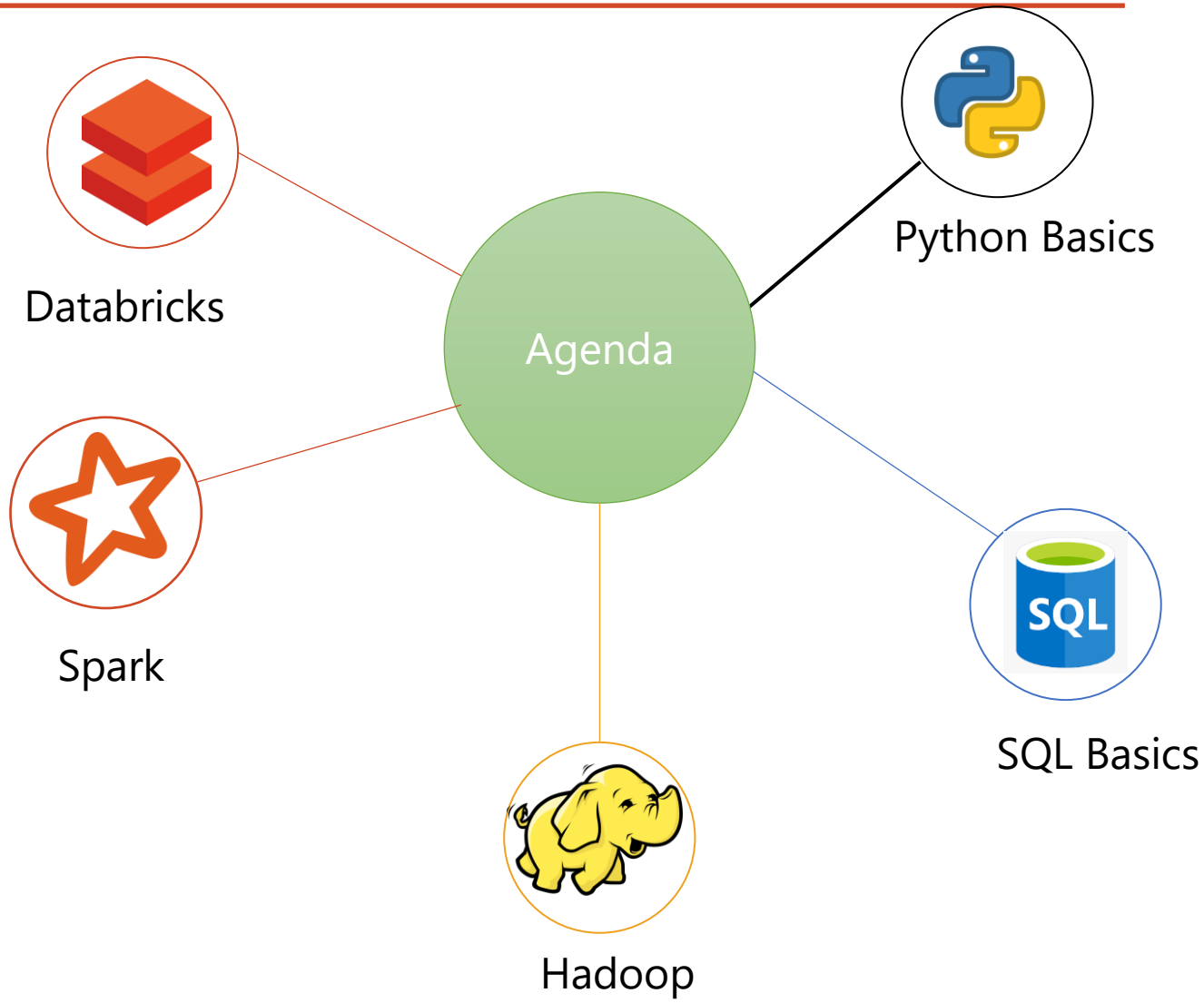
Data Bricks Training- Day2

By: Gaurav Gangwar

Agenda of the training

Topics to be covered:

- ❑Python Basics
- ❑SQL Basics
- ❑Hadoop
- ❑Apache Spark
- ❑Databricks



Agenda of Day-2

- **Introduction to Big Data.**
- **Introduction to Apache Spark.**
- **Spark Toolset.**

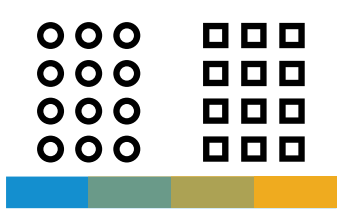
Introduction to Big Data


That how Huge Big Data is!

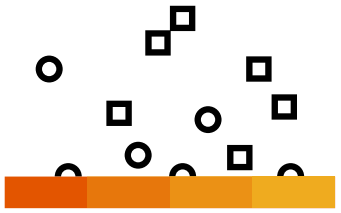
What is Big data?

- Big Data is a collection of data that is huge in volume, yet growing exponentially with time.
- It is a data with so large size and complexity that none of traditional data management tools can store it or process it efficiently.

