skillspeed

for the serious learner

Apache Spark and Scala

Module 1: Getting Started / Introduction to Scala

Course Topics

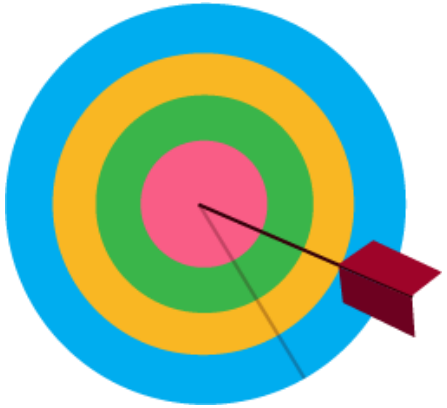
skillspeed
for the serious learner

Module 1	Module 2	Module 3	Module 4
Getting Started / Introduction to Scala	RDD and Spark Streaming	Scala Basics	SparkSQL – Real-time Analysis

Session Objectives

This session will help you to understand:

- Big Data
- IBM's Big Data Definition
- Some Big Data Examples
- Sparks Basics
- Why Spark ?
- Spark Components
- Scala Basics
- Why Scala ?
- Scala Job Trends
- Users of Scala
- Scala Frameworks
- Scala Usage
- Software Installation
- Scala Hands-on
- Scala community



