

Stream Processing Technology

Streaming Data Processing
Prajwol Sangat

Updated by Ting Chee Ming

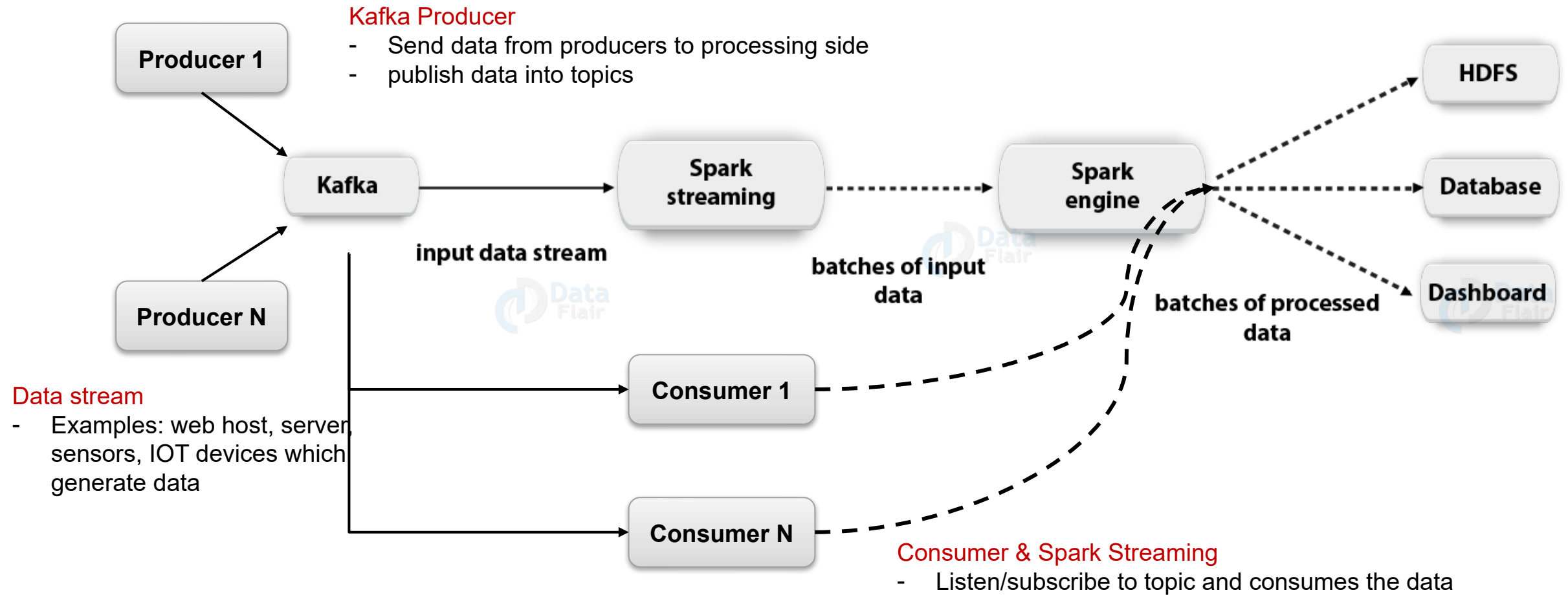


Streaming Platforms



Real-Time Streaming Architecture

Kafka-Spark Streaming Integration

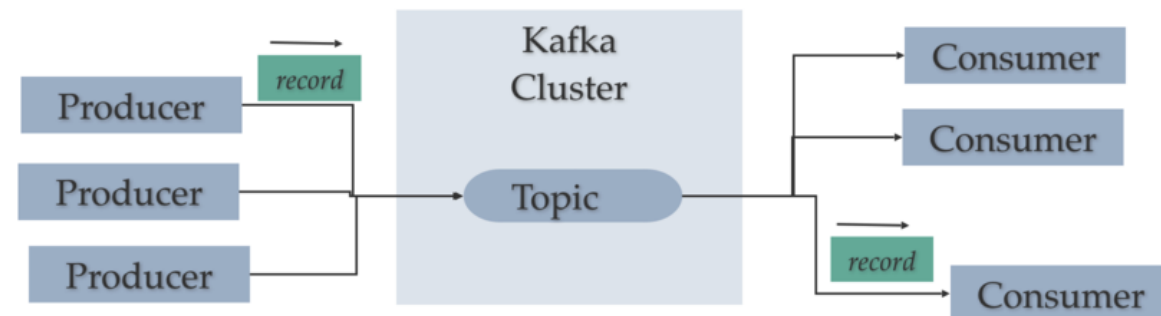


What is Apache Kafka?

- **Publish-subscribe messaging system**
- Scalable, Fault-tolerant
- Enables distributed applications
- Powers web-scale Internet companies such as LinkedIn, NetFlix, AirBnB, and many others.

