Events Written and Read in Milliseconds Rate
- Improve Production and Consumption of Data
- Distributed Storage System

Demo

Working with KafkaConnect Cluster on Kubernetes for Source and Sink Pipelines

Business Use-Case

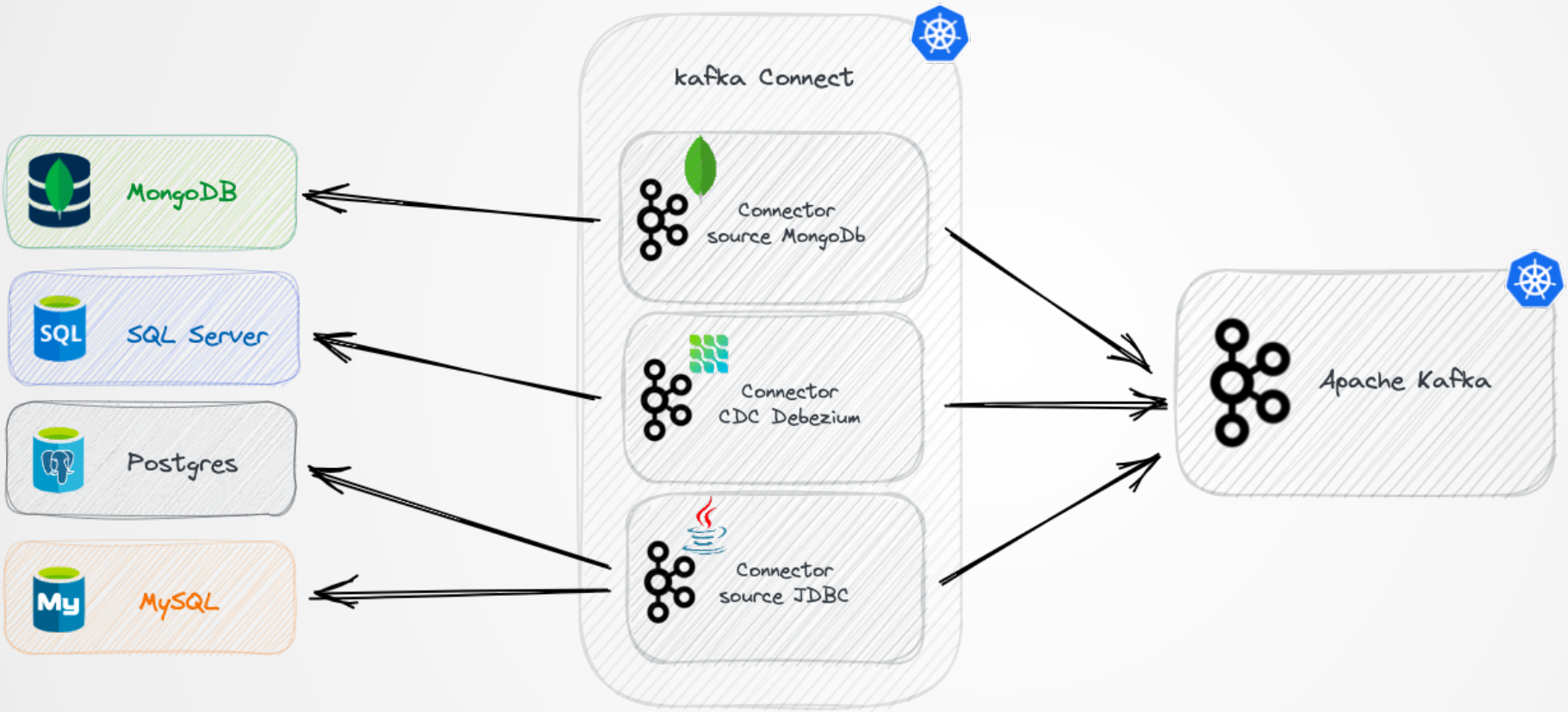


Business Use-Case Scenario

- Multiple Databases Technologies
- Data Integration Need for Analytics
- Complex Methods to Unify Data
- Cumbersome Migration
- Only Fragments of Data into DB and DW