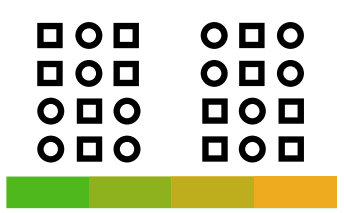
Form of Big Data



Structured data



Unstructured data



Semi-structured data

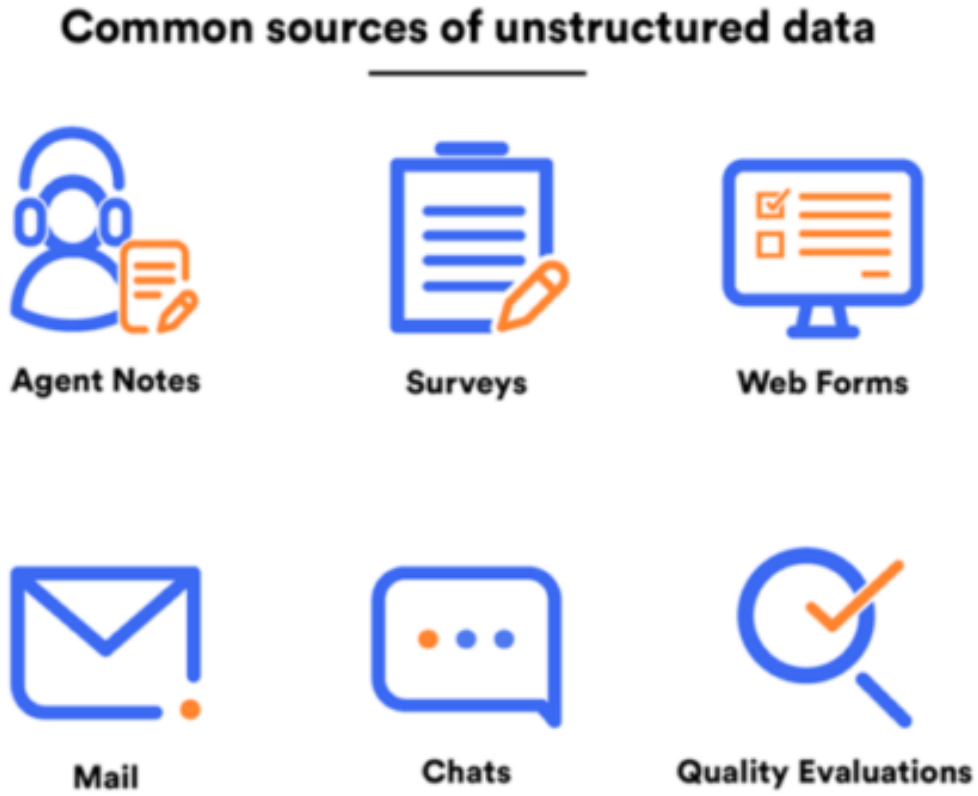
Structured Data

- Any data that can be stored, accessed and processed in the form of fixed format is termed as a ‘structured’ data.

Employee_ID	Employee_Name	Gender	Department	Salary_In_lacs
2365	Rajesh Kulkarni	Male	Finance	650000
3398	Pratibha Joshi	Female	Admin	650000
7465	Shushil Roy	Male	Admin	500000

Unstructured Data

- Unstructured data is information that either does not have a pre-defined data model or is not organized in a pre-defined manner. Unstructured information is typically text-heavy, but may contain data such as dates, numbers, and facts as well audio, video, pdf and survey data.



Semi Structure Data

- Semi-structured data has some characteristics of both structured and unstructured data.

The five V’s that define Big Data

Volume

- The name Big Data itself is related to a size which is enormous.
- Size of data plays a very crucial role in determining value out of data. Also, whether a particular data can actually be considered as a Big Data or not, is dependent upon the volume of data. Hence, **‘Volume’** is one characteristic which needs to be considered while dealing with Big Data solutions

The five V’s that define Big Data

Velocity

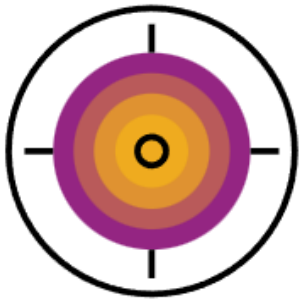
- The term **‘velocity’** refers to the speed of generation of data. How fast the data is generated and processed to meet the demands, determines real potential in the data.
- Big Data Velocity deals with the speed at which data flows in from sources like business processes, application logs, networks, and social media sites, sensors, Mobile devices, etc. The flow of data is massive and continuous.

The five V’s that define Big Data

Variety

- The next aspect of Big Data is its **variety**.
- Variety refers to heterogeneous sources and the nature of data, both structured and unstructured. During earlier days, spreadsheets and databases were the only sources of data considered by most of the applications.
- Nowadays, data in the form of emails, photos, videos, monitoring devices, PDFs, audio, etc. are also being considered in the analysis applications. This variety of unstructured data poses certain issues for storage, mining and analyzing data.

The five V’s that define Big Data



Veracity

- Veracity is a big data characteristic related to consistency, accuracy, quality, and trustworthiness. Data veracity refers to the biasedness, noise, abnormality in data.
- It also refers to incomplete data or the presence of errors, outliers, and missing values.

The five V’s that define Big Data

Value

- Value refers to the usefulness of gathered data for your business.
- Data by itself, regardless of its volume, usually isn't very useful — to be valuable, it needs to be converted into insights or information, and that is where data processing steps in

Source of Big Data

Some Example

- The **New York Stock Exchange** is an example of Big Data that generates about ***one terabyte*** of new trade data per day.