- ❑ **Producer** publish streams of data records into topics
- ❑ **Consumers** subscribe to the topics and process the stream of records
- ❑ Data in Kafka is stored in topic
 - **Topic** is category/feed name where records are stored
 - A topic is associated with a **log** – data structure on disk
 - Each topic is indexed and stored with timestamp

Kafka: Topics, Producers, and Consumers



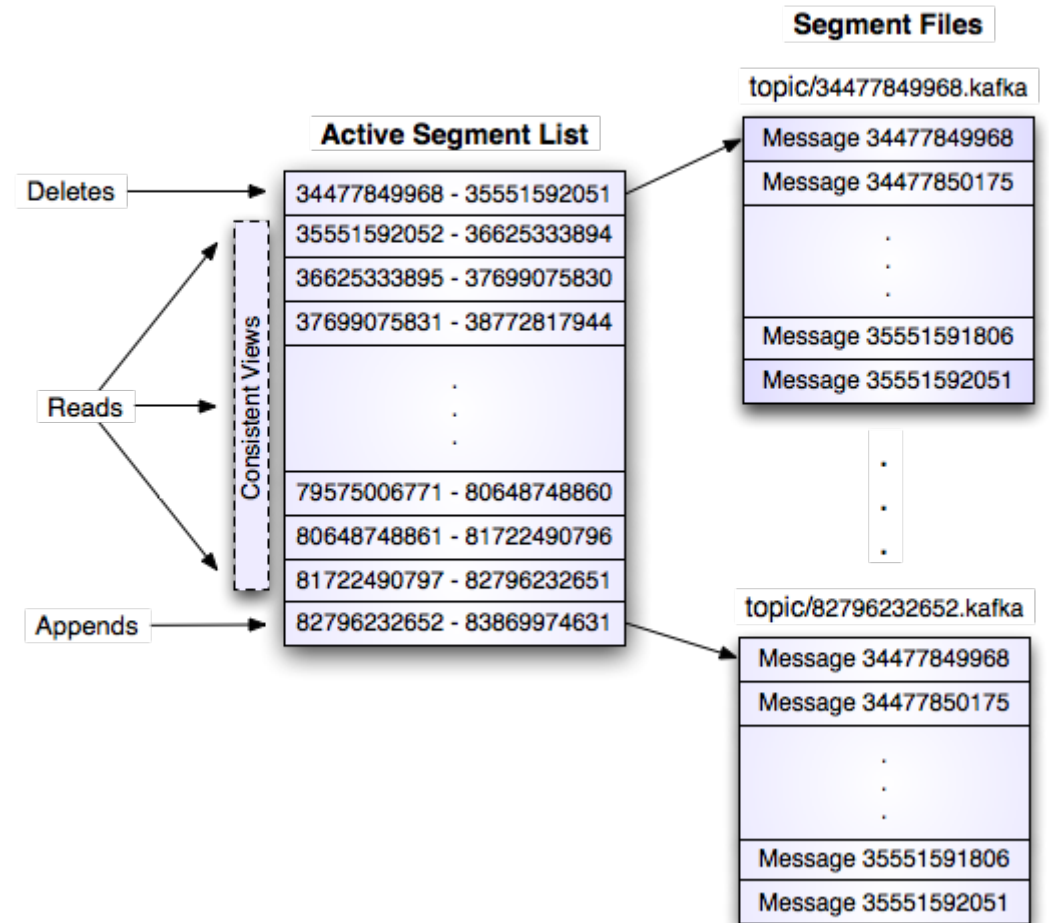
<https://dzone.com/articles/kafka-architecture>

