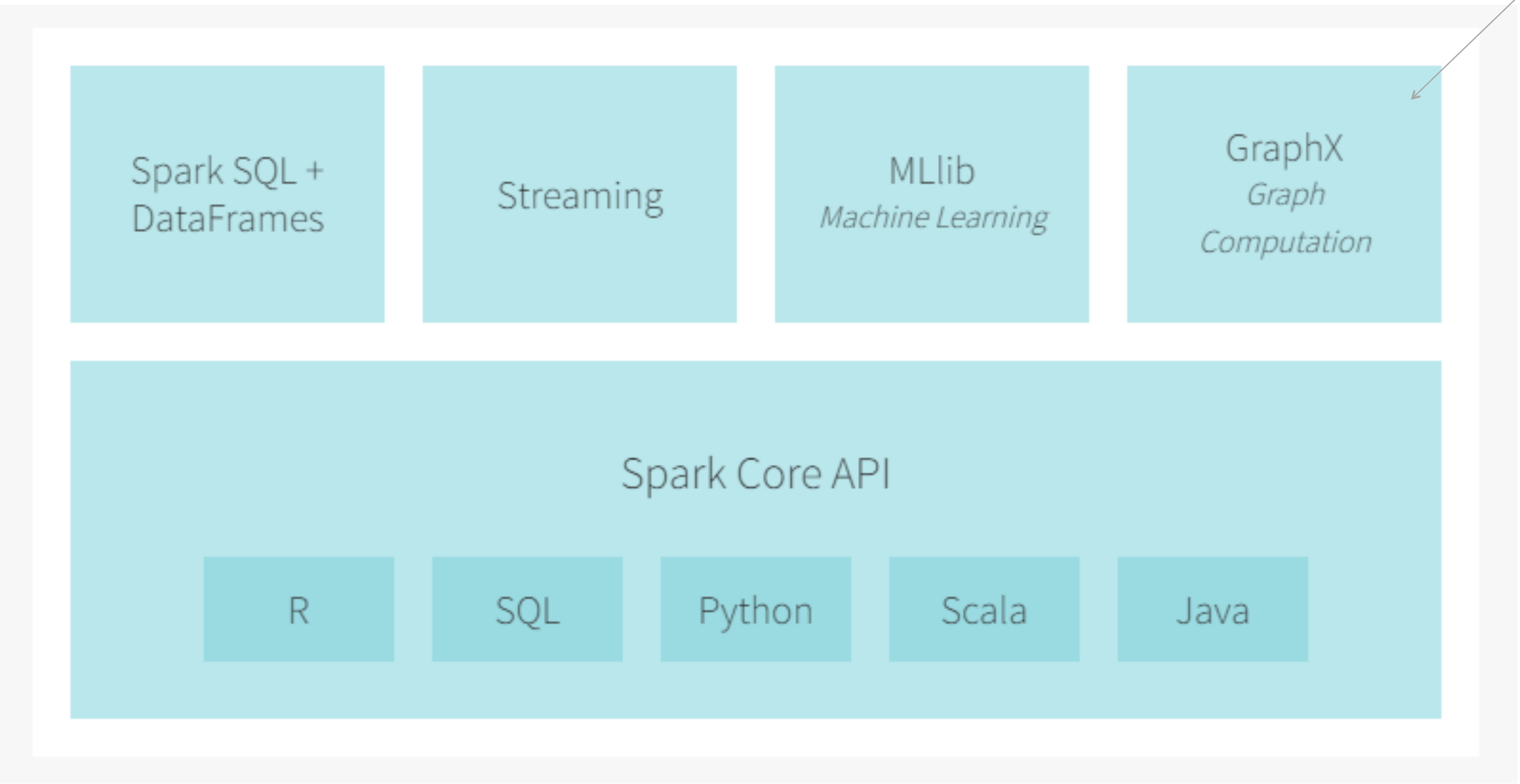
# Spark Workshop

Tao Ruangyam, ING Analytics – Frankfurt Hub

Istanbul 2020

# Spark Components



Spark SQL + DataFrames

Streaming

MLlib
*Machine Learning*

GraphX
*Graph Computation*

Spark Core API

R

SQL

Python

Scala

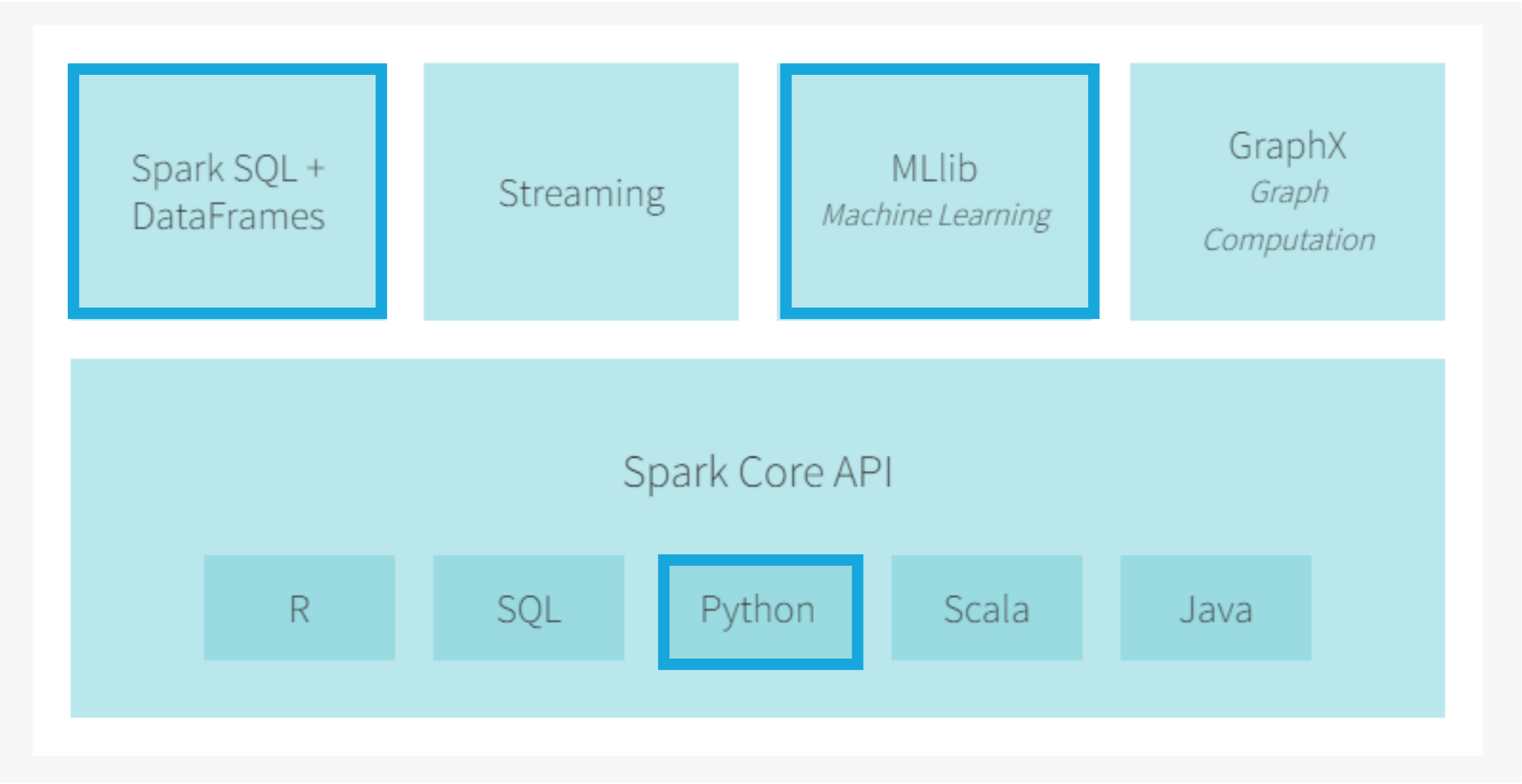
Java

Spark 3.0 is aimed to have **Neo4J's Cypher** in built.

