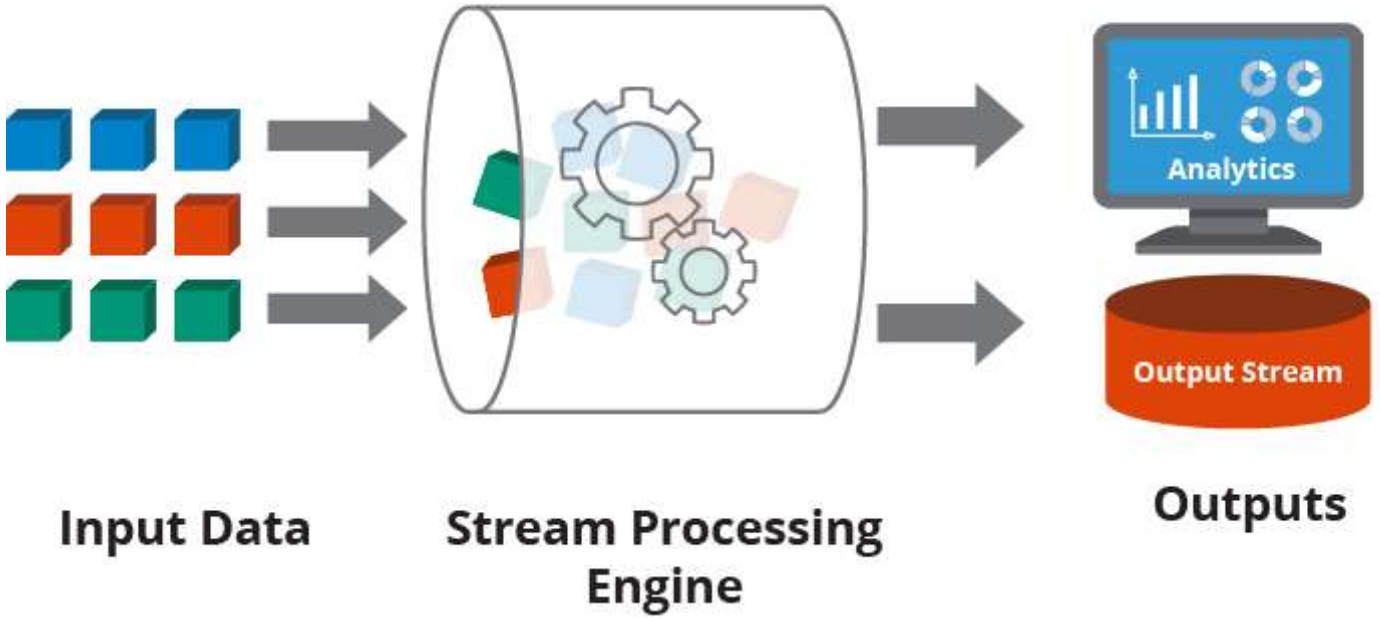
Apache Spark

Streaming

Stream processing

- Is a key requirement in many big data applications.
- The act of continuously incorporating new data to compute a result