- ❑ Kafka is run as a cluster comprised of one or more servers (called **brokers**)
 - A broker receive messages from producers and stores them on disk keyed by unique offset.
 - A broker allows consumers to fetch messages by topic, partition and offset

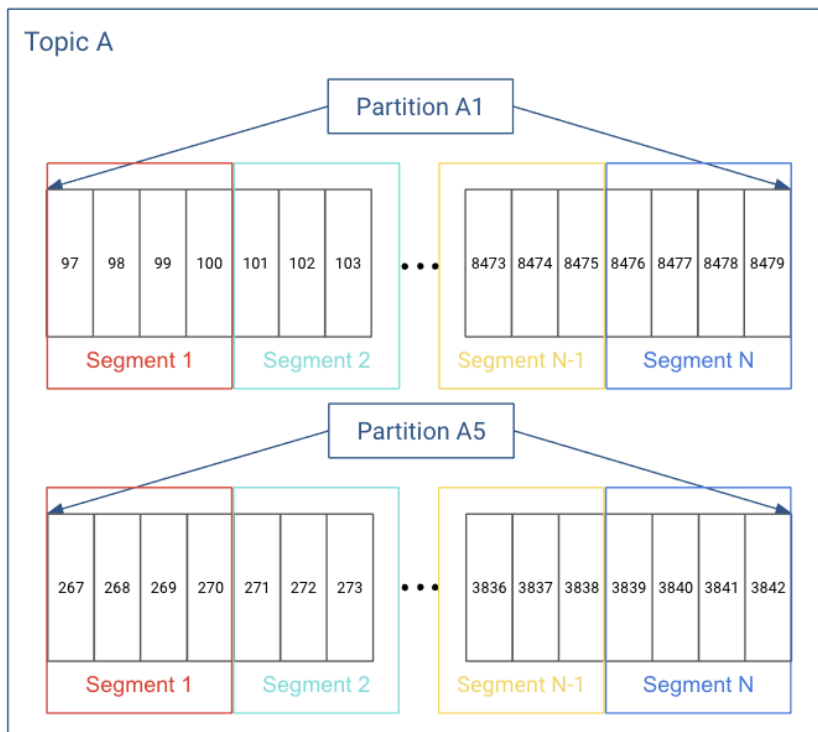
How does Kafka Work?

- ❑ Topics represent commit **log data structures** stored on disk
 - Topics are divided into **partitions**, each partition is further divided into **segments**
 - Each segment has a **log file** to store the actual message
 - Log – **time-ordered, sequence of messages that is continually appended** (each log entry can be array of bytes)
 - Partitions are **replicated and distributed** over servers in Kafka cluster (brokers) for **fault tolerance** (when a server in the cluster fails so messages remain available)
 - Messages stay around for a configurable period of time (i.e., 7 days, 30 days, etc.).
 - Can recover lost messages during time out or lost connection