Spark 2.x supports up to JDK8, But Spark 3 will support JDK11

# This workshop covers

APACHE Spark™



Spark SQL + DataFrames

Streaming

MLlib
*Machine Learning*

GraphX
*Graph Computation*

Spark Core API

R

SQL

Python

Scala

Java

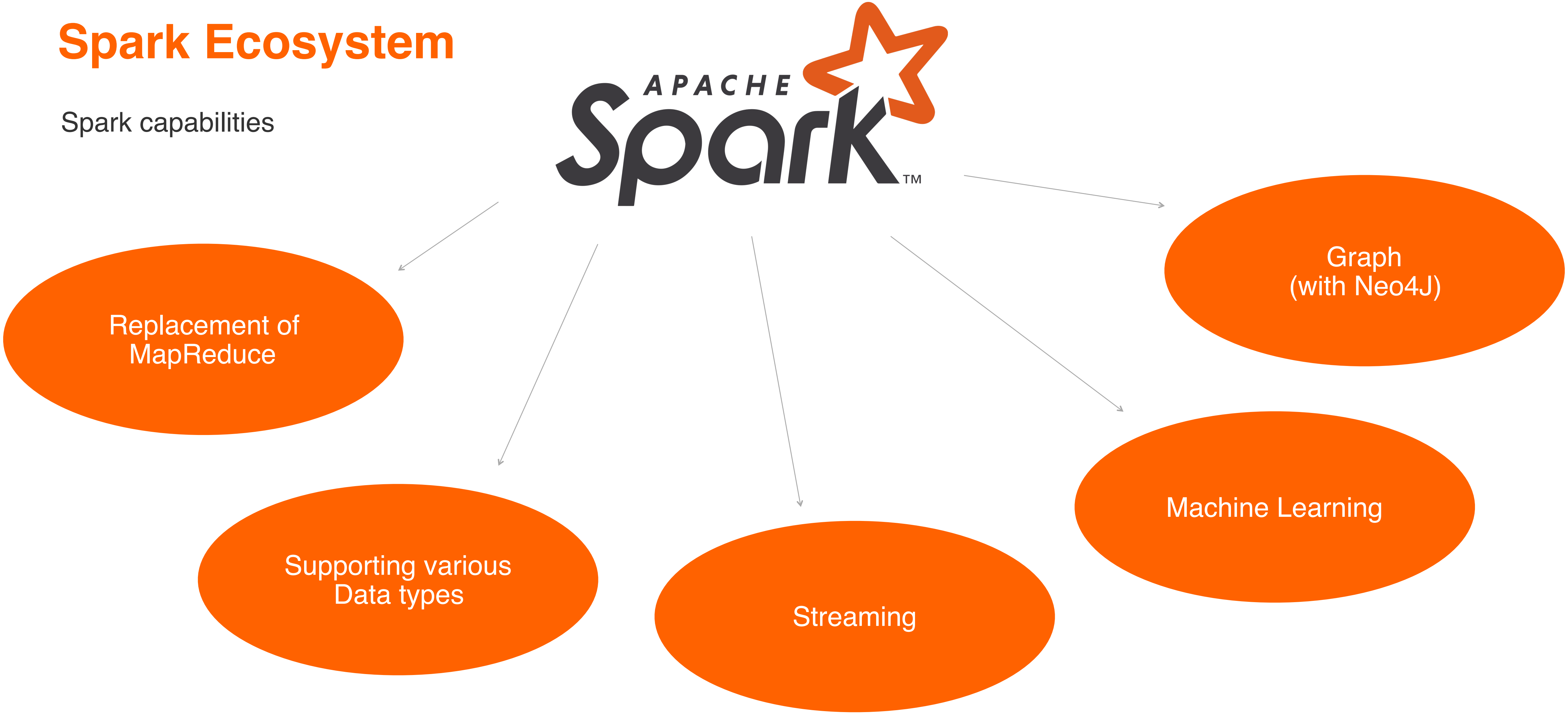
# Agenda

| Day 1 | | Coding? |
|---|---|---|
| 10:00 – 10:50 | Intro to Spark Ecosystem | - |
| 11:00 – 12:00 | Spark APIs | - |
| Lunch break | | |
| 13:00 – 13:50 | Spark APIs (cont) | Yes |
