Потоковая обработка данных (Kafka, Spark Streaming, Flink)

Андрей Кузнецов

03.12.2022

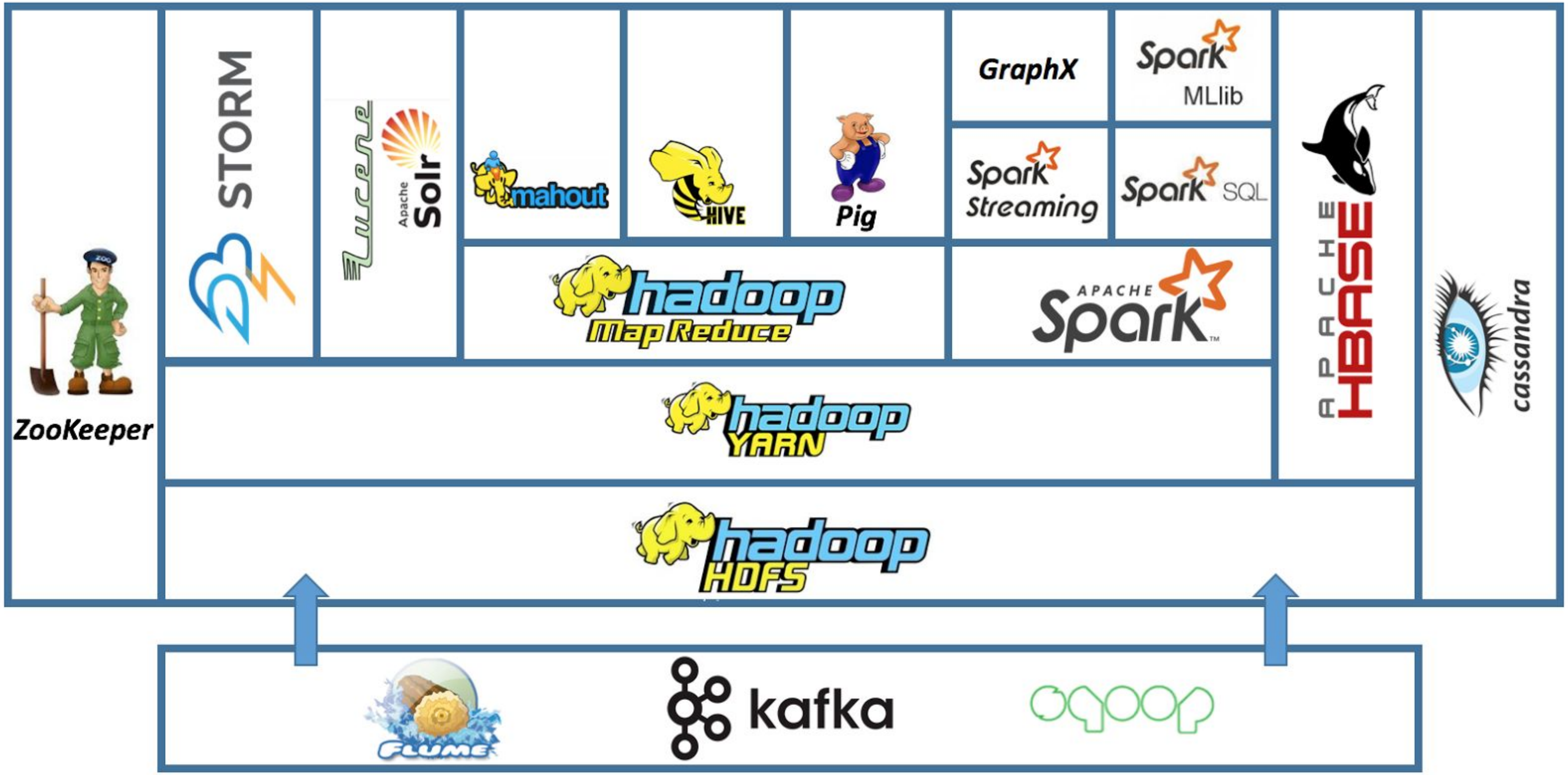
Структура курса

1. Введение в Большие Данные
2. Hadoop экосистема и MapReduce
3. SQL поверх больших данных
4. Инструменты визуализации при работе с Большими Данными
5. Введение в Scala
6. Устройство и API Spark
7. Approximate алгоритмы для больших данных
8. Потоковая обработка данных (Kafka, Spark Streaming, Flink)
9. Основы распределённой СУБД Apache Cassandra

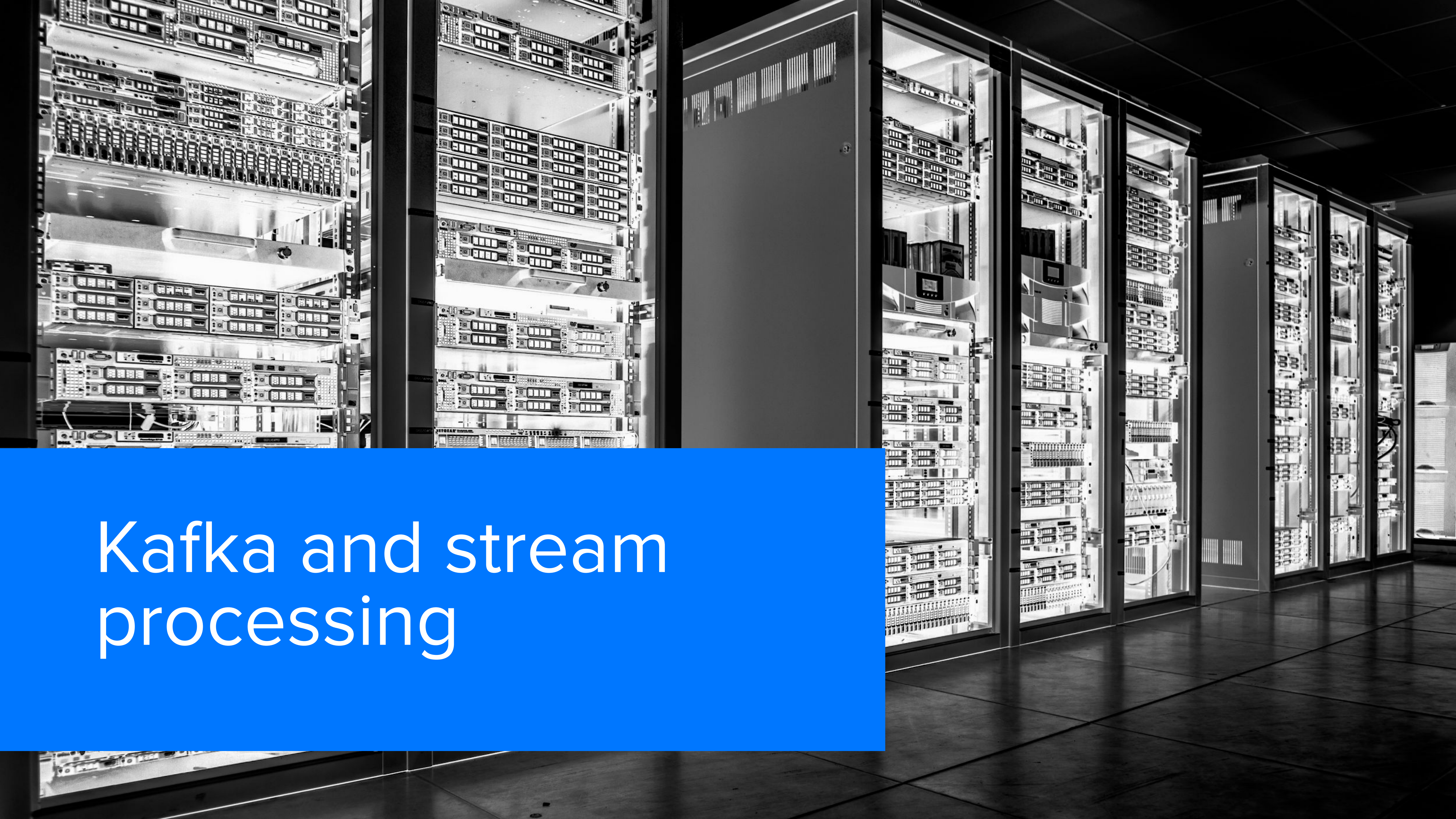
План занятия

1. Kafka and stream processing
2. Spark streaming
3. Workshop

Hadoop ecosystem



Kafka and stream processing



Streaming

Table 1.1 Classification of real-time systems

Classification	Examples	Latency measured in	Tolerance for delay
Hard	Pacemaker, anti-lock brakes	Microseconds–milliseconds	None—total system failure, potential loss of life
Soft	Airline reservation system, online stock quotes, VoIP (Skype)	Milliseconds–seconds	Low—no system failure, no life at risk
Near	Skype video, home automation	Seconds–minutes	High—no system failure, no life at risk