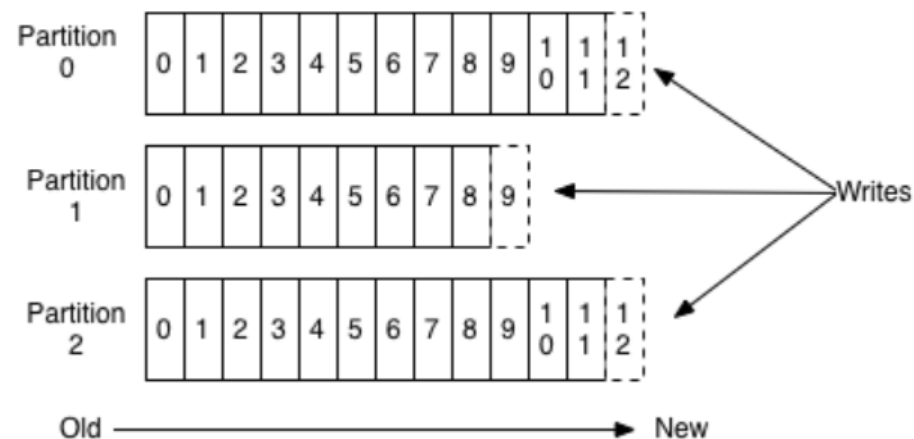
Kafka Log Implementation



Kafka Storage Structure



Anatomy of a Topic



Messages are written to it in an append-only fashion

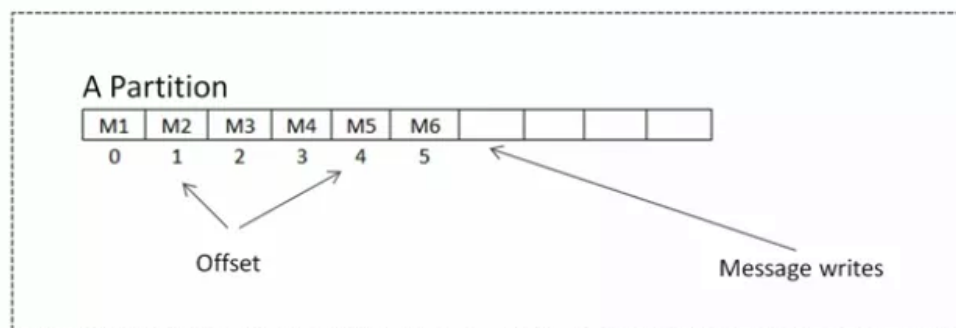
- ❑ **Topic** is logical grouping
- ❑ **Partition** is actual unit of data storage
- ❑ Every piece of data stored in segment file is a **message**
- ❑ Each message in partition is uniquely identified by an ID called **offset**

What Kafka doesn't do?

- Kafka does not have individual message IDs. Messages are simply addressed by their offset in the log.
- Kafka also does not track the consumers that a topic has or who has consumed what messages.
- There are no deletes. Kafka keeps all parts of the log for the specified time.