| 14:00 – 15:00 | Spark Memory management & Optimisation | - |

| Day 2 | | Coding? |
|---|---|---|
| 10:00 – 10:50 | Spark SQL | Yes |
| 11:00 – 12:00 | Spark ML | - |
| Lunch break | | |
| 13:00 – 13:50 | Spark ML (cont) | Yes |
| 14:00 – 15:00 | Wrap up, Best practices & Tips | Optional |

# Spark Ecosystem

Spark capabilities



Replacement of MapReduce

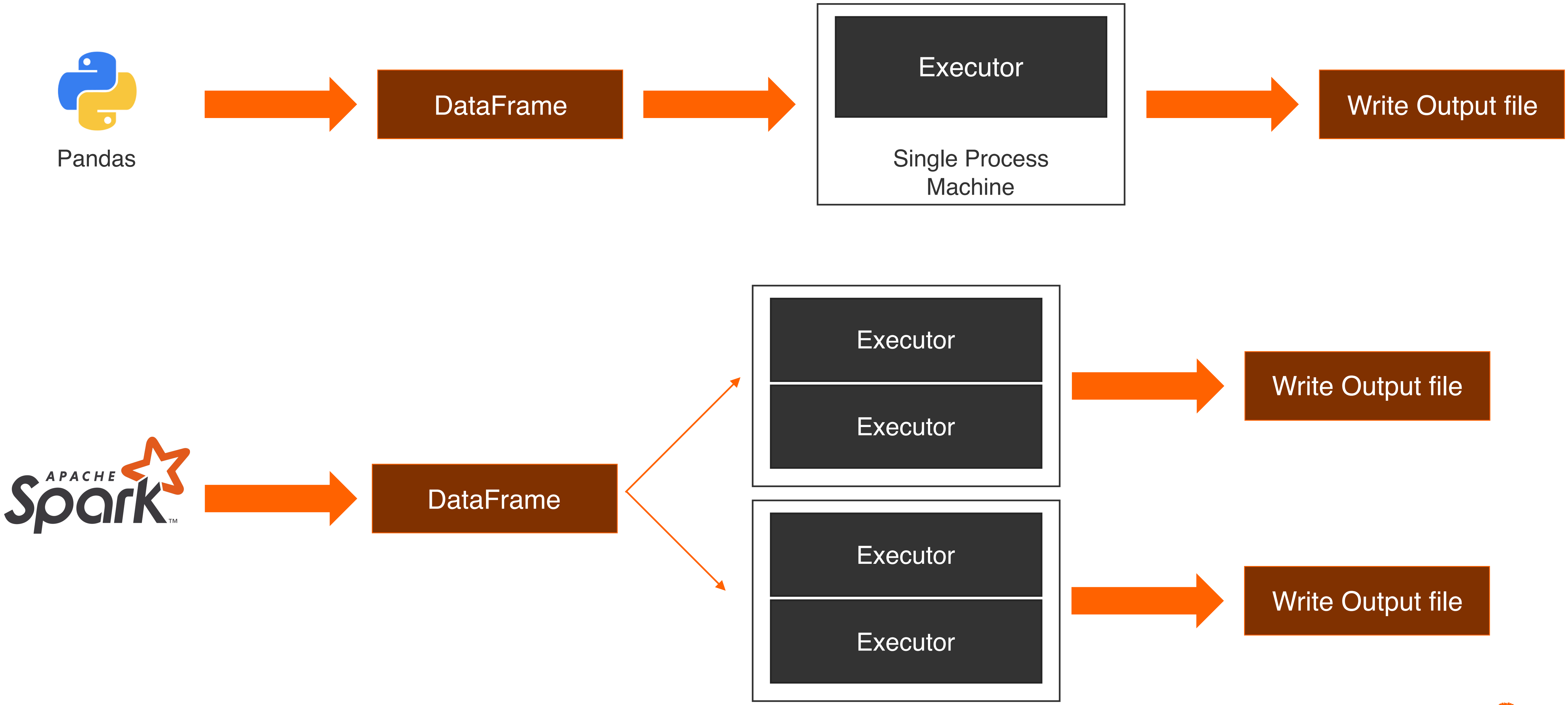
Supporting various Data types

Streaming

Machine Learning

Graph (with Neo4J)