Working with KafkaConnect Cluster on Kubernetes for Source and Sink Pipelines

Stack



Technology Stack

- Enterprise Data Hub – Apache Kafka
- NoSQL DB Document-Oriented Store:
  - MongoDB
- Traditional Databases:
  - SQLS Server
  - Postgres
  - MySQL

Advantages

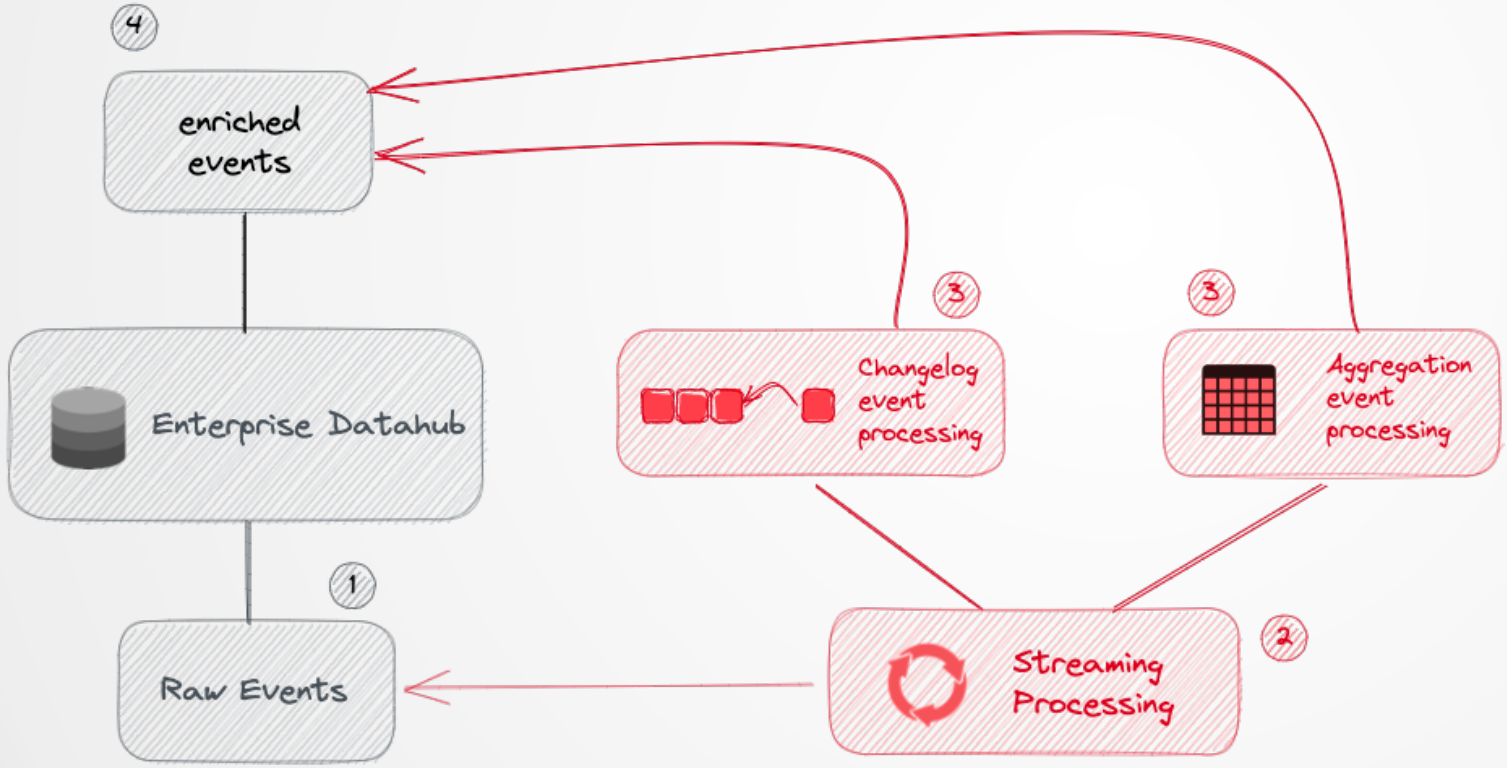
- Integration of Database in a Single Point
- Data Becoming an Immutable Event
- Deploy Data Pipelines using Configuration Files