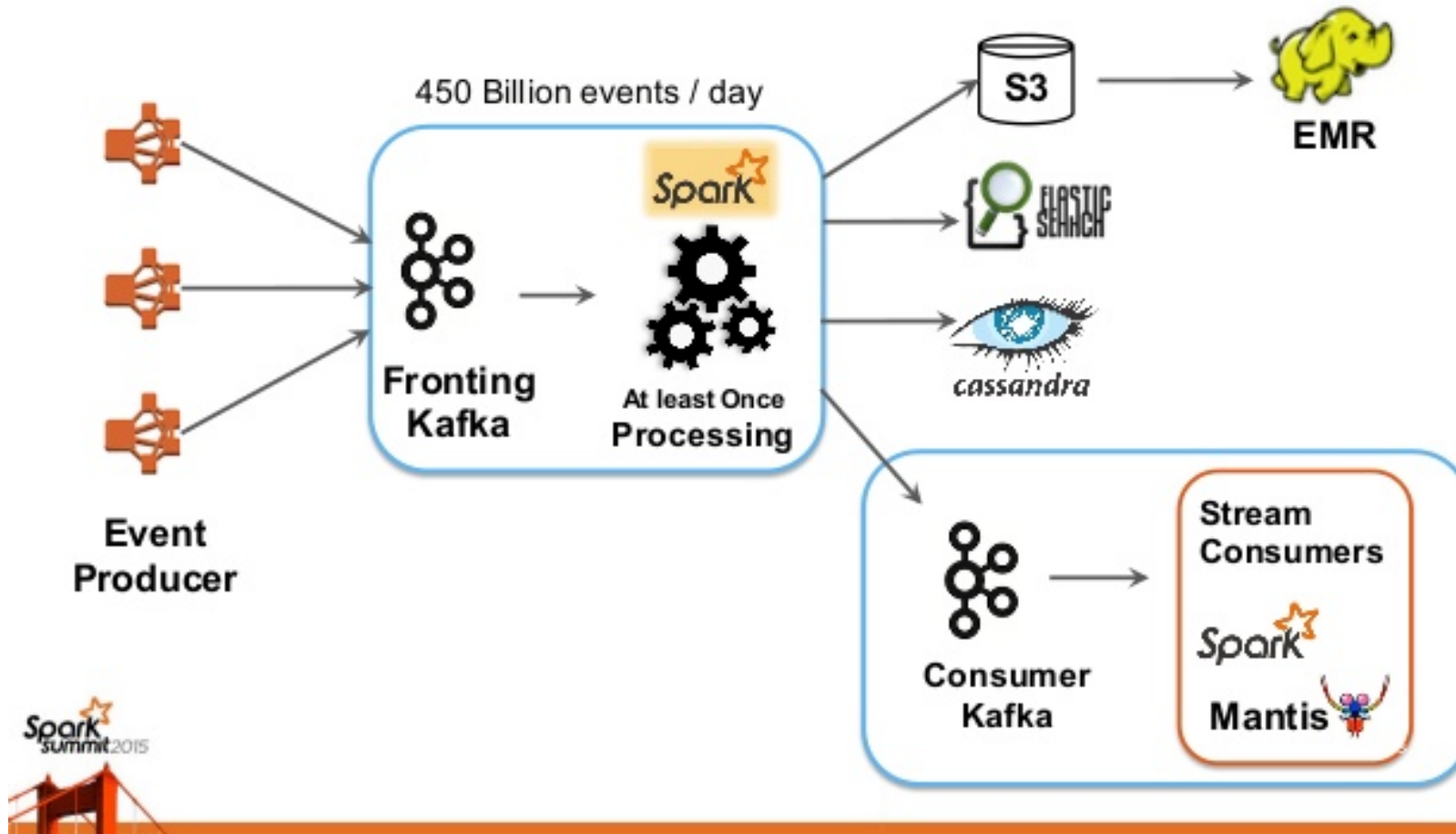
What is offset?

A sequence id given to messages as they arrive in a partition.



