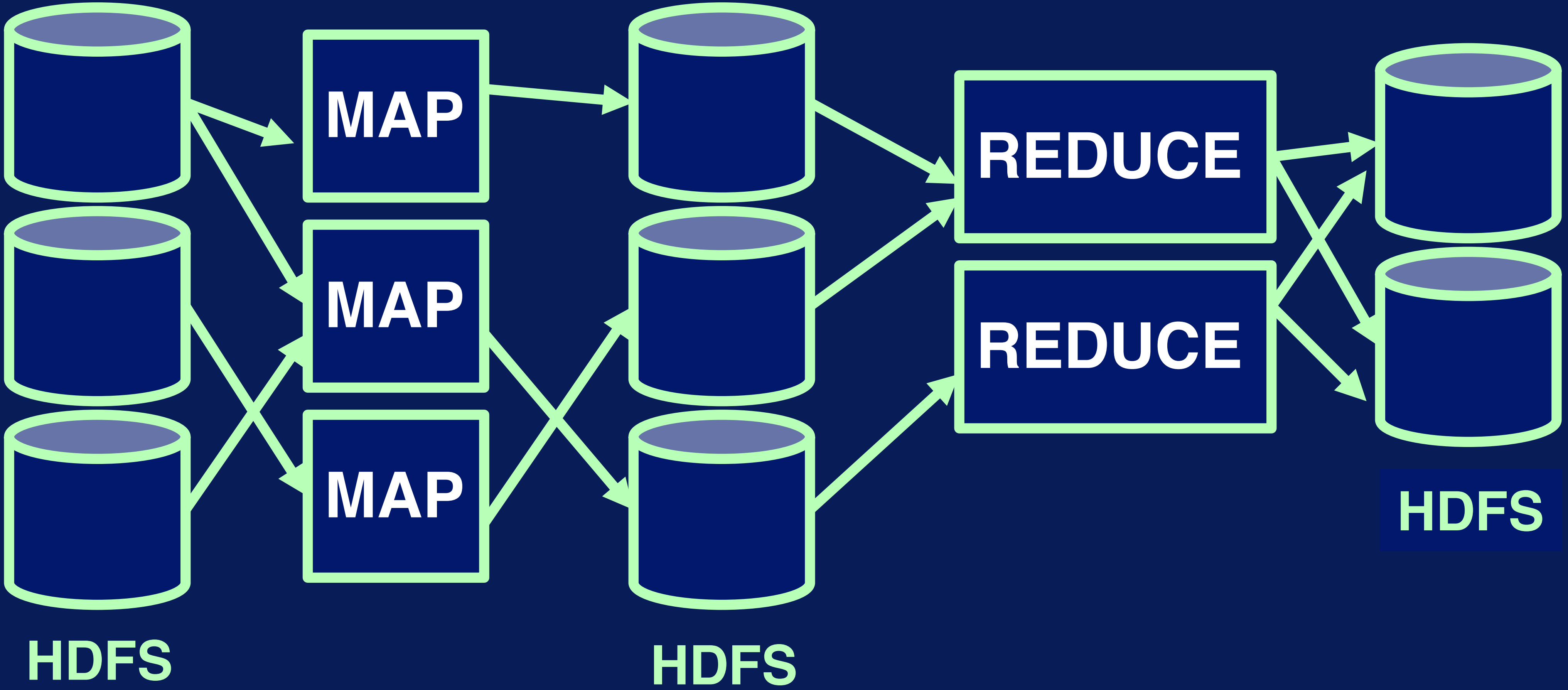
Getting Started with Spark: The Architecture and Basic Concepts

After this video you will be able to..