Demo

ETL in Real-Time using KSQLDB

Business Use-Case

1. API insert into EDH raw events
2. Event Processing Engine access the EDH to retrieve the new raw events
3. Apply the transformation that can be a join between streams [changelog event processing]
3. Apply the transformation that can be aggregation (count) of events producing a table [aggregation event processing]
4. write back into EDH the new enriched events in a new "table"

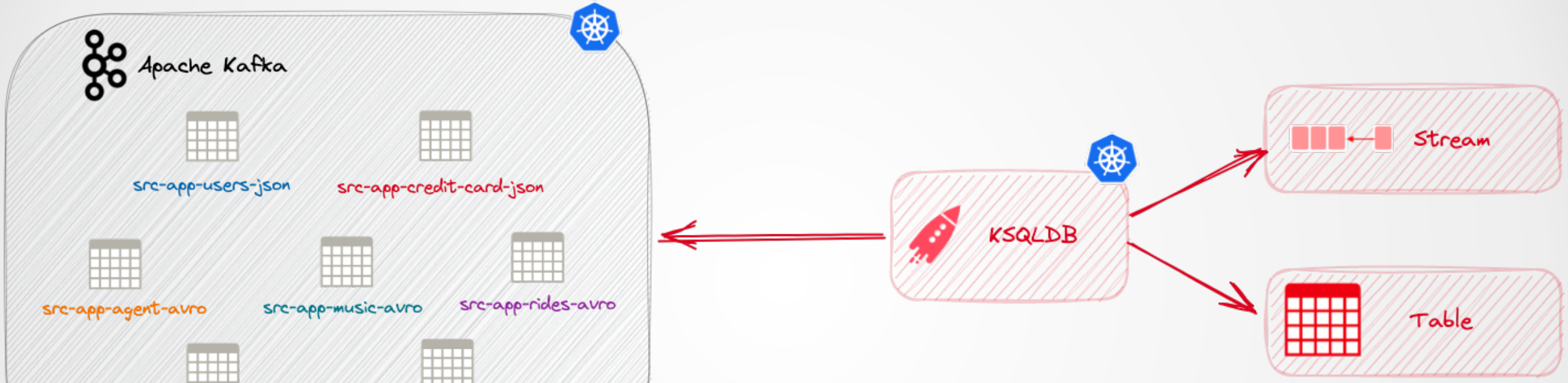


Business Use-Case Scenario

- Necessity for Process Events in Real-Time
- Create a Stream Pipeline that Guarantees Message Delivery
- Process Events using SQL Language
- Lightweight Deployment in Small, Mid and High Workloads
- Process using Changelog Events and Aggregated Transformations

