Emerging Paradigms for Big Data

Presentation
Big Data
Introduction
Big Data
Solutions

Fátima Leal

DCT DEPARTAMENTO CIÊNCIA E TECNOLOGIA

UPT UNIVERSIDADE PORTUCALENSE

Content

- Presentation
- Big Data Introduction
- Environment Installation
- Hands-on Activity

General Information

- Professors
  - Fátima Leal (fatimal@upt.pt)

- Professor's assistance*
  - Office 304 or via Zoom
  - The schedules are available in Moodle.

*We recommend sending an email to avoid mismatches. When possible, it could be presential or then, via Zoom.

Content

1. General principles
   a. Definition, characteristics, architectures and technologies for Big Data.
   b. Cloud computing and Big Data-related technologies
   c. Programming Paradigms for Big Data in the Age of Internet of Things
   d. Open Challenges
2. Cloud infrastructure
3. Virtualization
4. Cloud storage
   a. Apache Hive
   b. Apache Cassandra
   c. Apache Pig
5. Programming models
   a. Apache Hadoop - MapReduce

Evaluation

- **Regular Evaluation**
  - **Work 1**: 30 %
  - **Work 2**: 30 %
  - **Written test**: 40 %

  - Practical work with 2 components (60 %):
    - ***Component 1***: Kafka + Cassandra (30 %);
    - ***Component 2***: Spark (30 %).
  - Each work component has an oral presentation and demonstration;
  - Component 2 should include a paper or report
  - The practical work has a minimum of 9.5 in 20.

- **Resit examination**: Practical work (60 %) + Exam (40 %)

Evaluation

- **The detailed requirements of the practical work will be placed in Moodle**;
- Please, **read carefully** the **rules** and the **correction criteria** of the practical work;
- There is **no tolerance** concerning failing the **deadline;**
- **Submission** will be done through **Moodle**. Only **works delivered** on this platform **will be corrected**;
- The **oral presentation** is **mandatory;**
- **Intellectual honesty** - Academic fraud implies the work cancellation.

Project

- Develop a system for **anomaly detection** or **predictive maintenance** using Kafka, Spark, and Cassandra.
- The project involve collecting sensor data from various sources using Kafka, processing and analyzing the data using Spark Streaming, and storing the results in Cassandra.
- The anomaly detection could be performed using statistical methods or machine learning algorithms.
- You can choose one of the following datasets:
  - NASA Turbofan Jet Engine Data Set
  - NASA Bearing Dataset

Project

- **Component 1**
  - Kafka + Cassandra
  - Oral presentation and demonstration
  - Paper/report (Abstract + Introduction + Related-work)

- **Component 2**
  - Spark Processing and Machine Learning
  - Paper/report (Proposed Method + Results + Conclusions)