Event streams:

- Event streams are ordered
- Immutable data records
- Event streams are replayable

Event processing:

- Request-response
- Batch processing
- Stream processing

Streaming applications

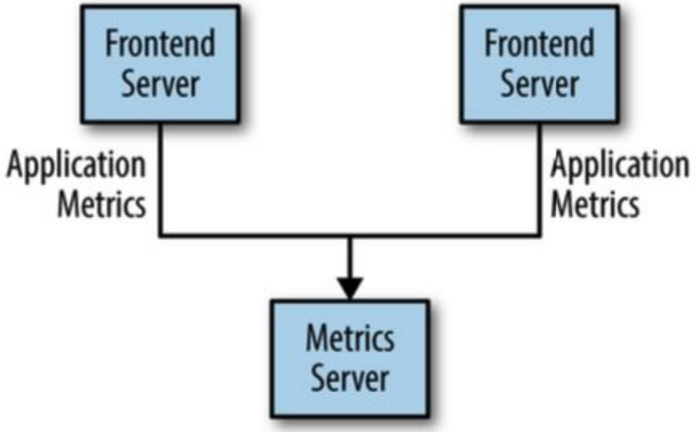
1. Device monitoring
2. Fault detection
3. Media recommendations
4. Faster loans
5. Fraud detection

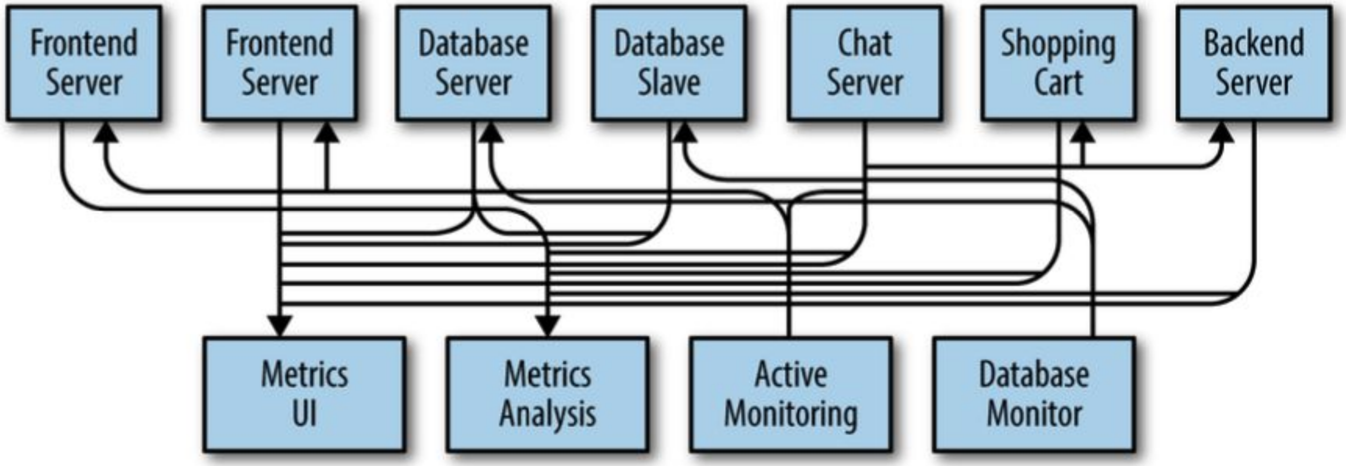
Why streaming?

Streaming data processing is a big deal in big data these days, and for good reasons; among them are the following:

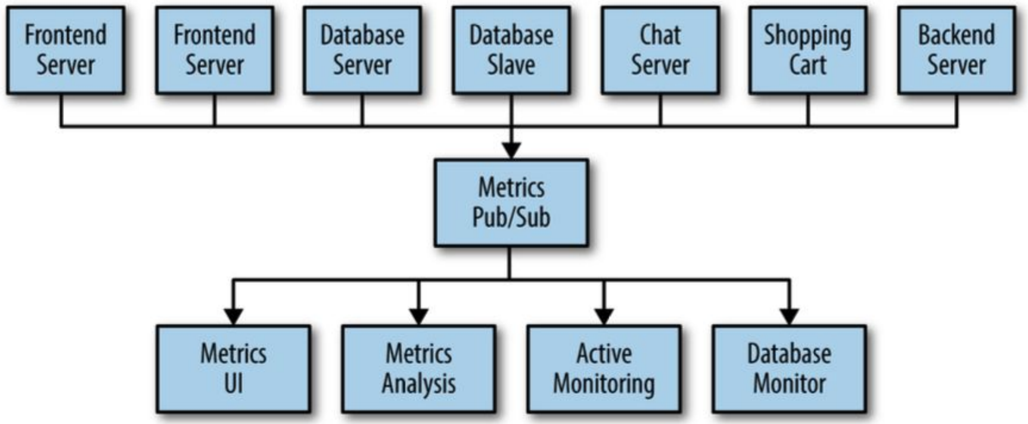
1. Businesses crave ever-more timely insights into their data, and switching to streaming is a good way to achieve lower latency
2. The massive, unbounded datasets that are increasingly common in modern business are more easily tamed using a system designed for such never-ending volumes of data.
3. Processing data as they arrive spreads workloads out more evenly over time, yielding more consistent and predictable consumption of resources.

