Spark Streaming

Let’s learn something!

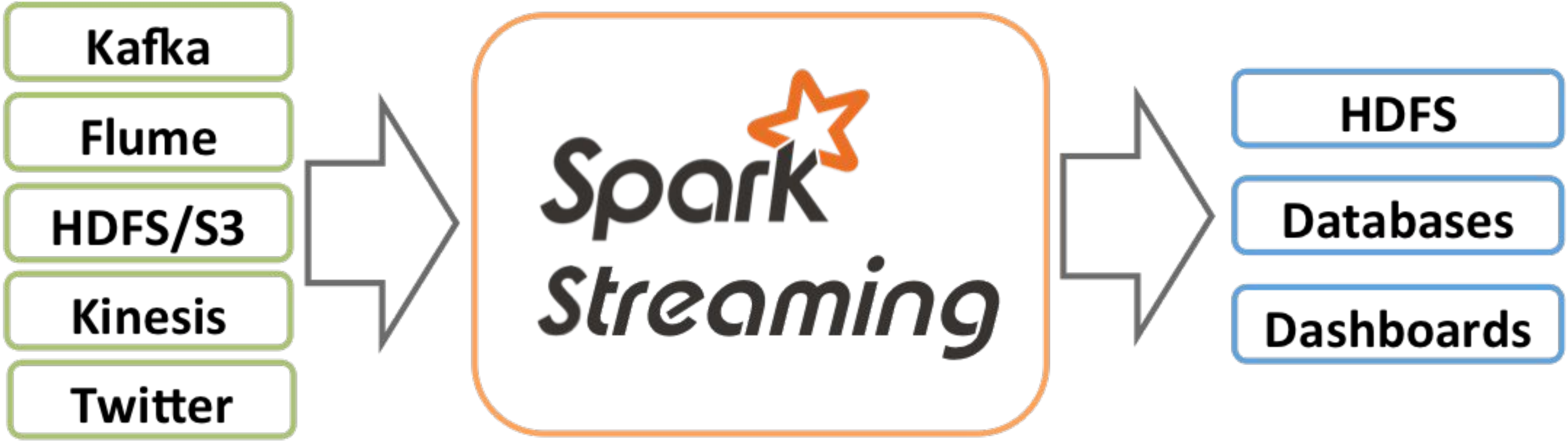
Python and Spark

- Spark Streaming is an extension of the core Spark API that enables scalable, high-throughput, fault-tolerant stream processing of live data streams.

Python and Spark

- Data can be ingested from many sources like Kafka, Flume, Kinesis, or TCP sockets, and can be processed using complex algorithms expressed with high-level functions like map, reduce, join and window.

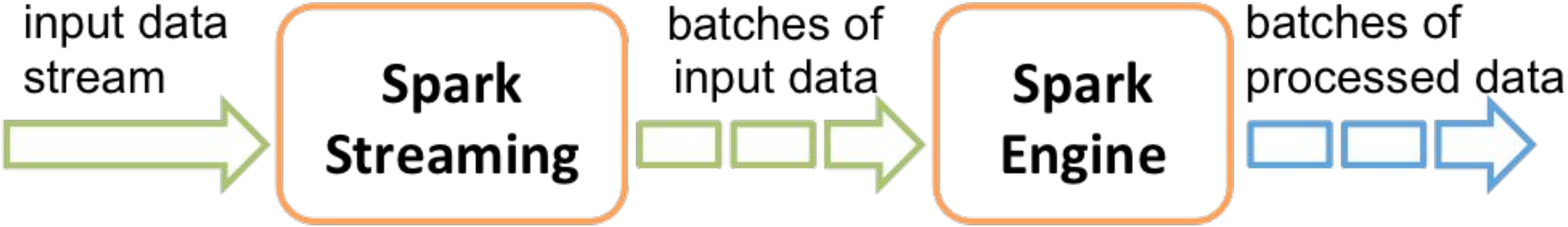
Python and Spark



Python and Spark

- Internally, Spark Streaming receives live input data streams and divides the data into batches, which are then processed by the Spark engine to generate the final stream of results in batches.

Python and Spark



Python and Spark

- The various possible data sources (Kafka, Flume,Kinesis, etc…) can not realistically be shown in a single computer setting.
- If your place of work necessitates use of one of these sources, Spark provides full integration guides.

Python and Spark

- Keep in mind not every source version is available with the Python API.
- Let’s jump to the documentation to show you where you can find additional information on Spark Streaming!

Python and Spark

- Because we will be using Spark Streaming and not structured streaming (still experimental and in Alpha) we need to use some older “RDD” syntax.
- This stems from using a SparkContext instead of a SparkSession.

Python and Spark

- We will be building a very simple application that connects to a local stream of data (an open terminal) through a socket connection.
- It will then count the words for each line that we type in.

Python and Spark

- The steps for streaming will be:
  - Create a SparkContext
  - Create a StreamingContext
  - Create a Socket Text Stream
  - Read in the lines as a “DStream”

Python and Spark

- The steps for working with the data:
  - Split the input line into a list of words
  - Map each word to a tuple: (word,1)
  - Then group (**reduce)** the tuples by the word (**key)** and sum up the second argument (the number one)

Python and Spark

- That will then provide us with a word count in the form **(‘hello’,3)** for each line.
- As a quick note, the RDD syntax relies heavily on lambda expressions, which are just quick anonymous functions.