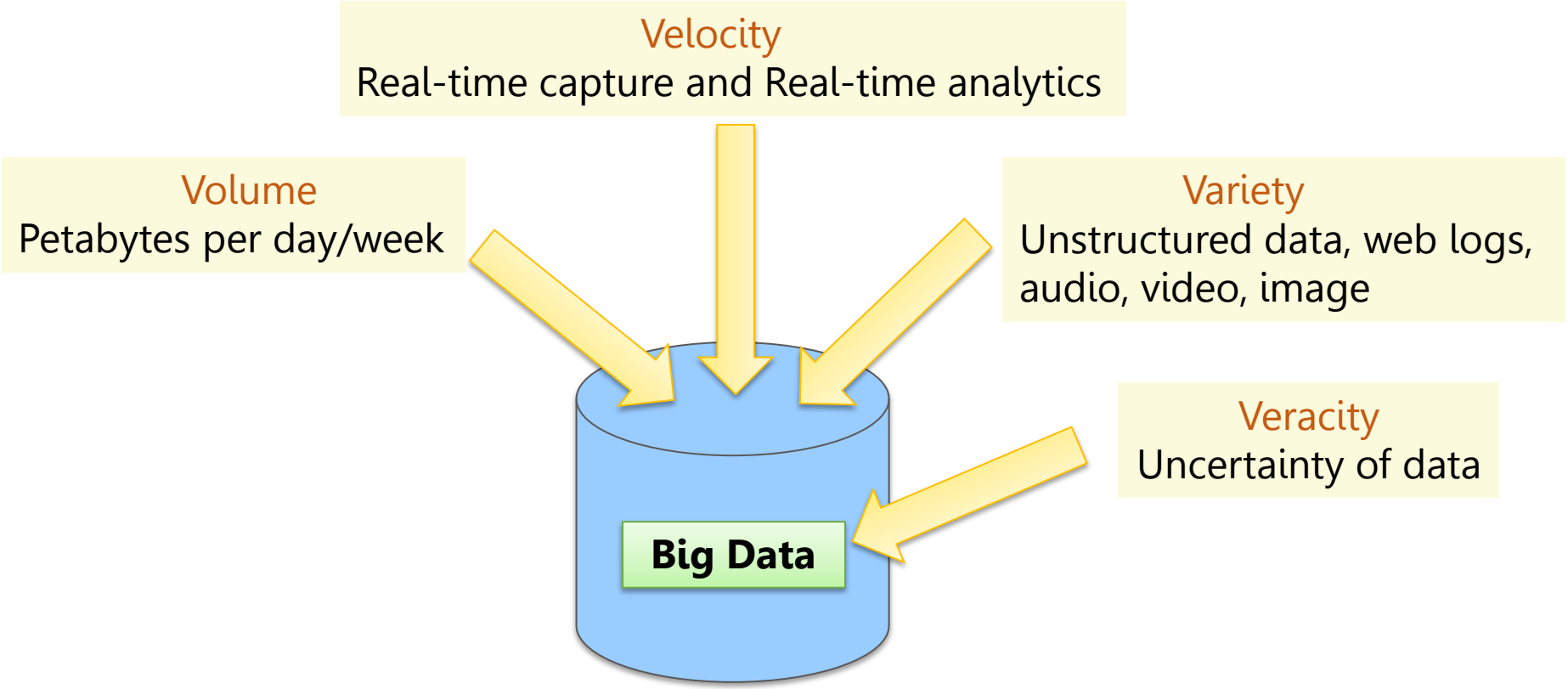
Introduction to Big Data

Big data is a broad term for data sets so large or complex that traditional data processing applications are inadequate

The challenges of big data includes: analysis, capture, data curation, search, sharing, storage, transfer, visualization, and information privacy

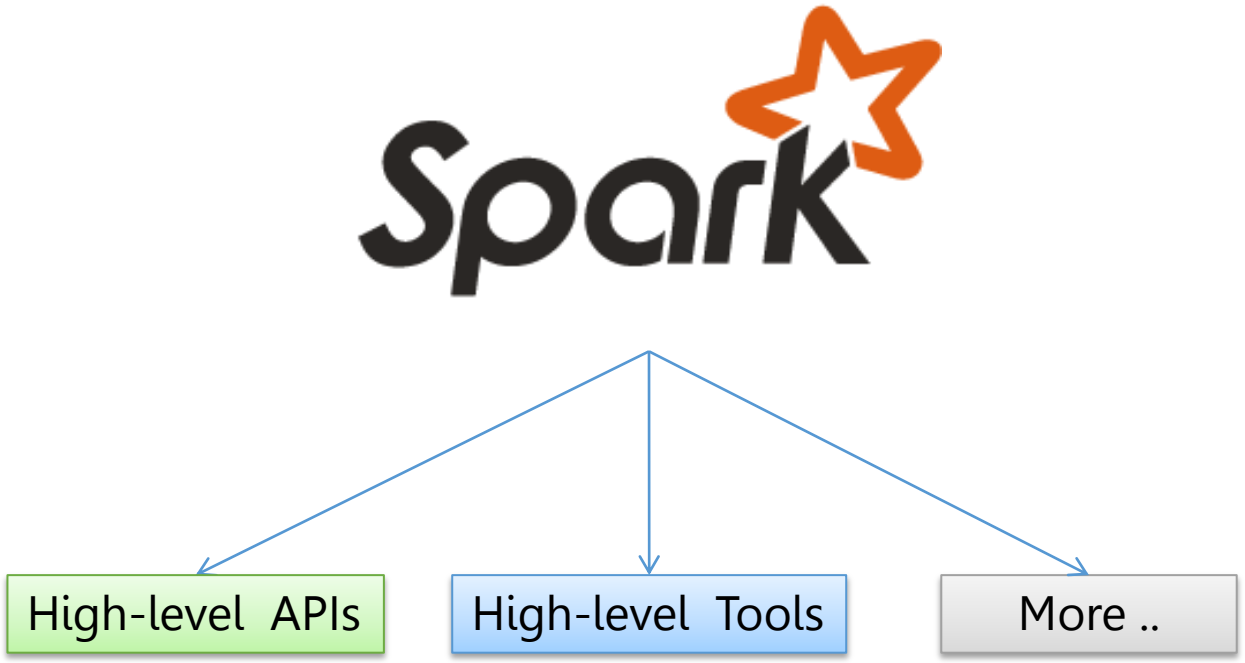
IBM's Big Data Definition

- IBM's Definition – Big Data Characteristics
- http://www.ibmbigdatahub.com/infographic/four-vs-big-data/