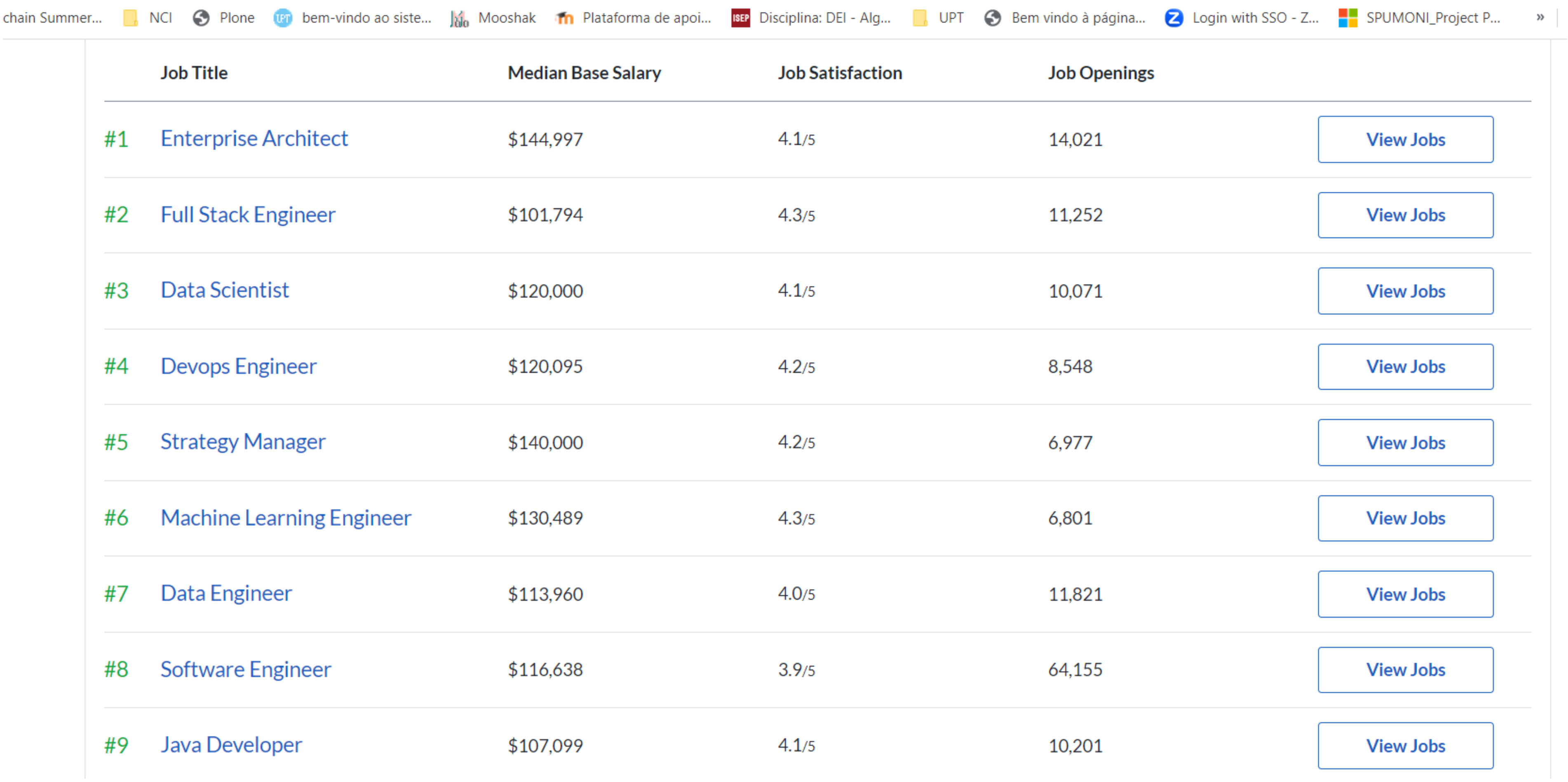
- Groups of 2 students

Planification

Class	Date	Content	Assessments
1	23 March	Presentation. General Principles of Big Data	
2	23 March	Development environment (instalation)	
3	30 March	General Principles of Big Data Apache Kafta and Cassandra Installation. Kafka introduction	
4	19 April	Cloud Infrastructure. Apache Cassandra Introduction. Conection with Kafta	
5	19 April	Apache Kafta + Cassandra	
6	4 May	Virtualization. Apache Spark configuration	
7	4 May	Apache Spark	Component 1
8	10 May	Cloud Storage. Apache Spark + Kafta + Cassandra	
9	10 May	Apache Spark + Kafta + Cassandra	
10	31 May	Programming Models.	
11	31 May	Apache Spark + Kafta + Cassandra	
12	1 june	Project	
13	1 june	Project	
14	17 june	Project	Component 2
15	17 june	Presentations	Written test

Welcome to Big Data

Data Analysis Lab



	Job Title	Median Base Salary	Job Satisfaction	Job Openings	
#1	Enterprise Architect	$144,997	4.1/5	14,021	View Jobs
#2	Full Stack Engineer	$101,794	4.3/5	11,252	View Jobs
#3	Data Scientist	$120,000	4.1/5	10,071	View Jobs
#4	Devops Engineer	$120,095	4.2/5	8,548	View Jobs
#5	Strategy Manager	$140,000	4.2/5	6,977	View Jobs
#6	Machine Learning Engineer	$130,489	4.3/5	6,801	View Jobs
#7	Data Engineer	$113,960	4.0/5	11,821	View Jobs
#8	Software Engineer	$116,638	3.9/5	64,155	View Jobs
#9	Java Developer	$107,099	4.1/5	10,201	View Jobs

https://www.glassdoor.com/List/Best-Jobs-in-America-LST_KQ0,20.htm

Big Data Introduction

What is
Big Data?

What is Big Data?

- Big Data is characterized by a set of Vs…

Big Data is “Vig”!

What is Big Data?

What is Big Data?

What is Big Data?

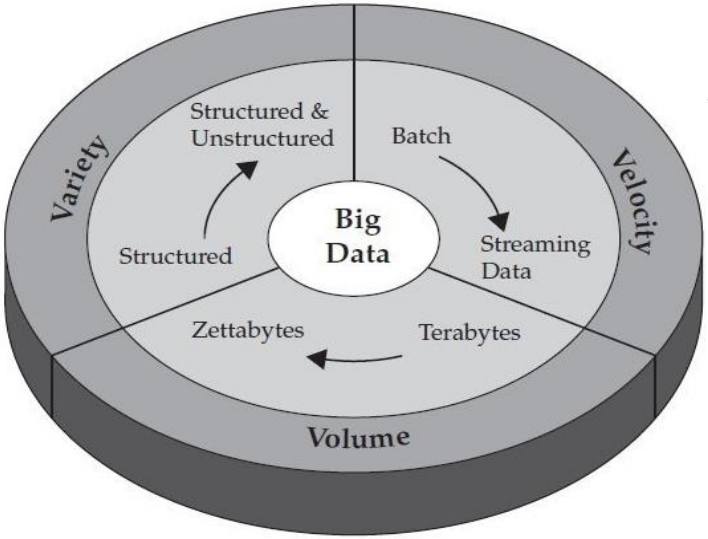
What is Big Data?