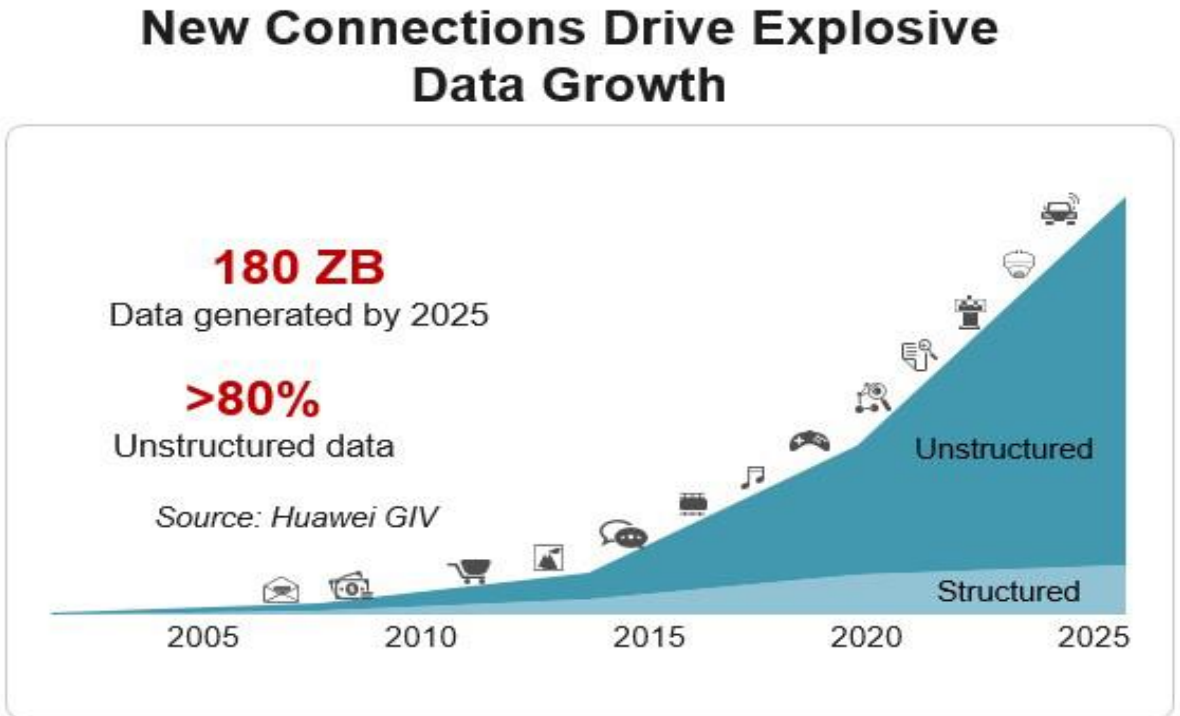
Some Example

- The statistic shows that ***500+terabytes*** of new data get ingested into the databases of social media site **Facebook**, every day.
- This data is mainly generated in terms of photo and video uploads, message exchanges, putting comments etc.

Some Example

- A single **Jet engine** can generate ***10+terabytes*** of data in ***30 minutes*** of flight time. With many thousand flights per day, generation of data reaches up to many ***Petabytes.***

Data Growth Over the Year



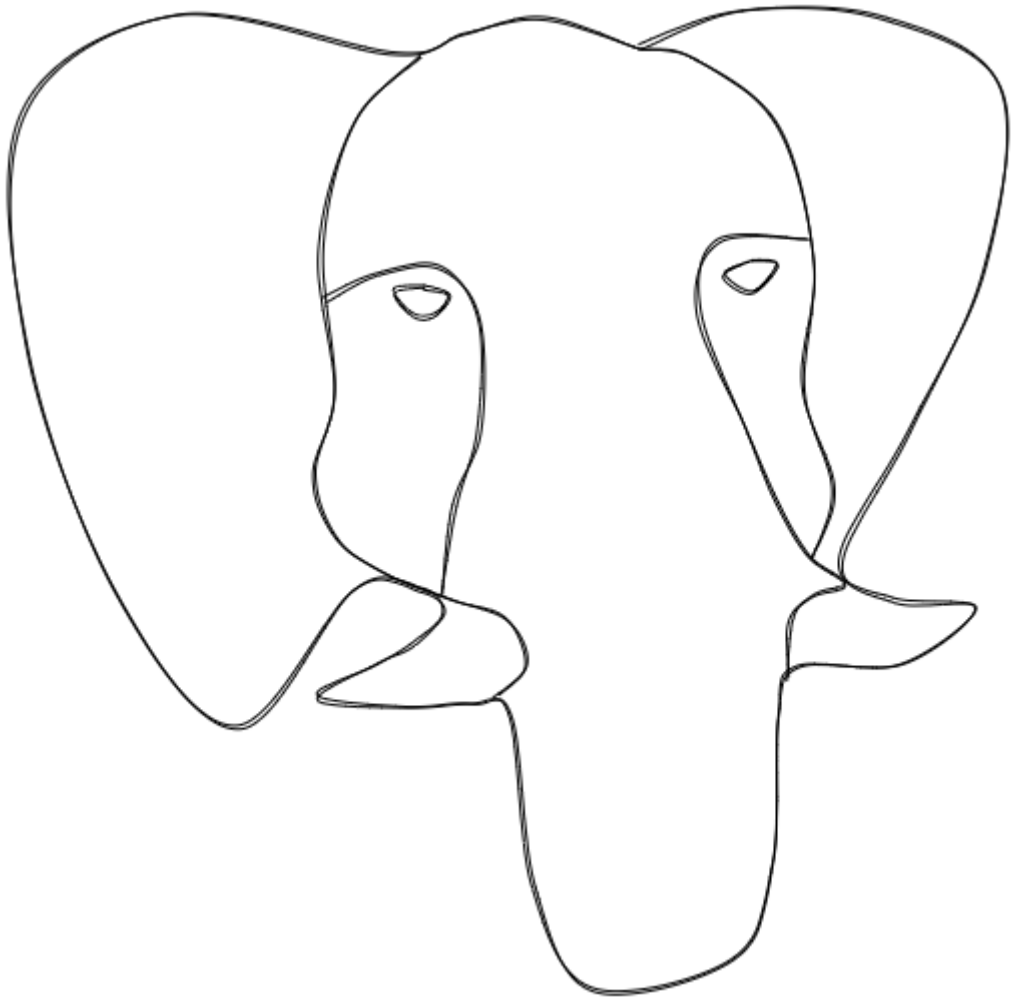
1 PB
Daily production data
Digital interconnected factories

64 TB
Daily training data
Self-driving vehicles

How to Process this Much Data

How to Process this Data.
Current System is not Capable?