ETL in Real-Time using KSQLDB

Stack



Technology Stack

- Enterprise Data Hub – Apache Kafka
- Events Processor using SQL - KSQLDB

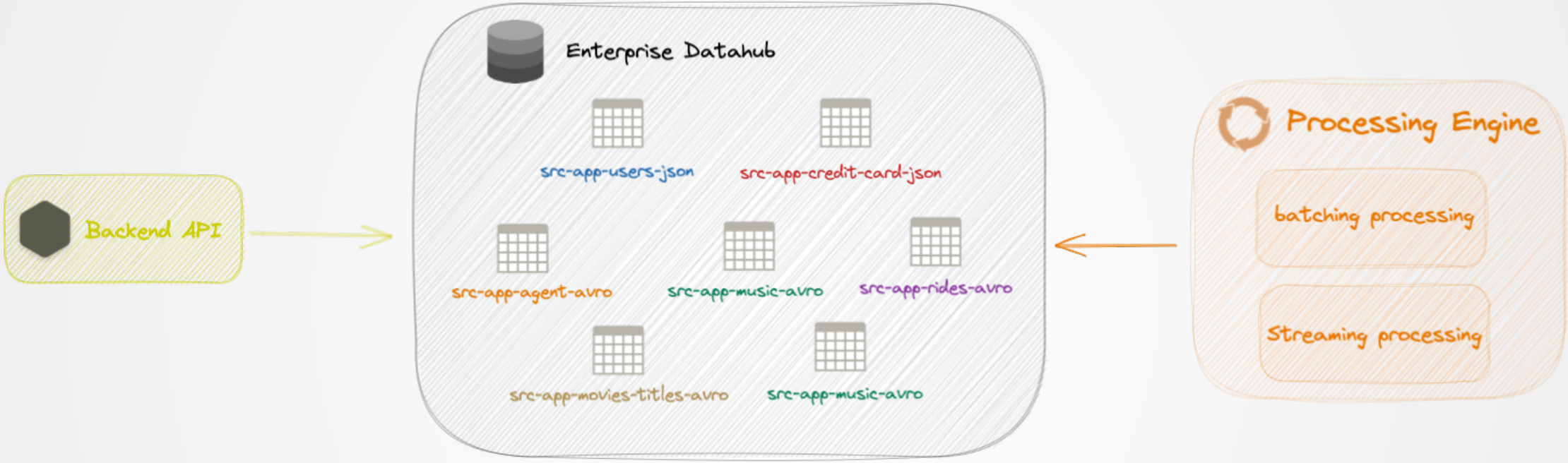
Advantages

- Processing Events [EOS]
- Using Query to Process Event Streams
- Ability to Join Between Tables and Streams
- Write a Changelog Processing = Stream
- Write an Aggregation Processing = Table

Demo

Data Enrichment in Near Real-Time using Apache Spark

Business Use-Case



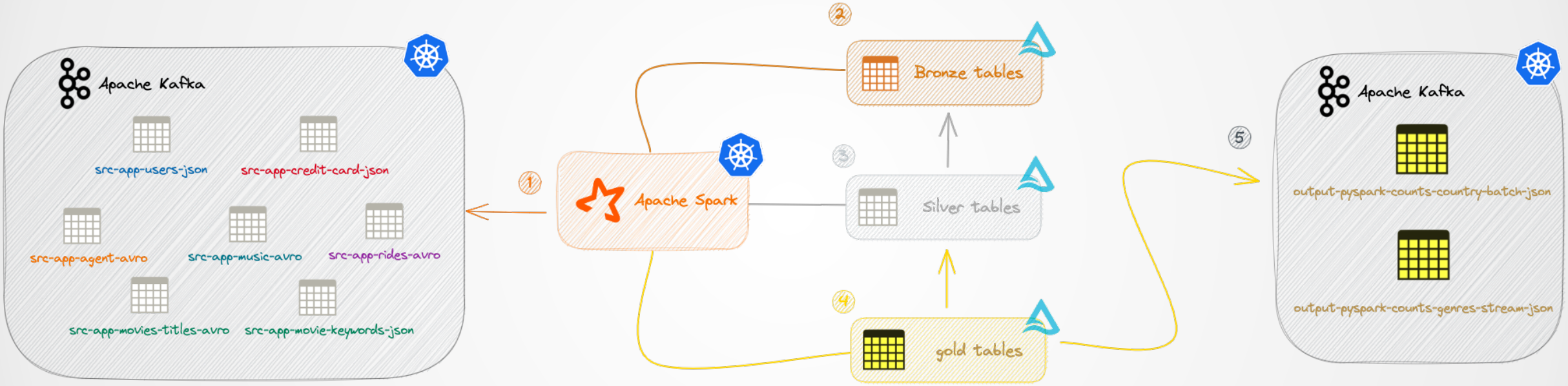
Business Use-Case Scenario

- Process Humongous Amount of Events
- Capability to Perform Process in Batch & Stream
- Fully Integrated with Most Famous EDHs
- The Best Processing Engine in the Planet