- Big data is an abstract concept: it does not involve just large amount of data

- As we could see, in “3Vs” model:
  - **Volume**: generation and collection of massive data
  - **Velocity**: data collection and analysis, *etc.*, must be rapidly and timely conducted
  - **Variety**: indicates the various types of data (semi-structured and unstructured data such as audio, video, webpage, and text, as well as traditional structured data)

Big Data 3V's

- First Big Data definition

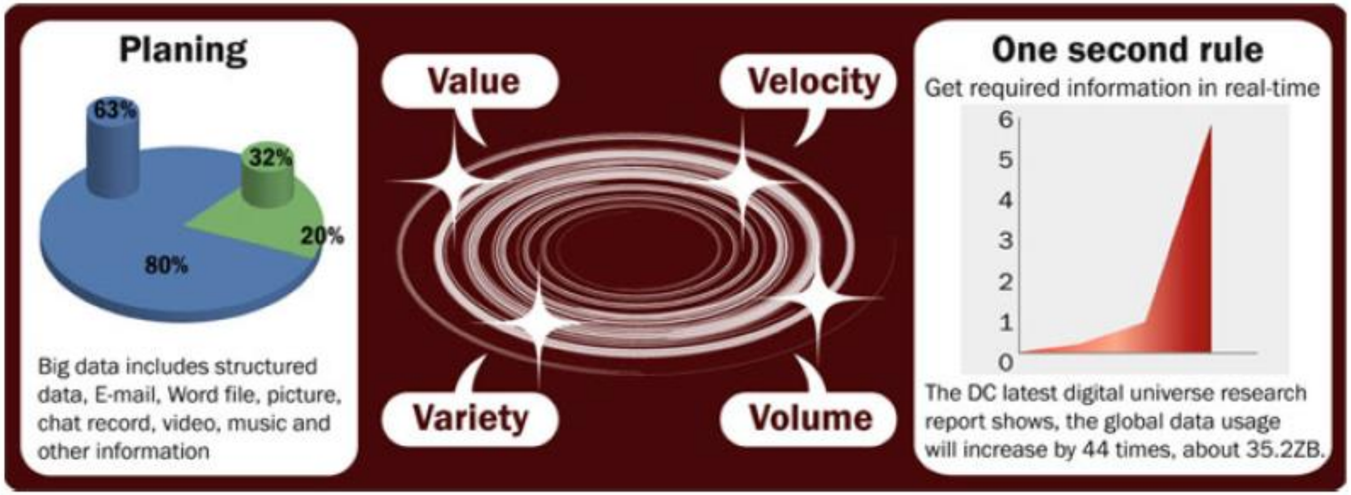


What is Big Data?

- Value from Big Data? Is it possible?

- Big data technologies describe a new generation of technologies and architectures, designed to economically extract value from very large **volumes** of a wide **variety** of data, by enabling the high-**velocity** capture, discovery, and/or analysis

Discover value from datasets



Big data includes structured data, E-mail, Word file, picture, chat record, video, music and other information

The DC latest digital universe research report shows, the global data usage will increase by 44 times, about 35.2ZB.

What is Big Data?

- **To generate value, the high volume**, **velocity,** and **variety** of data must be processed with **advanced tools** (analytics and algorithms) to reveal **meaningful information**.