Hadoop

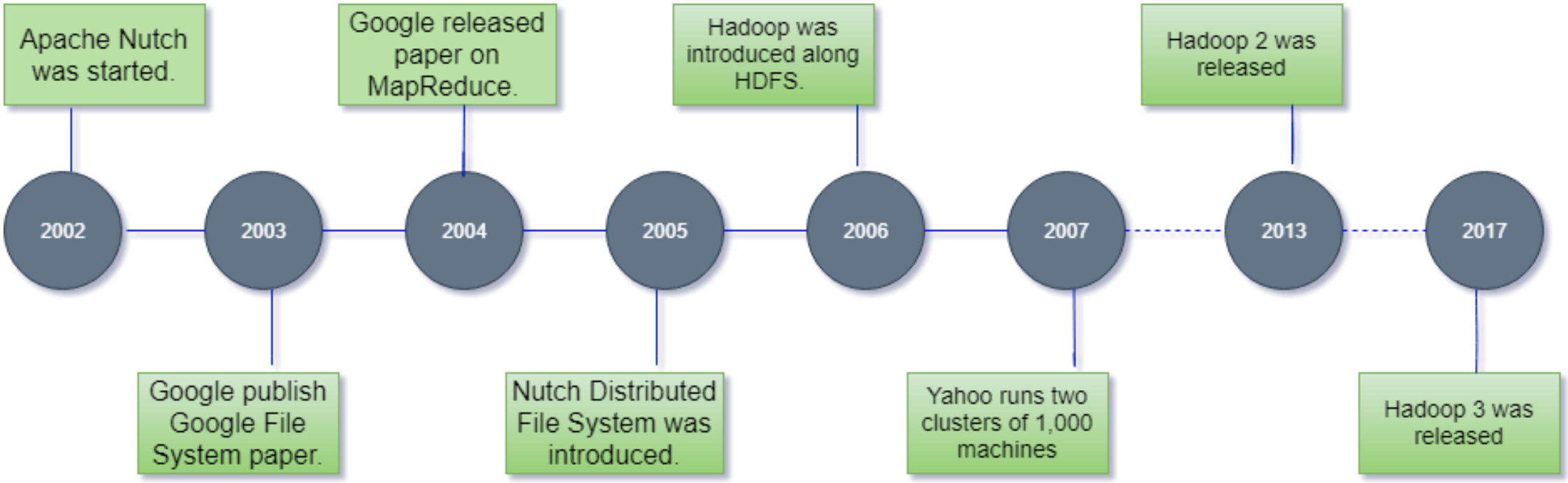


What is Hadoop?

- The Apache Hadoop software library is a framework that allows for the distributed processing of large data sets across clusters of computers using simple programming models.
- It is designed to scale up from single servers to thousands of machines, each offering local computation and storage.

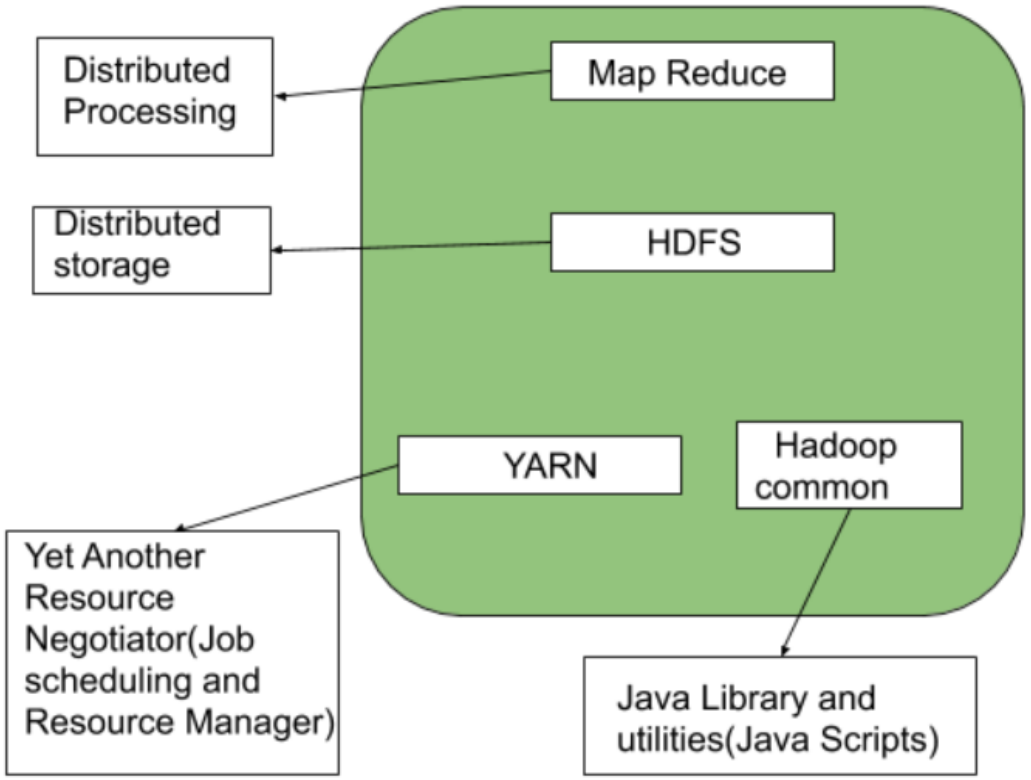
History of Hadoop

The Hadoop was started by Doug Cutting and Mike Cafarella in 2002. Its origin was the Google File System paper, published by Google.