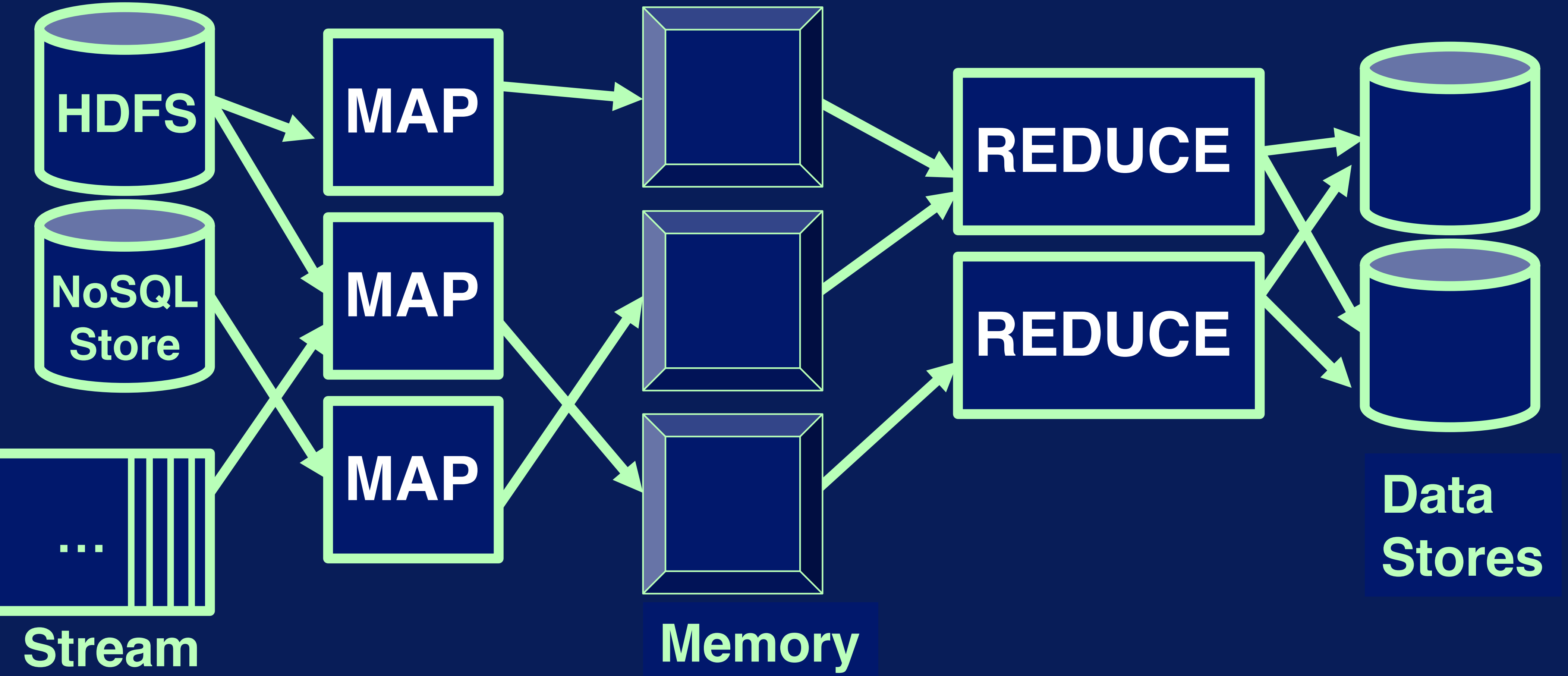
- Describe how Spark does in-memory processing using the RDD abstraction
- Explain the inner workings of the Spark architecture
- Summarize how Spark manages and executes code on Clusters

What does in memory processing mean?