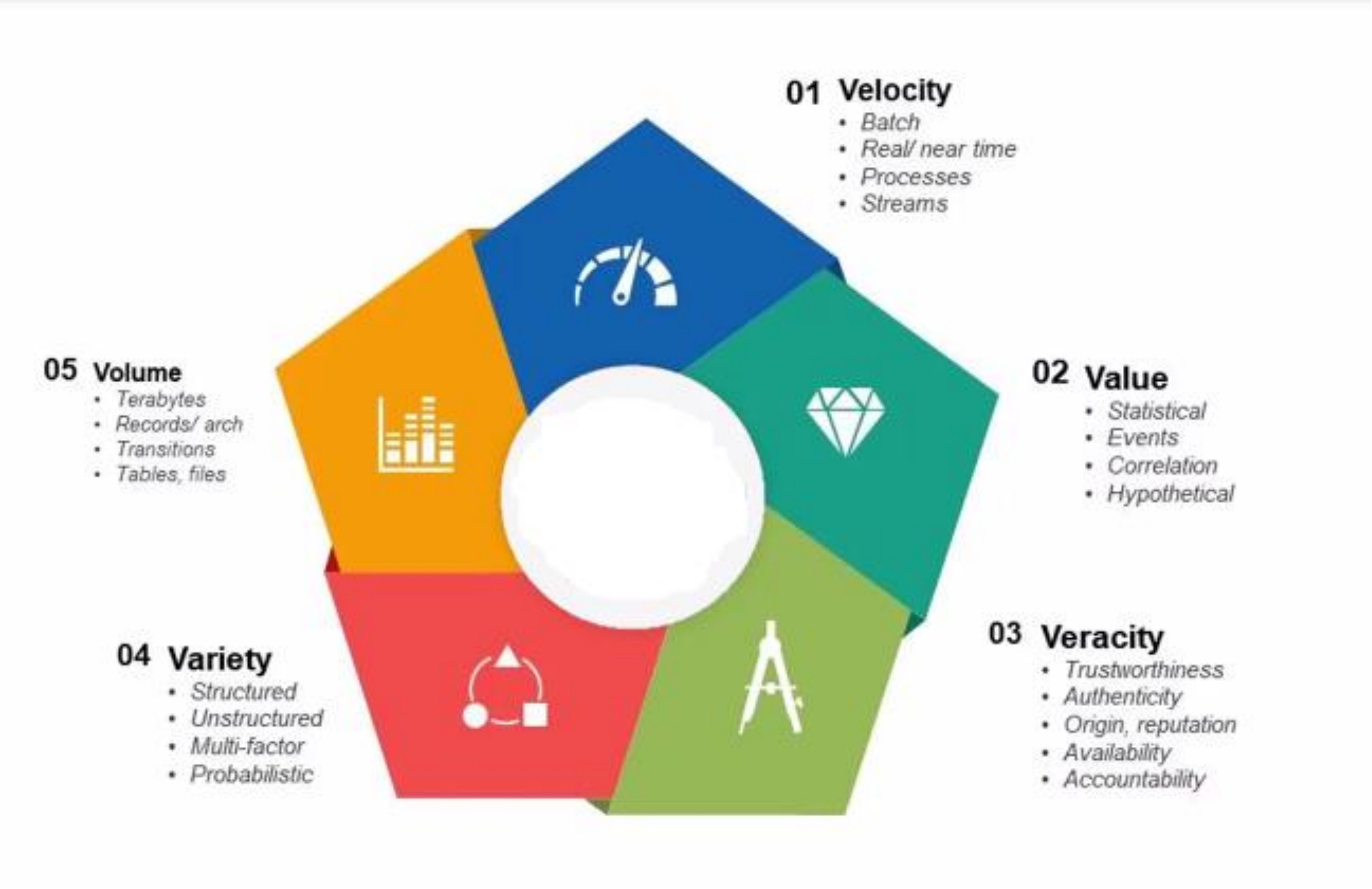
- **Veracity** refers to the quality of the data that is being analysed.

- High veracity data has many records that are valuable to analyse and that contribute in a meaningful way to the overall results.

- Low veracity data, on the other hand, contains a high percentage of meaningless data.

- The non-valuable data is noise.

What is Big Data?



01 Velocity
- *Batch*
- *Real/ near time*
- *Processes*
- *Streams*

02 Value
- *Statistical*
- *Events*
- *Correlation*
- *Hypothetical*

03 Veracity
- *Trustworthiness*
- *Authenticity*
- *Origin, reputation*
- *Availability*
- *Accountability*

04 Variety
- *Structured*
- *Unstructured*
- *Multi-factor*
- *Probabilistic*

05 Volume
- *Terabytes*
- *Records/ arch*
- *Transitions*
- *Tables, files*

Big Data 5V’s

- **Volume**: amount of data
- **Velocity**: data generation and data processing
- **Variety**: multiple nature