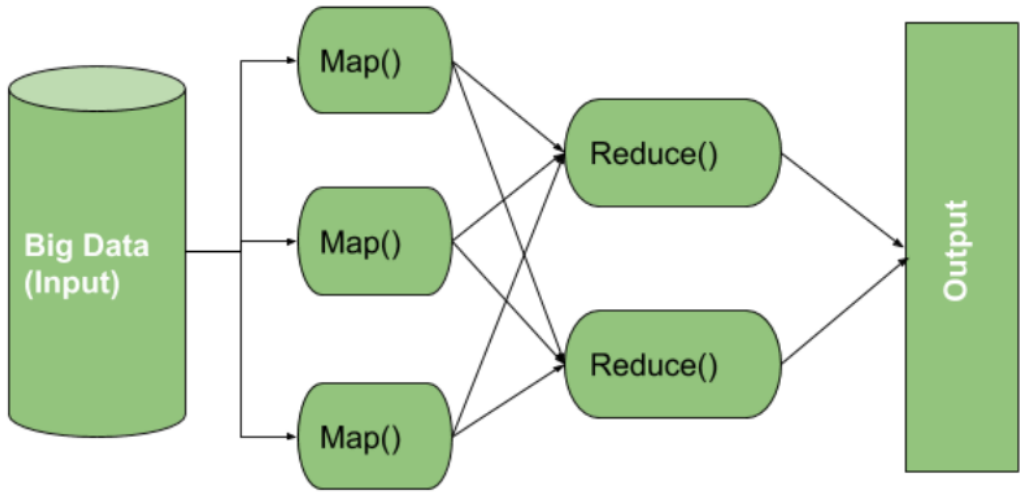
Architecture of Hadoop

- MapReduce
- HDFS(Hadoop distributed File System)
- YARN(Yet Another Resource Framework)
- Common Utilities or Hadoop Common



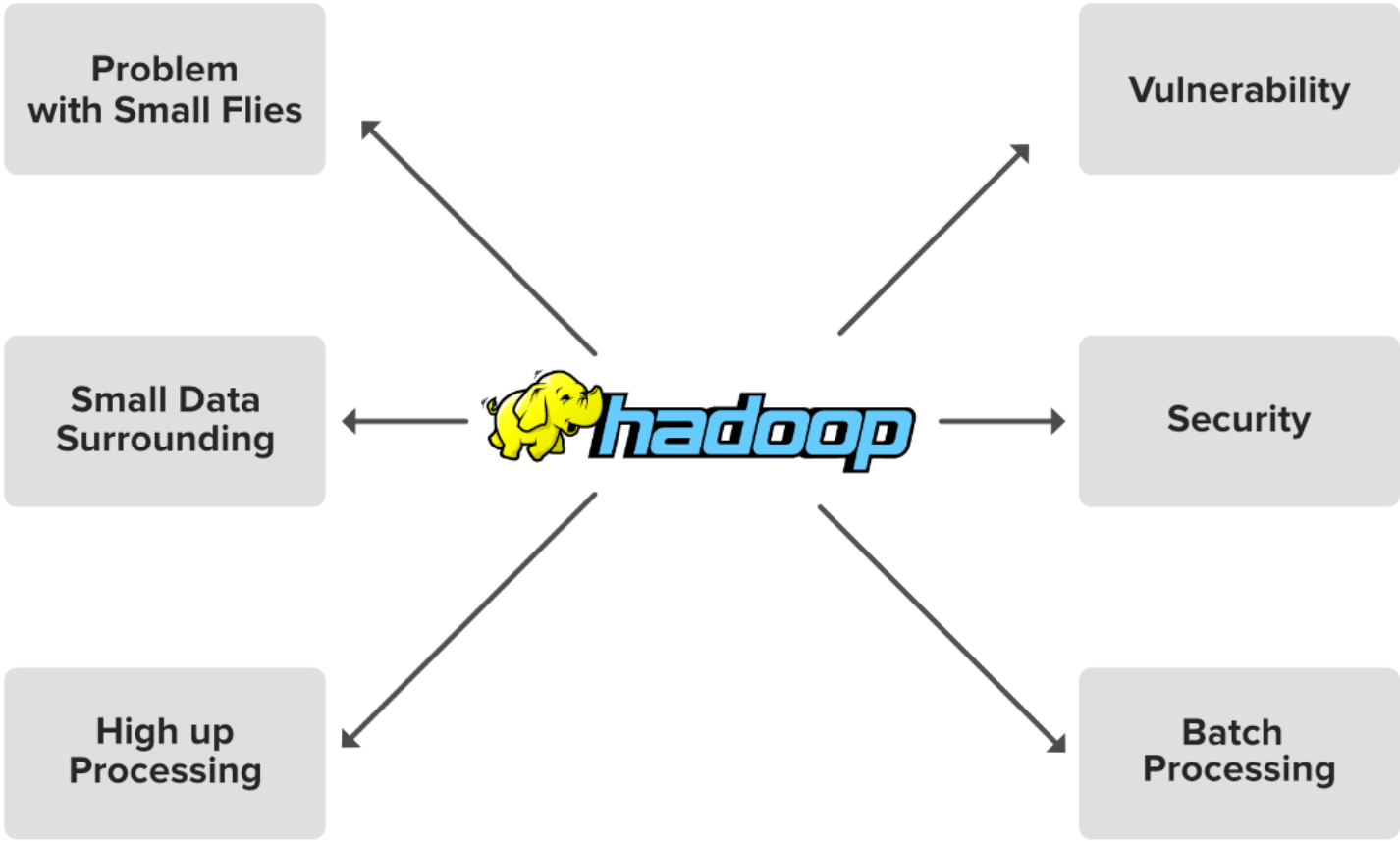
Architecture of Hadoop

- MapReduce nothing but just like an Algorithm or a data structure that is based on the YARN framework.
- The major feature of MapReduce is to perform the distributed processing in parallel in a Hadoop cluster which.
- Makes Hadoop working so fast. When you are dealing with Big Data, serial processing is no more of any use.
- MapReduce has mainly 2 tasks which are divided phase-wise:



Big Data Tools

Limitation of Hadoop



How to Process this Much Data

How to Process this Data.
With Slow processing System?

Introduction to Apache Spark

What is Apache Spark?

- Apache Spark is an open-source, distributed processing system used for big data workloads.
- It utilizes in-memory caching, and optimized query execution for fast analytic queries against data of any size.
- It provides development APIs in Java, Scala, Python and R.
- And supports code reuse across multiple workloads—batch processing, interactive queries, real-time analytics, machine learning, and graph processing.