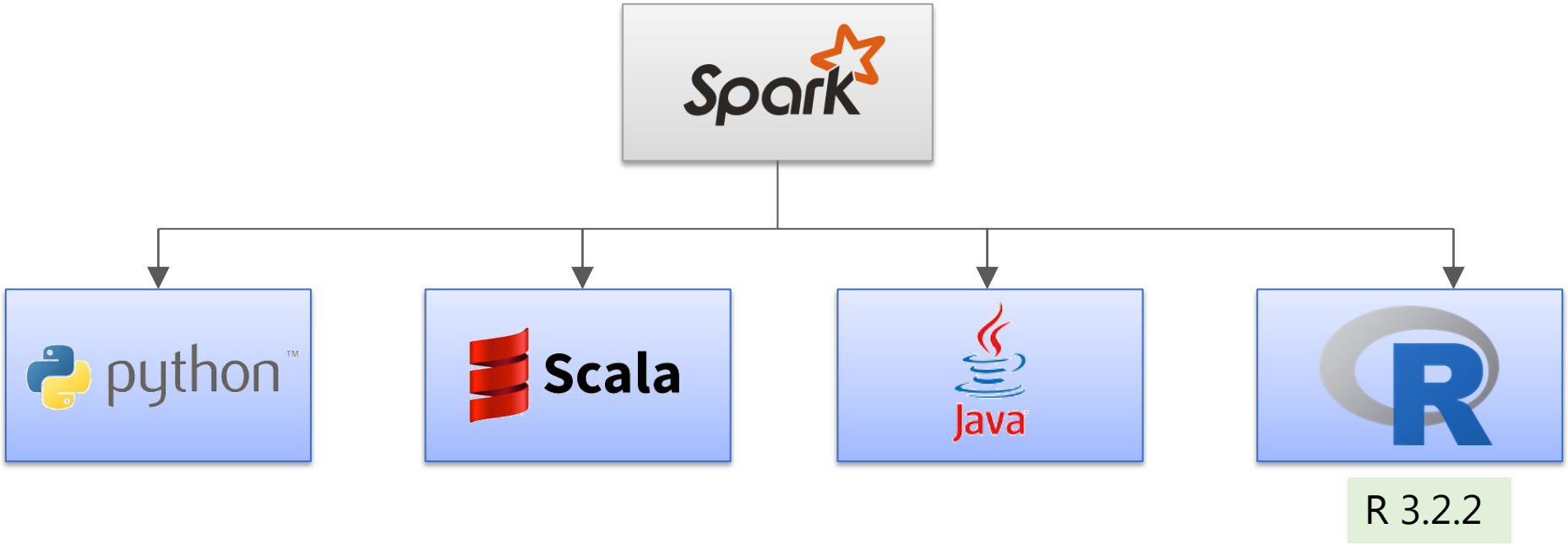
Spark Basics

- Apache Spark is a general-purpose cluster in-memory computing system which is used for data analytics
- It provides high-level APIs in Java, Scala and Python and an optimized engine that supports general execution graphs
- Apache Spark Provides various high level tools like Spark SQL for structured data processing, R programming Language for analyzing large datasets and MLlib for Machine Learning etc.

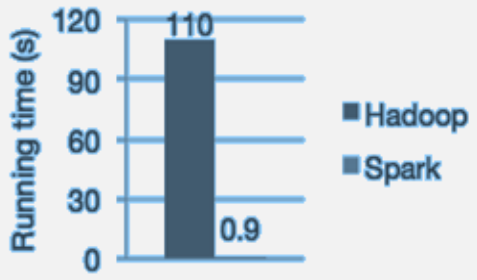


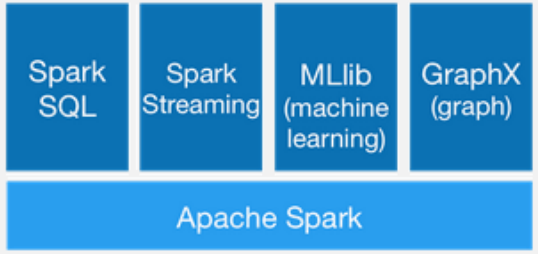
Spark Basics (Cont'd)

Spark framework is polyglot – It can be programmed in several programming languages (Java, Scala ,R 3.2.2 and Python supported)