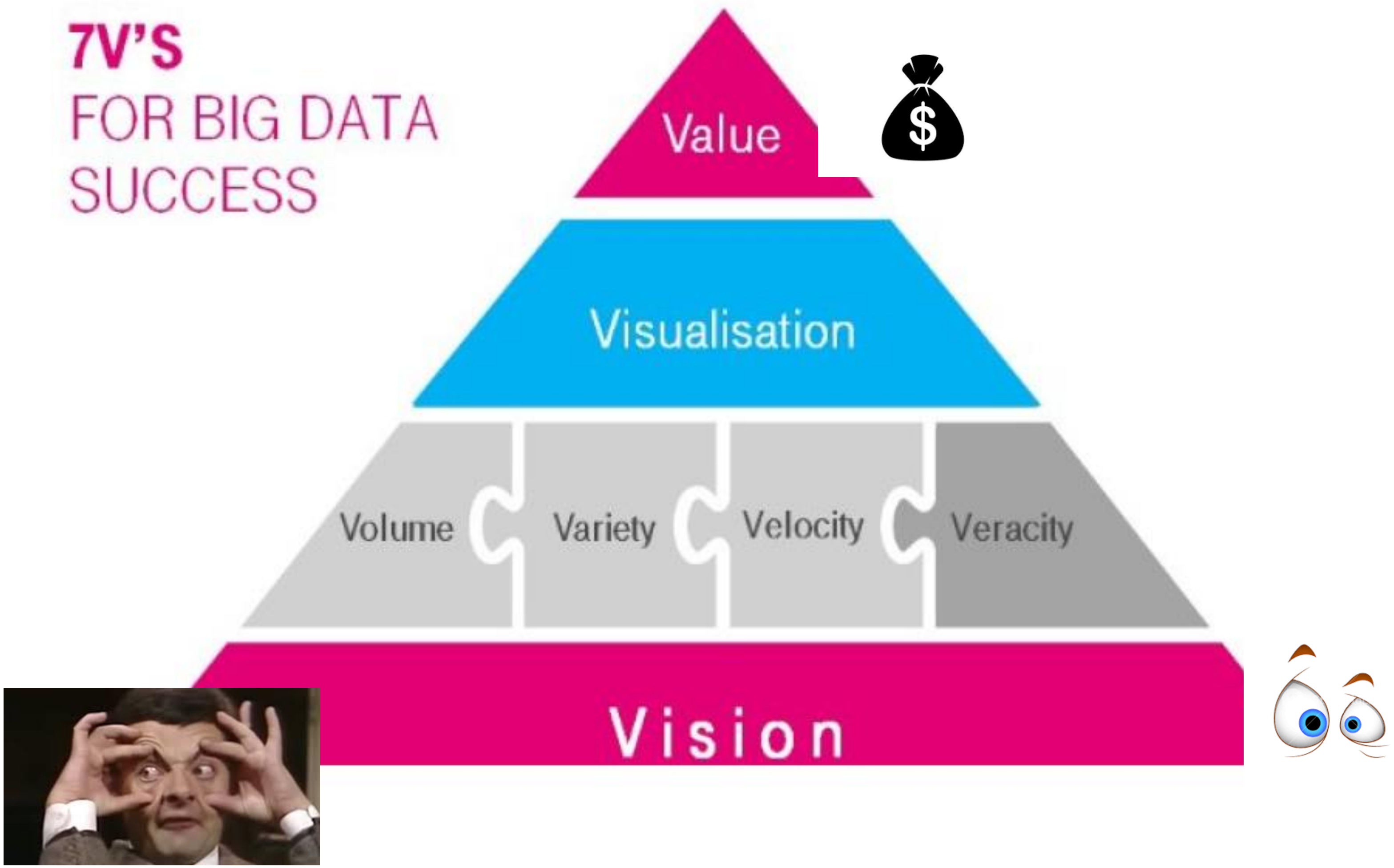
Academic Vision

- **Veracity**: data source reliability
- **Value**: its potential to generate value.

Industrial Vision

In summary: Big Data is “Vig”!

Big Data 7V's



7V'S
FOR BIG DATA
SUCCESS

Value

Visualisation

Volume

Variety

Velocity

Veracity

Vision

Big Data Challenges

- **Storage**:
  - Traditional data management and analytics systems are based on the relational database management system (RDBMS)
  - RDBMSs only apply to **structured data**
  - RDBMSs are increasingly utilizing more and more **expensive hardware**
  - RDBMSs **cannot handle** the huge volume and heterogeneity of big data

Big Data Challenges

- **Data representation**: Data representation aims to make data more meaningful for computer analysis and user interpretation. Improper data representation will reduce the value of the original data and may even obstruct effective data analysis.

- **Data Life Cycle Management**: Pervasive sensors and computing are generating data at unprecedented rates and scales which the current storage system does not support. We must decide which data shall be stored and which data shall be discarded.

- **Analytical Mechanism:** the analytical system of big data shall process masses of heterogeneous data within a limited time. Traditional RDBMSs are strictly designed with a lack of **scalability** and **expandability**.