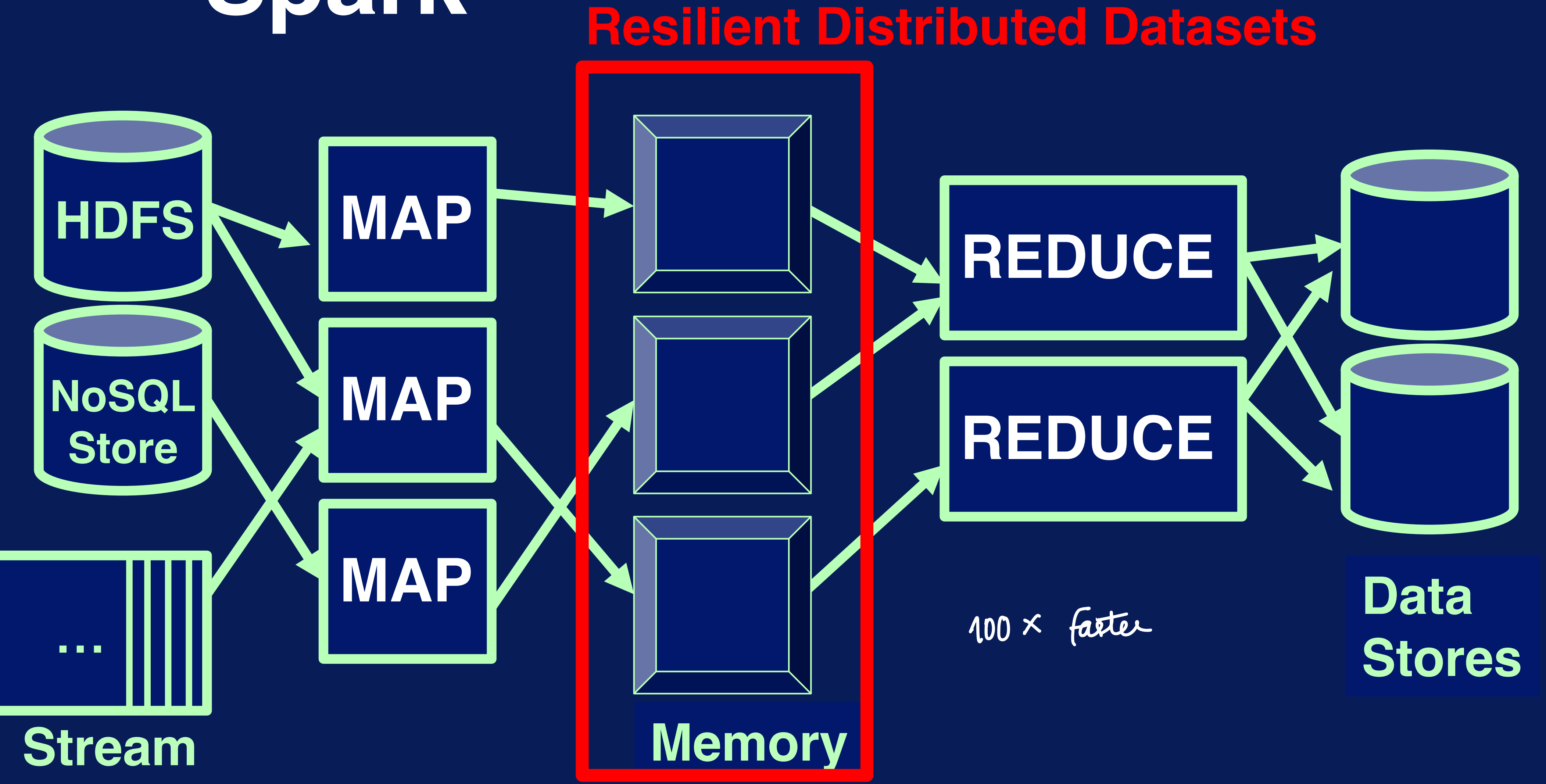
MapReduce



Spark


HDFS
NoSQL Store
...
Stream
MAP
MAP
MAP
Memory
REDUCE
REDUCE
Data Stores

Spark



100 × faster

Resilient Distributed **Datasets**

Dataset

Data storage created from: HDFS, S3, HBase, JSON, text, Local hierarchy of folders