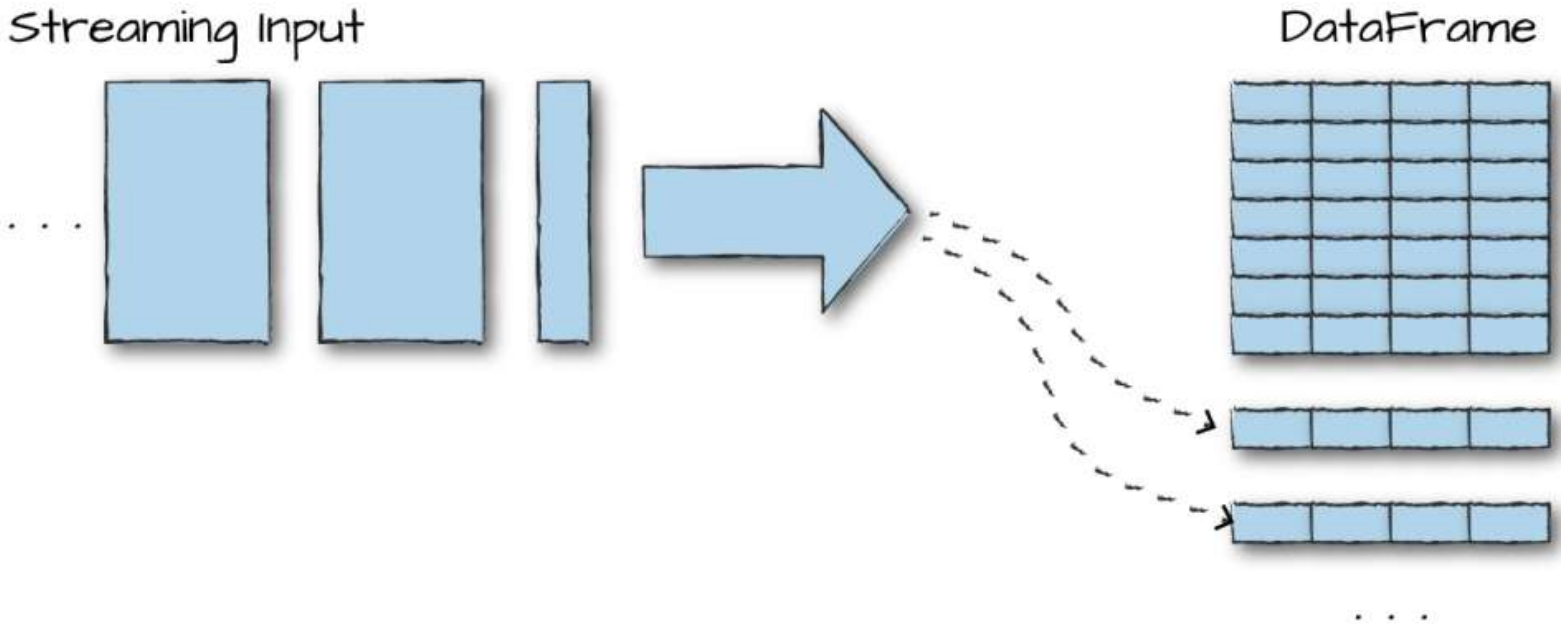


Stream processing - Use Cases

- Notifications and alerting
- Real-time reporting
- Online machine learning

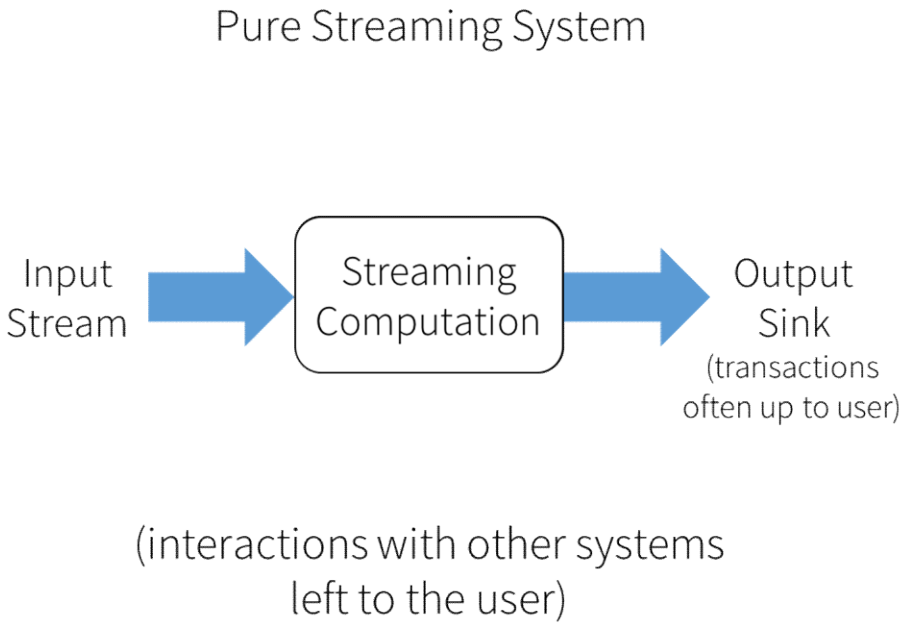
Structured Streaming Basics

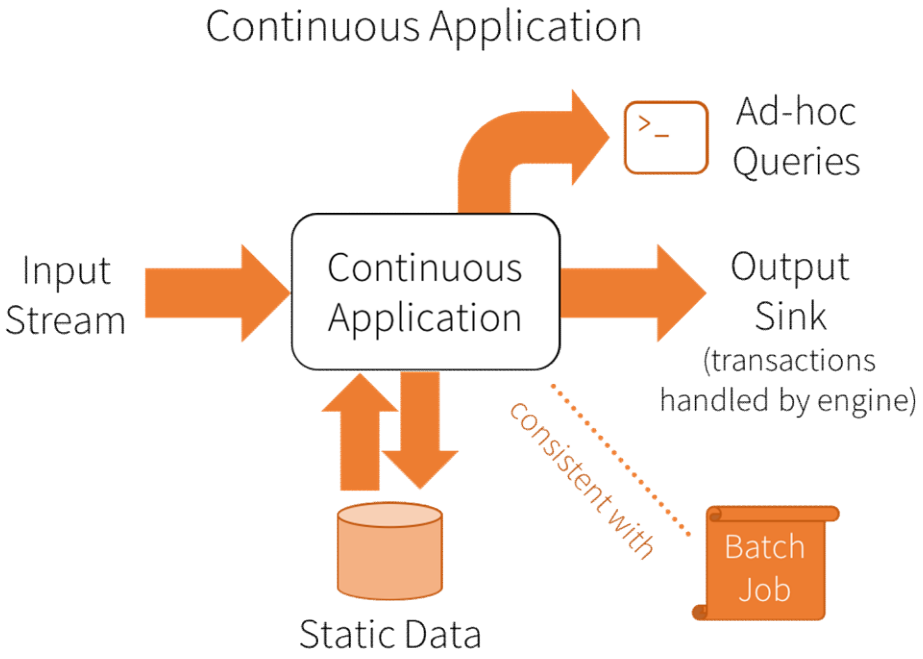
- To treat a stream of data as a table to which data is continuously appended
- The job then periodically checks for new input data, process it, updates some internal state
- In simplest terms, Structured Streaming is “your DataFrame, but streaming.”



Continuous Application

- Structured Streaming enables users to build
  - Continuous applications
- Continuous application is an end-to-end application that
  - Reacts to data in real time





Core Concepts

- Transformations and Actions
  - Maintains the same concept of transformations and actions
- Input Sources
  - Apache Kafka
  - Files on a distributed file system like HDFS or S3
- Sinks
  - Apache Kafka
  - Almost any file format
  - A console sink for testing
  - A memory sink for debugging

Core Concepts

- Output Modes
  - Append (only add new records to the output sink)
  - Update (update changed records in place)
  - Complete (rewrite the full output)
- Triggers
  - Define when data is output
  - By default will look for new input records as soon as it has finished processing the last group of input data
  - Also supports triggers based on fixed interval