Kafka and big data at web-scale companies

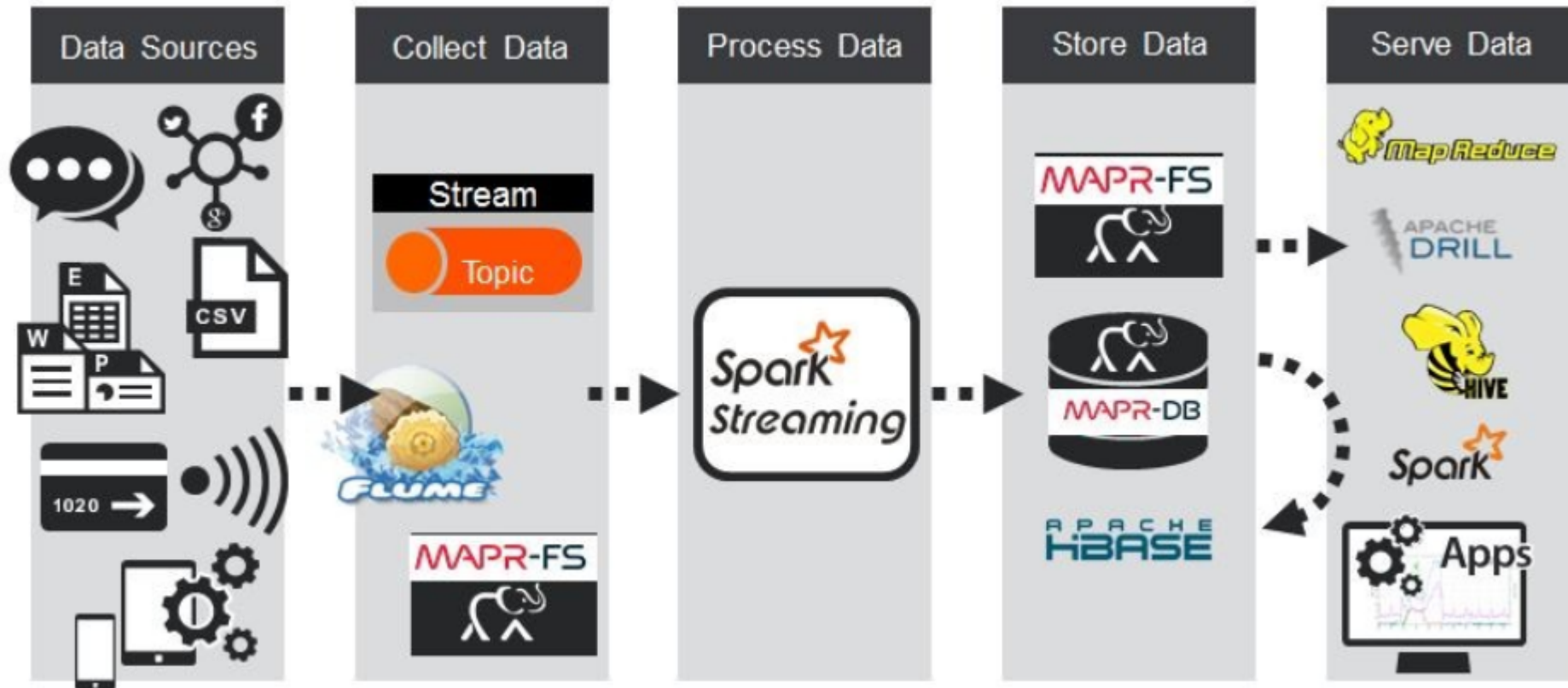
- Big Data ingestion at Netflix



<https://www.slideshare.net/SparkSummit/spark-and-spark-streaming-at-netflix-sedakar-daxini>