# Spark vs Single-Machine



Pandas

DataFrame

Executor

Single Process Machine

Write Output file

APACHE Spark

DataFrame

Executor

Executor

Executor

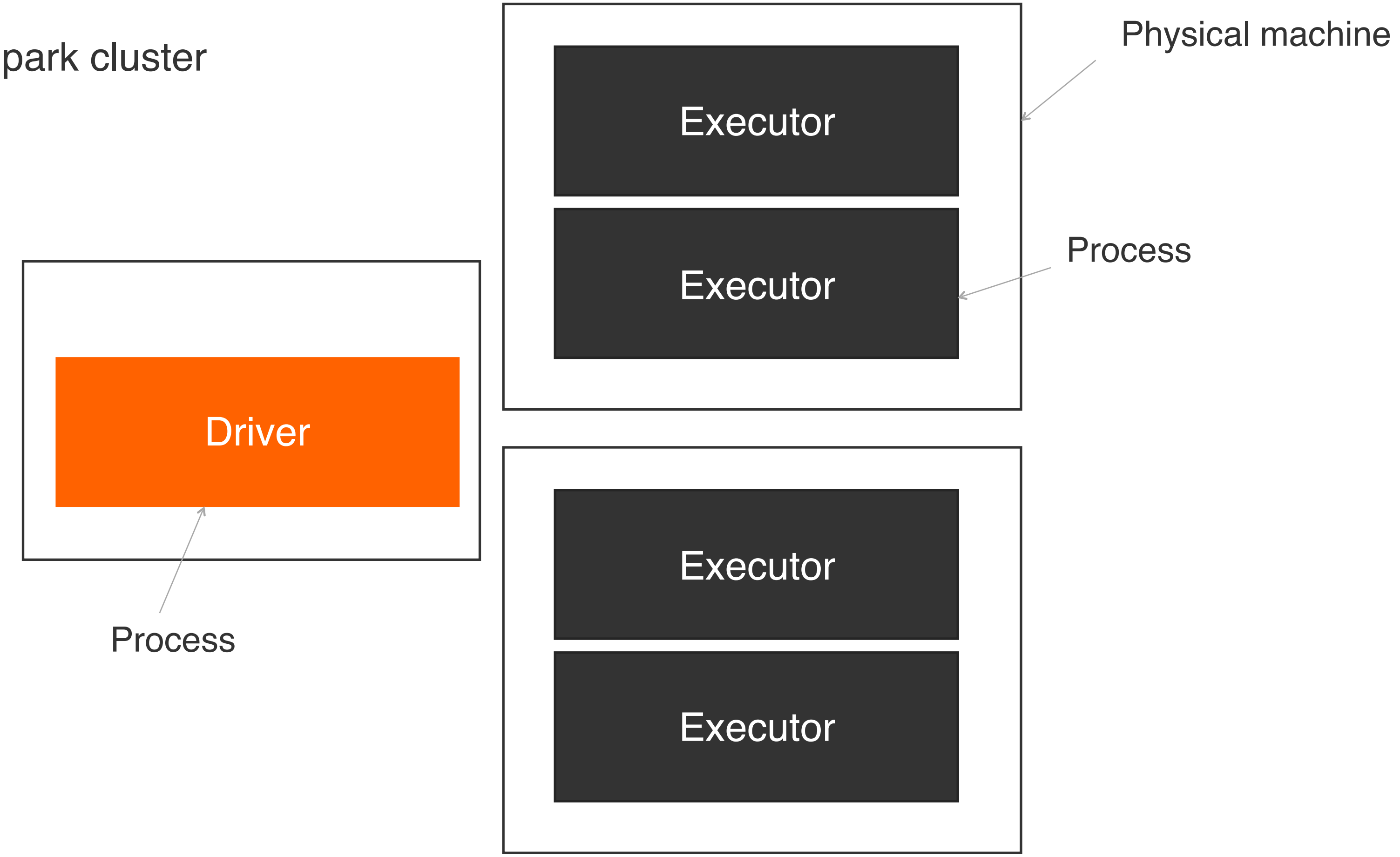
Executor

Write Output file

Write Output file

# Spark Environment

Orchestration model of Spark cluster



Physical machine

Process

Executor

Executor

Driver

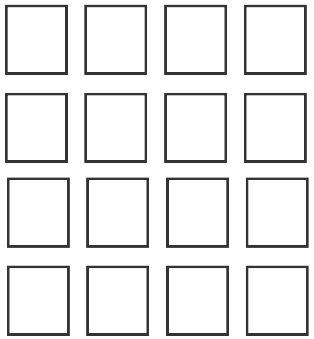