History of Apache Spark

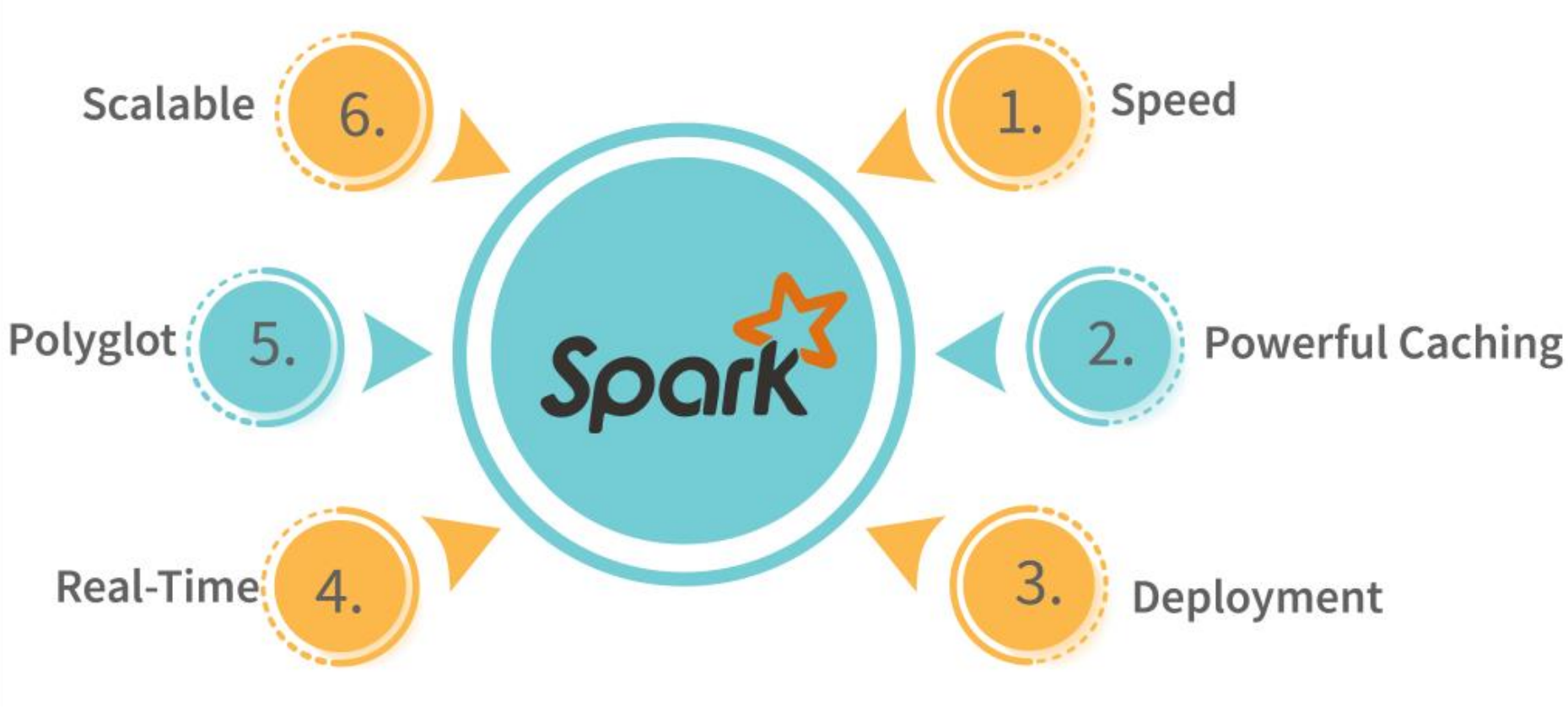
- Apache Spark started in 2009 as a research project at UC Berkley's AMPLab, a collaboration involving students, researchers, and faculty, focused on data-intensive application domains.
- The goal of Spark was to create a new framework, optimized for fast iterative processing like machine learning, and interactive data analysis, while retaining the scalability, and fault tolerance of Hadoop MapReduce.
- The first paper entitled, "Spark: Cluster Computing with Working Sets" was published in June 2010.
- Launch in 2014, Spark can run standalone, on Apache Mesos, or most frequently on Apache Hadoop.

Hadoop vs Spark

S.No	Hadoop	Spark
1.	Hadoop is an open source framework which uses a MapReduce algorithm	Spark is lightning fast cluster computing technology, which extends the MapReduce model to efficiently use with more type of computations.
2.	Hadoop's MapReduce model reads and writes from a disk, thus slow down the processing speed	Spark reduces the number of read/write cycles to disk and store intermediate data in-memory, hence faster-processing speed.
3.	Hadoop is designed to handle batch processing efficiently	Spark is designed to handle real-time data efficiently.
4.	Hadoop is a high latency computing framework, which does not have an interactive mode	Spark is a low latency computing and can process data interactively.
5.	With Hadoop MapReduce, a developer can only process data in batch mode only	Spark can process real-time data, from real time events like twitter, facebook
6.	Hadoop is a cheaper option available while comparing it in terms of cost	Spark requires a lot of RAM to run in-memory, thus increasing the cluster and hence cost.
7.	The PageRank algorithm is used in Hadoop.	Graph computation library called GraphX is used by Spark.

Time for 10 Min Break

Features of Spark



Features of Spark

- **Speed:** Spark performs up to 100 times faster than MapReduce for processing large amounts of data. It is also able to divide the data into chunks in a controlled way.
- **Powerful Caching:** Powerful caching and disk persistence capabilities are offered by a simple programming layer.
- **Deployment:** Mesos, Hadoop via YARN, or Spark's own cluster manager can all be used to deploy it.
- **Real-Time:** Because of its in-memory processing, it offers real-time computation and low latency.
- **Polyglot:** In addition to Java, Scala, Python, and R, Spark also supports all four of these languages. You can write Spark code in any one of these languages. Spark also provides a command-line interface in Scala and Python.