Kafka and big data at web-scale companies

- **BP OIL USE CASE :**

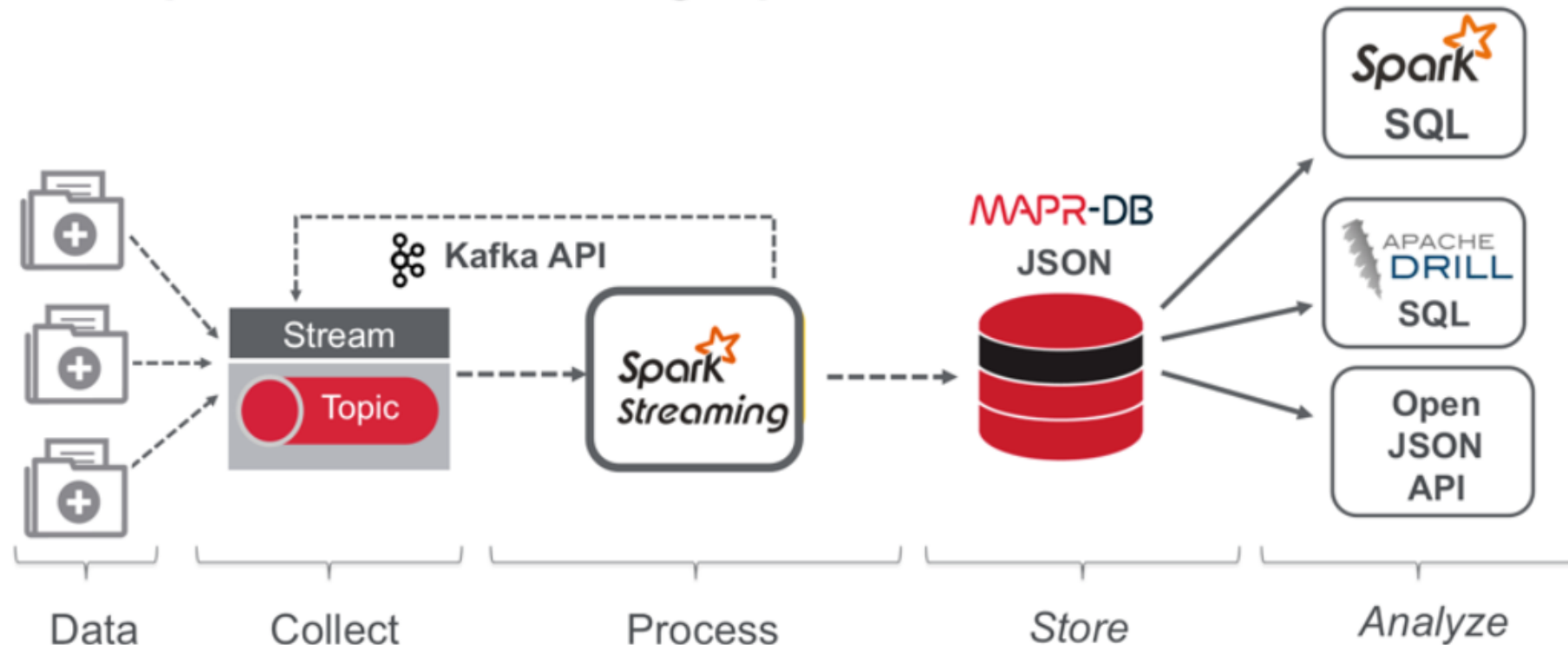


<https://www.linkedin.com/pulse/real-time-streaming-data-pipelines-apache-kafka-spark-steven-murhula/>

Kafka and big data at web-scale companies

- Transform, Store and Explore Healthcare Dataset

Example Stream Processing Pipeline



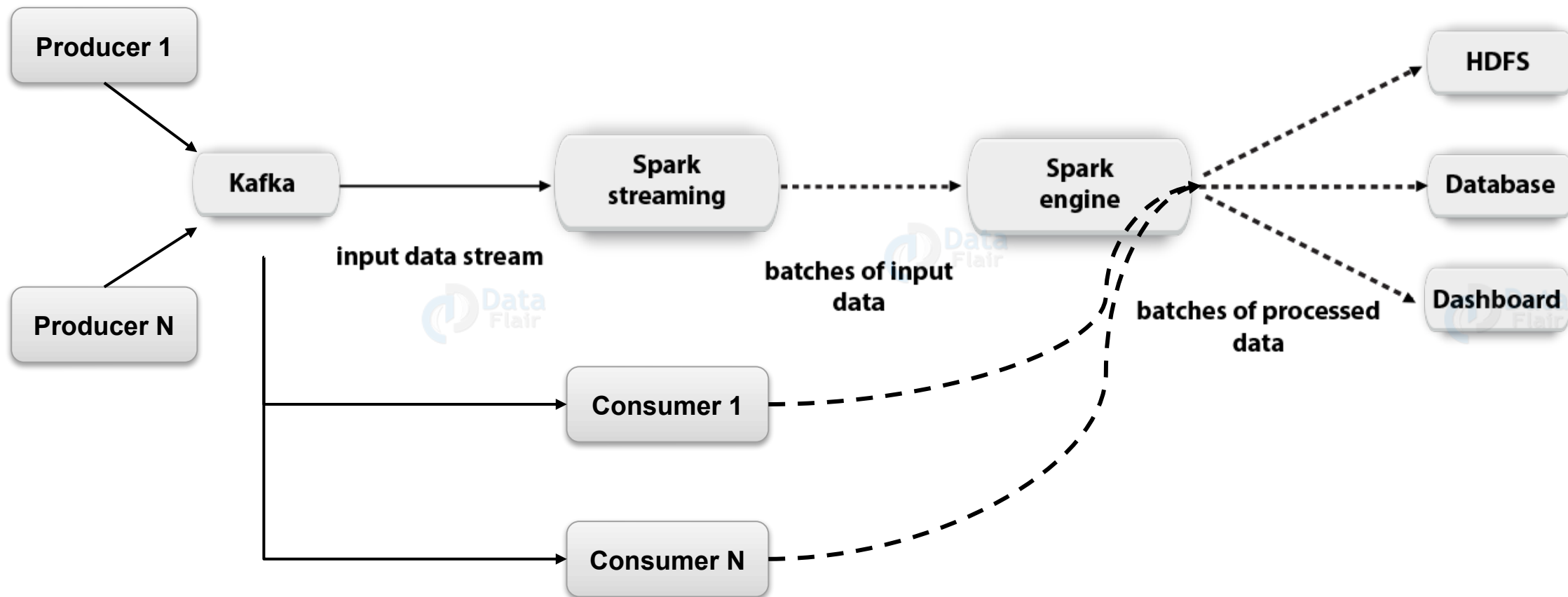
<https://mapr.com/blog/streaming-data-pipeline-transform-store-explore-healthcare-dataset-mapr-db/>