Executor

Executor

Process

# Spark Environment

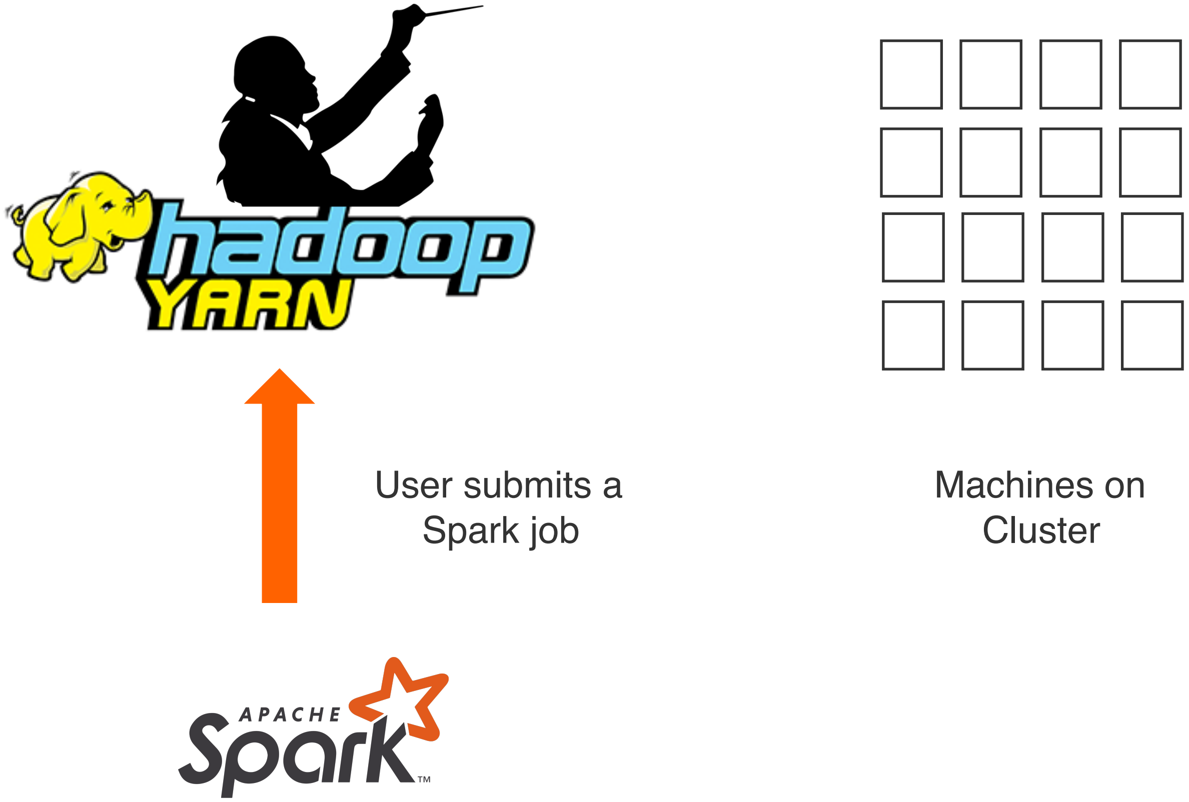
Orchrestration model of Spark cluster



Machines on
Cluster

# Spark Environment

Orchestration model of Spark cluster

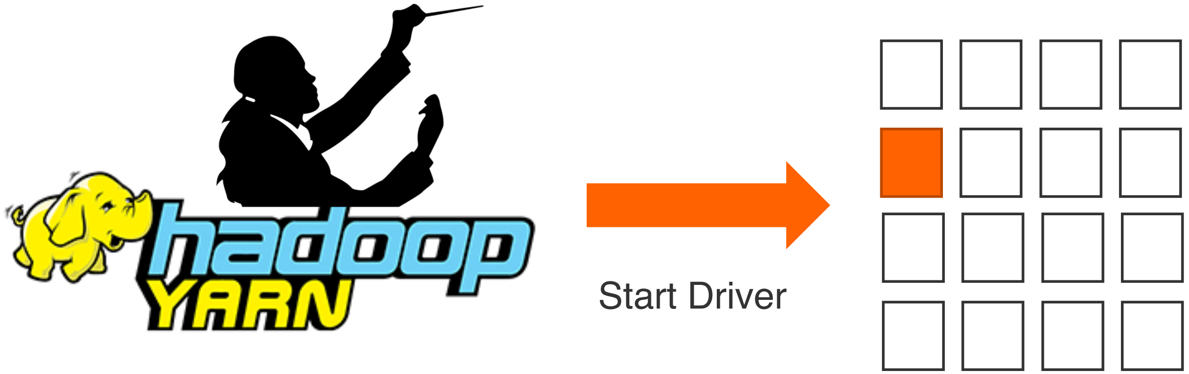


User submits a Spark job

Machines on Cluster

# Spark Environment

Orchrestration model of Spark cluster

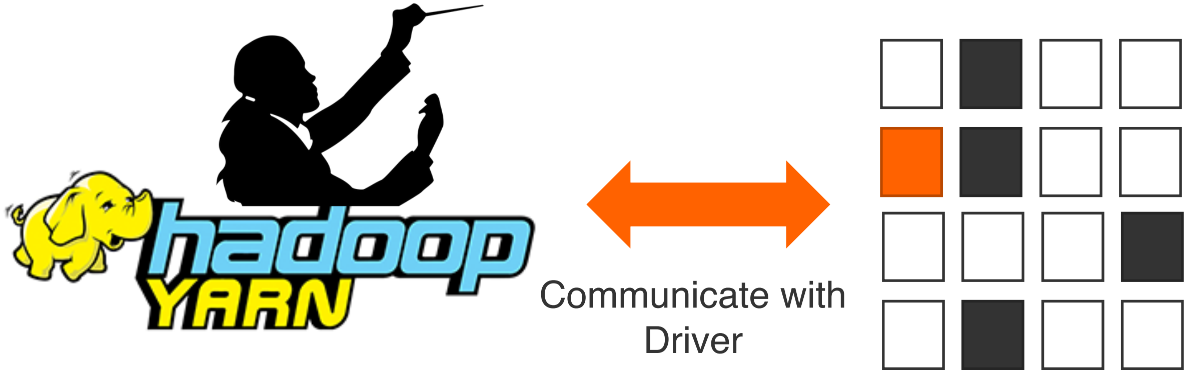


