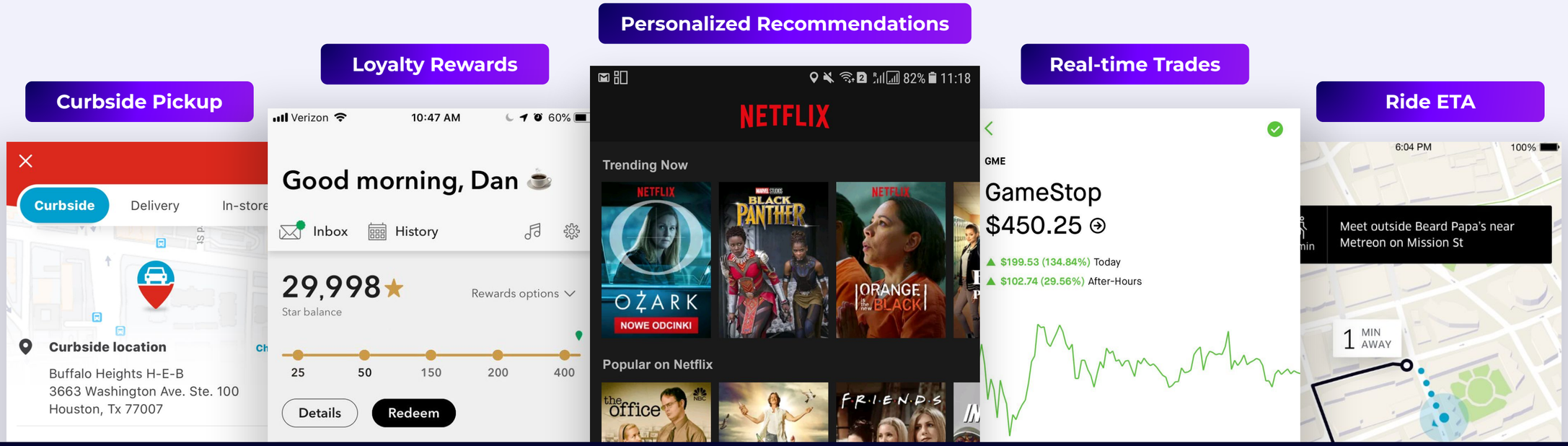
Leveraging Data Streaming Platform (DSP) for Analytics & GenAI

Jun Rao, co-founder @ Confluent

Apache Kafka has ushered in the data streaming era...



APACHE kafka®

>70%
of the Fortune 500

>100,000+
Organizations

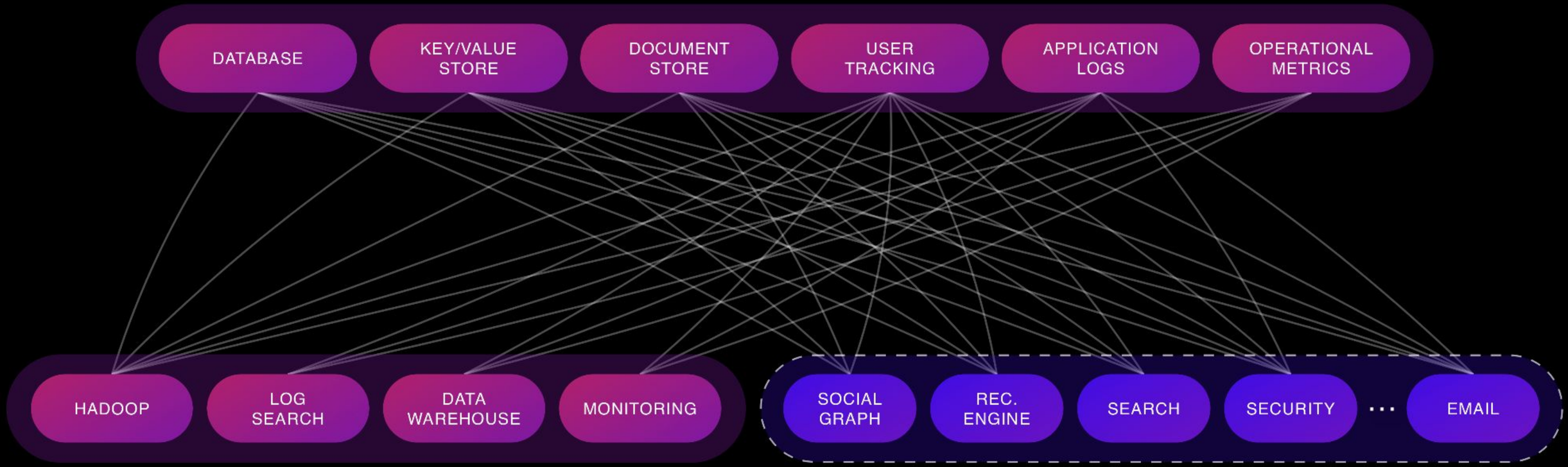
>41,000
Kafka Meetup Attendees

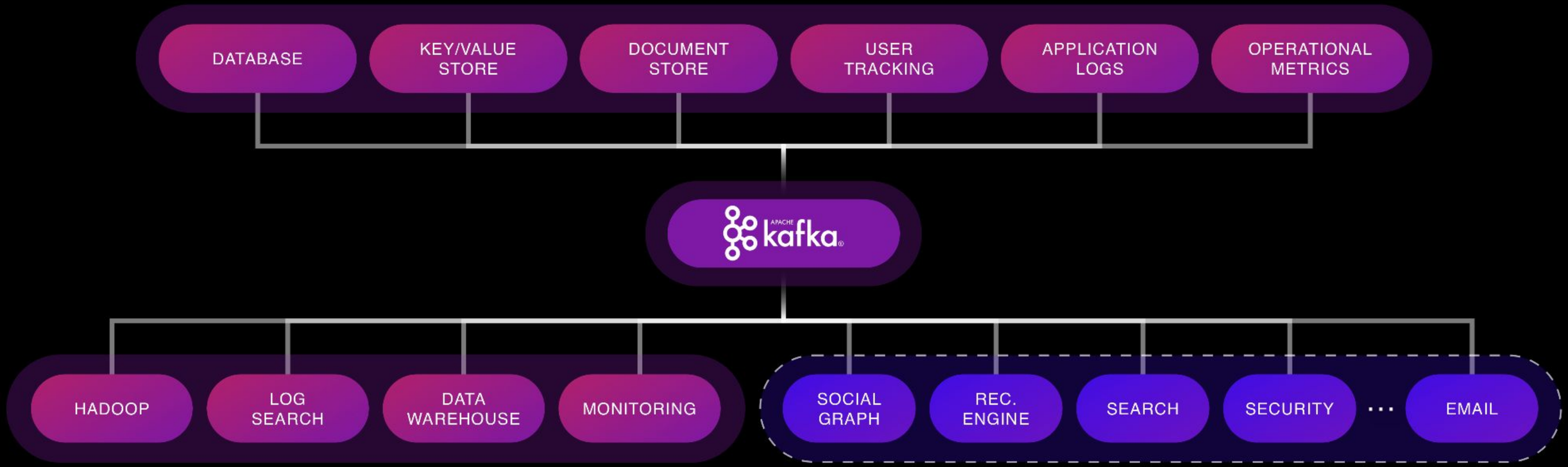
>200
Global Meetup Groups

>750
Kafka Improvement Proposals (KIPs)