Or created transforming another RDD

inmutable
create new

Resilient **Distributed** Datasets

Distributed

Distributed across the cluster of machines

Divided in partitions, atomic chunks of data

Resilient Distributed Datasets

Resilient

Recover from errors, e.g. node failure, slow processes

Track history of each partition, re-run

* it is common to have node failures

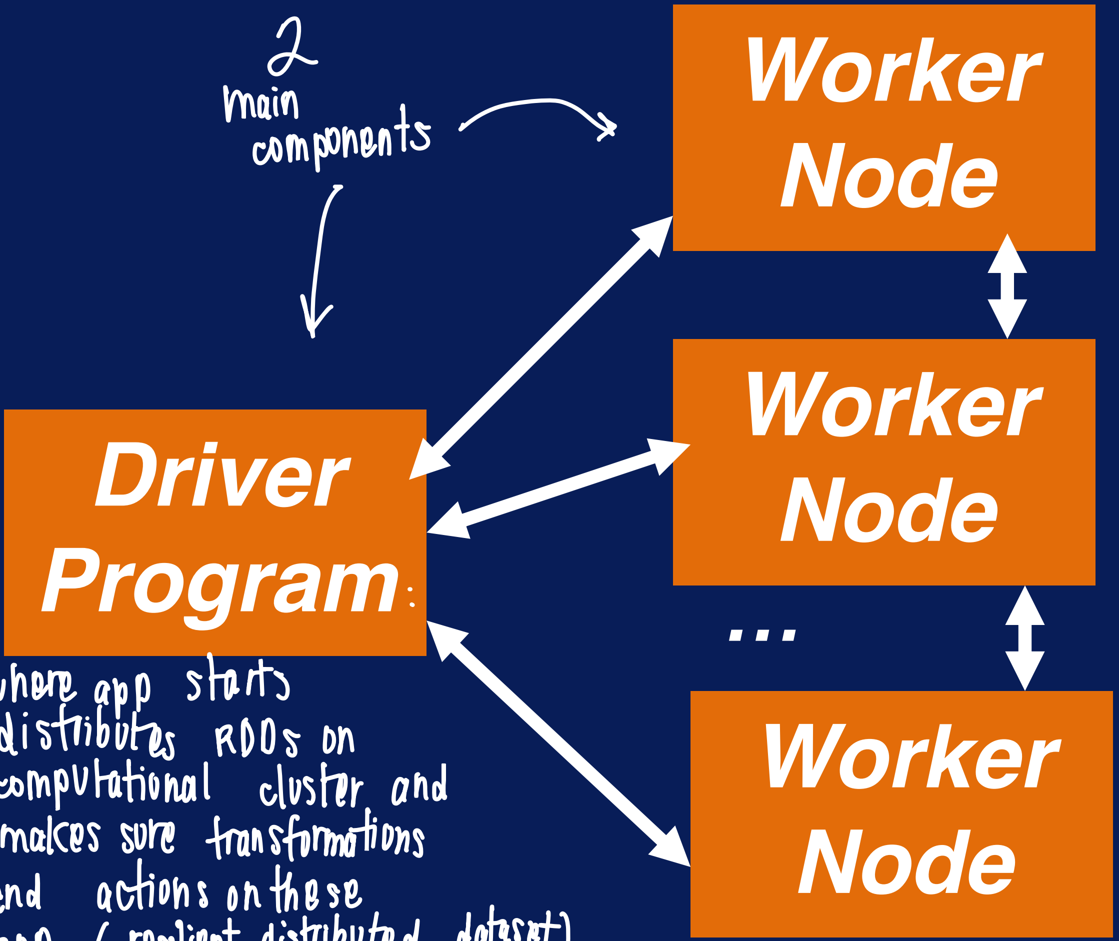
↓

its important to recover without losing work already done

spark knows which are the partitions needed to recreate the partition in case it gets lost

Spark Architecture

2 main components



Worker Node

Worker Node

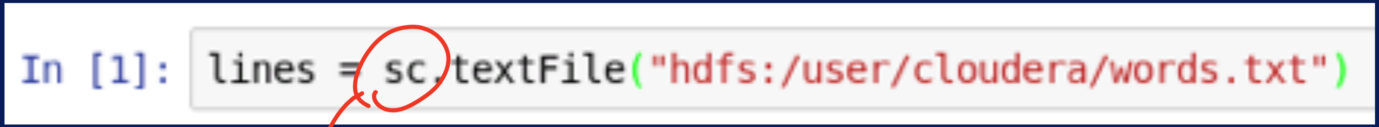
...

Worker Node

Driver Program:

- where app starts
- distributes RDDs on computational cluster and makes sure transformations and actions on these RDDs (resilient distributed dataset) are performed