Data Enrichment in Near Real-Time using Apache Spark

Stack

1. Ingest data from kafka topics
2. create a structured streaming and record a bronze table in delta [raw events]
3. access the bronze table and transform data into silver table [transform events]
4. from silver tables create gold table with enriched and insightful data [output events]
5. record the gold table inside kafka using structured streaming



Technology Stack

- Enterprise Data Hub – Apache Kafka
- The Most Powerful Processing Engine - Apache Spark

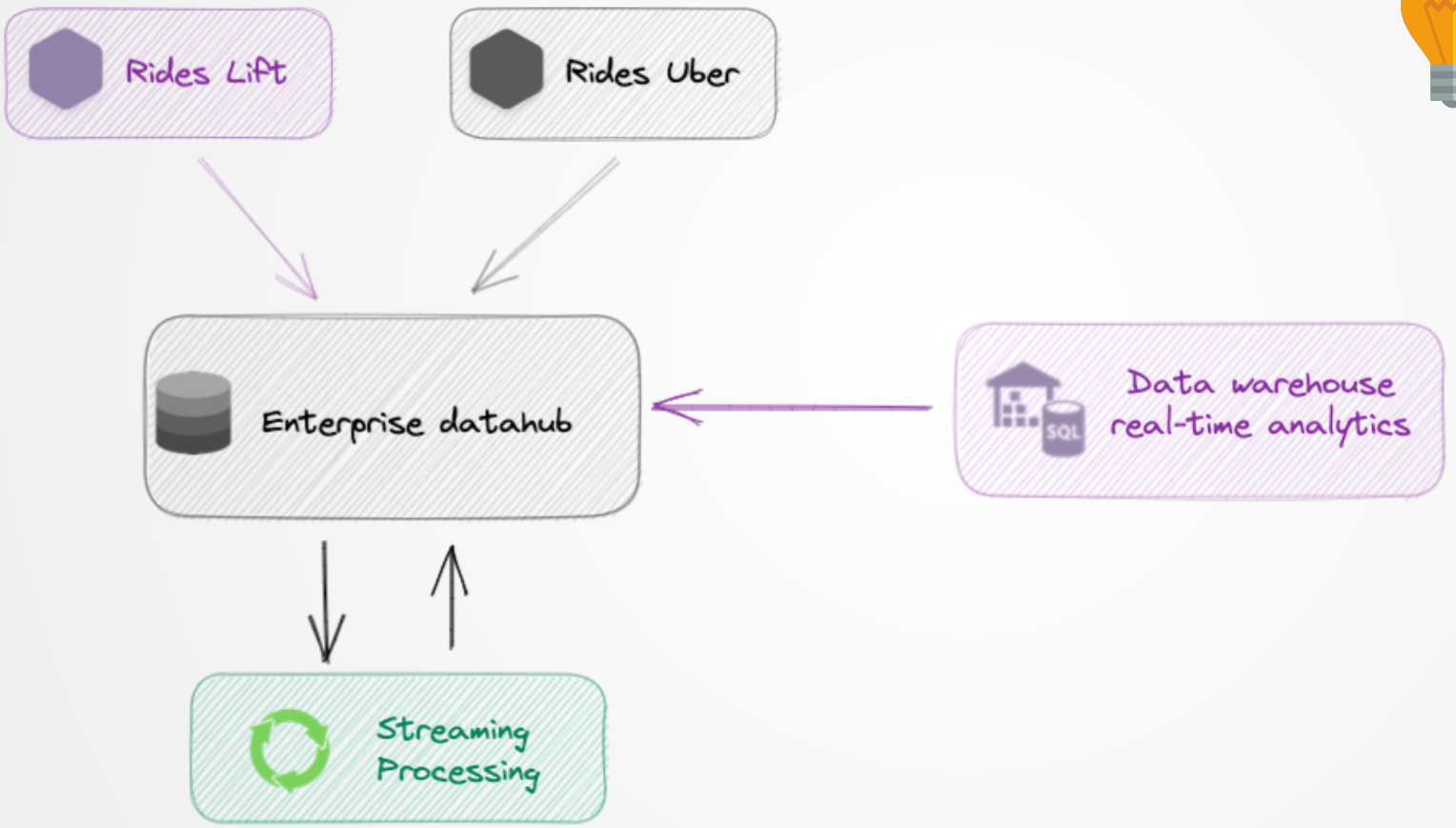
Advantages

- Processing Events in [EOS]
- Ability to Perform Complex Transformations
- Writing Pipeline in Python and SQL - PySpark Engine