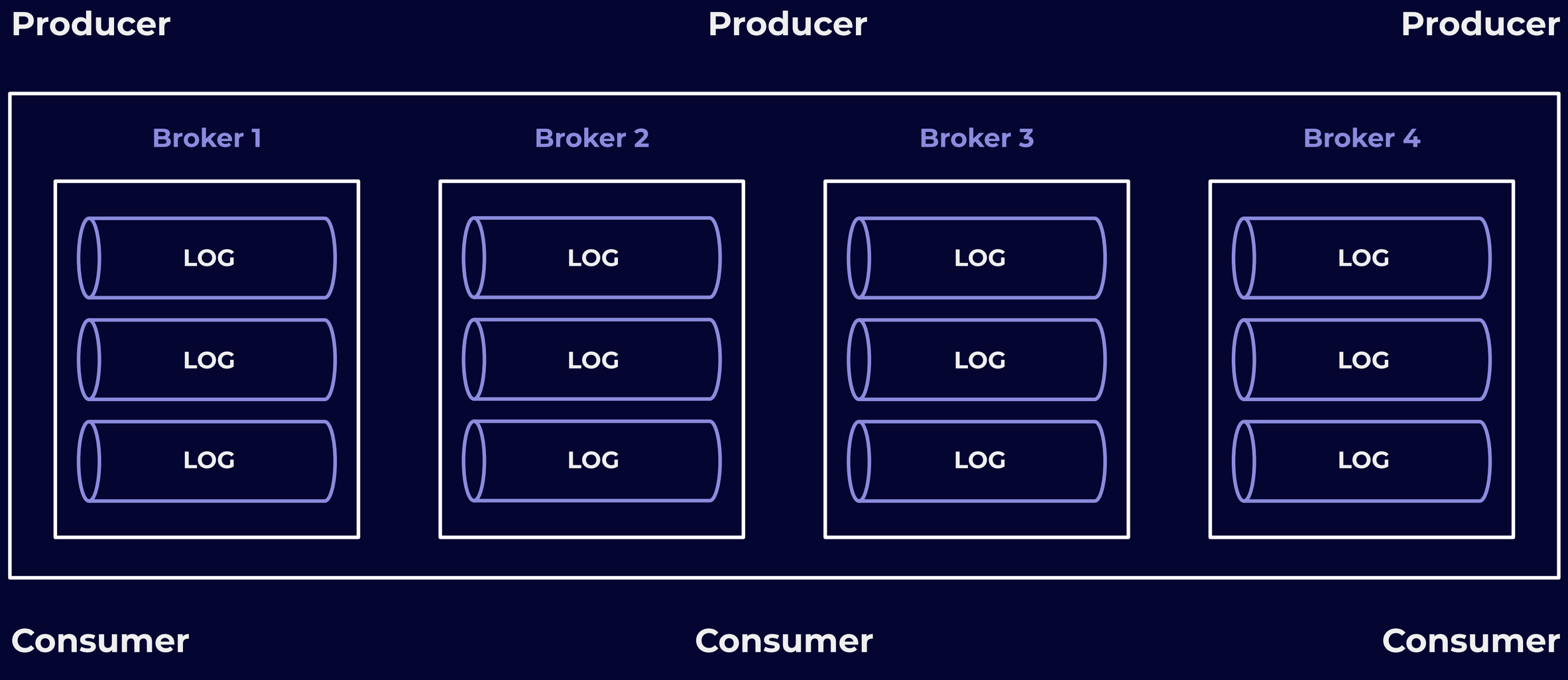
>12,000
Jiras for Apache Kafka

>32,000
Stack Overflow Questions

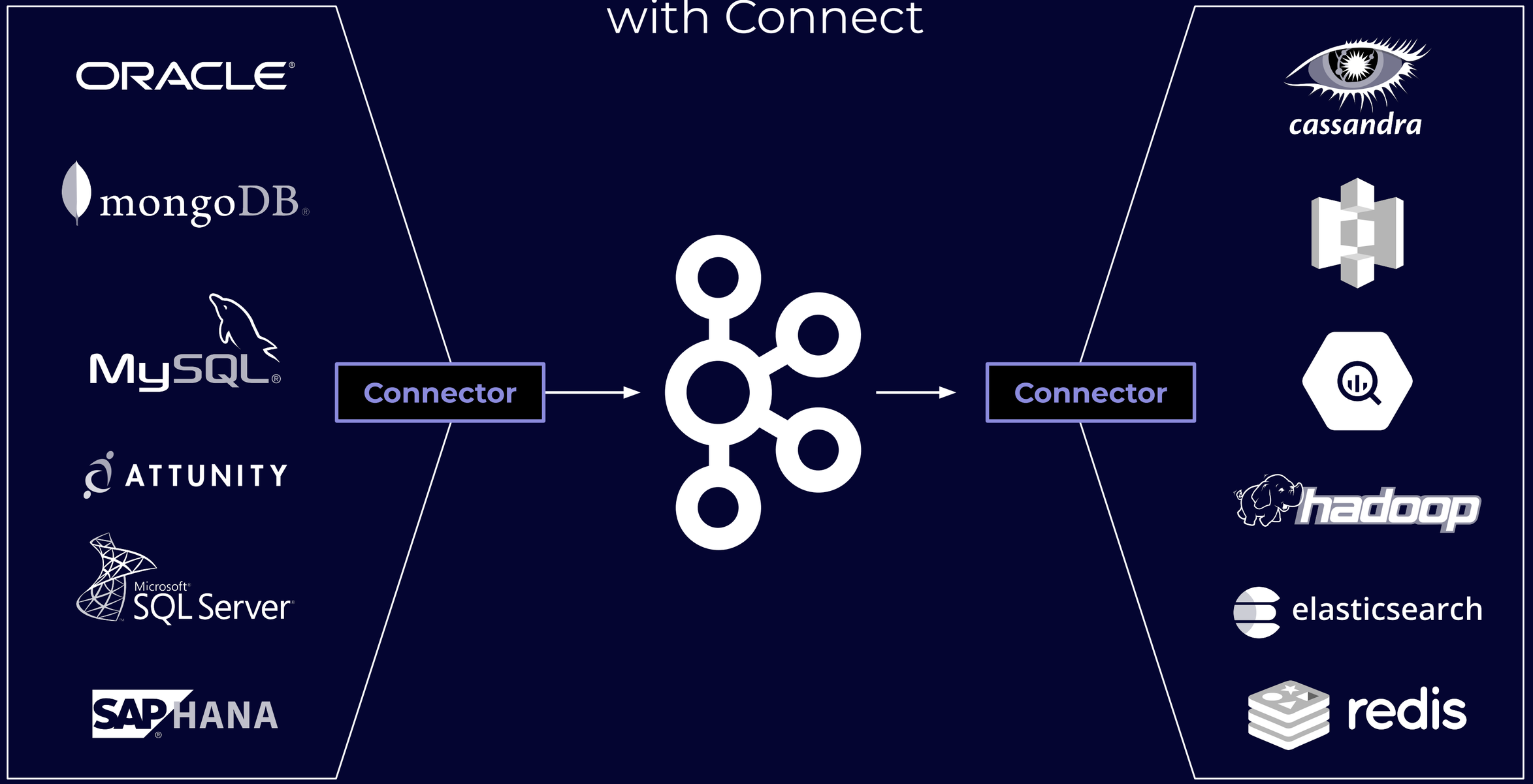

DATABASE
KEY/VALUE STORE
DOCUMENT STORE
USER TRACKING
APPLICATION LOGS
OPERATIONAL METRICS
HADOOP
LOG SEARCH
DATA WAREHOUSE
MONITORING
SOCIAL GRAPH
REC. ENGINE
SEARCH
SECURITY
• • •
EMAIL


DATABASE
KEY/VALUE STORE
DOCUMENT STORE
USER TRACKING
APPLICATION LOGS
OPERATIONAL METRICS
APACHE kafka
HADOOP
LOG SEARCH
DATA WAREHOUSE
MONITORING
SOCIAL GRAPH
REC. ENGINE
SEARCH
SECURITY
• • •
EMAIL

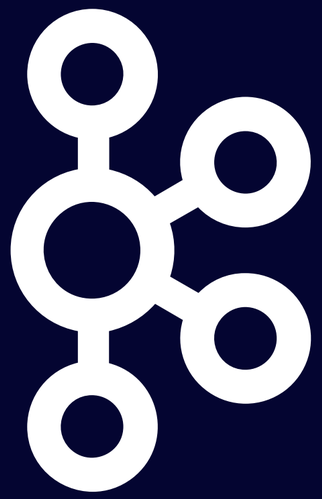
Kafka as Storage: Log at Scale

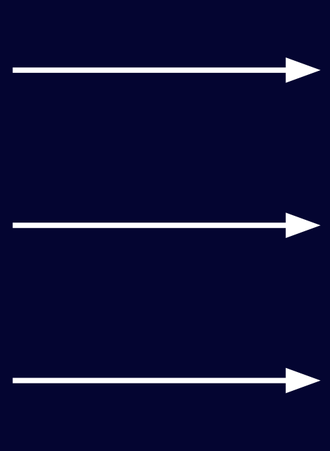


Extensive Integration with Connect



Continuous Data Processing