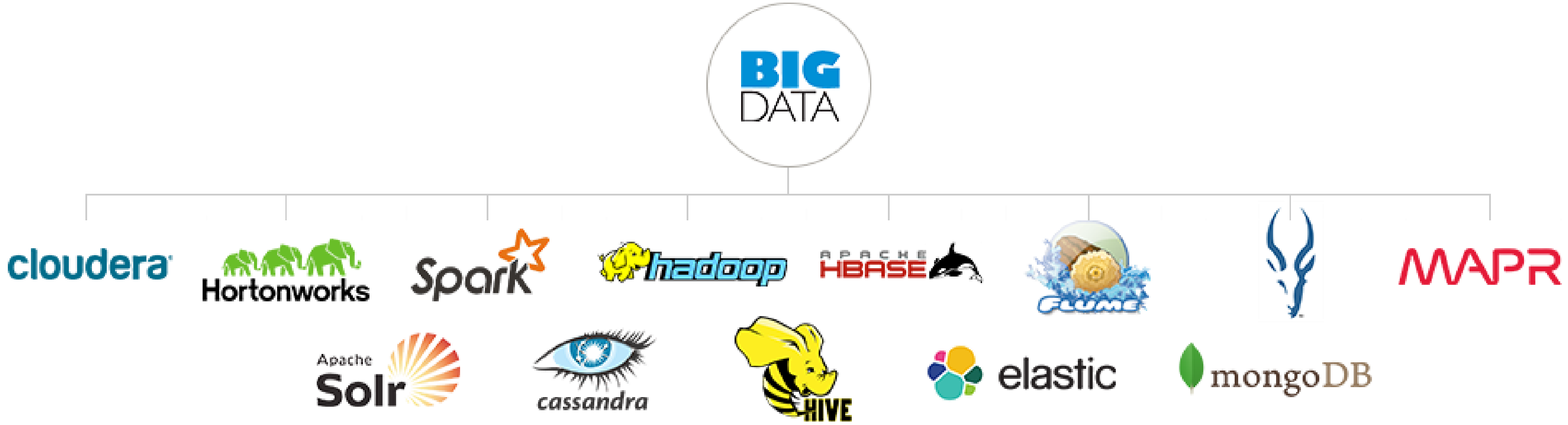
Big Data Challenges

- **Data Confidentiality:** analysis of big data challenges privacy
- **Energy Management:** the energy consumption of mainframe computing systems has drawn much attention from both economy and environment perspectives. Processing, storage, and transmission of big data will inevitably consume more and more electric energy
- **Expendability and Scalability**: analytical algorithm must be able to process increasingly expanding and more complex datasets
- **Cooperation**: analysis of big data is an interdisciplinary research

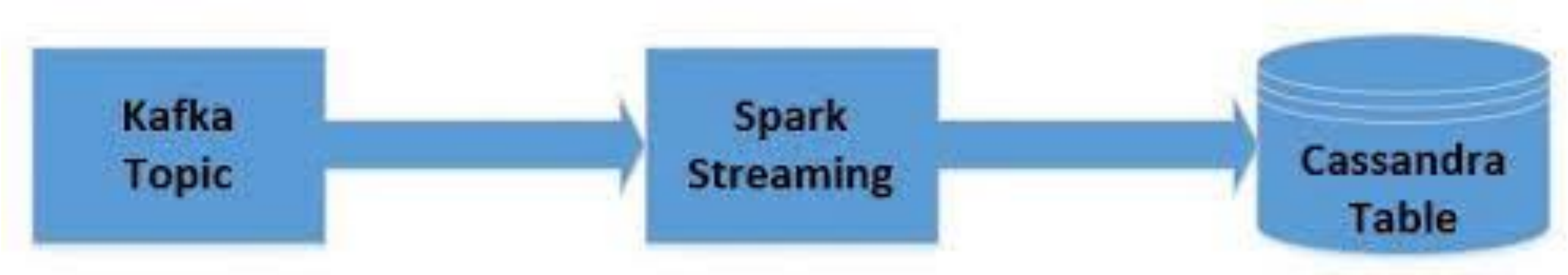
Big Data Challenges - Summary

- **Scalability**: property of a system to handle a growing amount of work by adding resources to the system
- **Load balancing:** process of distributing a set of tasks over a set of resources
- **Fault tolerance:** property of a system to continue operating properly in the case of failure of some of its components
- **Efficiency:** system performance
- **Data stream processing:** real-time systems