Demo

Using a OLAP System for Analytical Queries

Business Use-Case



Business Use-Case Scenario

- Events Landing into EDH
- Urge to Understand Customer Behavior
- React Rapidly to Events
- Necessity to Process New Arrival Events
- Perform Analytics Queries at Scale
- Complex Challenge to Deliver Analytical Dashboards in Real-Time

Using a OLAP System for Analytical Queries

Stack

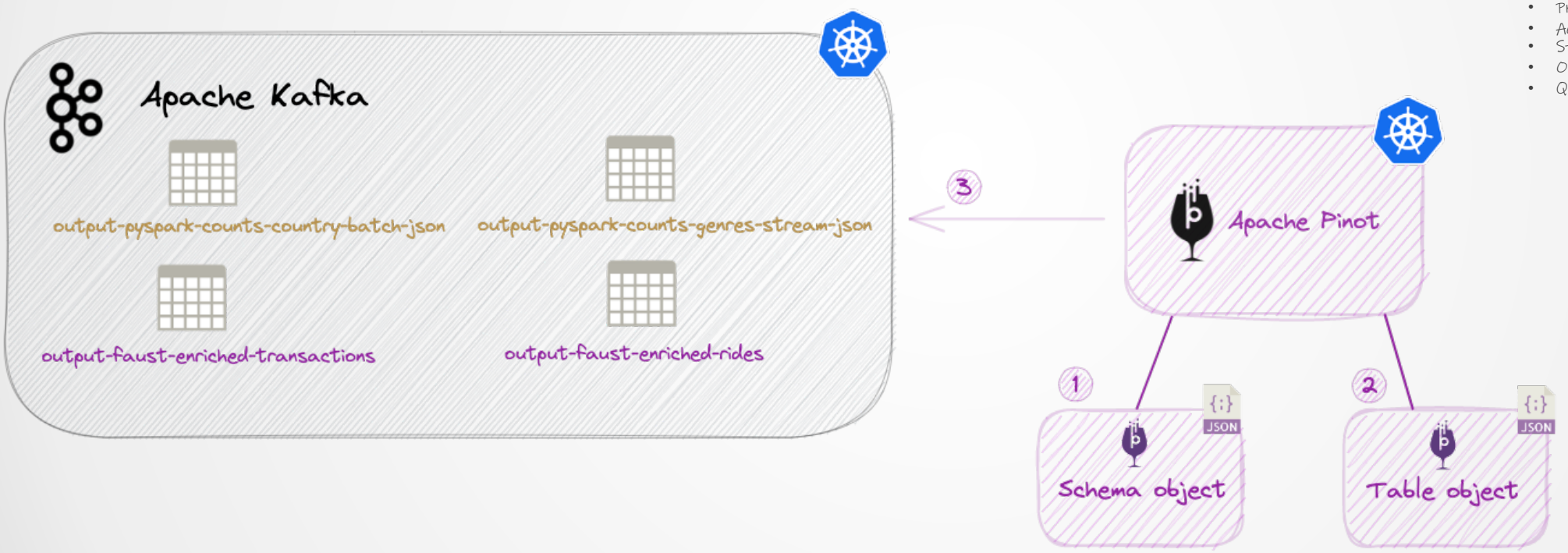
1. create a schema object from the topic we want to read
2. create a configuration file for the table as real-time process
3. copy and execute in pinot coordinator the cmd to create the table an access kafka

Technology Stack

- Enterprise Data Hub – Apache Kafka
- Low Latency Olap System - Apache Pinot

Advantages

- Processing Events using [EOS]
- Act as Consumer
- Store Data in Pinot Tables
- Outstanding Compression Performance
- Query in Low Latency Million Events per Second