Event-driven Microservice

Event-driven Microservice

Event-driven Microservice

Kafka Streams

```
val fraudulentPayments: KStream[String, Payment] = builder
  .stream[String, Payment](“payments-kafka-topic”)
  .filter((_ ,payment) => payment.fraudProbability > 0.8)
fraudulentPayments.to(“fraudulent-payments-topic”)
```

Apache Flink

```
CREATE STREAM fraudulent_payments AS
  SELECT * FROM payments
  WHERE fraudProbability > 0.8;
```

Why Developers Choose Flink

Elastic Scalability

Flink is capable of supporting stream processing workloads at tremendous scale

Language Flexibility

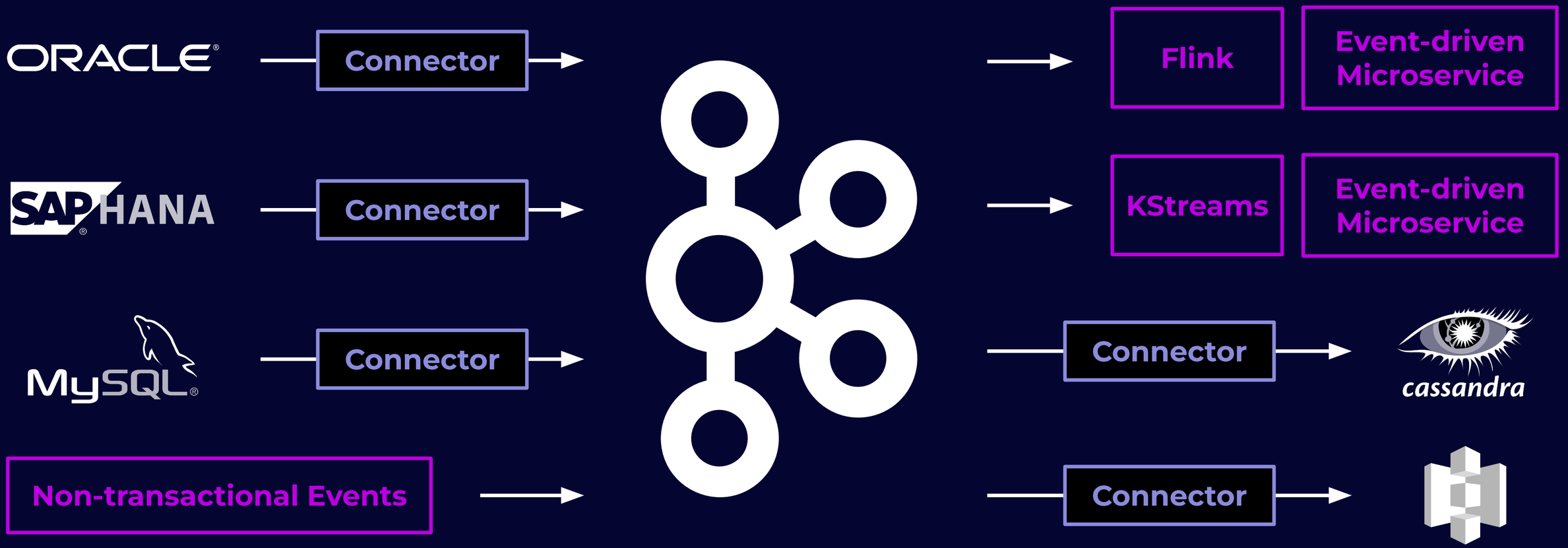
Flink supports Java, Python, & SQL, enabling developers to work in their language of choice