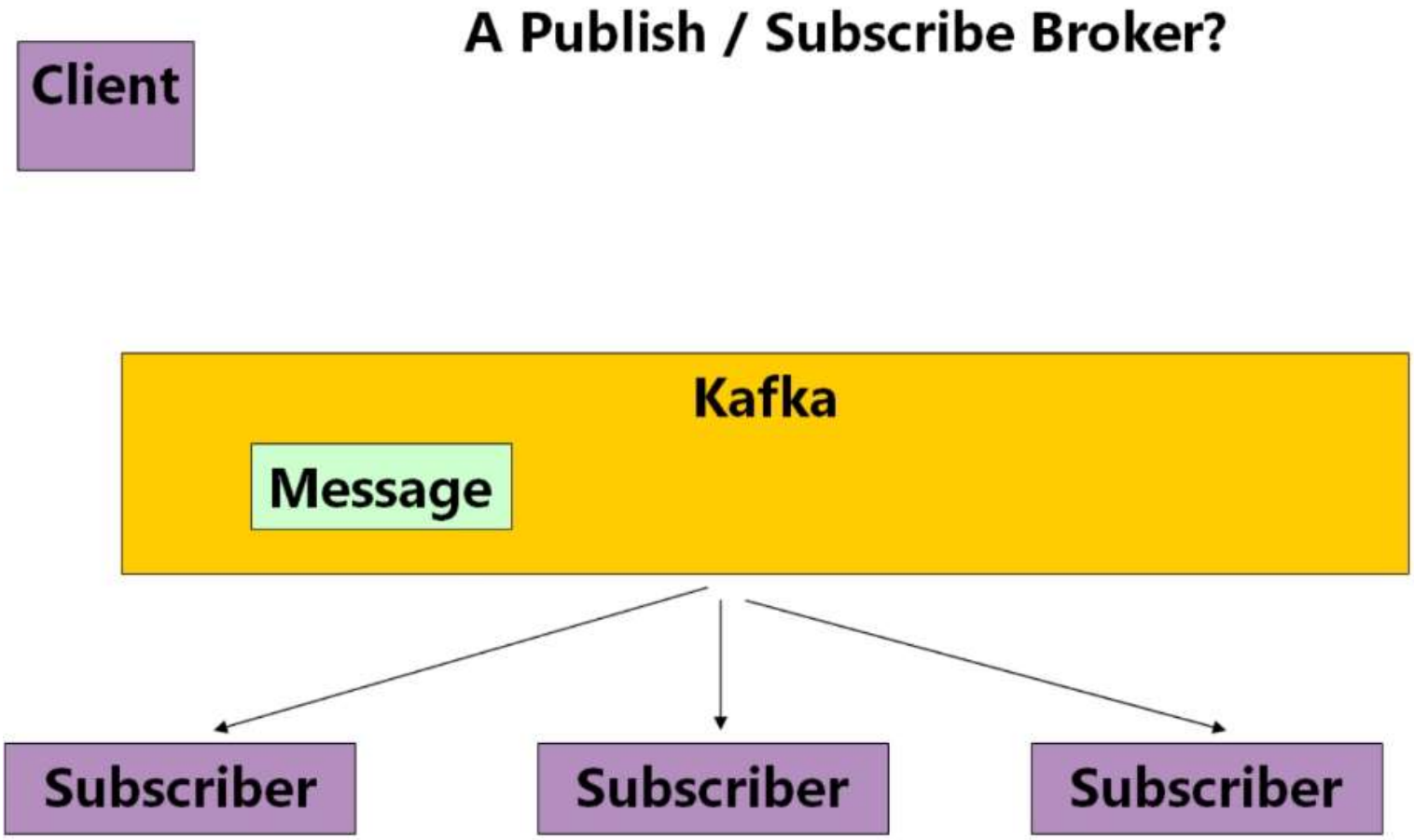
2 Versions of Streaming

DStreams - traditional model based on RDD API