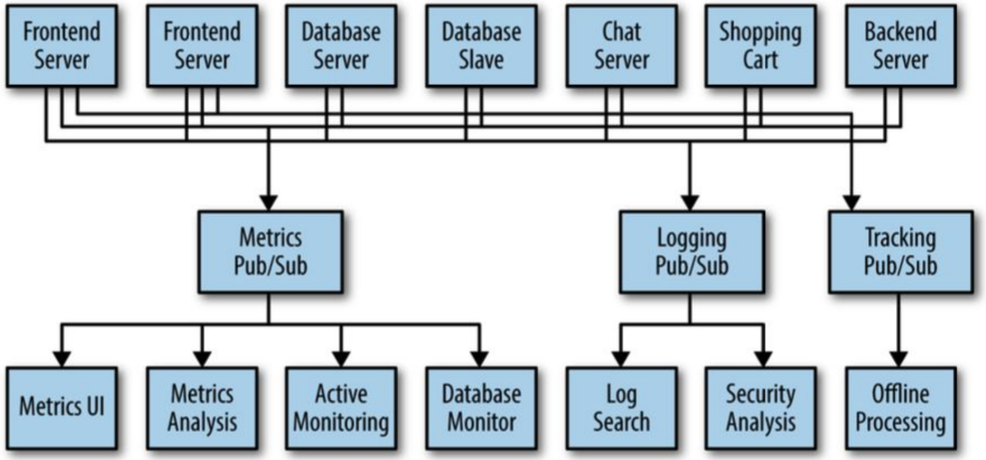
Real-time systems



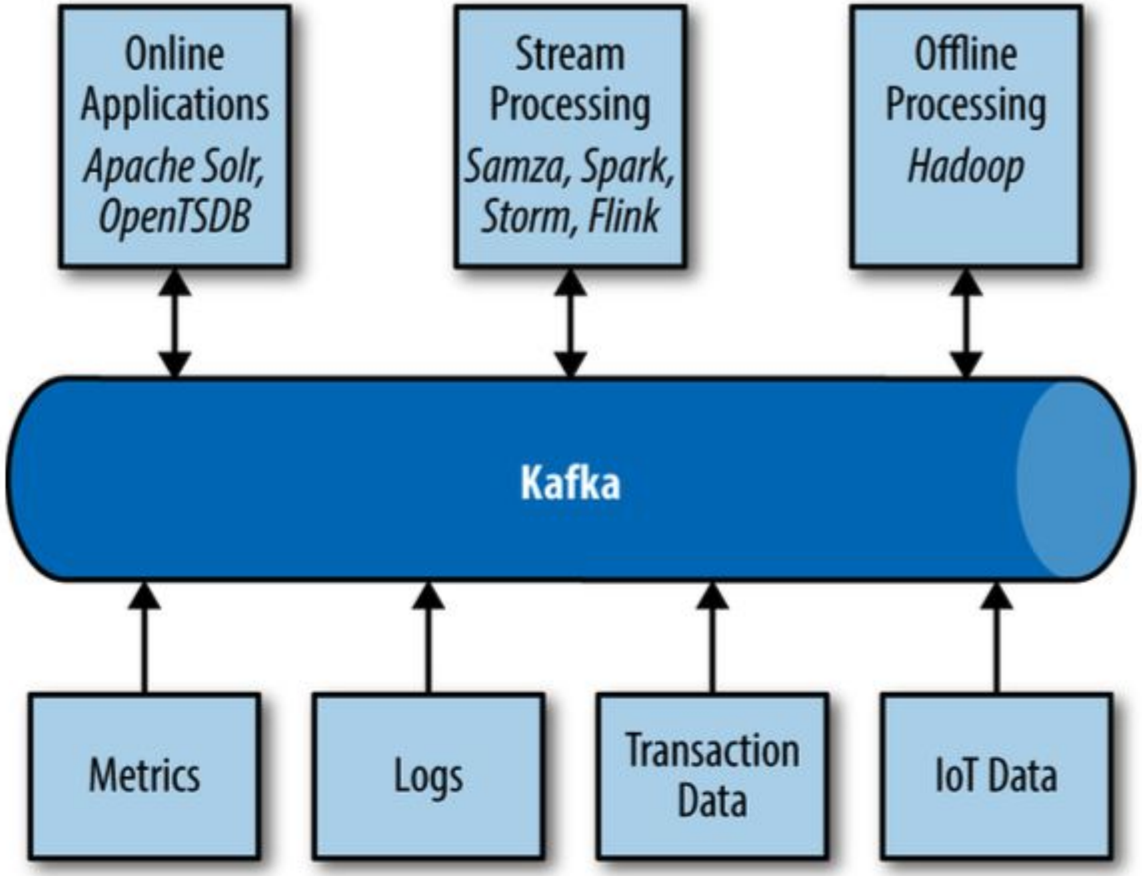


Real-time systems



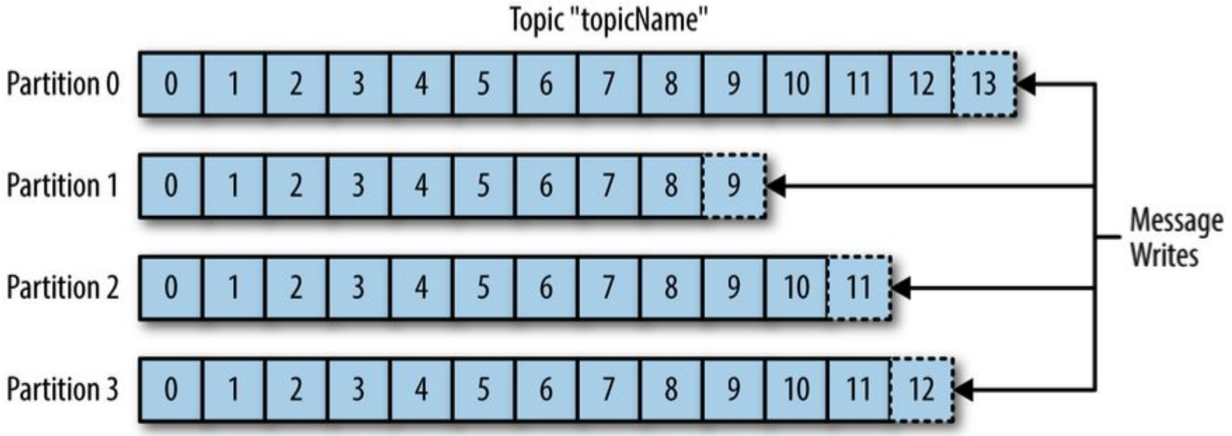


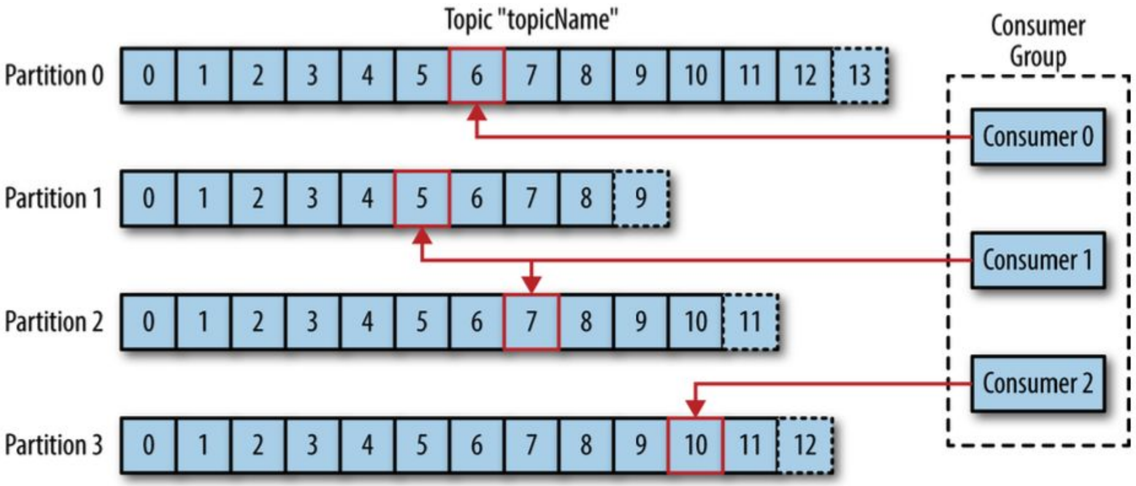
Kafka



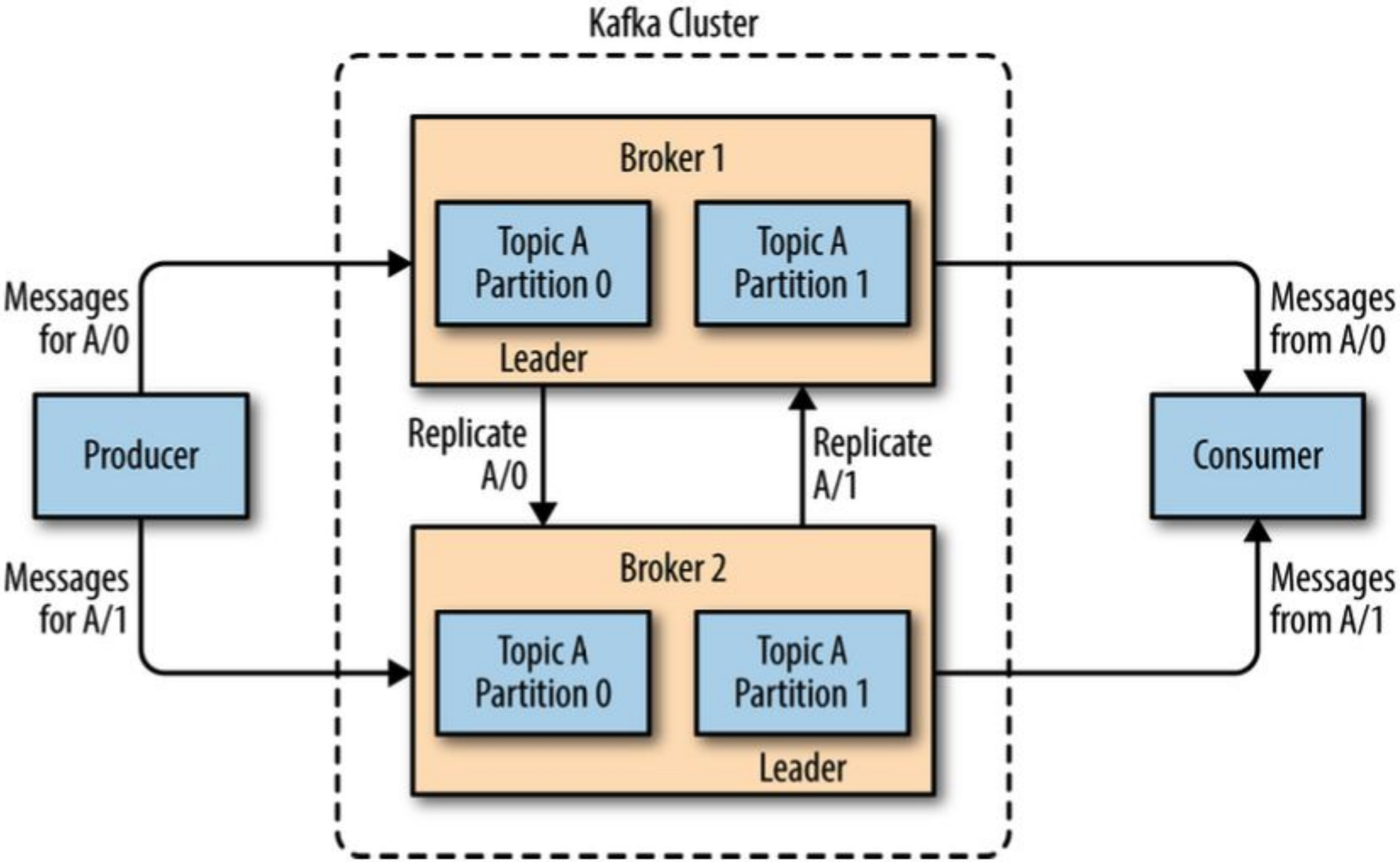
Online Applications
Apache Solr, OpenTSDB
Stream Processing
Samza, Spark, Storm, Flink
Offline Processing
Hadoop
Kafka
Metrics
Logs
Transaction Data
IoT Data