Unified Processing

Flink supports stream processing, batch processing, and ad-hoc analytics through one technology

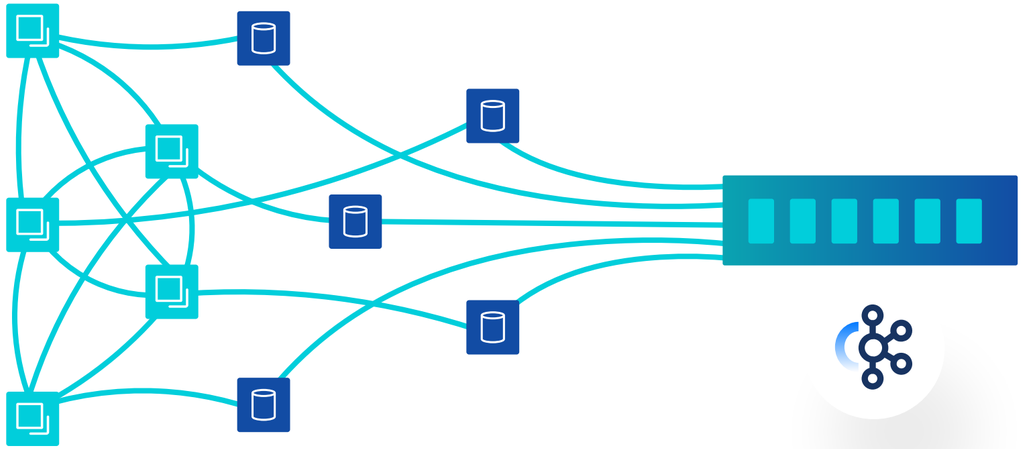
Flink is a top 5 Apache project and has a very active community

Complete Data Streaming Platform



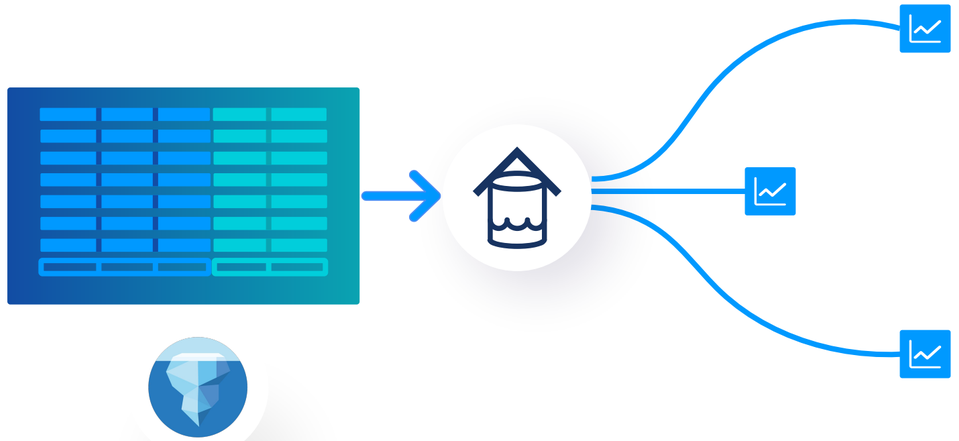
Bridging between Operation and Analytics

OPERATIONAL ESTATE



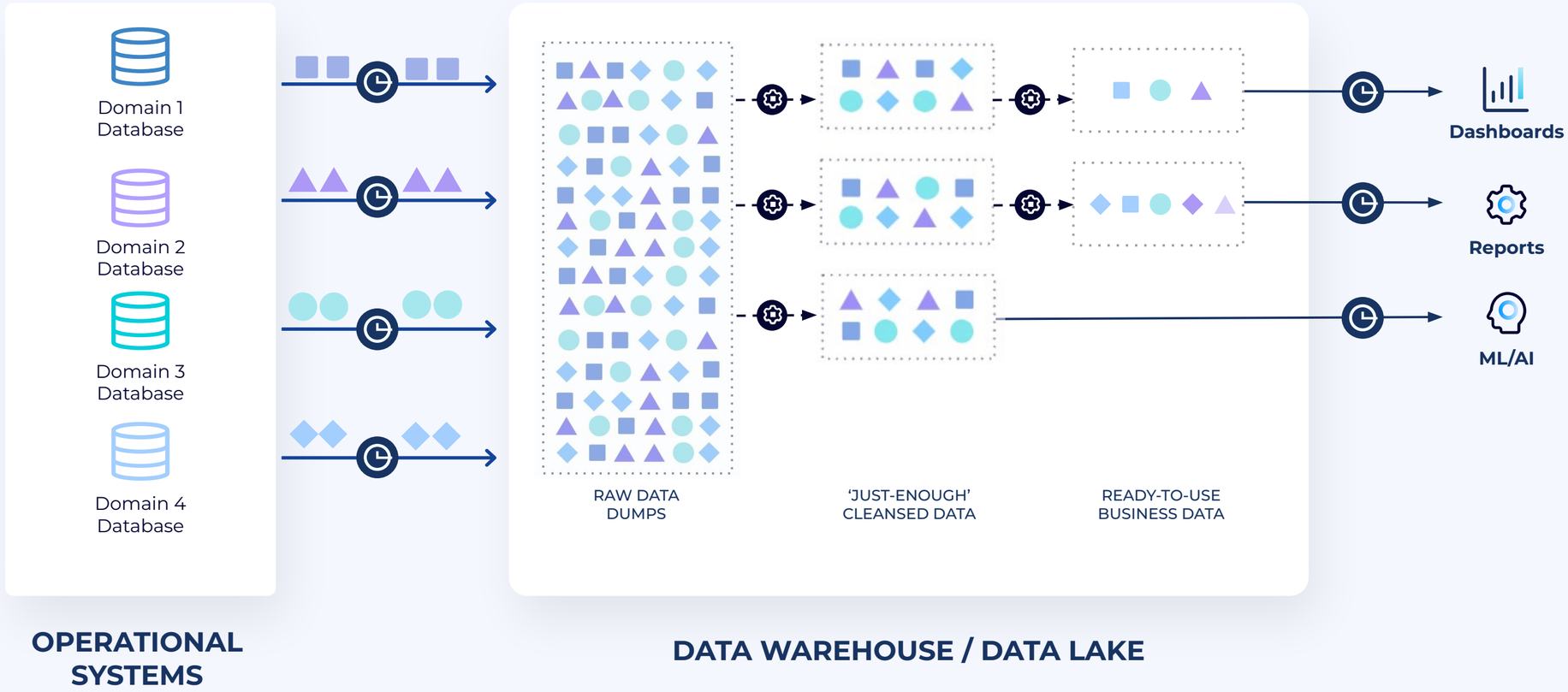
Apache Kafka is the standard to connect and organize business data as *data streams*