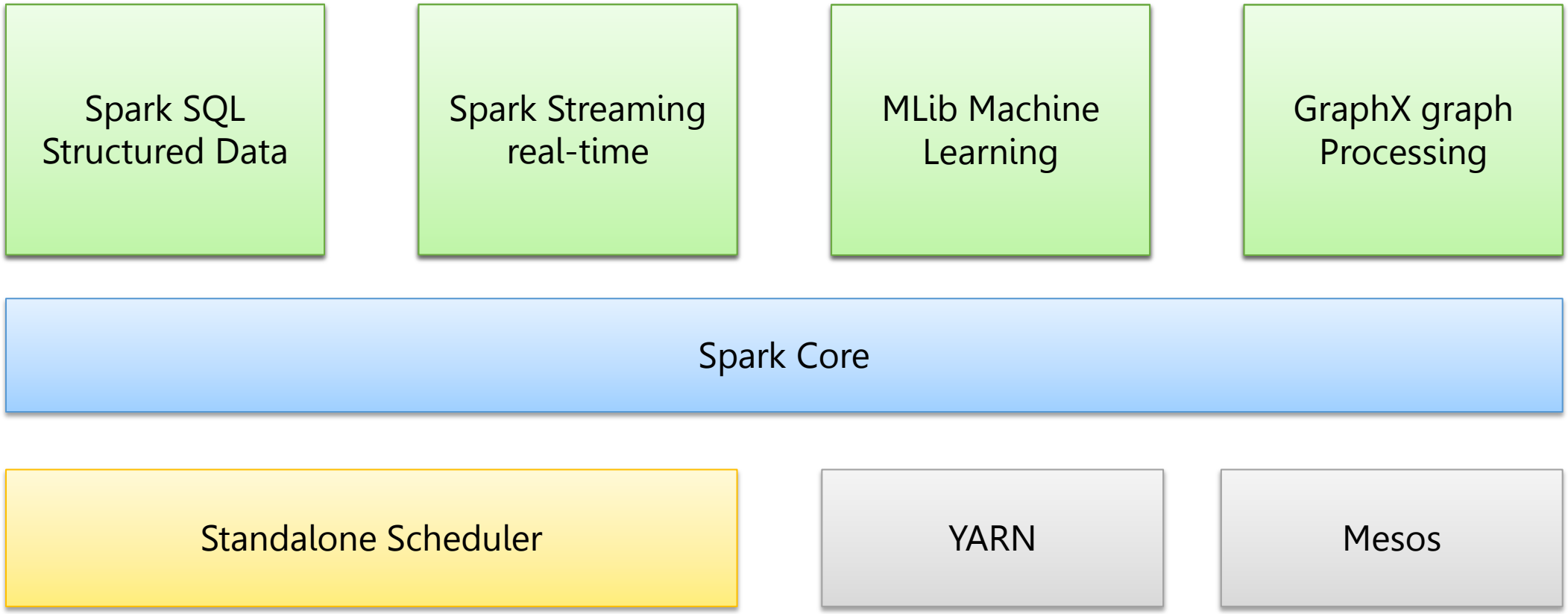
Why Spark?