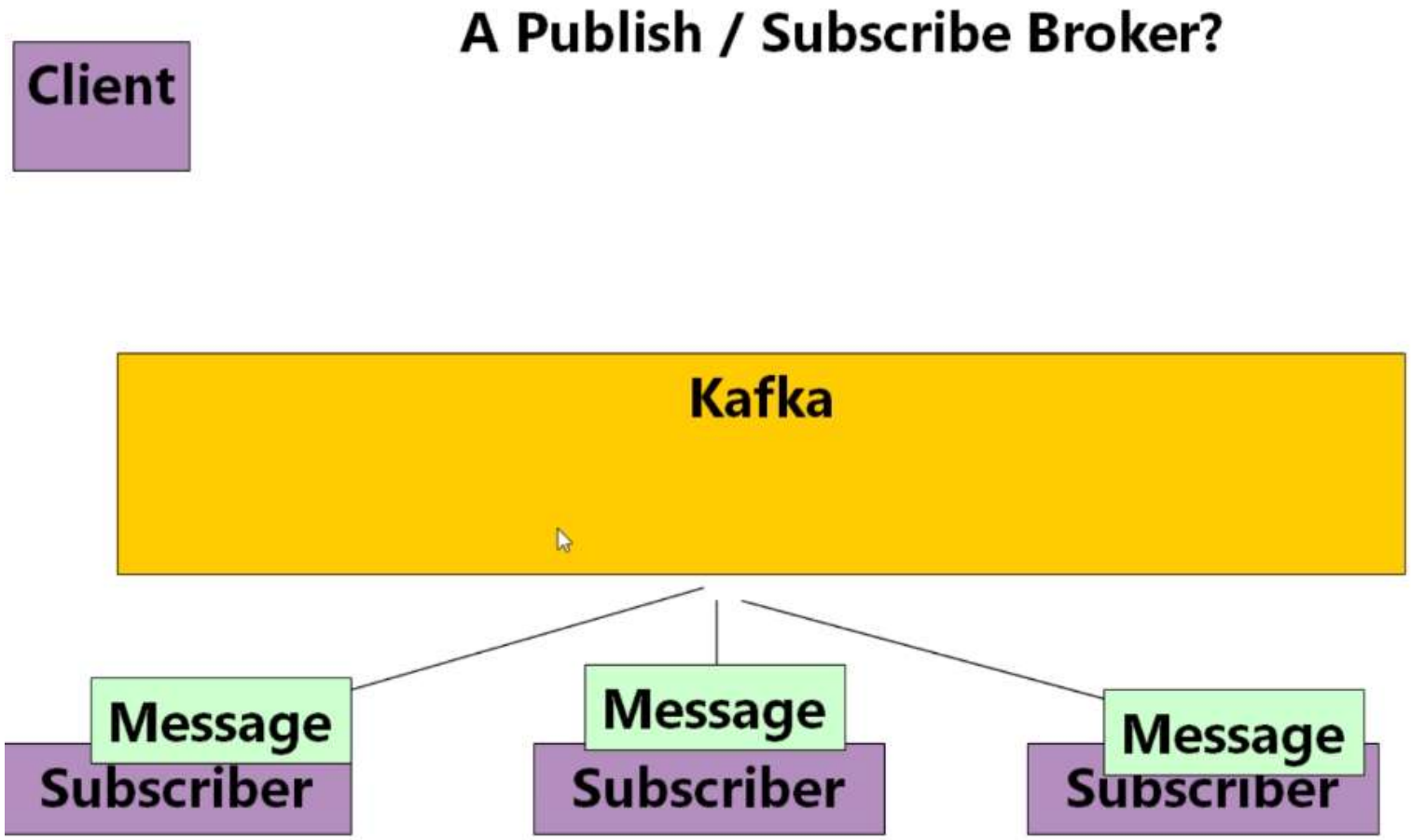
Structured Streaming – new model based on SparkSQL engine

What is Kafka?