Start Driver

Machines on Cluster

# Spark Environment

Orchrestration model of Spark cluster



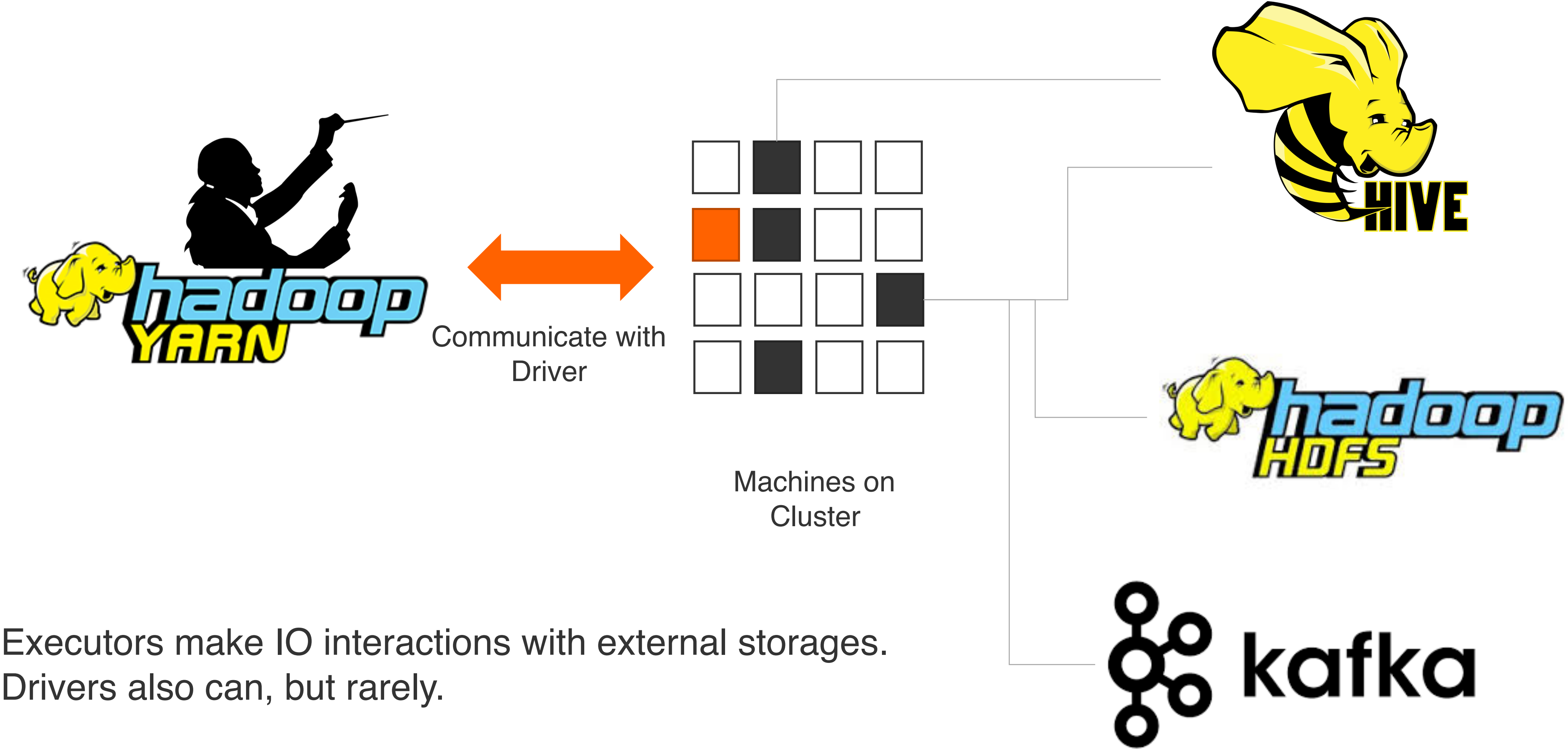
Communicate with Driver

Machines on Cluster

Driver assigns executors, define tasks and distribute them to executors

# Spark Environment

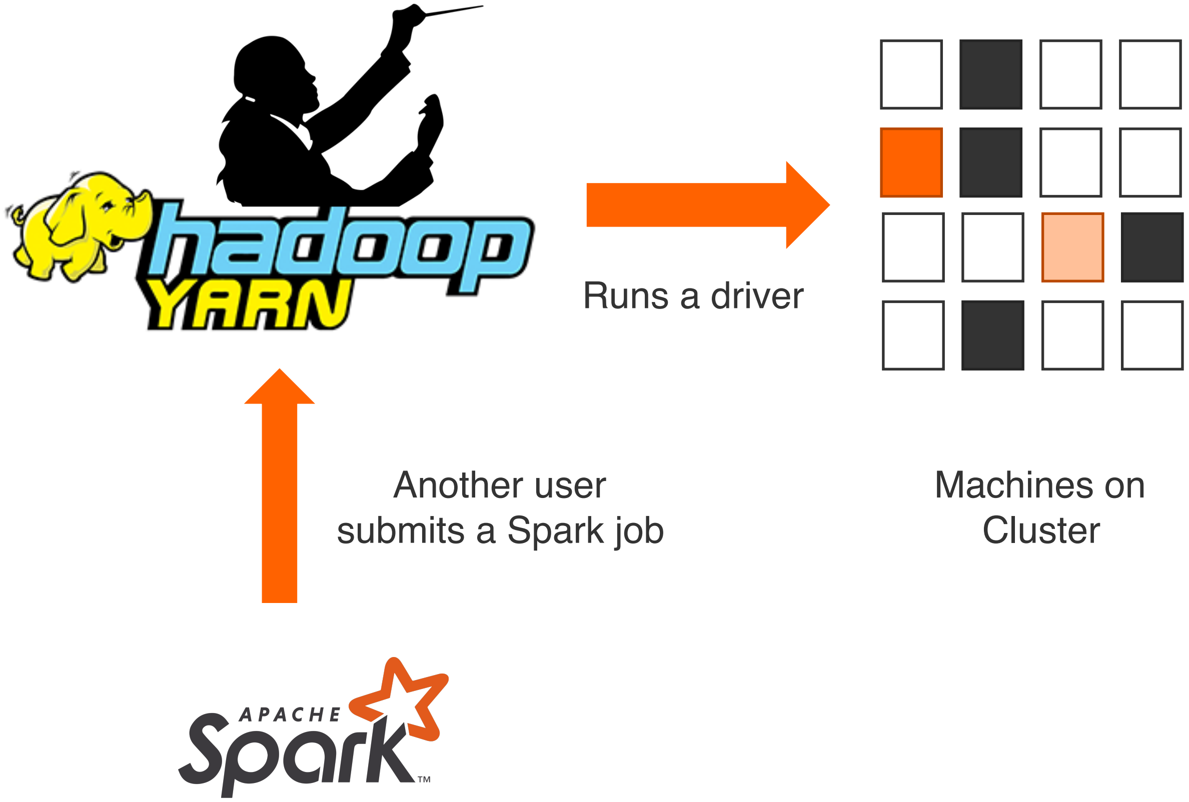
Orchrestration model of Spark cluster