ANALYTICAL ESTATE

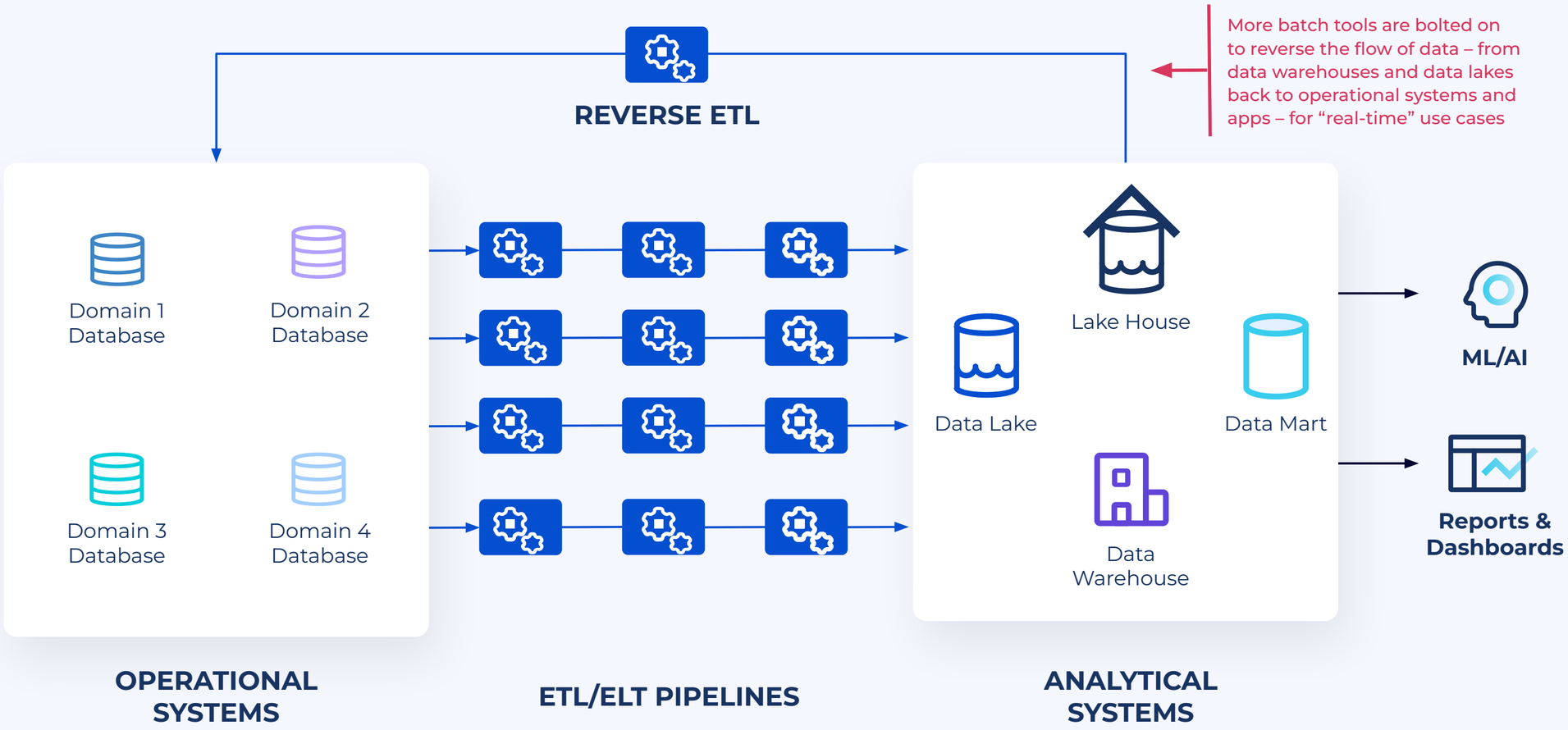


Apache Iceberg is the standard for managing *tables* that feed the analytical estate