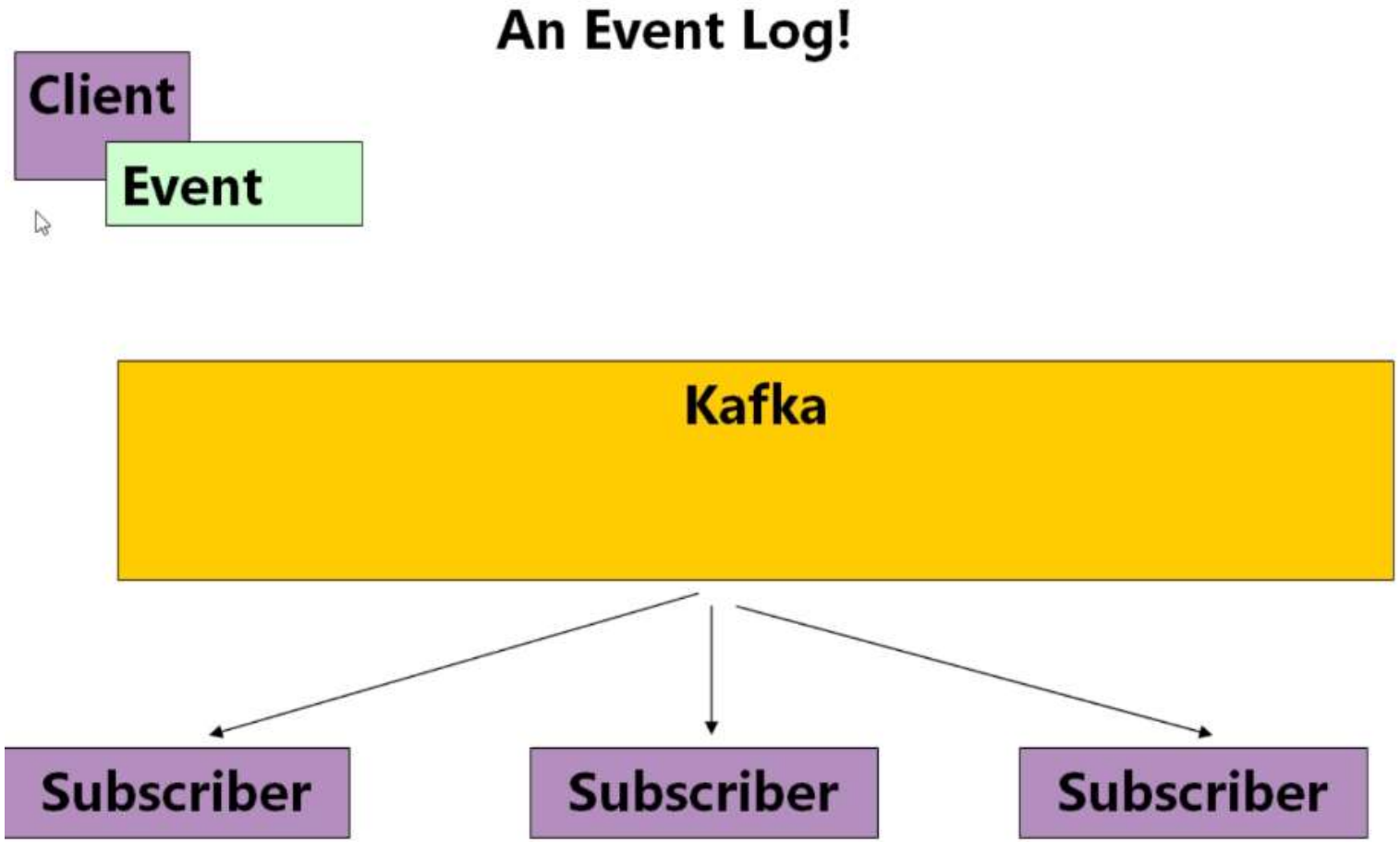
What is Kafka?



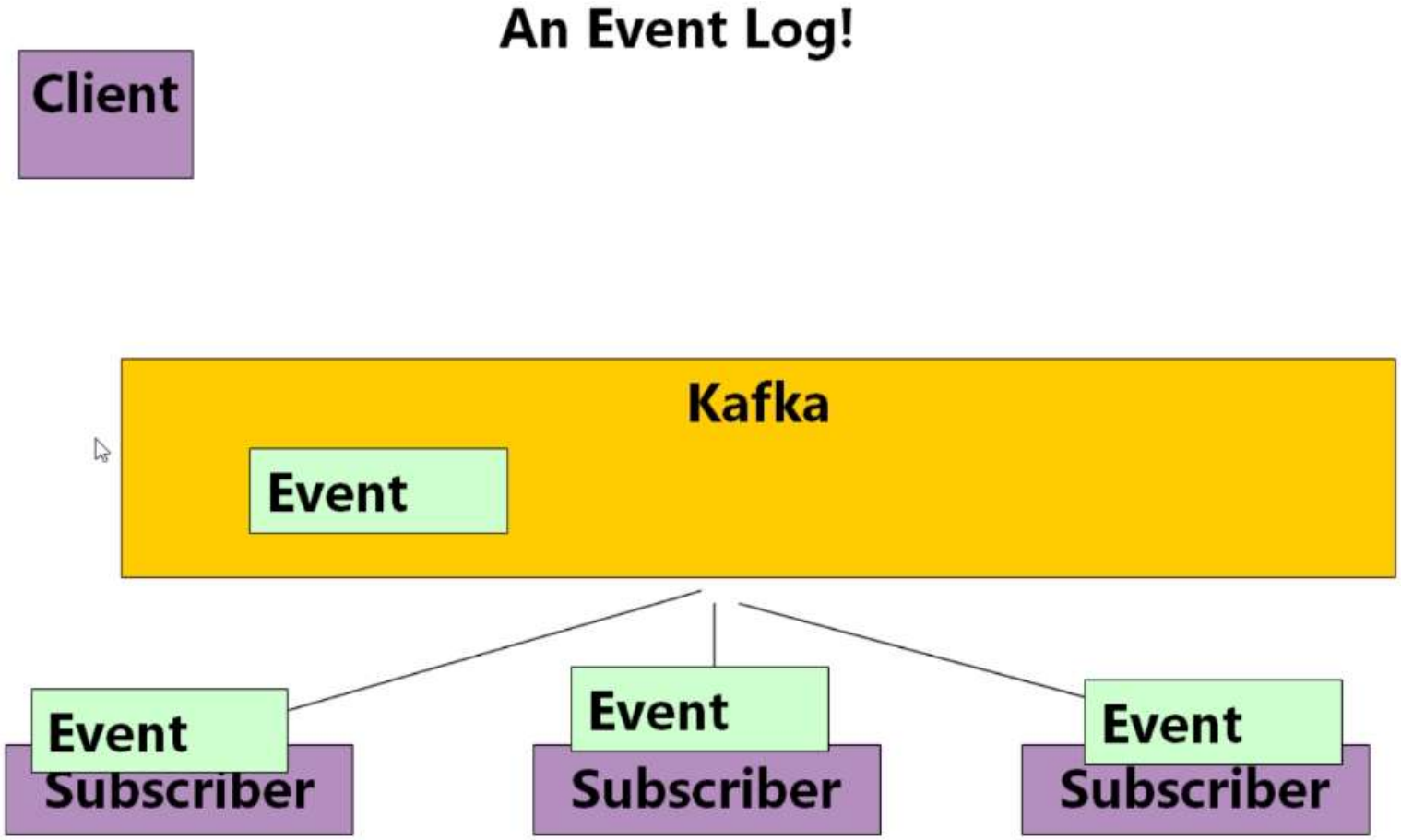
What is Kafka?