Executors make IO interactions with external storages.
Drivers also can, but rarely.

# Spark Environment

Orchrestration model of Spark cluster



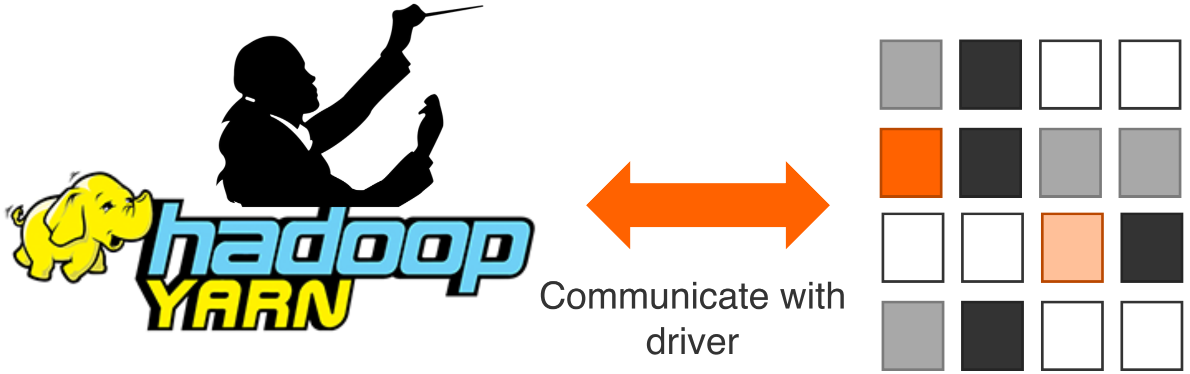
Runs a driver

Another user
submits a Spark job

Machines on
Cluster

# Spark Environment

Orchrestration model of Spark cluster



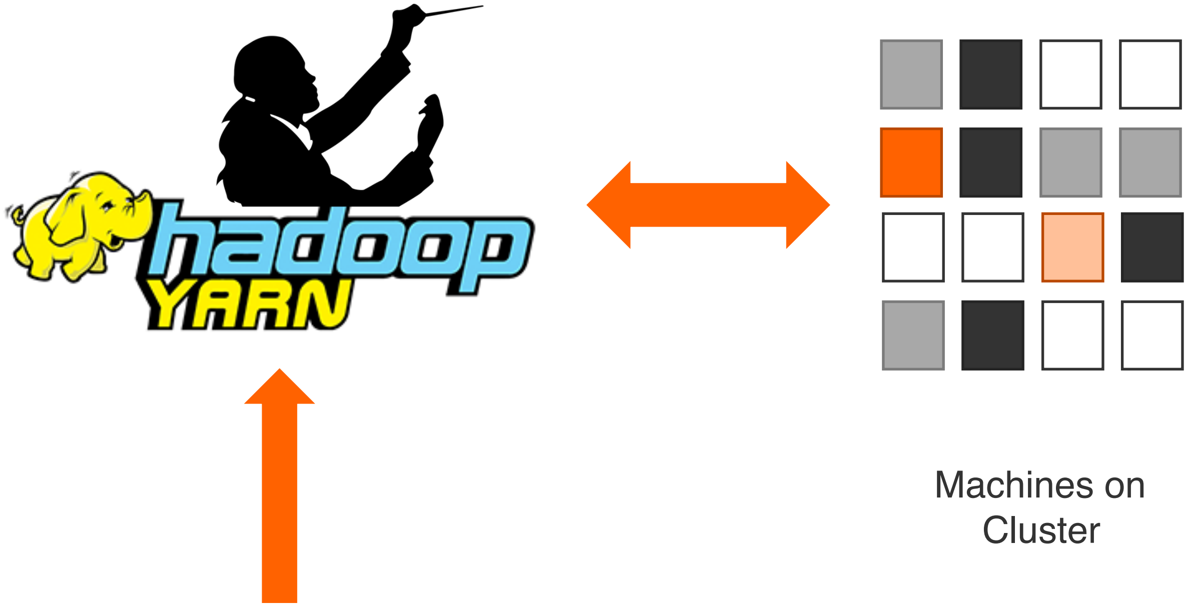
Machines are allocated to run new executors