The conventional Extract, Load, Transform (ELT) architecture

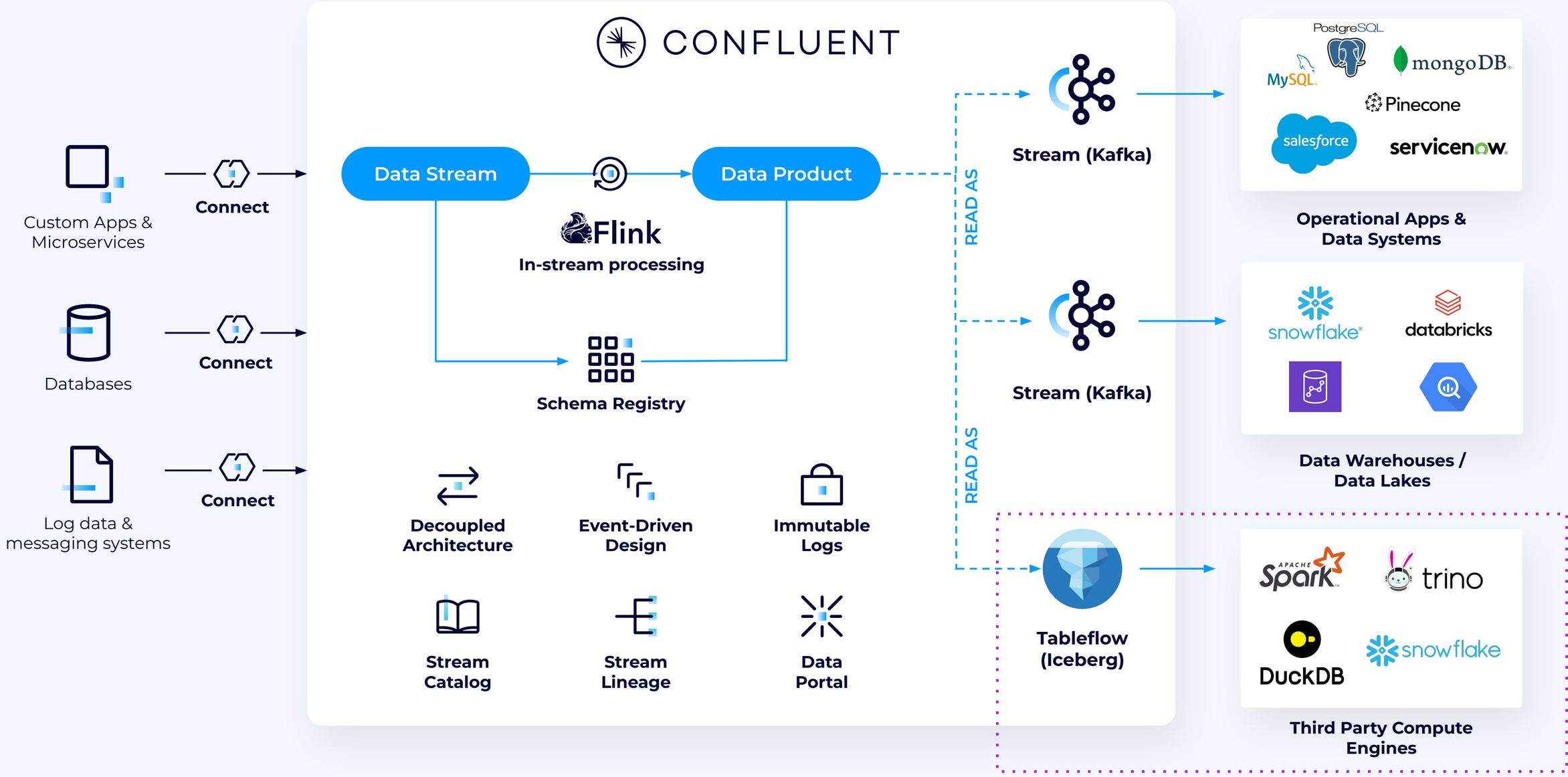


Modern applications need data to flow ‘upstream’ too

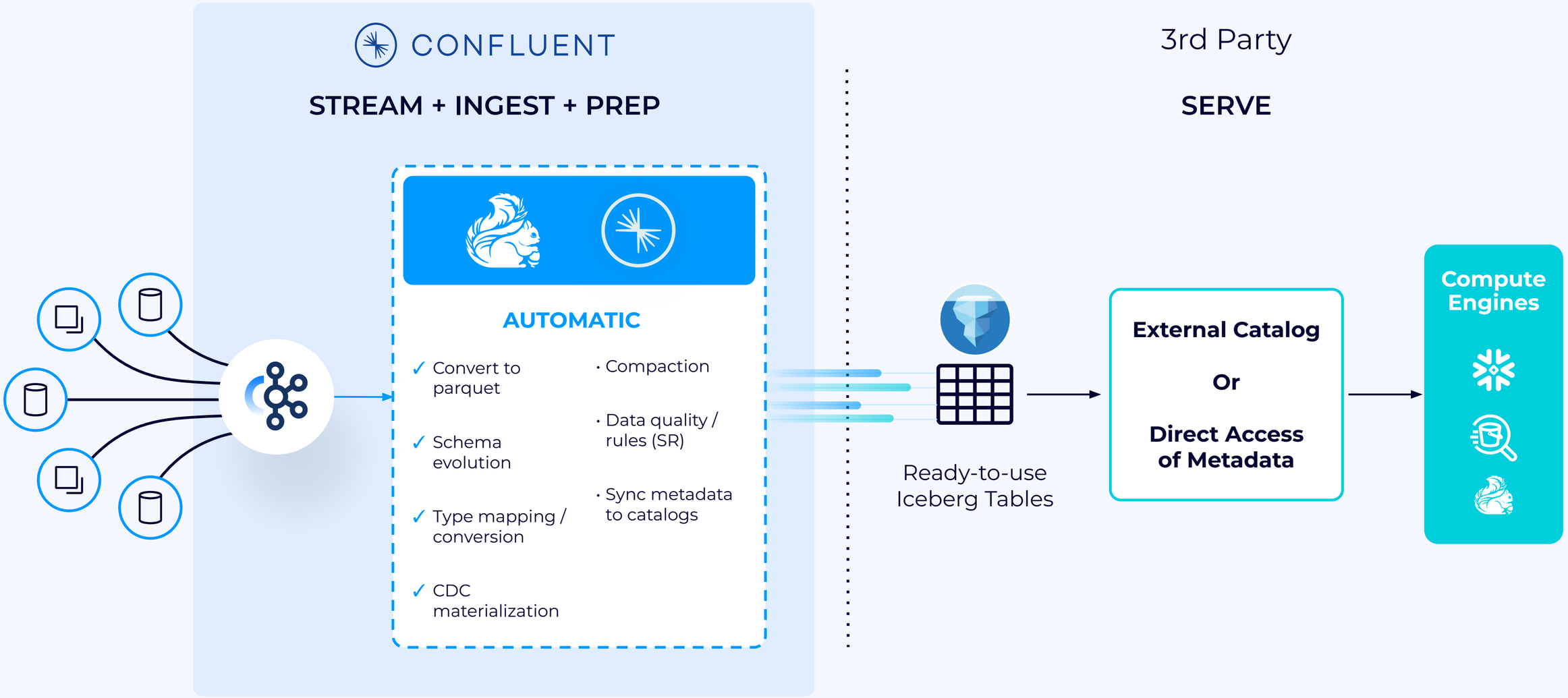


Shift Left with DSP

Write Your Data Once, Read It as a Stream or Table

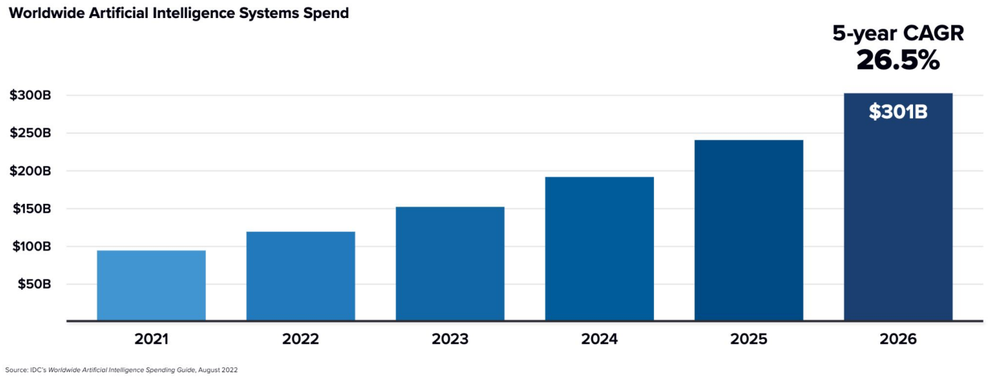


Confluent's Tableflow simplifies converting streaming data to Apache Iceberg tables