Driver Program

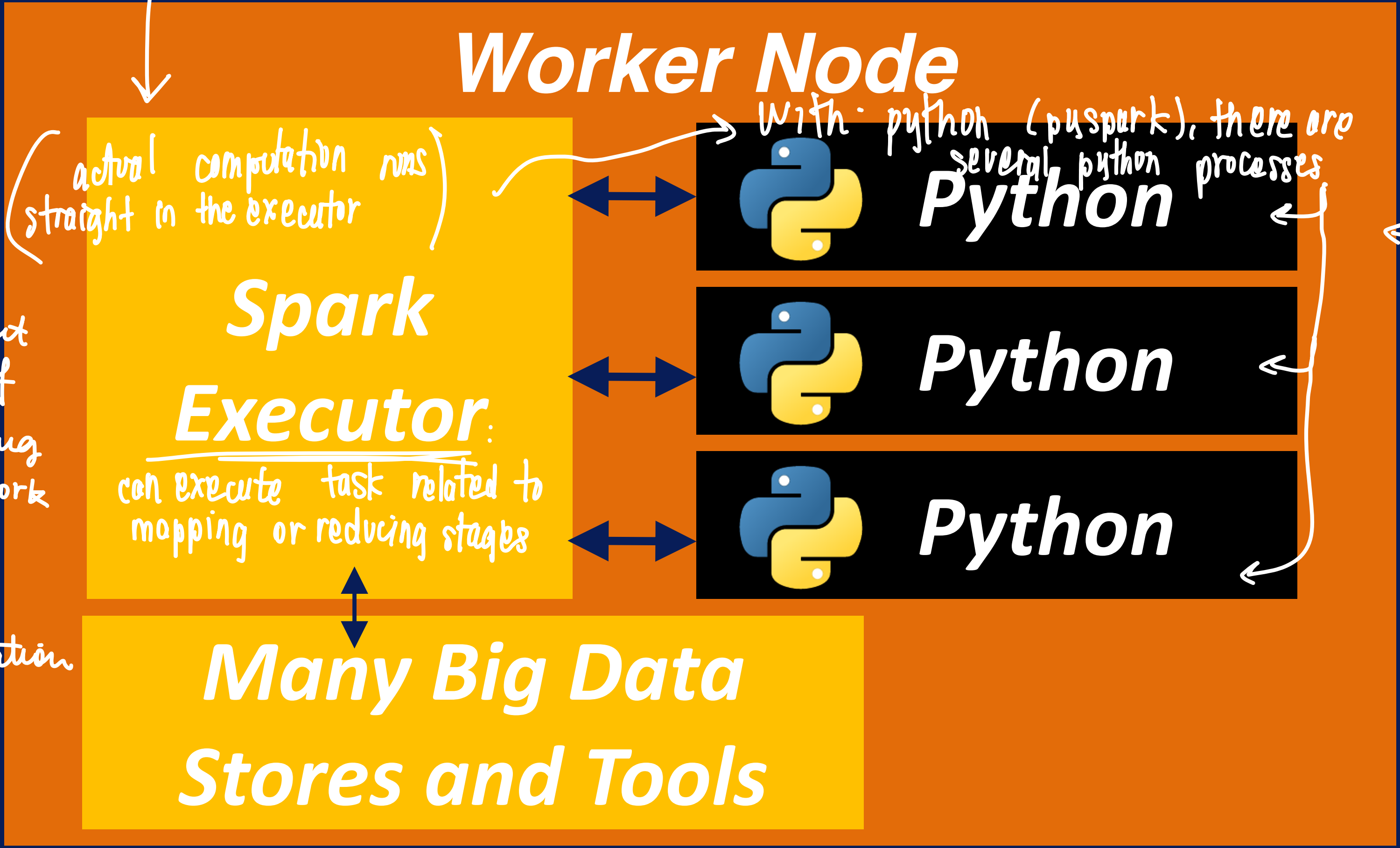


```
In [1]: lines = sc.textFile("hdfs:/user/cloudera/words.txt")
```

- creates a connection to a spark cluster / or local spark through a Spark context object

default: sc (spark context)

it keeps a running java virtual machine



Worker Node

actual computation runs straight in the executor

Spark Executor: can execute task related to mapping or reducing stages

with python (pyspark), there are several python processes

Python

Python

Python

Many Big Data Stores and Tools

* the most important point of this computing framework is to bring the computation to data