Kafka topics

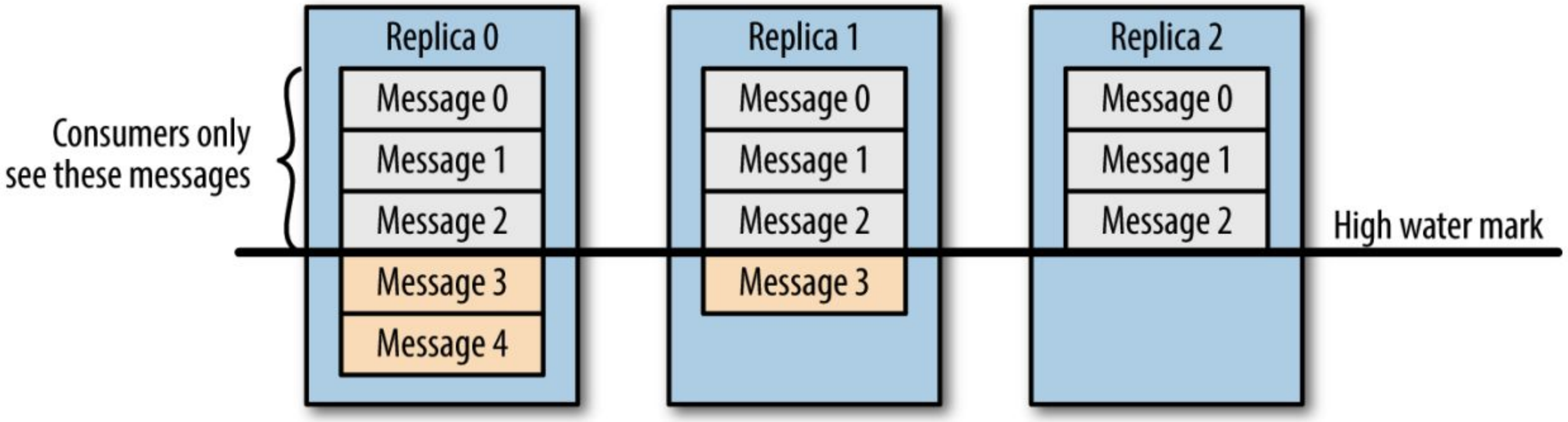




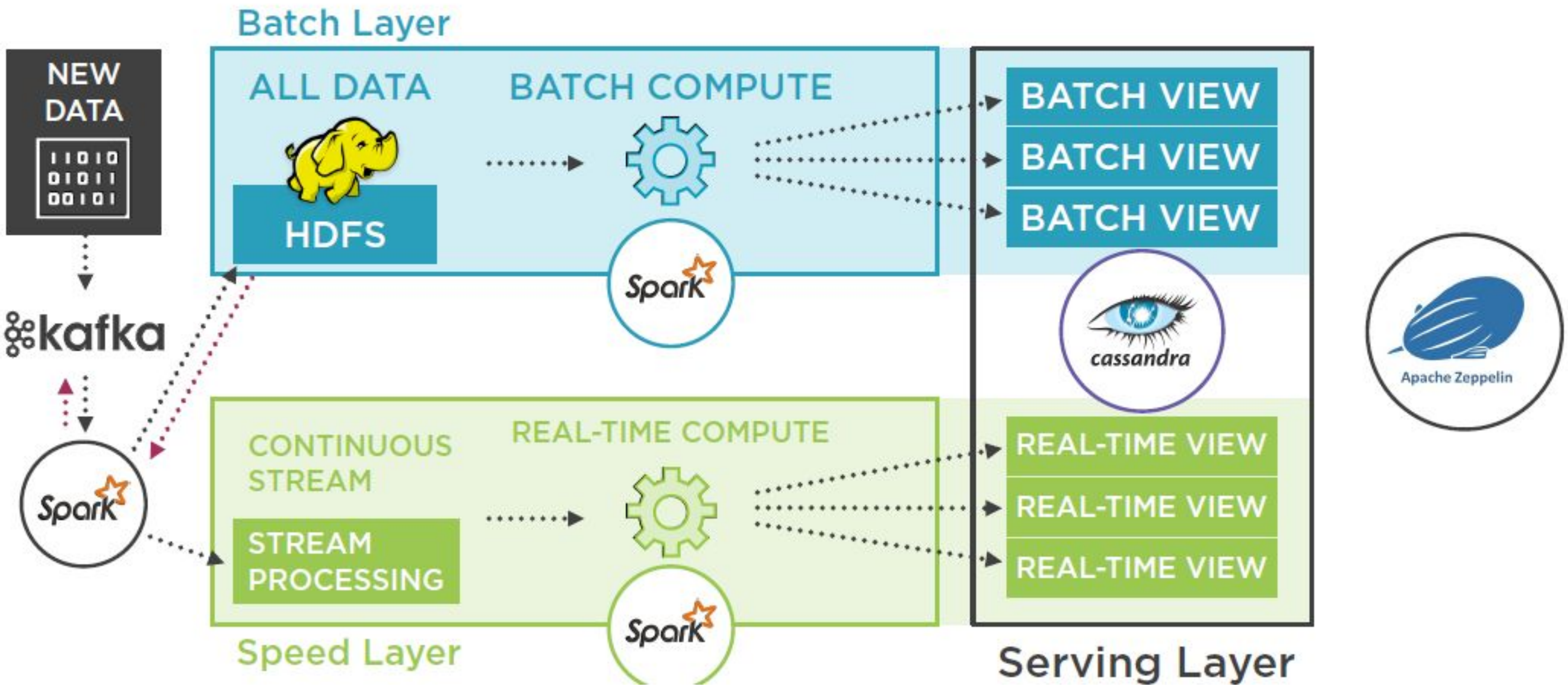
Kafka overview



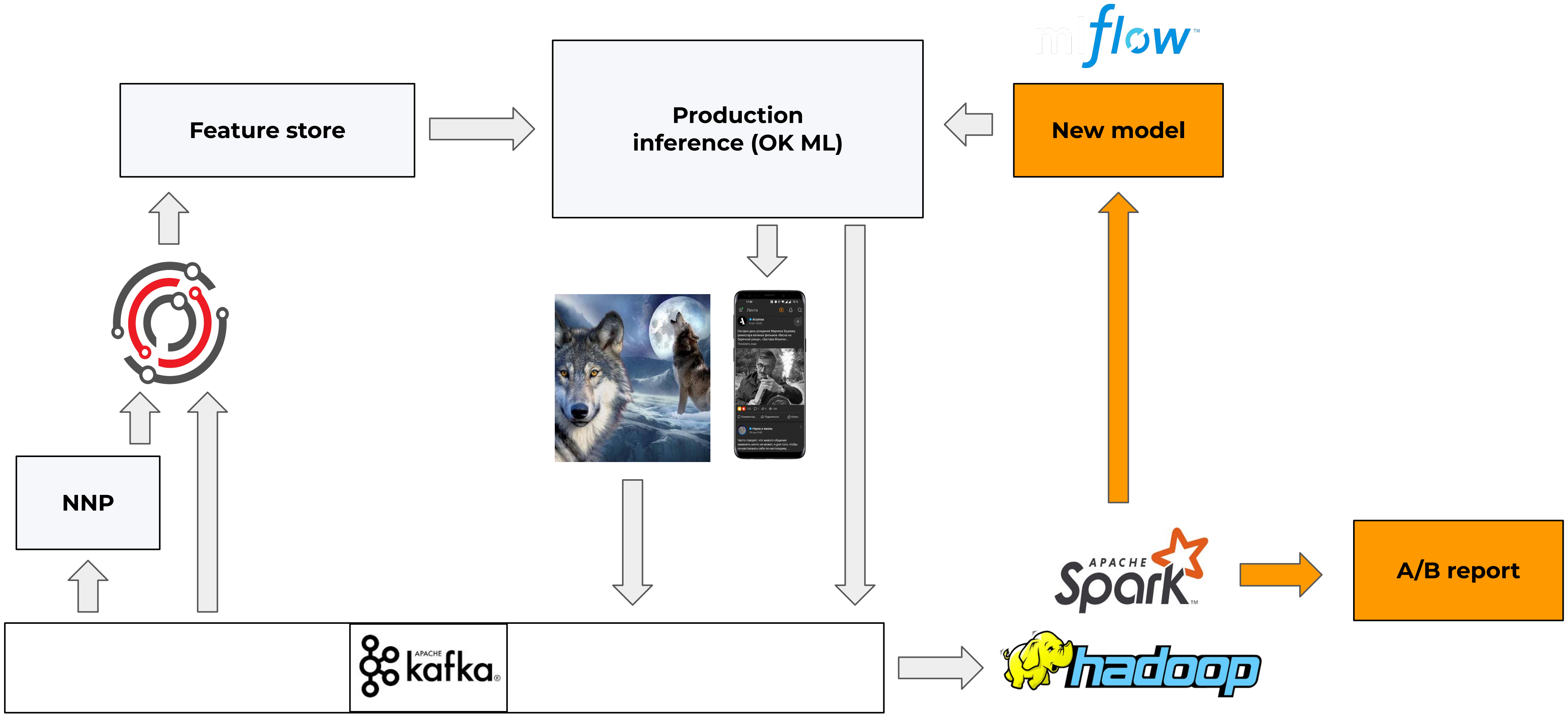
Kafka replication



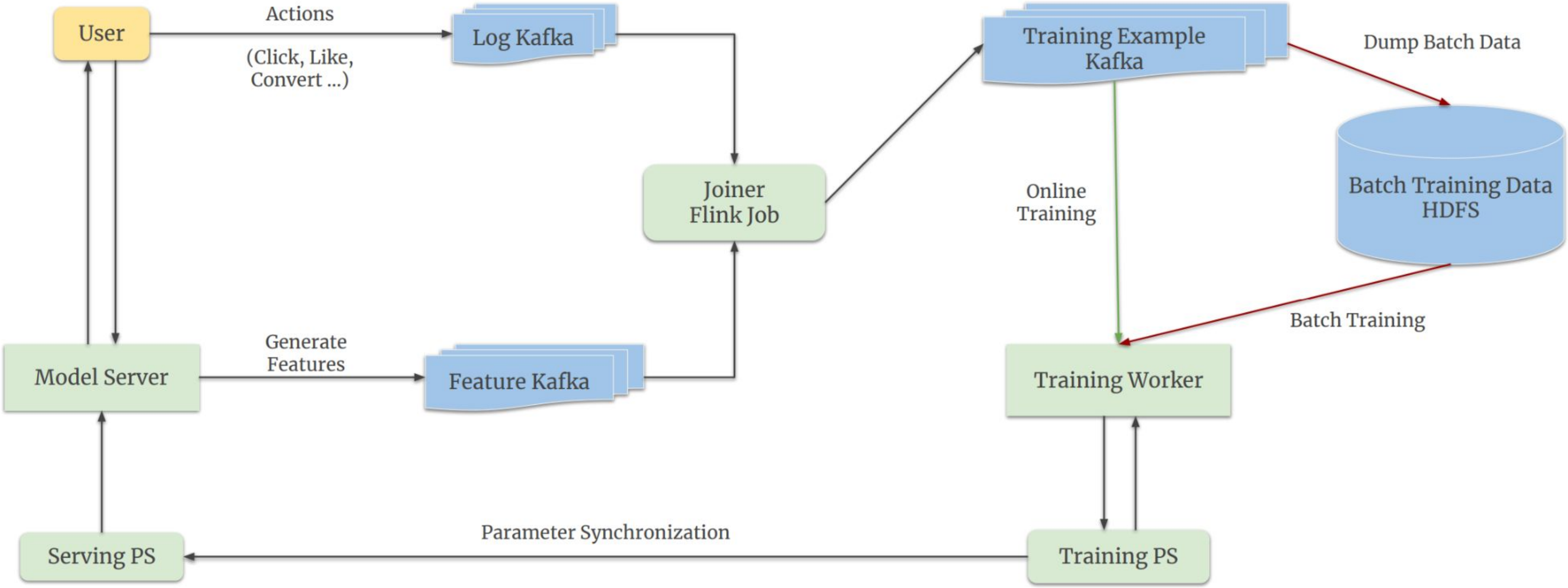
Processing layers. Lambda