Should you use Apache Kafka?

- Kafka fits a class of problem that a lot of web-scale companies and enterprises have, but just as the traditional message broker is not a one size fits all, neither is Kafka.
- If you're looking to build a set of resilient data services and applications, Kafka can serve as the source of truth by collecting and keeping all of the "facts" or "events" for a system.

Real-Time Streaming Architecture

Kafka-Spark Streaming Integration

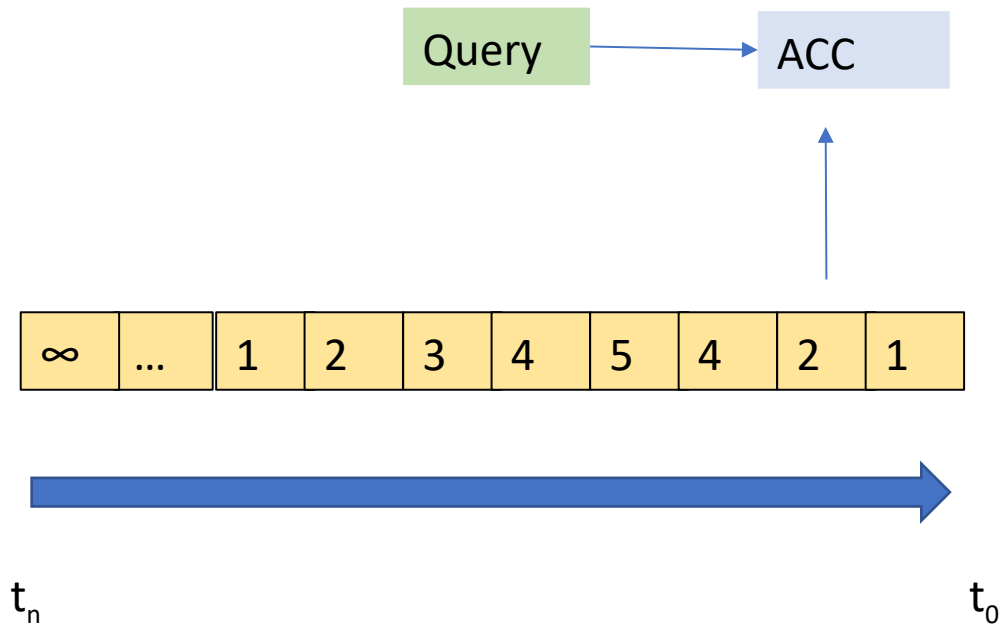


DEMO

- What is the total number of attendees up to tx?

Assume memory can only keep 5 tuples.

Keep an accumulator for the sum. There is no need to keep the tuples.



Thank You

Questions?