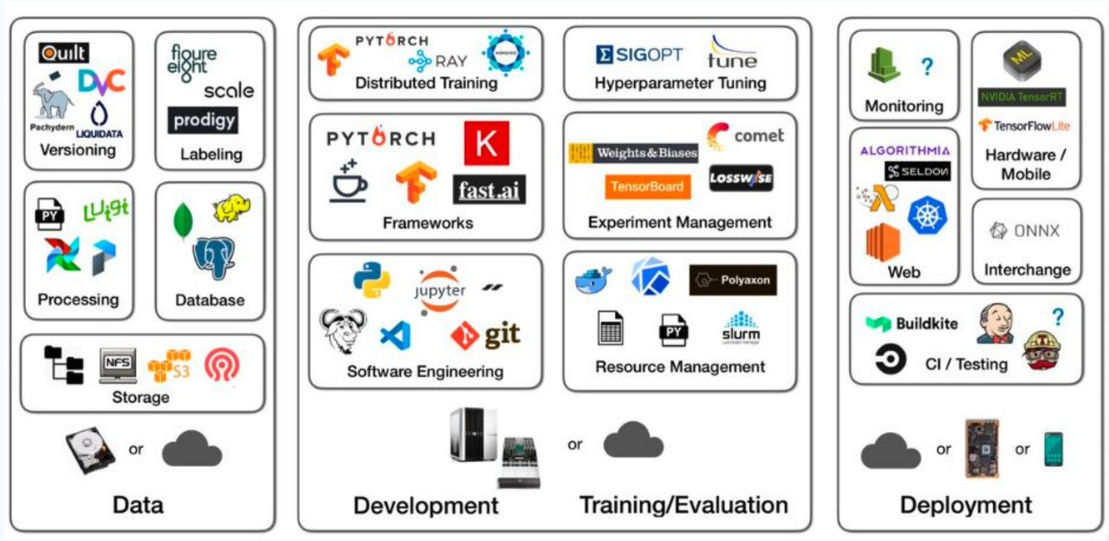
The Age of AI

Global Spending on AI to Exceed $301 Billion by 2026

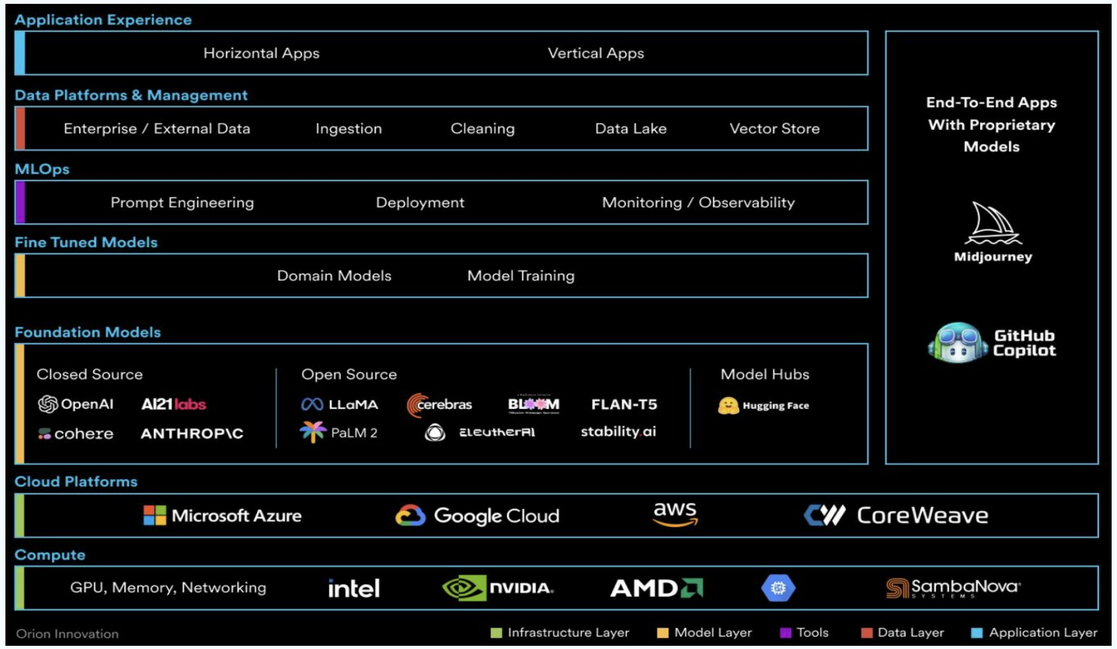


Source: IDC's *Worldwide Artificial Intelligence Spending Guide*, August 2022

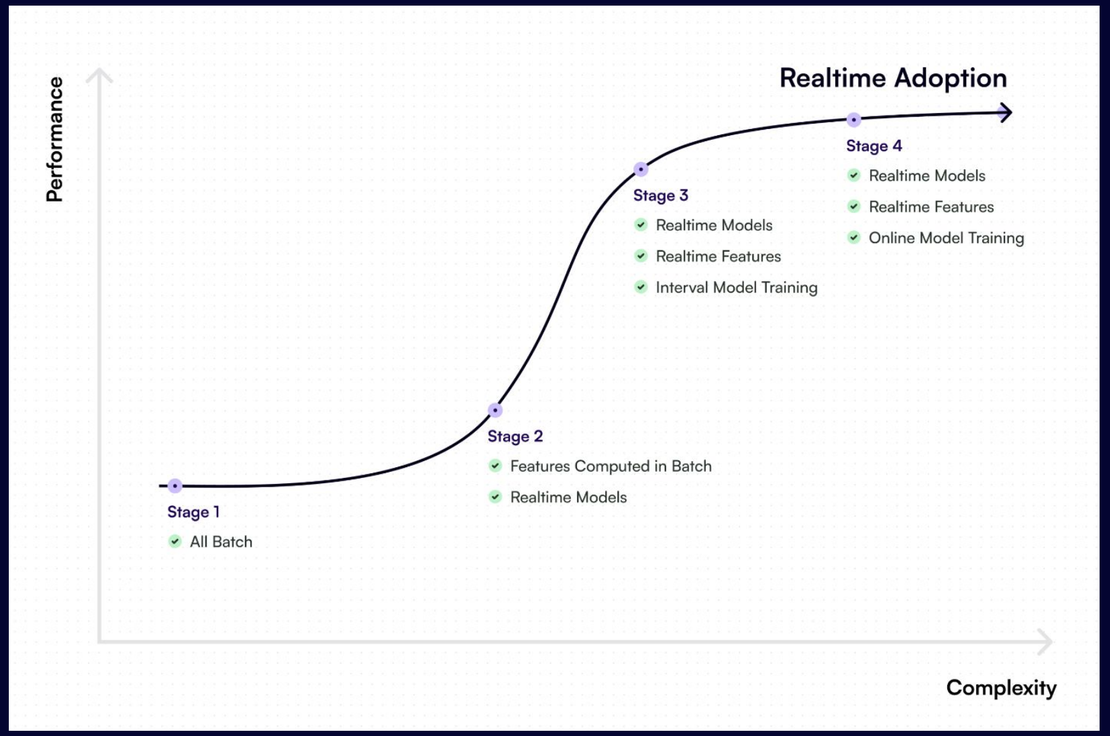
Predictive AI



Generative AI



Real-time AI



Data-Centric AI

- AI is another way to extract information from data
- Model tuning: fresh private data
- Inference needs to be real-time
  - Need real-time data for inputs and Retrieval-augmented generation (RAG)