OK Feed ML architecture



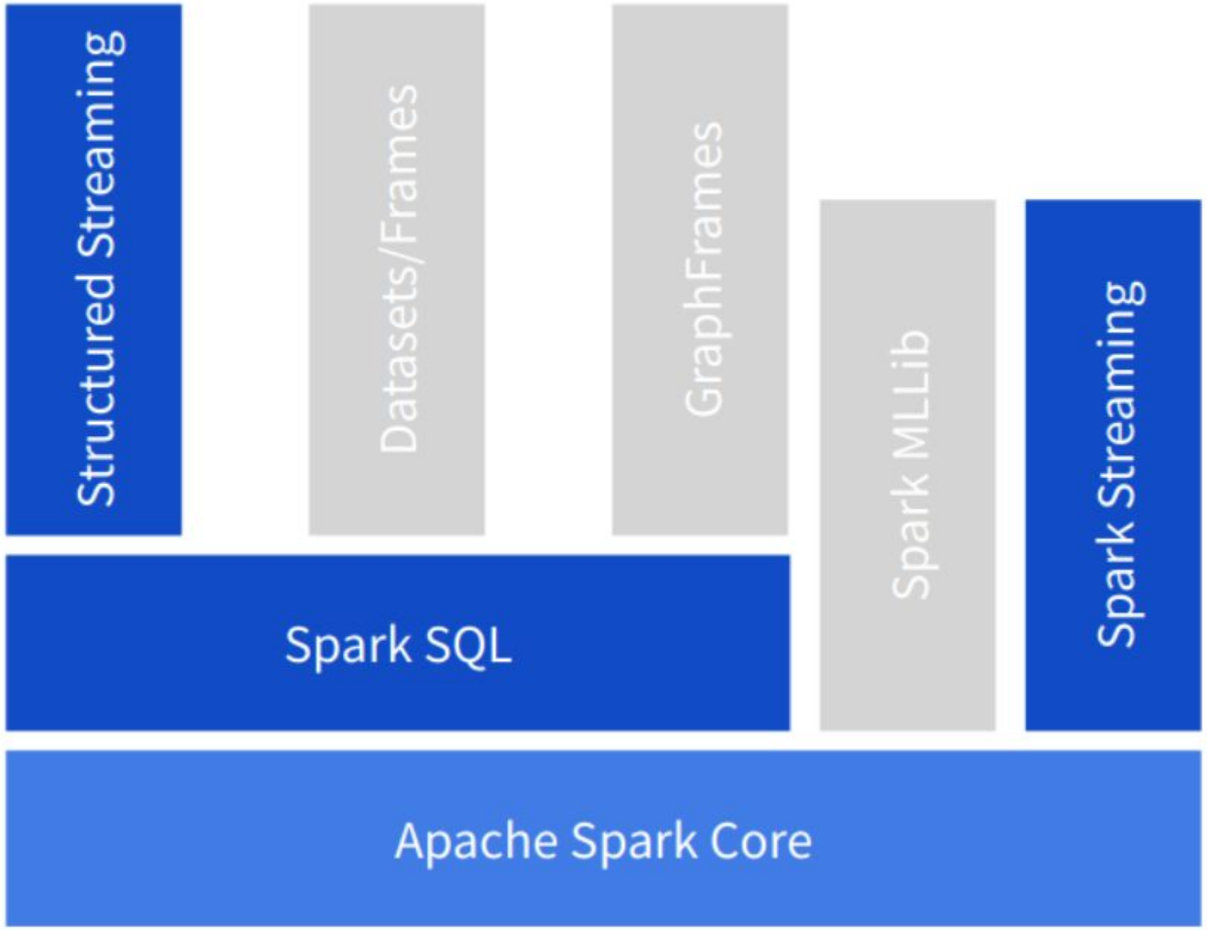
TikTok reco arch



https://arxiv.org/pdf/2209.07663.pdf

Spark streaming

Spark streaming



Streaming generic arch

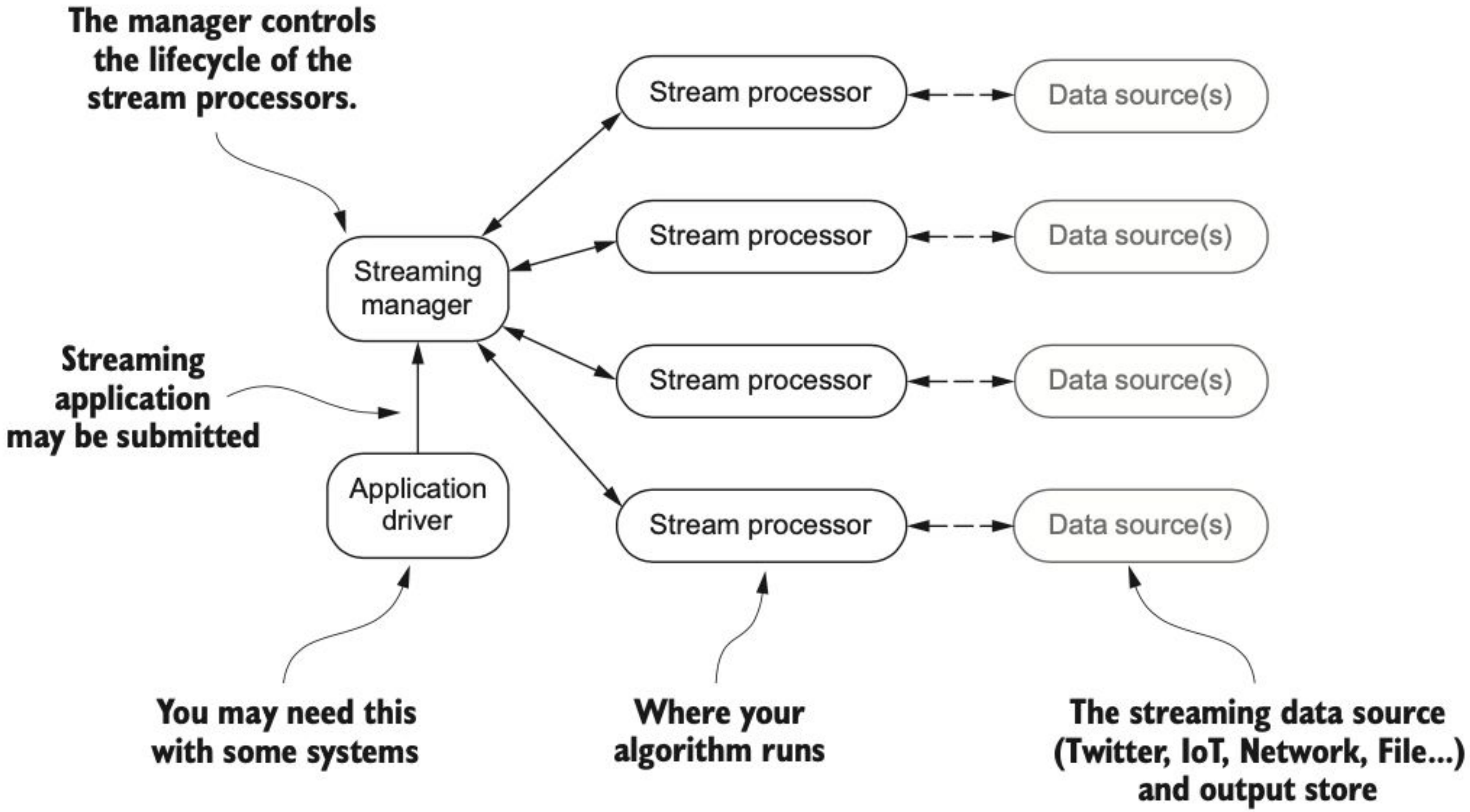