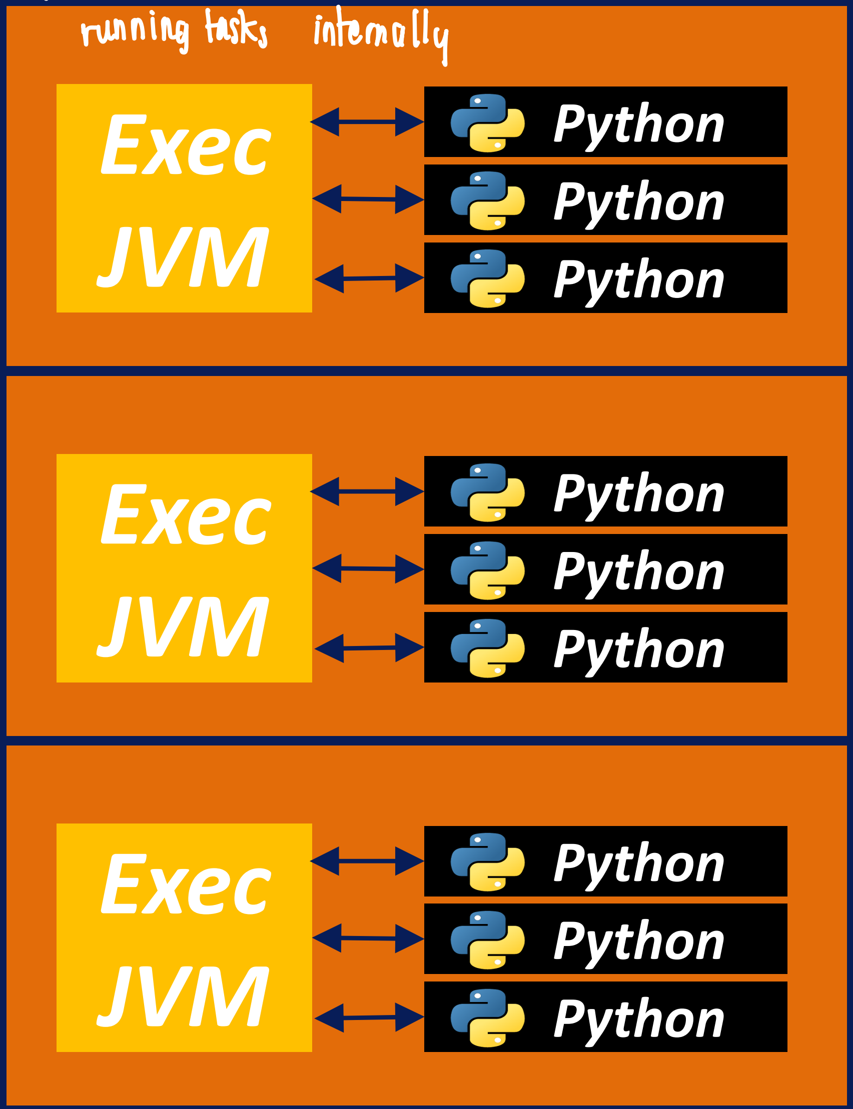
driver manages a potentially large number of nodes : worker nodes

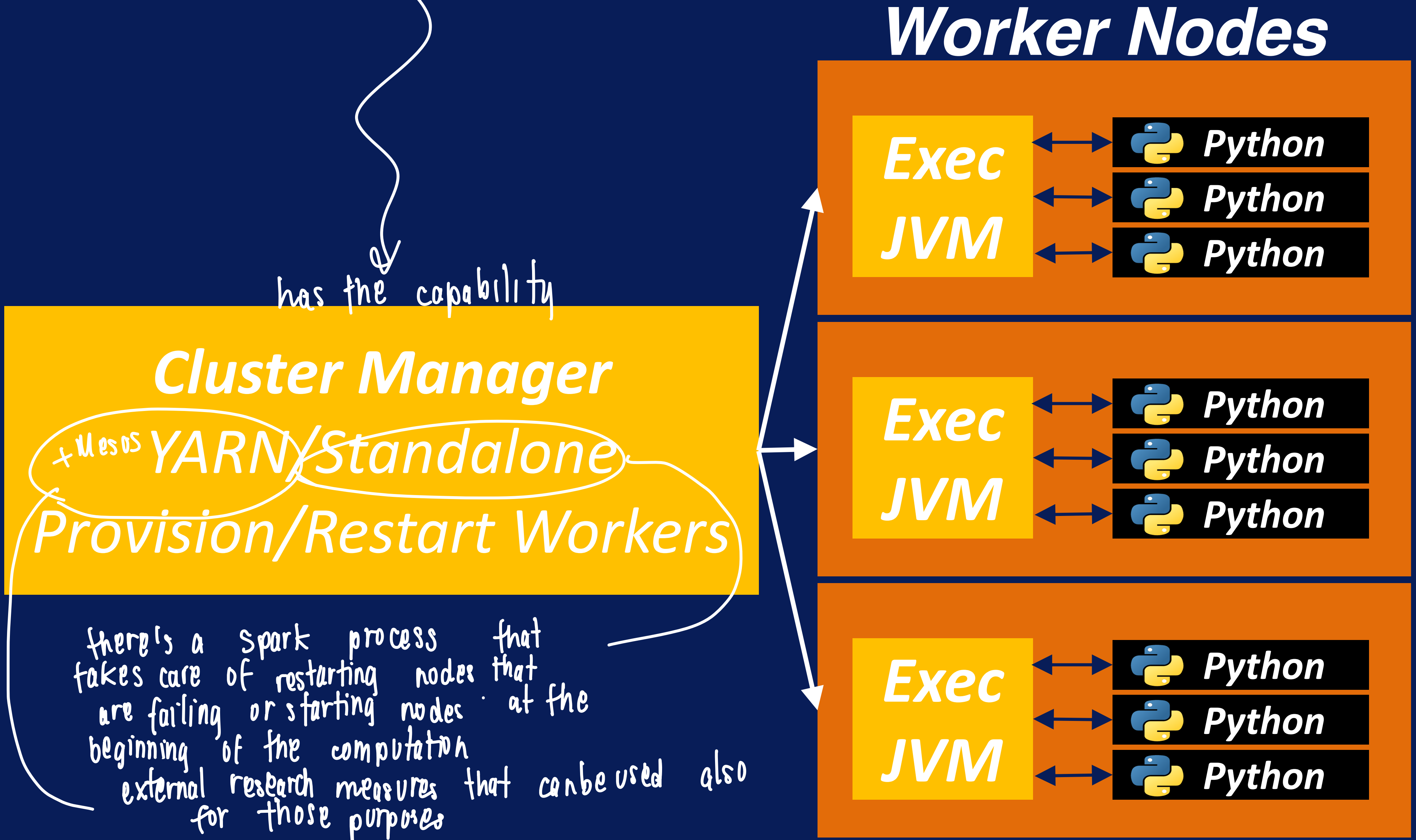
* on a local computer, we can assume there's only one worker node ; where operations execute