What is Kafka?



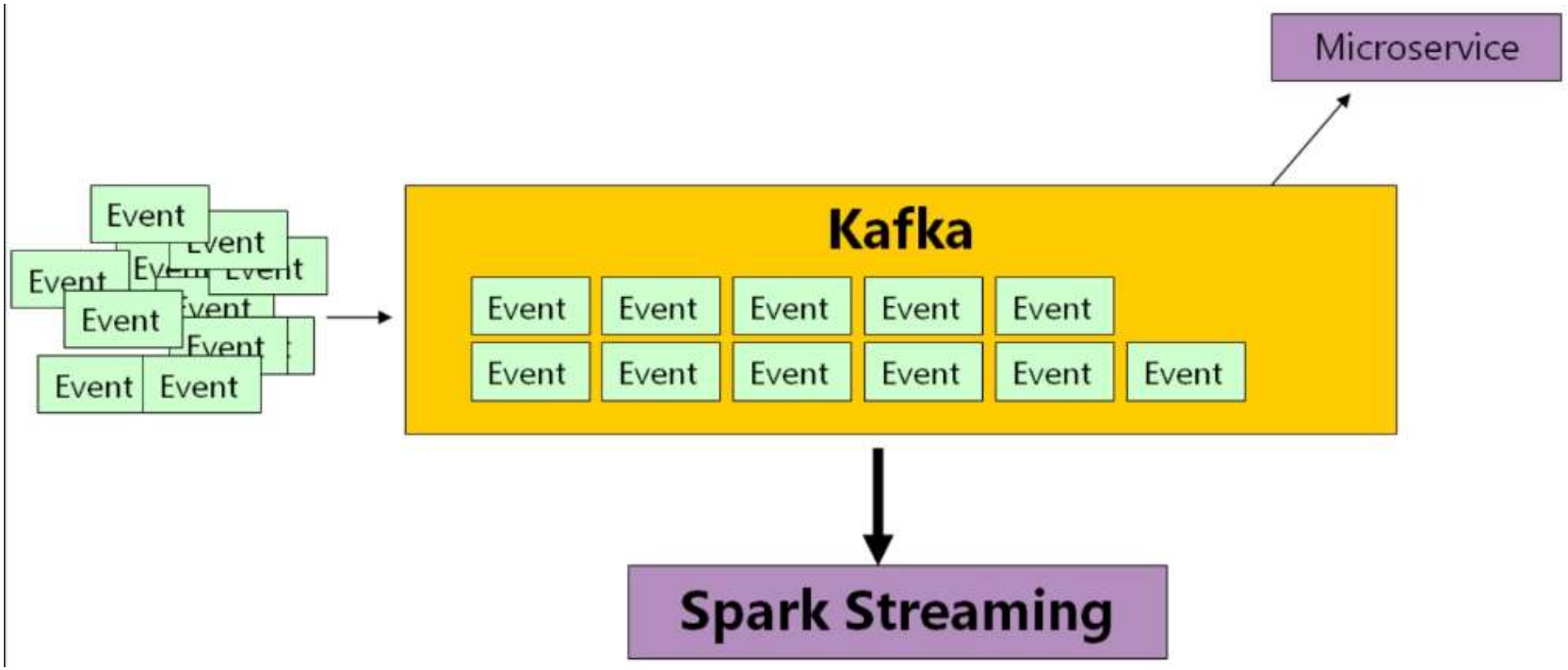
What is Kafka?