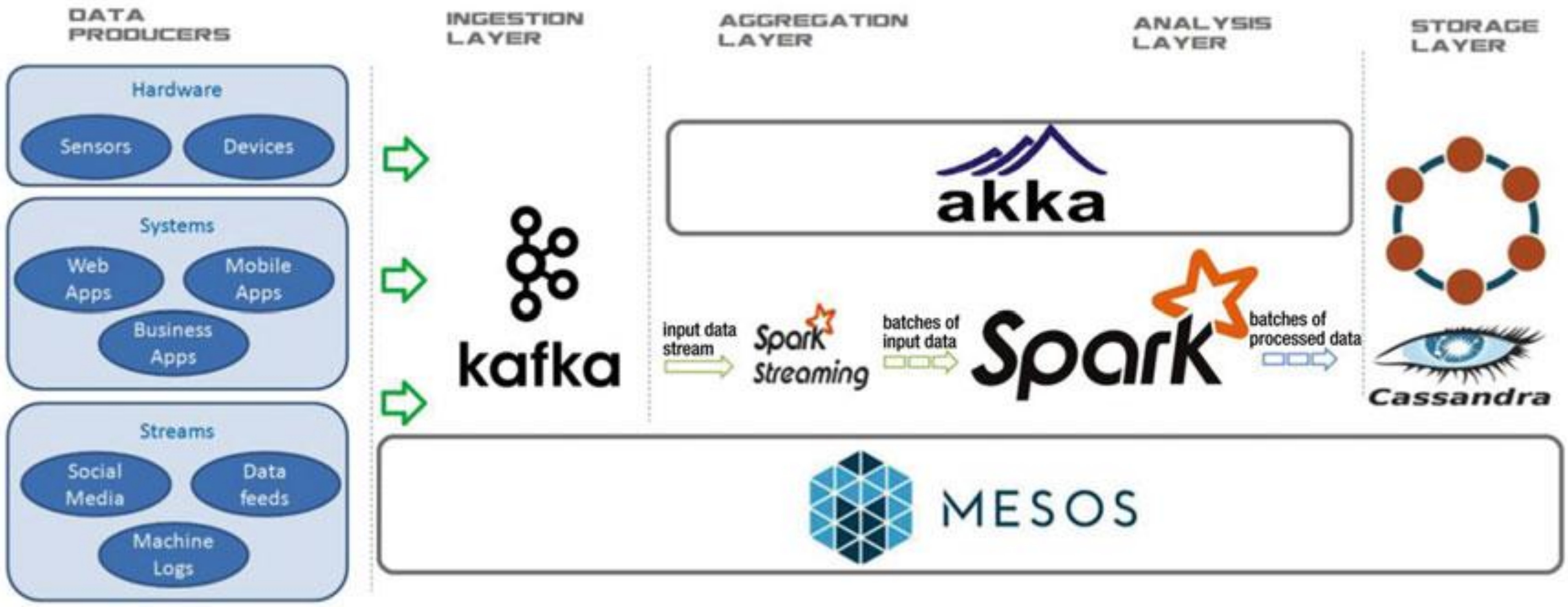
Big Data Solutions



Big Data Solutions



Big Data Solutions



Big Data Solutions

- What is Kafka?
- What is Cassandra?
- What is Spark?

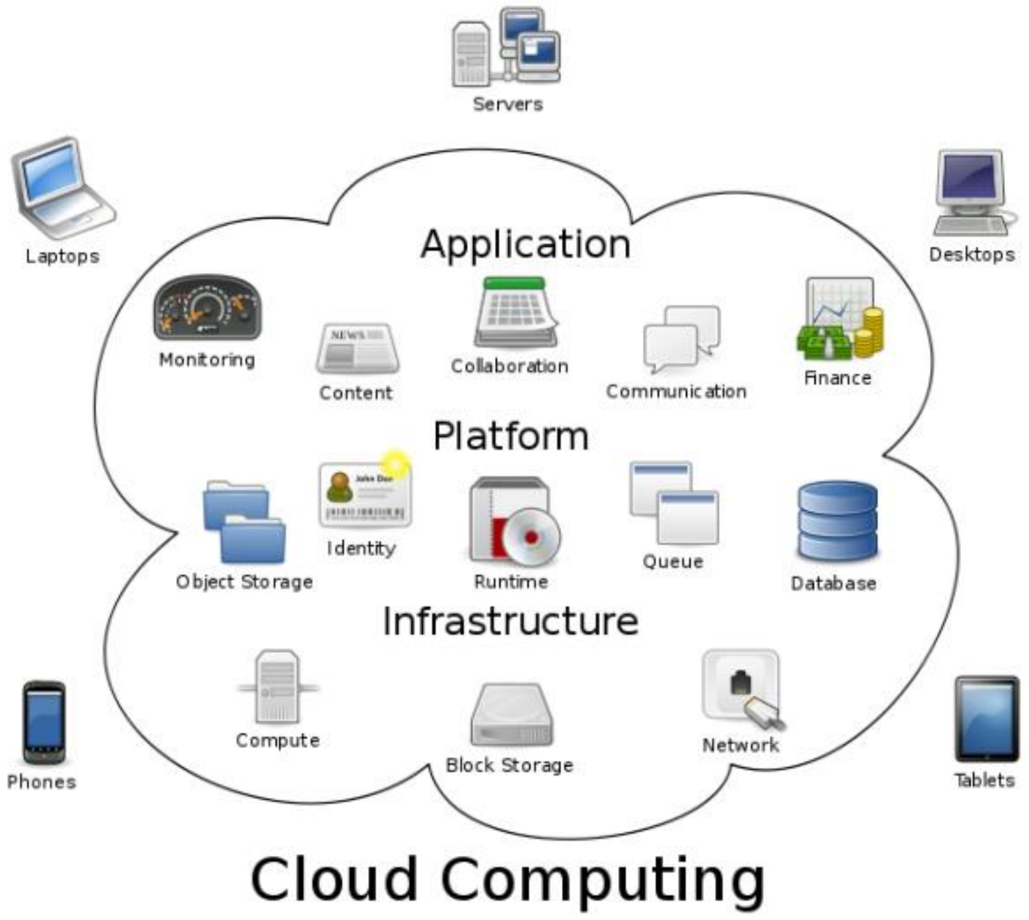
Big Data Solutions

- **Apache Kafka** is a real-time publish-subscribe solution messaging system: open source, distributed, partitioned, replicated, commit-log based with a publish-subscribe schema.

- **Apache Cassandra** is a NoSQL database management system designed to handle large amounts of data across many servers, providing high availability with no failures.