many

Worker Nodes

* it is important to have a system that can automatically manage provisioning and restarting of these nodes





Worker Nodes

has the capability

Cluster Manager

+Mesos YARN/Standalone

Provision/Restart Workers

Exec JVM ↔ Python, Python, Python

Exec JVM ↔ Python, Python, Python

Exec JVM ↔ Python, Python, Python

there's a spark process that takes care of restarting nodes that are failing or starting nodes at the beginning of the computation

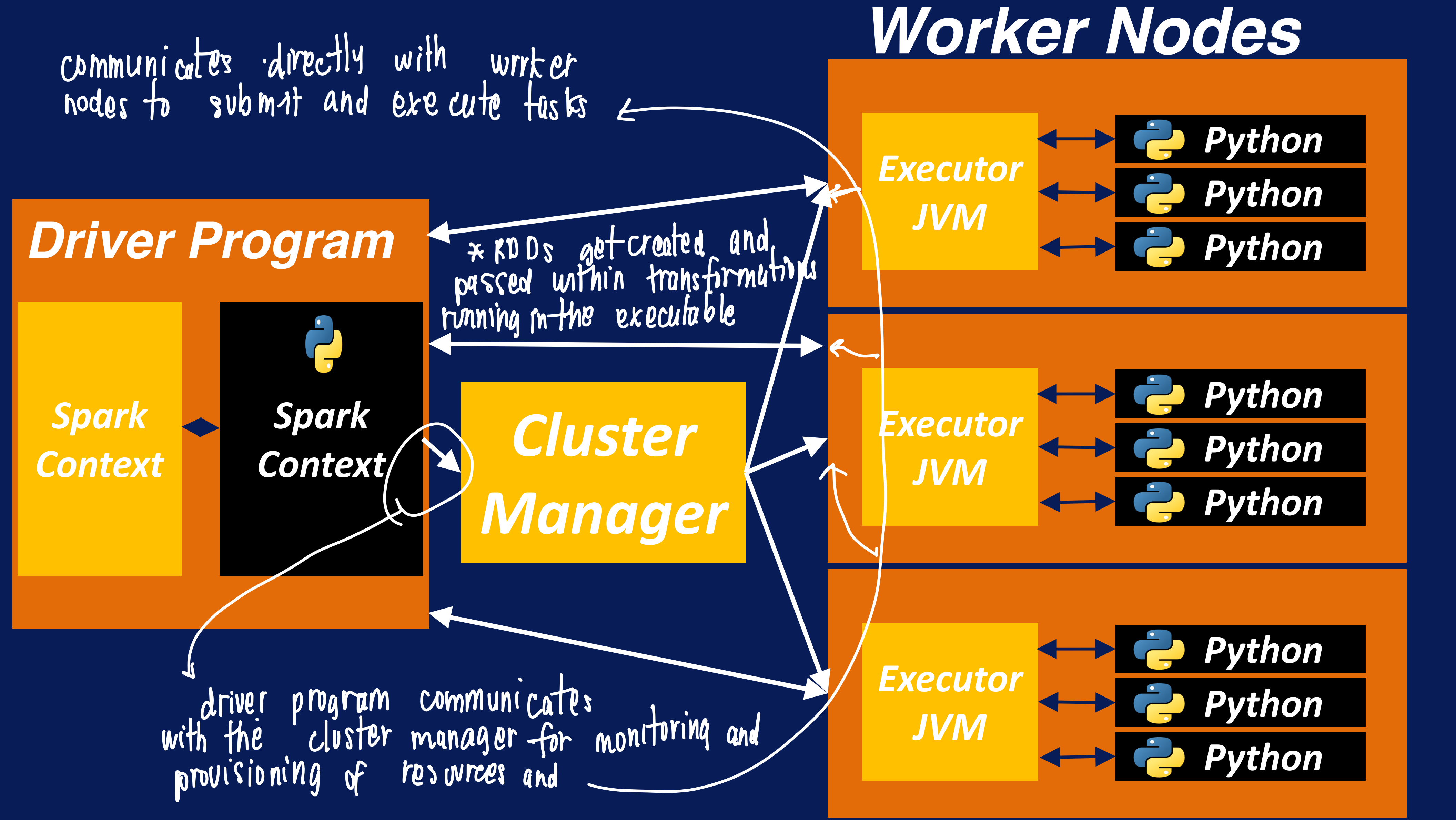
external research measures that can be used also for those purposes