Kafka is

- A distributed Event Ledger / Log
- This makes it perfect for streaming applications

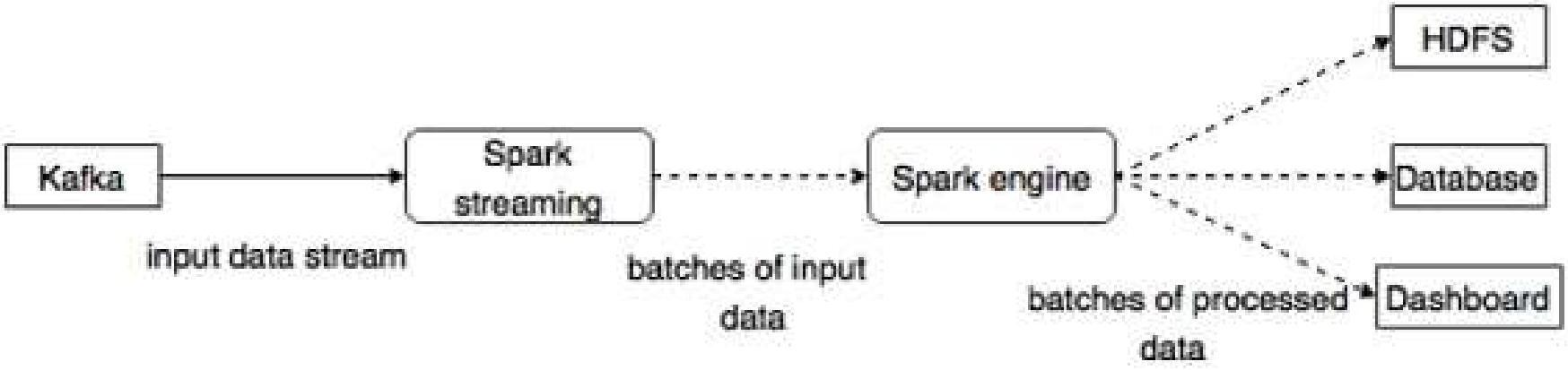
Kafka for Streaming



Install Kafka

- Refer:
  - https://kafka.apache.org/quickstart
- wget https://apachemirror.wuchna.com/kafka/2.8.0/kafka_2.13-2.8.0.tgz
- tar -xzf kafka_2.13-2.8.0.tgz
- cd kafka_2.13-2.8.0
- bin/zookeeper-server-start.sh config/zookeeper.properties &
- bin/kafka-server-start.sh config/server.properties &
- bin/kafka-console-producer.sh --topic topic1 --bootstrap-server localhost:9092
- bin/kafka-console-consumer.sh --topic topic1 --from-beginning --bootstrap-server localhost:9092

Integration with Spark



Hands-On: Stream Kafka events using Spark

- Connect to terminal
- Activate VENV
  - source /spark_venv_311/bin/activate
- Run Kafka Producer
  - python ~/Python-Scala-Spark-Training/Exercises/Spark/Spark-Streaming/kafka_producer_csv.py
- Open another terminal and activate VENV again
- Run Kafka Consumer
  - python ~/Python-Scala-Spark-Training/Exercises/Spark/Spark-Streaming/kafka_streaming_csv_demo.py

Thanks