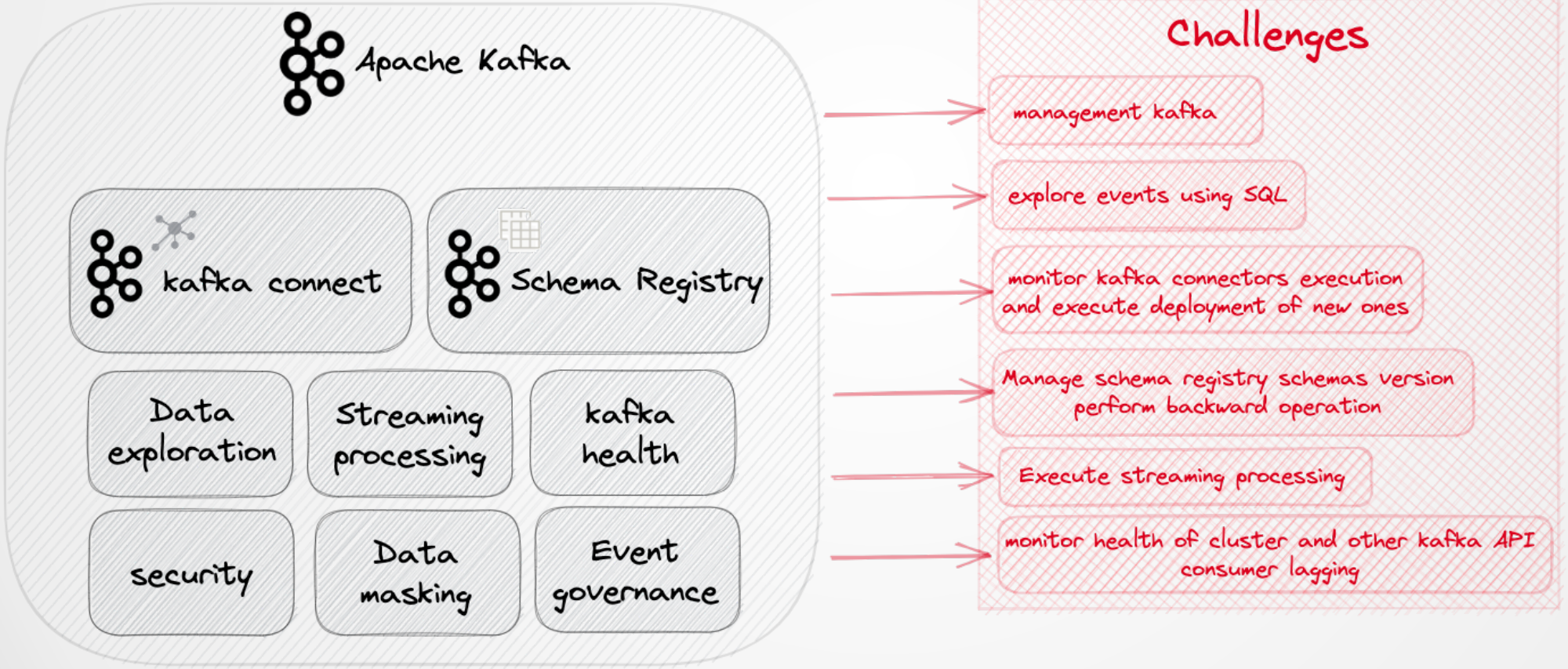
Demo

Orchestrating and Managing a Real-Time Pipelines with Lenses

Business Use-Case

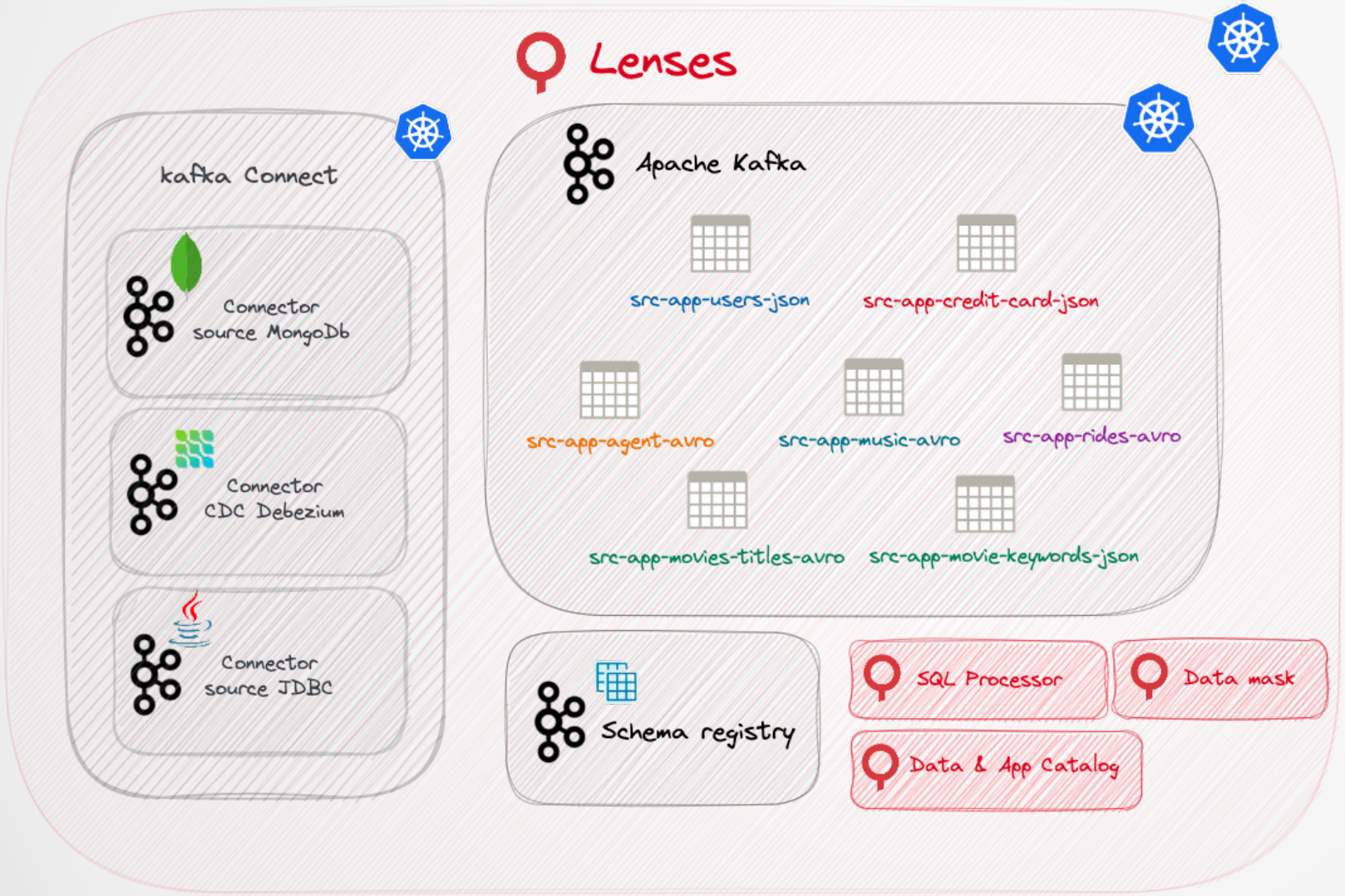
Business Use-Case Scenario

- Events Landing into EDH
- Monitoring Flow of Data and Events
- Observability is Complex



Orchestrating and Managing a Real-Time Pipelines with Lenses

Business Use-Case



Technology Stack

- Enterprise Data Hub – Apache Kafka
- 360° Kafka Observability - Lenses

Advantages

- Monitoring Health of Kafka Clusters
- Deploy Kafka Connectors
- Explore Events using SQL
- Understand and Interact with Schema Registry
- Create a Topology for Lineage
- Development using Data and App Catalog

Demo


Our greatest glory is not in never failing, but in rising up every time we fail.
Ralph Waldo Emerson
quotefancy