Machines on Cluster

**NOTE:** One physical machine can run more than one executors.

# Spark Environment

Orchestration model of Spark cluster



Machines on Cluster

1. Driver takes forever to start
2. Though the driver starts, during the run the job halts.

Large job submitted, requiring lots of resources beyond the supply.

# Contributors and Providers of Spark

Official merger began in 2019

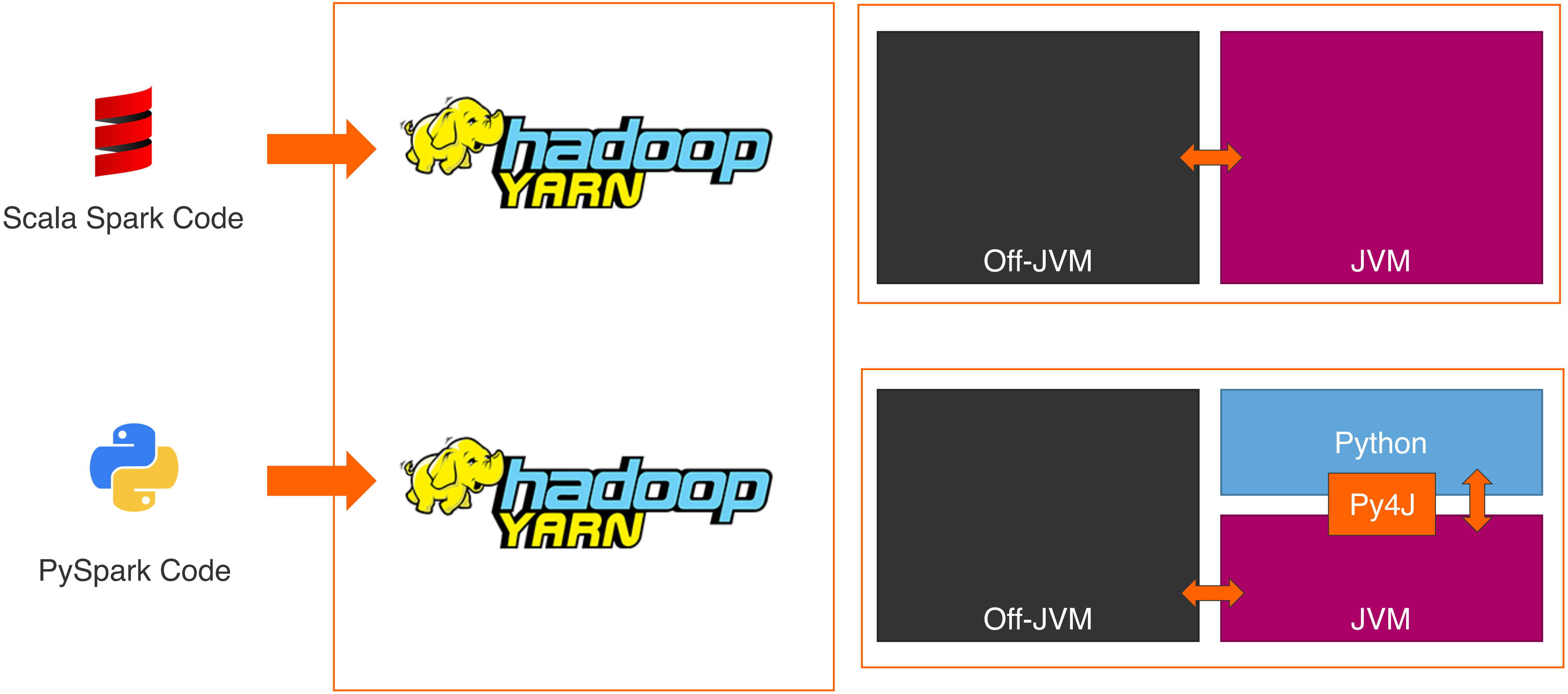
| databricks | CLOUDERA | HORTONWORKS® |
|---|---|---|
| Contributor of Spark codebase | Vendor of Spark distribution **(CDH)** | Vendor of Spark distribution **(HDP)** |
| MLLib | Parquet (collab. with **Twitter**) | ORC (collab. with **Facebook**) |
| DeltaLake | Hue | NiFi |
| MLFlow | Impala | |

# Spark is 72% written in Scala

| | With Scala | With Python |
|---|---|---|
| **Performance** | 10x faster,<br>more memory efficient | |
| **Object serialisation** | All types, case classes are natively supported | Native python types, not with numpy types |
| **RDD/Dataframe API** | Y | Y |
| **Typed Dataset API** | Y | N |
| **Notebook** | Yes, Zeppelin | Yes, Jupyter |
| **Types error** | Compilation time | Runtime only |

# PySpark vs Scala Spark

Executor Memory Region



Scala Spark Code

PySpark Code

Off-JVM

JVM

Off-JVM

Python

Py4J

JVM