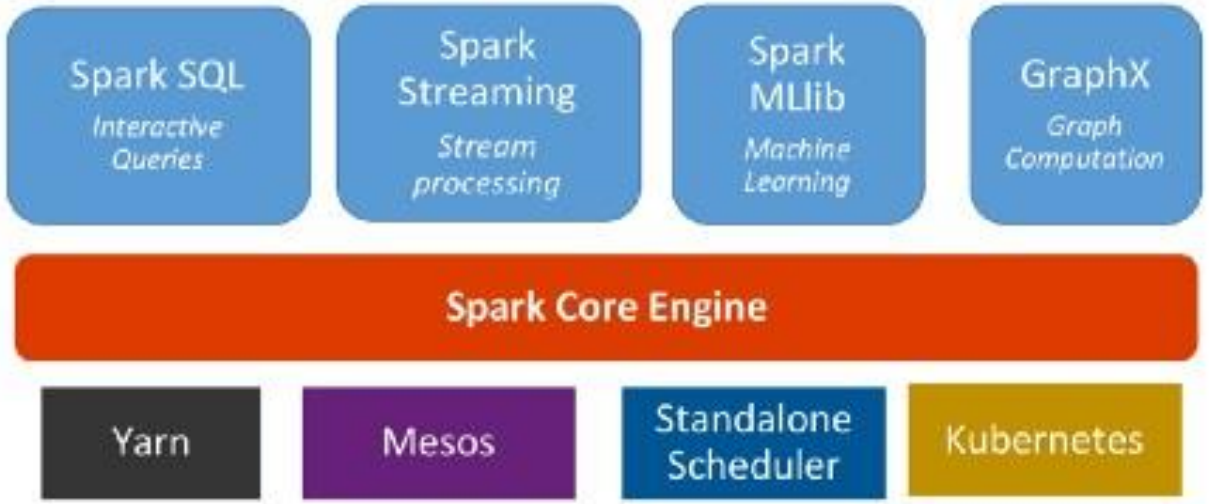
Spark's Basic Architecture

Spark

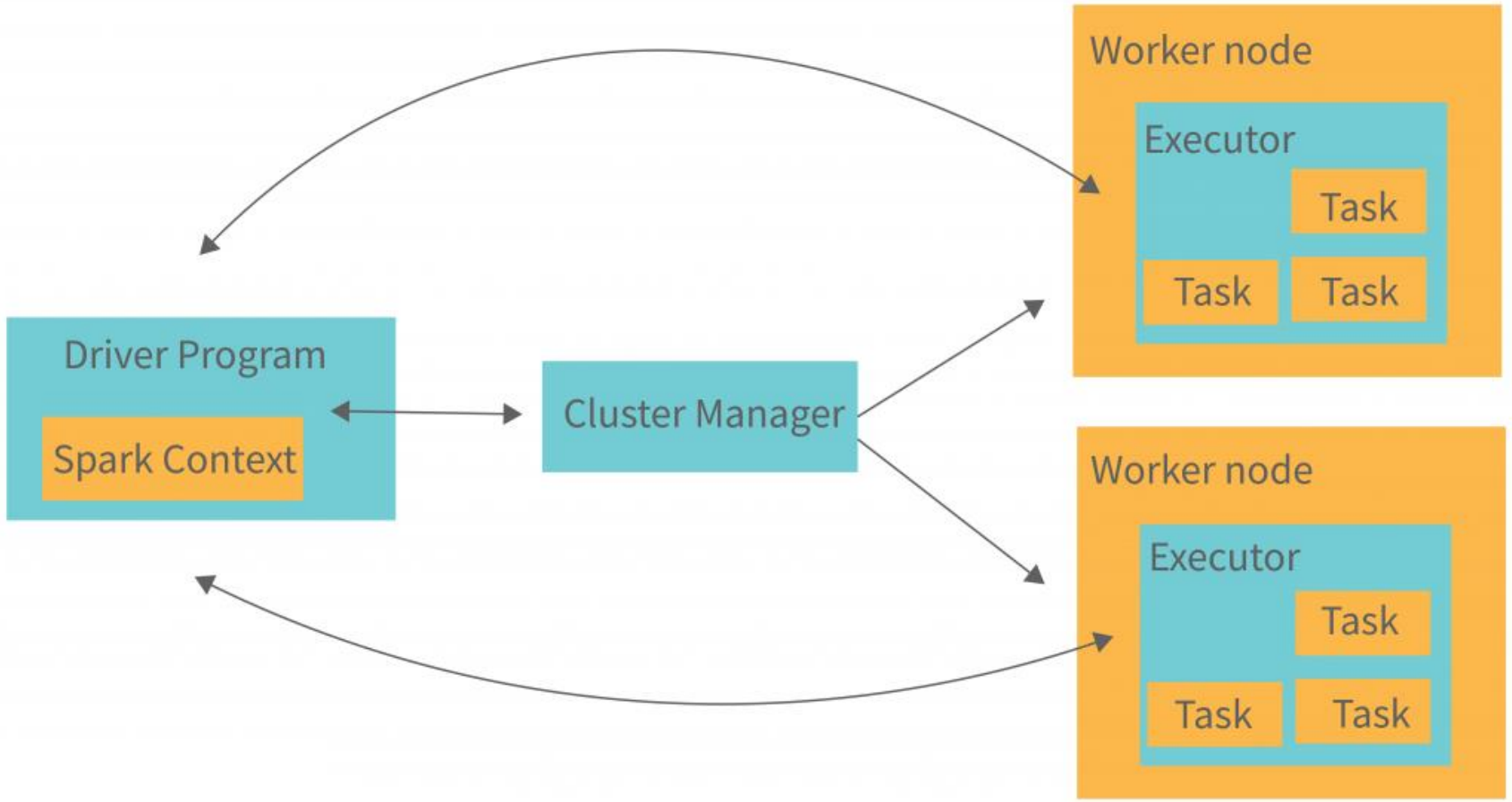
Unified, open source, parallel, data processing framework for Big Data Analytics

Spark Unifies:

- Batch Processing
- Real-time processing
- Stream Analytics
- Machine Learning
- Interactive SQL



Spark's Basic Architecture



Cluster Manager Types

- The system currently supports several cluster managers:
- **Standalone** – a simple cluster manager included with Spark that makes it easy to set up a cluster.
- **Apache Mesos** – a general cluster manager that can also run Hadoop MapReduce and service applications. (Deprecated)
- **Hadoop YARN –** the resource manager in Hadoop 2 and 3.
- **Kubernetes –** an open-source system for automating deployment, scaling, and management of containerized applications.

