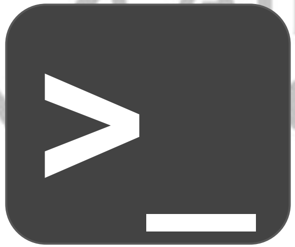
# Data Sources

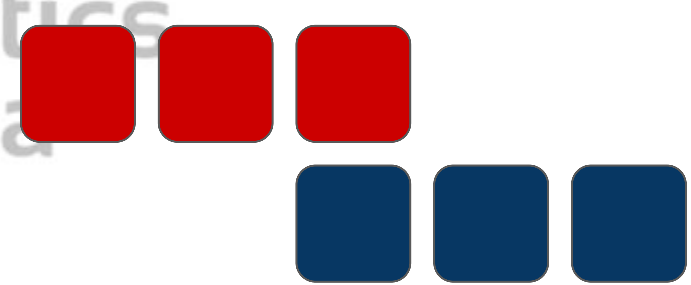
# Data Sources

- *Basic sources:* Sources directly available in the StreamingContext API.

**File Streams**



**TCP Sockets**



**Queues of RDD**

# File Streams

File Streams

- For reading data from files on any file system compatible with the HDFS API
- File streams do not require running a receiver
- For simple text files, the easiest method is **StreamingContext.textFileStream(dataDirectory)**
- **fileStream** is not available in the Python API only **textFileStream** is available

# textFileStream

## textFileStream(*DataDirectory*)

- Creates an input stream from new files that enters a specific  directory

```
def simple_text_to_stream(ssc):
      ssc.textFileStream(‘/data’).pprint()
```

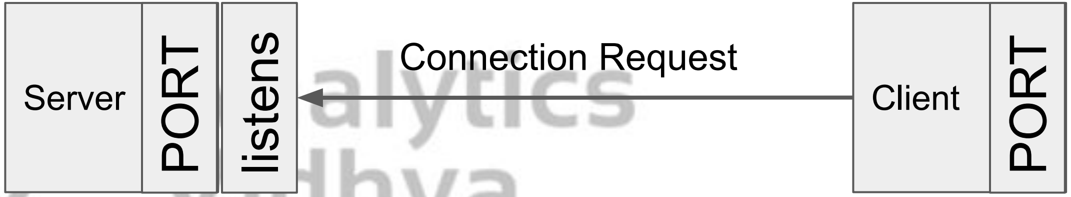
**Parameters**

- **dataDirectory**: filepath for a folder with new files being added after the start of the stream

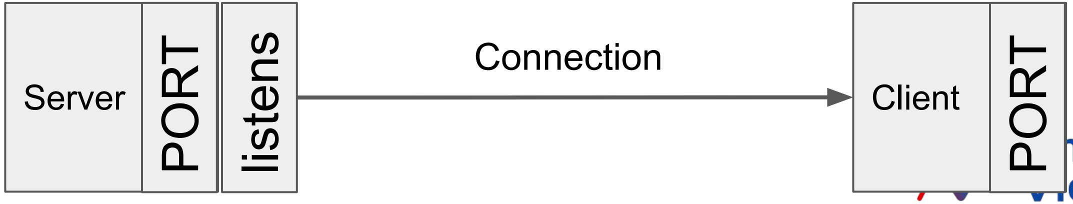
# TCP Sockets

**TCP Sockets**

- Normally, a server runs on a specific computer and has a socket that is bound to a specific port number



- Client tries to make a connection with the server on a specific port number
- Upon acceptance, the server gets a new socket bound to the same local port

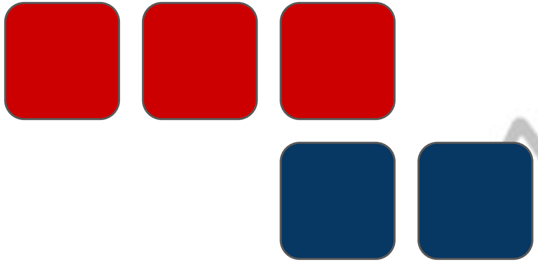


# Socket Stream Sources

- In this example we will create a Spark Socket Stream with the following lines

```
sc = SparkContext()
ssc = StreamingContext(sc, 10)
Socket_stream = ssc.socketTextStream(“127.0.0.1”, 9999)
```

# Queues of RDD



Queues of RDD

- For testing a Spark Streaming application with test data
- Each RDD pushed into the queue will be treated as a batch of data in the DStream, and processed like a stream.