Which cluster manager?

* choosing

how to pick the right cluster manager for your organization :

http://www.agildata.com/apache-spark-cluster-managers-yarn-mesos-or-standalone/



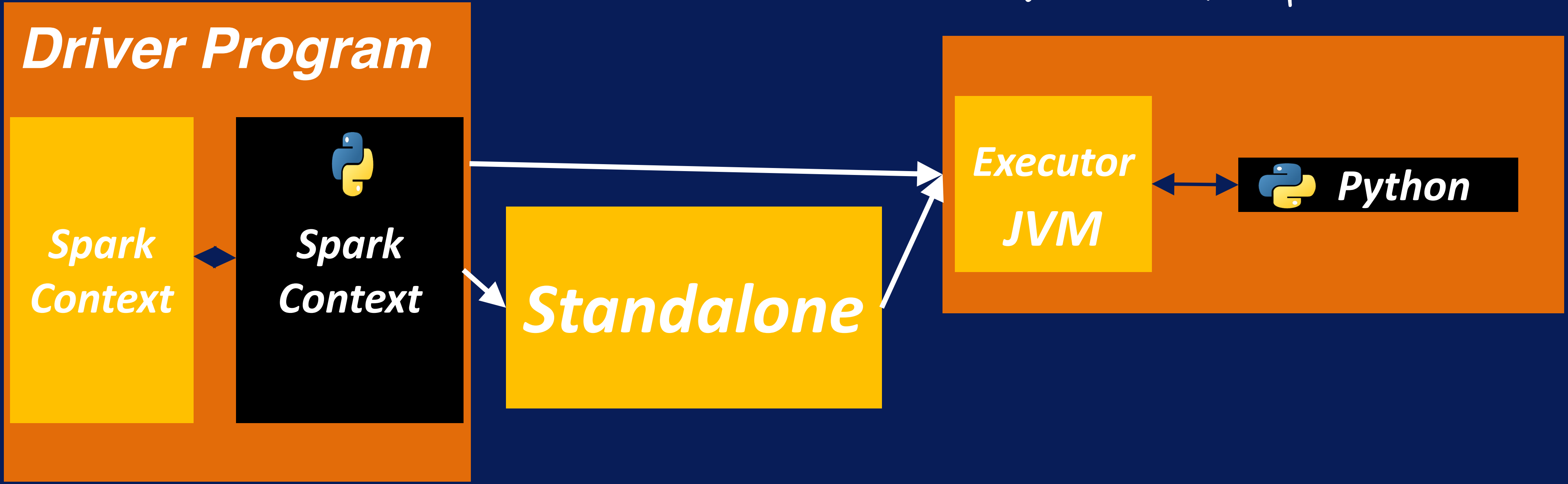
communicates directly with wrrker nodes to submit and execute tasks

* RDDs get created and passed within transformations running in the executable

driver program communicates with the cluster manager for monitoring and provisioning of resources and

Cloudera VM

everything runs locally
(a single machine)
⌊ with driver program
⌊ executor JVM
⌊ and single pyspark process


Driver Program
Spark Context
Spark Context
Standalone
Executor JVM
Python