Figure 4.4 Generic streaming analysis architecture you will find with many products on the market

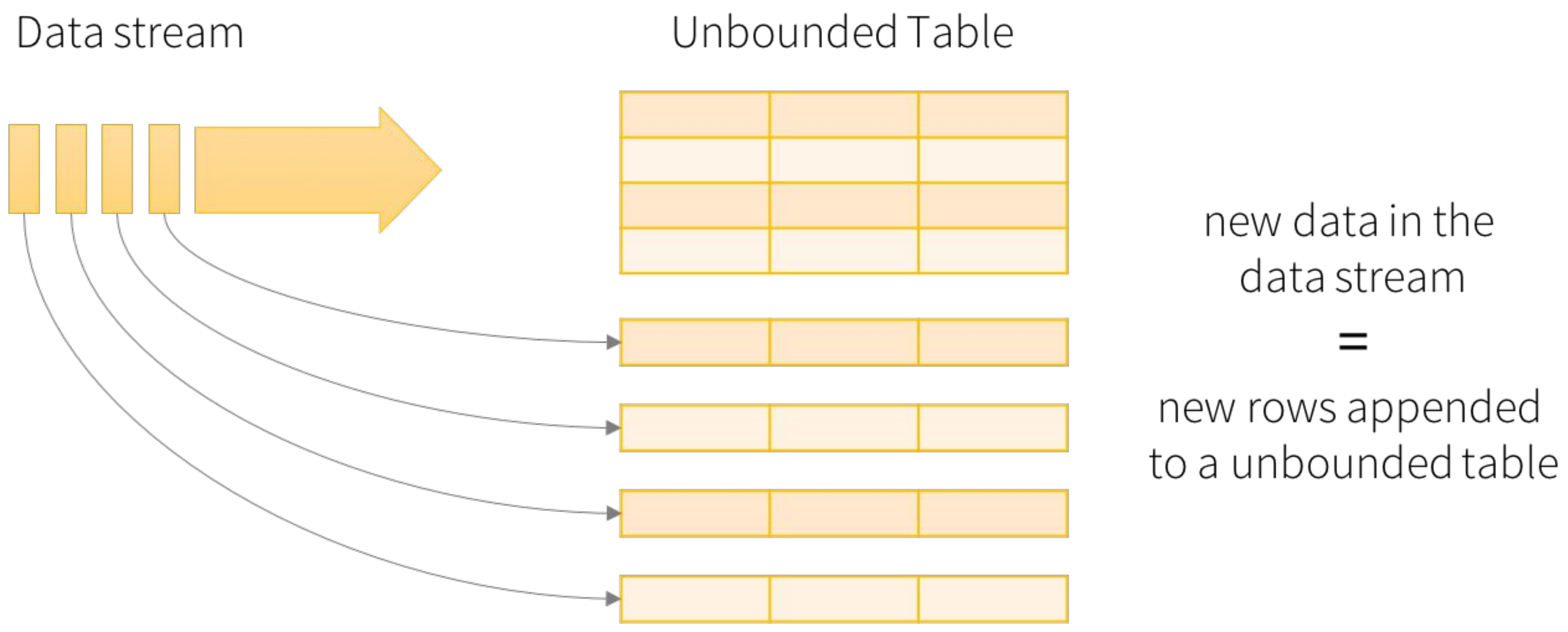
Streaming pipeline

- File source
- Kafka source
- Socket source (for testing)
- Rate source (for testing)