Pages / Apache Flink Home / Flink Improvement Proposals

FLIP-437: Support ML Models in Flink SQL

Created by Martijn Visser, last modified by Timo Walther on Apr 05, 2024

Discussion thread	https://lists.apache.org/thread/9z94m2bv4w265xb5l2mrnh4lf9m28ccn
Vote thread	https://lists.apache.org/thread/9z94m2bv4w265xb5l2mrnh4lf9m28ccn
JIRA	FLINK-34992 - FLIP-437: Support ML Models in Flink SQL OPEN
Release	

Please keep the discussion on the mailing list rather than commenting on the wiki (wiki discussions get unwieldy fast).

Motivation

ML developers spend significant time on data cleaning, preprocessing, ingestion for ML training and inference with two sets of frameworks (e.g., Spark, Flink for data tasks, Tensorflow, PyTorch for ML tasks). Usually these frameworks are deployed in separate platforms, meaning developers have to rely on external orchestration systems and storage to stitch them into a cohesive workflow. Separating data processing tasks from the ML tasks also adds complexity to change management, data governance and lineage tracking etc. The rapid evolution of AI and GenAI is significantly influencing the data industry, steering it towards a unified streaming data platform architecture for almost all market players. In fact, ML is essentially another way of extracting insights from data, which logically is no different from the traditional data processing & analytics, but with more intensive computation requirements. Ideally there should be an unified set of APIs to describe the data processing and ML tasks for a more cohesive user experience. As the declarative APIs (SQL) is the common tongue for data processing and analytics, the natural evolution should be to add SQL support for ML tasks.

Public Interfaces

Public interfaces changes include new SQL syntax changes proposed below for model operations as well as new catalog model and catalog changes to operate on models.

Catalog Model (New)

```
/** Interface for a model in a catalog. */
@PublicEvolving
public interface CatalogModel {

    /**
     * Get the unresolved input schema of the model.
     *
     * @return unresolved input schema of the model.
     */
    Schema getInputSchema();
```

AI @ OSS Flink

```
CREATE MODEL `product_reviews_classifier`
INPUT (review STRING)
OUTPUT (rating INT)
WITH(
  'type' = 'remote',
  'task' = 'classification',
  'provider' = 'OPENAI',
  'endpoint' = 'https://api.openai.com/v1/llm/v1/chat',
  'api_key' = 'my_key',
  'prompt' = 'generate a rating between 1 to 5 for the product review'
)
```

```
INSERT INTO PredictionResults
SELECT review, rating
FROM product_reviews,
LATERAL TABLE(ML_PREDICT(`product_reviews_classifier`, review, rating))
```

Remote Model EA

CONFLUENT

Supported Features

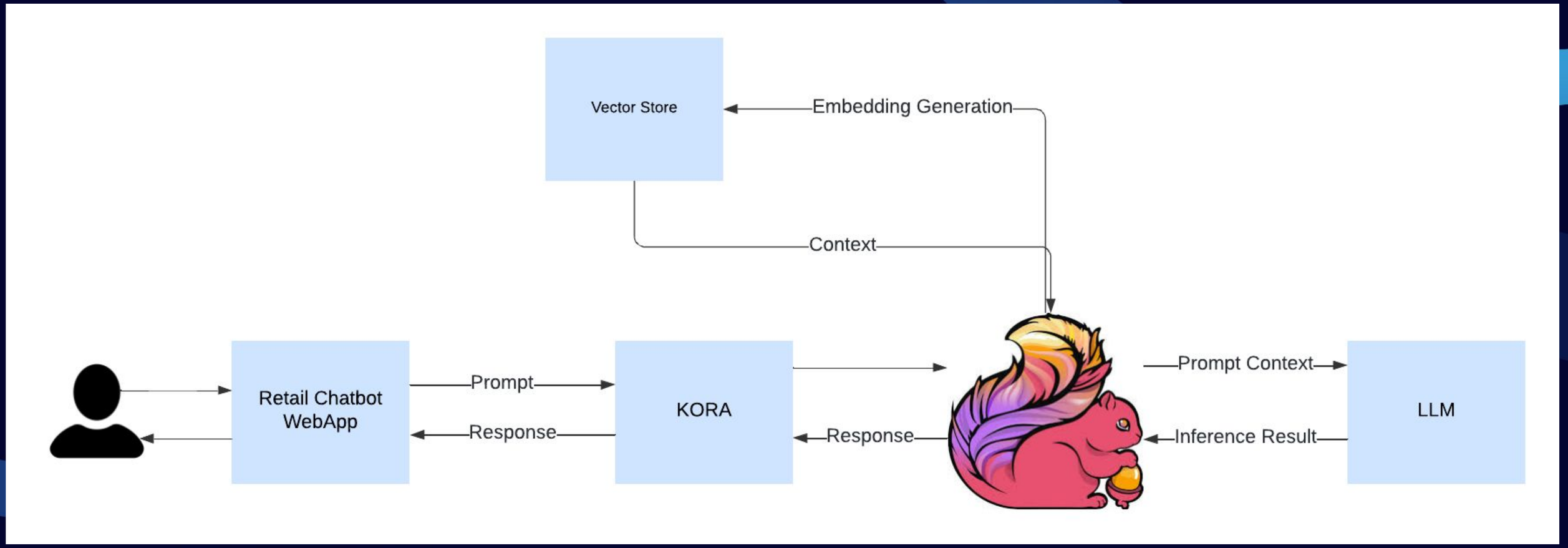
- Model DDL (Create / Delete / Alter)
- Model Inference (ML_PREDICT)
- Model Evaluation (ML_EVALUATE)
- Model Versioning

Supported AI Platforms

- OpenAI
- AWS Sagemaker, bedrock
- Azure ML
- GCP Vertex AI, Google Gemini AI

CONFLUENT

LLM - RAG on Confluent Cloud



Thank You and Happy Streaming