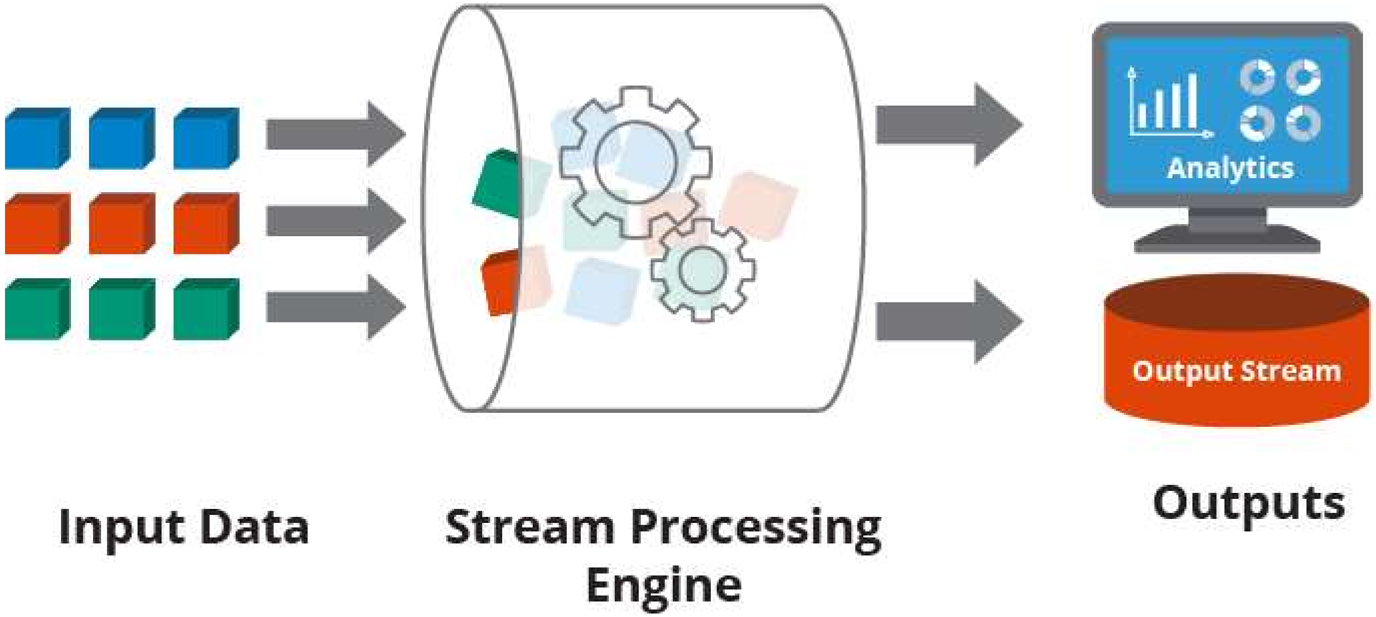
Apache Spark

Streaming

Stream processing

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

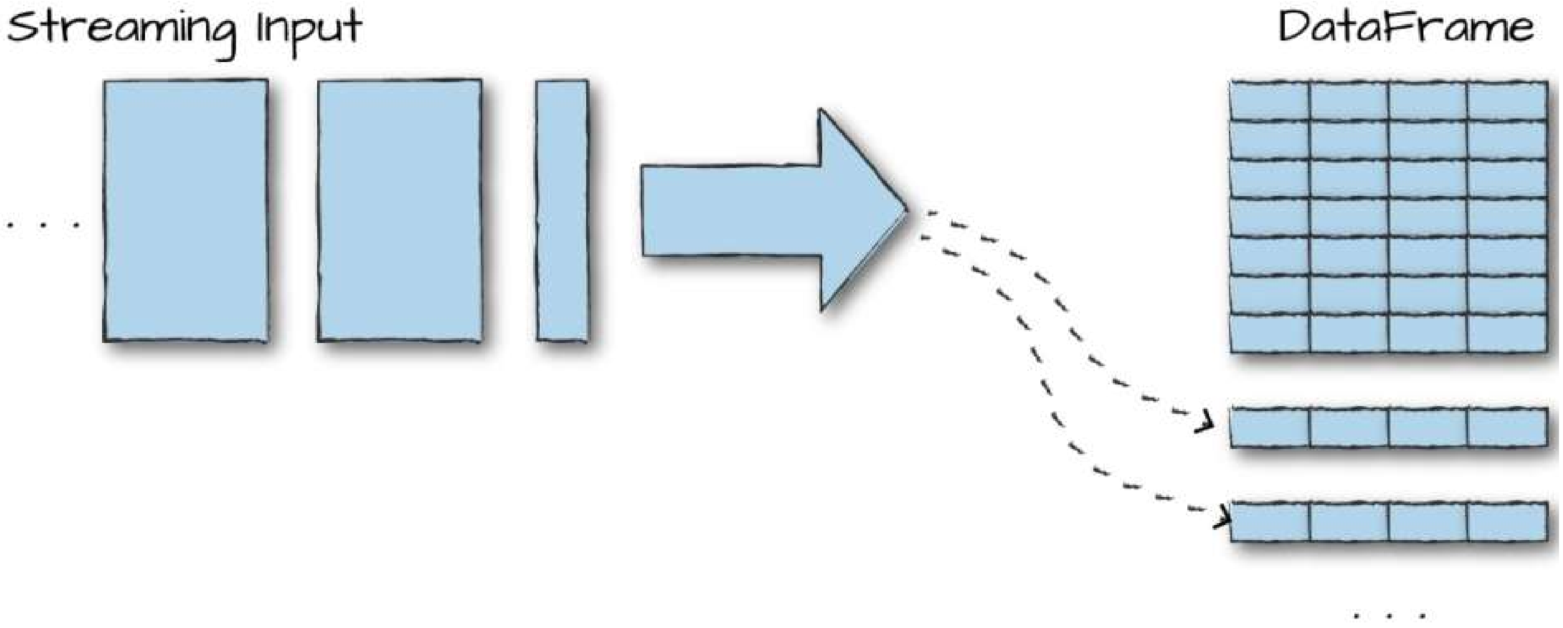


Stream processing - Use Cases

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

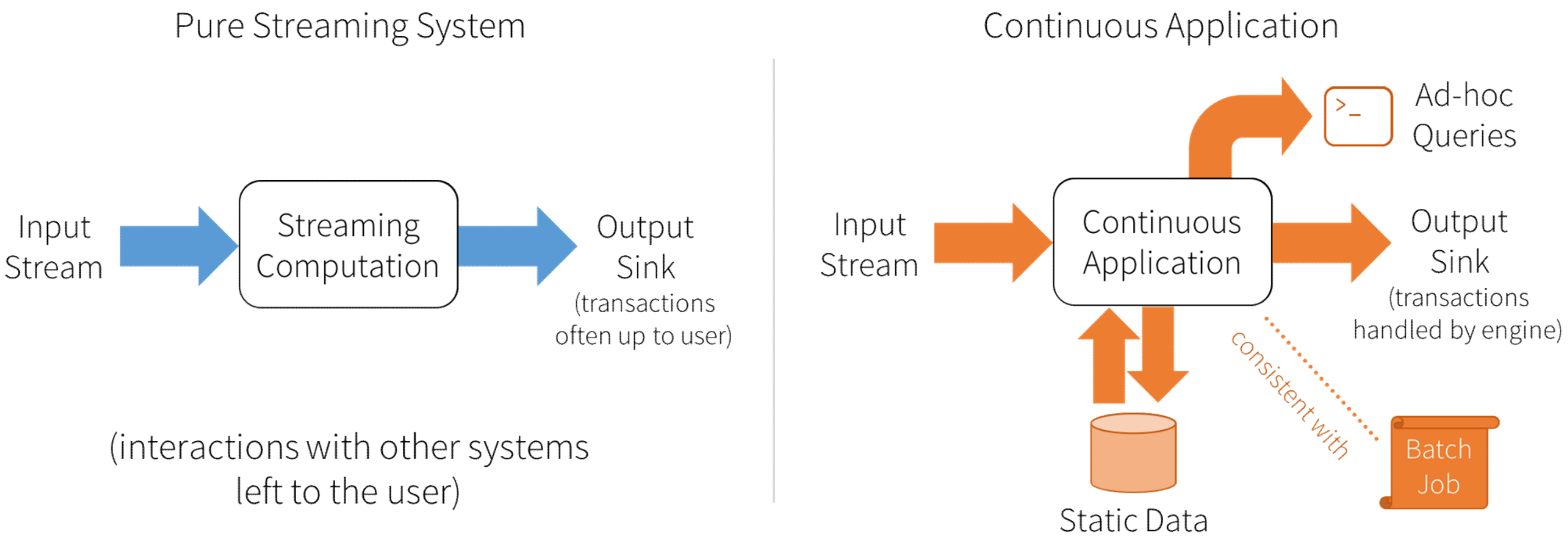
Structured Streaming Basics

- 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



Pure Streaming System

Input Stream

Streaming Computation

Output Sink

(transactions often up to user)

(interactions with other systems left to the user)

Continuous Application

Input Stream

Continuous Application

Output Sink

(transactions handled by engine)

>_

Ad-hoc Queries

Static Data

consistent with

Batch Job

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

Thanks