- Transformations
- Aggregations

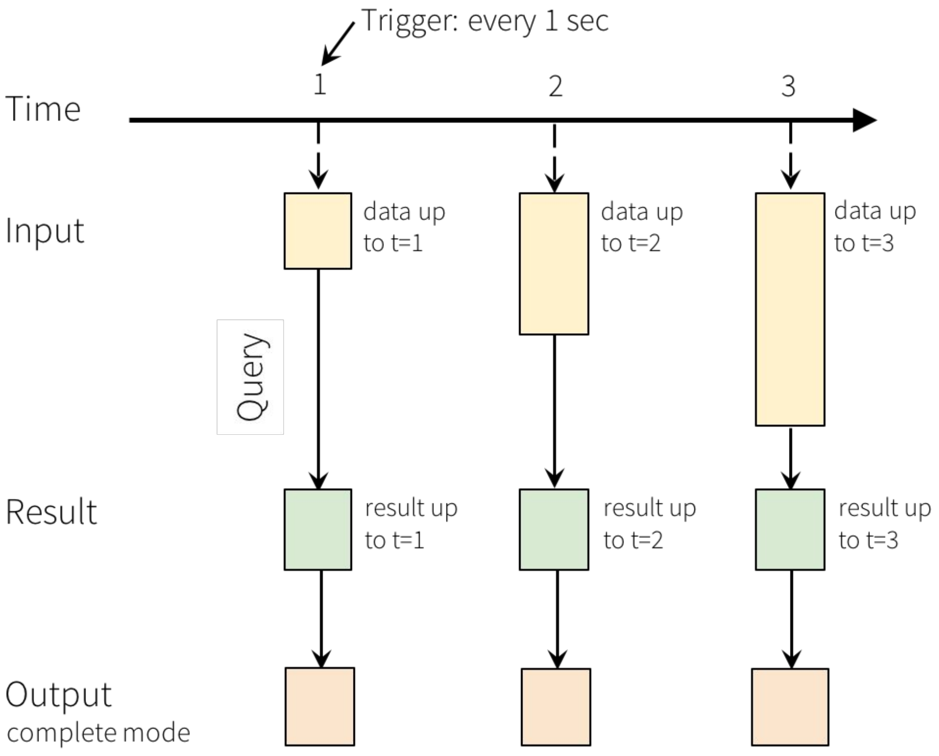
- File source
- Kafka source
- Socket source (for testing)
- Rate source (for testing)

Spark streaming abstract model



Data stream as an unbounded table

Spark streaming modes



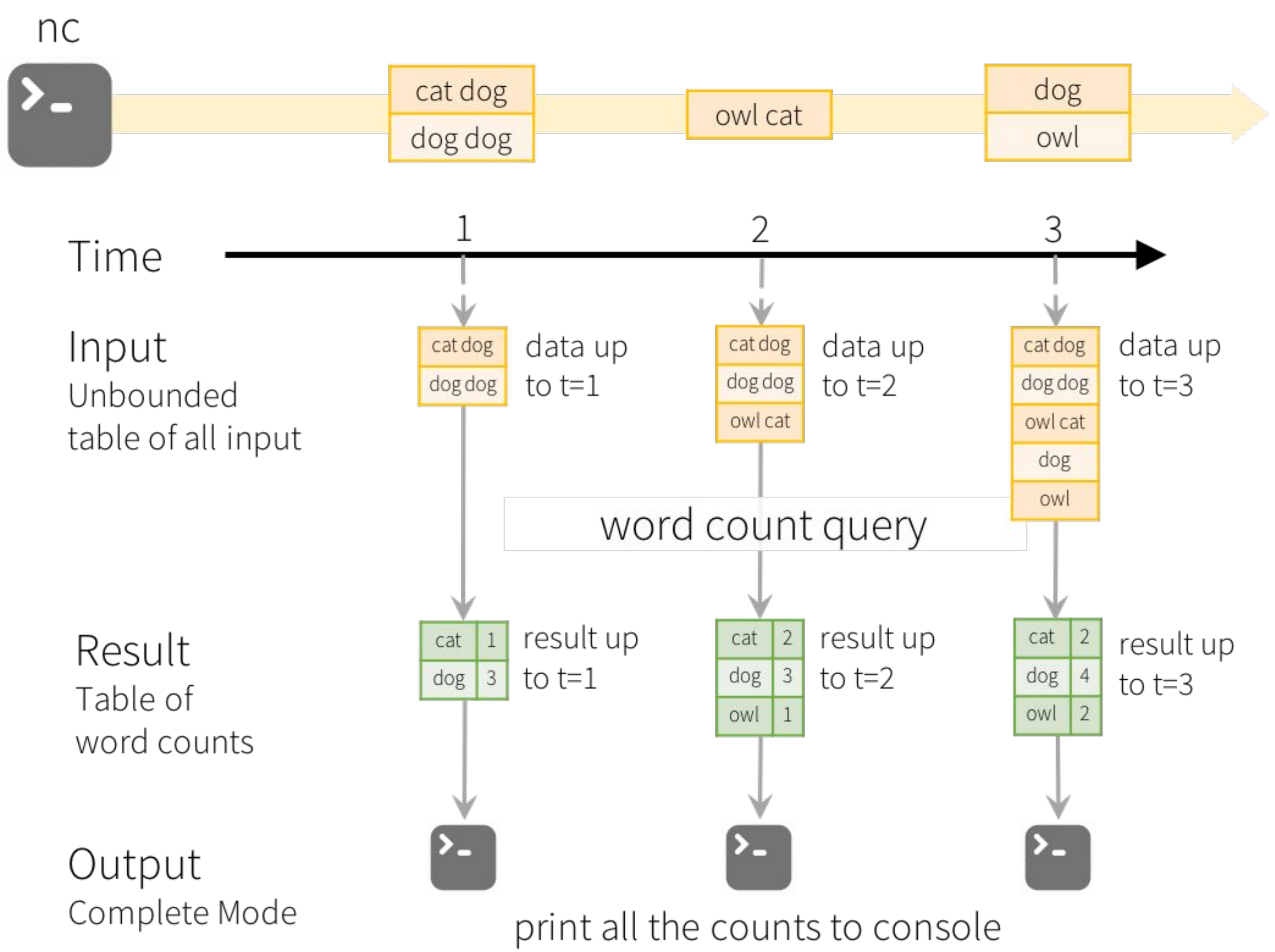
Programming Model for Structured Streaming

Complete Mode - The entire updated Result Table will be written to the external storage.

Append Mode - Only the new rows appended in the Result Table since the last trigger will be written to the external storage.