# queueStream

## **queueStream(*rdds, oneAtATime=True, default=None*)**

- Creates an input stream from a queue of RDD’s or list

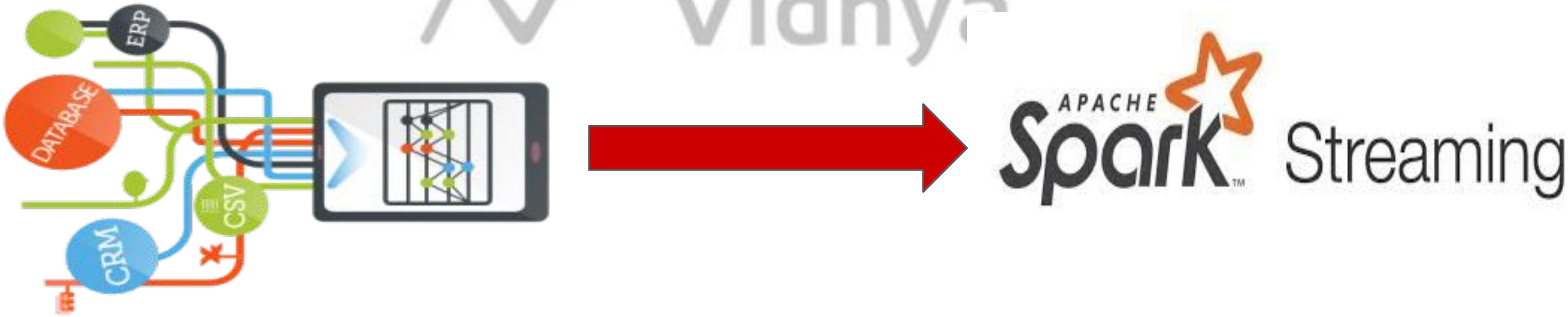
```
def queue_example(ssc):
        ssc.queueStream[range(5), [‘a’,’b’], [‘c’]], oneAtATime=True).pprint()
```

### Parameters

- **Rdds**: queue of rdds
- **oneAtATime** - Pick one rdd each time or pick all of them once
- **Default** - The default rdd is no more in rdds

# Data Sources

- *Basic sources:* Sources directly available in the StreamingContext API.
- *Advanced sources:* Available through extra utility classes



# Advanced Sources

- Use of external non Spark libraries
- Advanced sources are not available in Spark-Shell
- If you want to use them, download the  the corresponding Maven artifact JAR

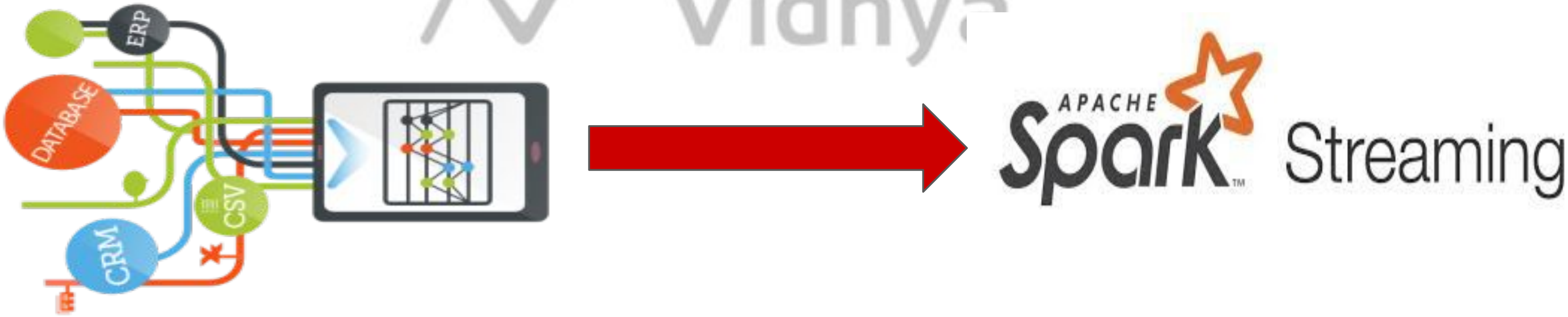
Some of these advanced sources are as follows:

APACHE
kafka

amazon
Kinesis

# Data Sources

- *Basic sources:* Sources directly available in the StreamingContext API.
- *Advanced sources:* Available through extra utility classes
- *Custom sources:* Available through extra utility classes



# Custom Sources

- This is not supported in **Python**
- Input DStreams can be created out of Custom data sources
- All you have to do is implement a user-defined receiver

# Thank You