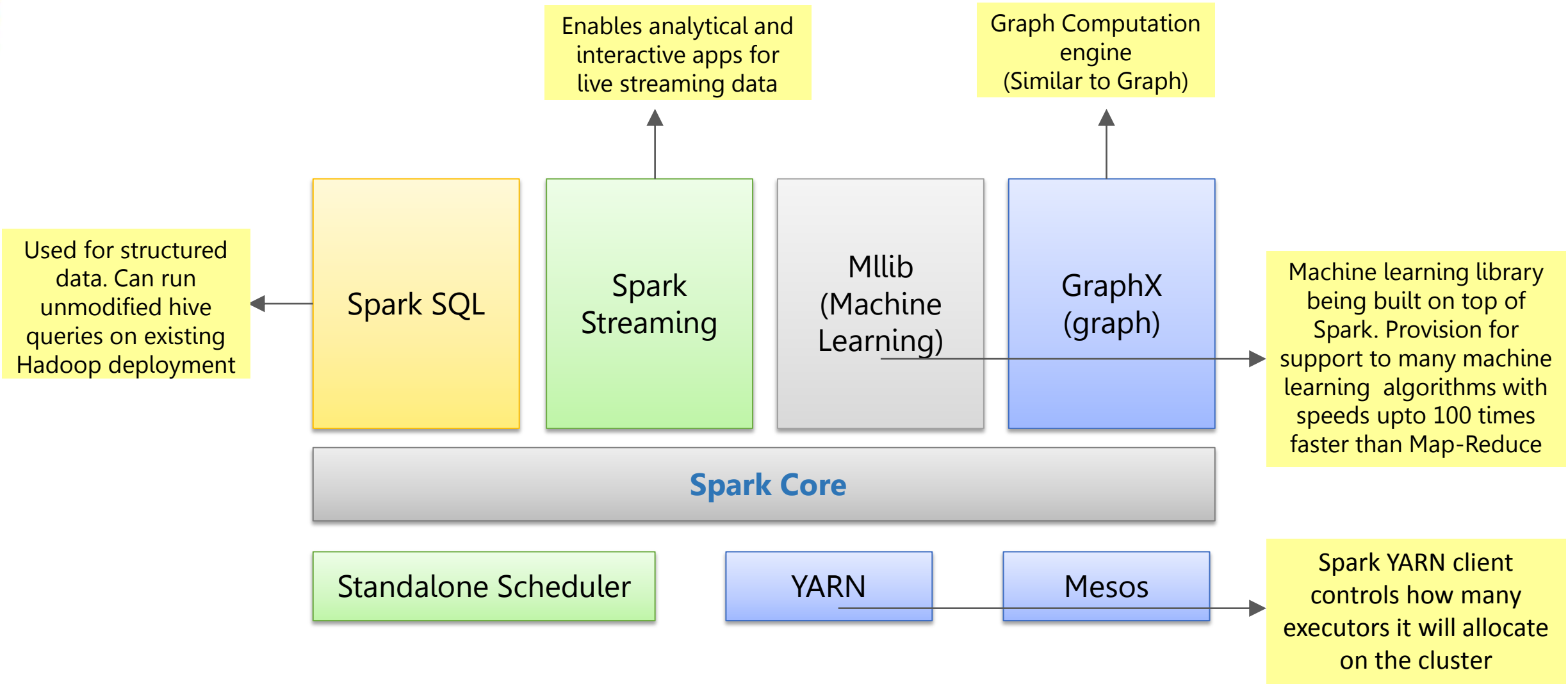


Speed Run programs up to 100x faster than Hadoop Map Reduce in memory	Generality Combine SQL ,streaming and complex analytics into one platform	Runs Everywhere Spark runs on Hadoop, Mesos.standalone or in cloud

Spark Components



Spark Components (Cont'd)



Introduction to Scala (Cont'd)

- Scala is a general purpose programming language, multiparadigm object oriented, functional, scalable
- Aimed to implement common programming patterns in a concise, elegant, and type-safe way
- Supports both object-oriented and functional programming styles, thus helping programmers to be more productive
- Publicly released in January 2004 on the JVM platform and a few months later on the .NET platform

Introduction to Scala (Cont'd)

Scala is Statically Typed

- Statically typed language binds the type to a variable for its entire scope

```
scala> val x = "skillspeed"
x: String = skillspeed

scala> x = 10
<console>:8: error: reassignment to val
       x = 10
         ^
```

- Dynamically typed languages bind the type to the actual value referenced by a variable .Example : python

```
>>> x="skillspeed"
>>> x=20
>>> x=34.5
```

- Fully supports Object Oriented Programming
- Everything is an object in Scala
- Unlike Java, Scala does not have primitives
- Supports "static" class members through Singleton Object Concept
- Improved support for OOP through Traits

Why Scala?

- Scala is pure object-oriented language. Conceptually, every value is an object and every operation is a method-call
- Scala is also a functional language and supports immutable data structures
- Many big data technologies use Scala like Spark, Kakfka, Storm, Akka, Scalding and web frameworks like Play

Why Scala? (Cont'd)

Scala code compared to Java code

Java	Scala

Java Code
 	
List<String> list = new ArrayList<String>();
list.add("1");
list.add("2");
list.add("3");
	Scala Code
	
	val list = List("1", "2", "3")

Questions

skillspeed
for the serious learner

thank you!