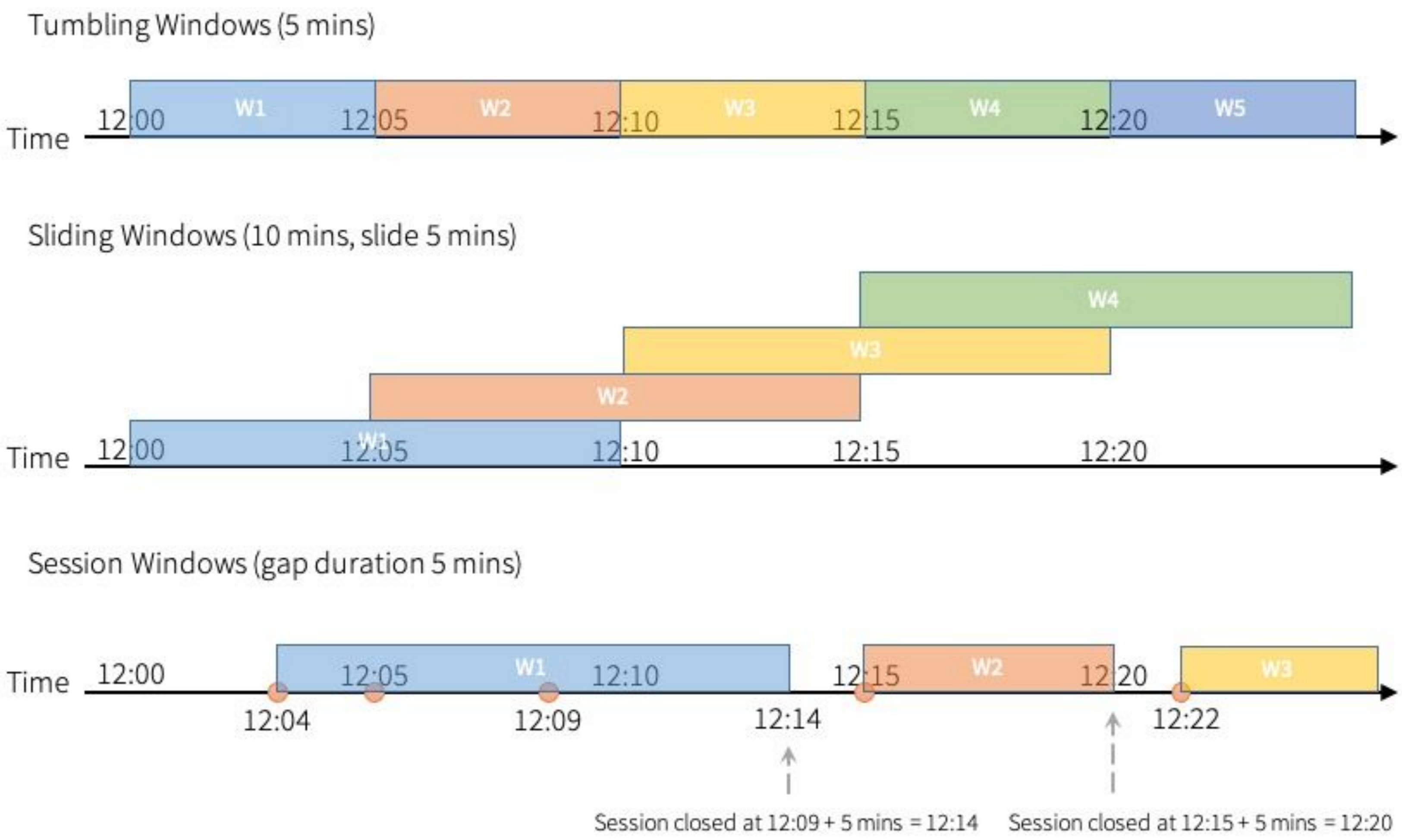
Update Mode - Only the rows that were updated in the Result Table since the last trigger will be written to the external storage.

Spark streaming example



Model of the Quick Example

Spark streaming windows

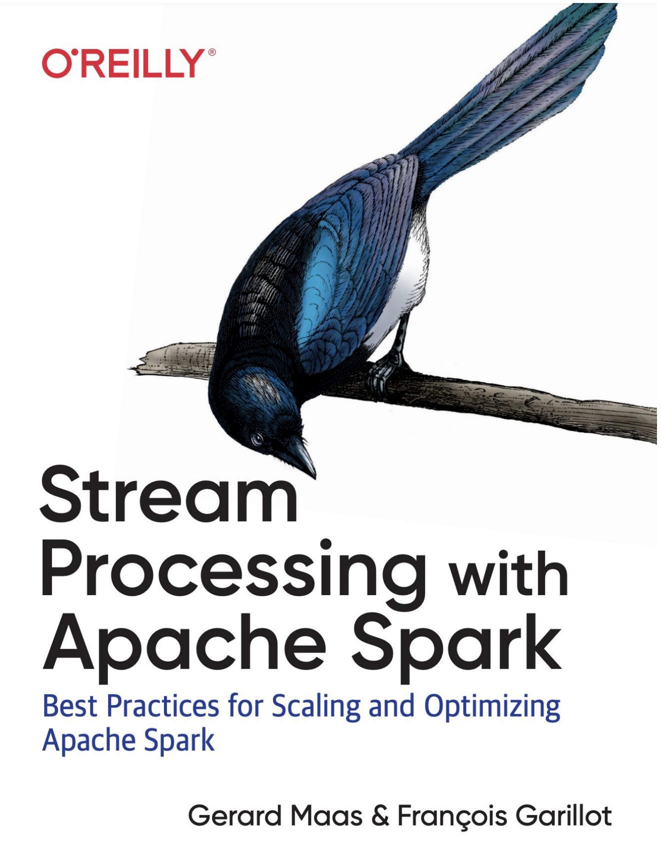


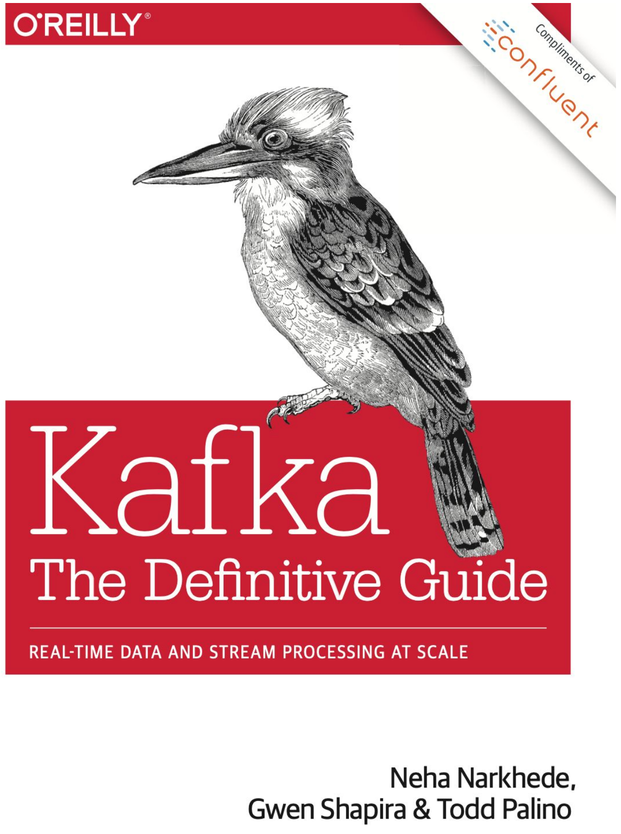
Alternative streaming frameworks

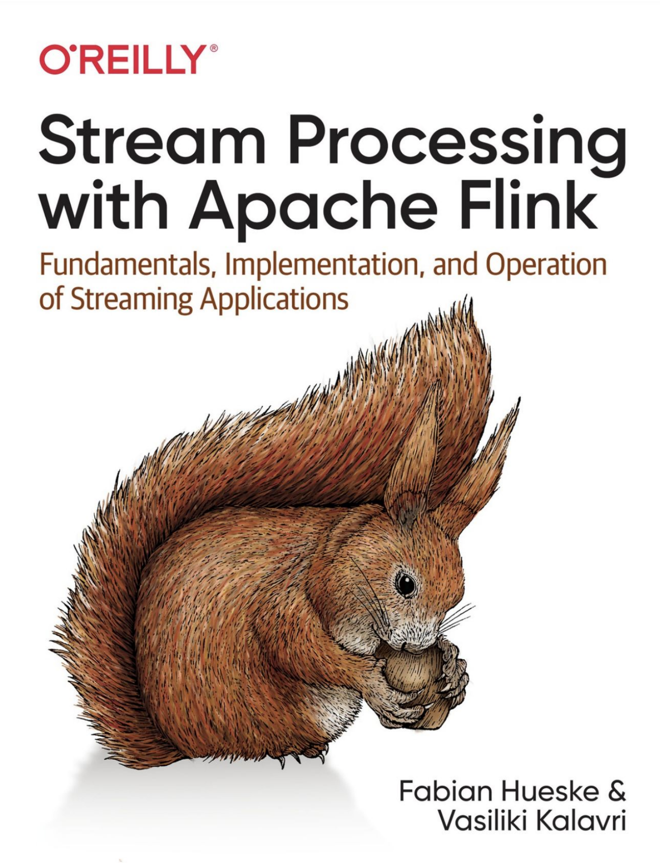
- Apache Storm
- *Apache Samza*
- **Apache Flink**
- Amazon Kinesis Streams
- Apache Apex
- Apache Flume

Workshop

Recommended literature







Recommended literature