SparkContext/Session

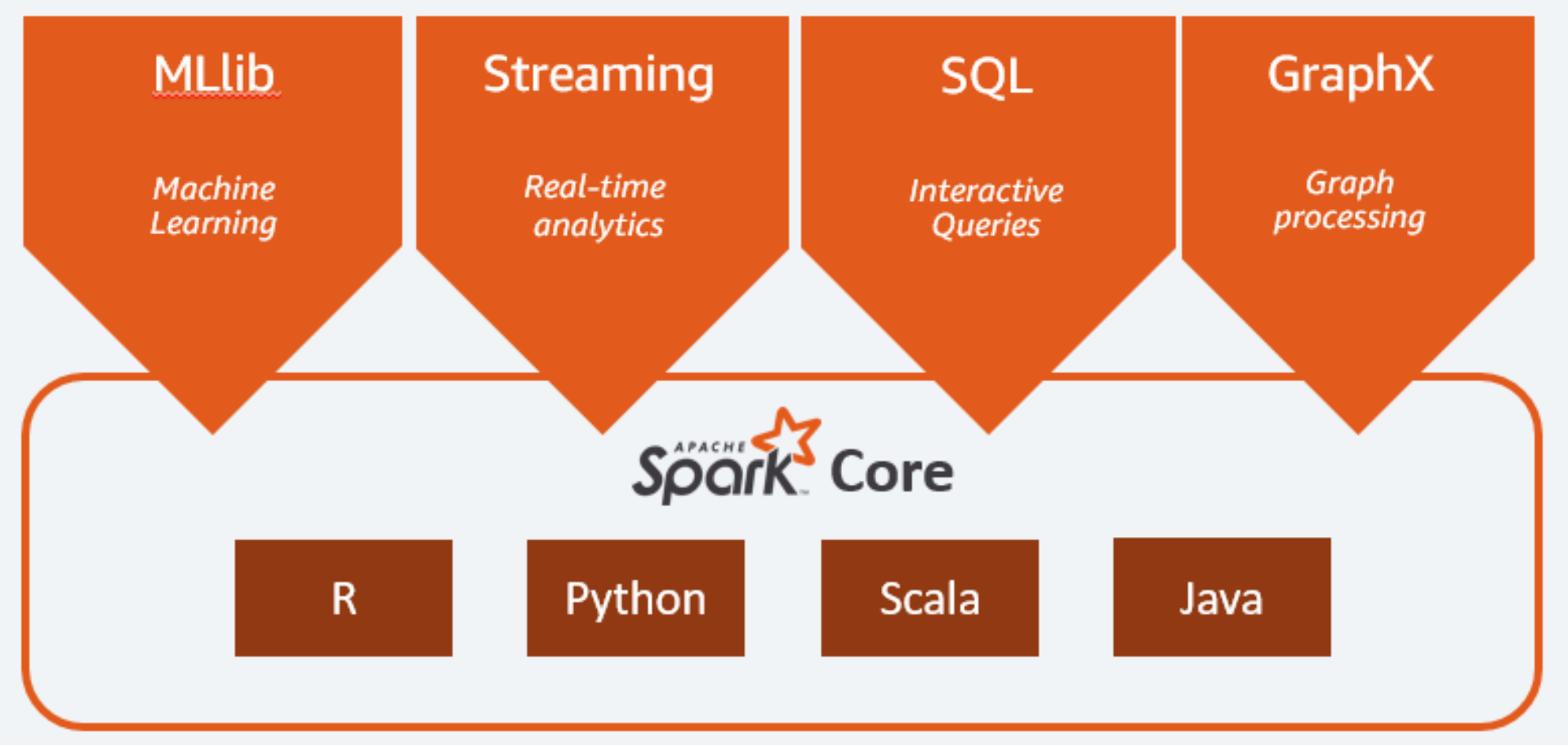
- **Spark Context** is an entry point to Spark and defined in org.apache.spark package since 1.x and used to programmatically create Spark RDD, accumulators and broadcast variables on the cluster. Since Spark 2.0 most of the functionalities (methods) available in Spark Context are also available in Spark Session. Its object sc is default available in spark-shell and it can be programmatically created using Spark Context class.

```
val conf = new SparkConf().setAppName("insightData").setMaster("local[1]")
val sparkContext = new SparkContext(conf)
```

- **Spark Session** introduced in version 2.0 and is an entry point to underlying Spark functionality in order to programmatically create Spark RDD, Data Frame and Data Set. It's object spark is default available in spark-shell and it can be created programmatically using Spark Session builder pattern.

```
val spark = SparkSession.builder() .master("local[1]")
.appName("HelloWorld").getOrCreate();
```

Spark Components Languages



Low Level API(RDD)

RDDs can be created only in two ways:

- Either parallelizing an already existing dataset, collection in your drivers and external storages which provides data sources like Hadoop Input Formats (HDFS,HBase,Cassandra..) or
- By transforming from already created RDDs.

Agenda for Tomorrow

- Introduction to Databricks
- Working Session on Databricks


Thank
You!