- **Apache Spark** is an infrastructure engine can be attached to powerful tools like Apache Kafka and Apache Cassandra to produce data science pipelines. Simultaneously, it is a data science laboratory because it represents an engine for machine learning in both a laptop and a productive cluster, from a few data kilobytes up to what the hardware capacity allows. Likewise, you can build models based on sample data and then apply them in larger datasets.

Big Data Solutions

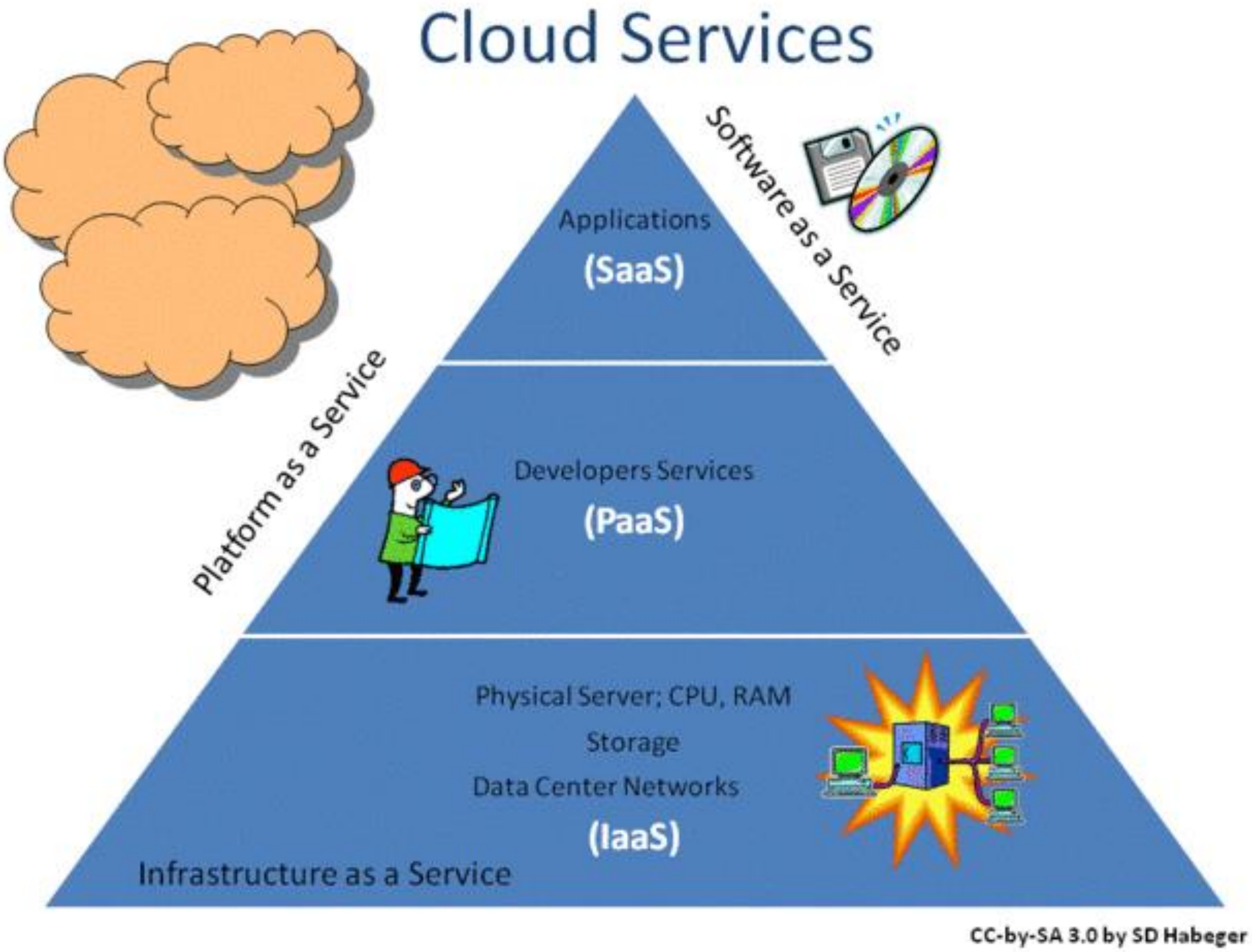


Big Data Solutions: Cloud Computing

- **Infrastructure as a service** (IaaS)
  - Offering hardware related services using the principles of cloud computing. These could include storage services (database or disk storage) or virtual servers.
  - Amazon EC2, openstack.

- **Platform as a Service** (PaaS)
  - Offering a development platform on the cloud.
  - Adds a middleware like data bases into the cloud environment

- **Software as a service** (SaaS)
  - Including a complete software offering on the cloud. Users can access a software application hosted by the cloud vendor on pay-per-use basis. This is a well-established sector.
  - Dropbox, slack, Adobe creative cloud



Cloud Services

Applications

(SaaS)

Software as a Service

Developers Services

(PaaS)

Platform as a Service

Physical Server; CPU, RAM

Storage

Data Center Networks

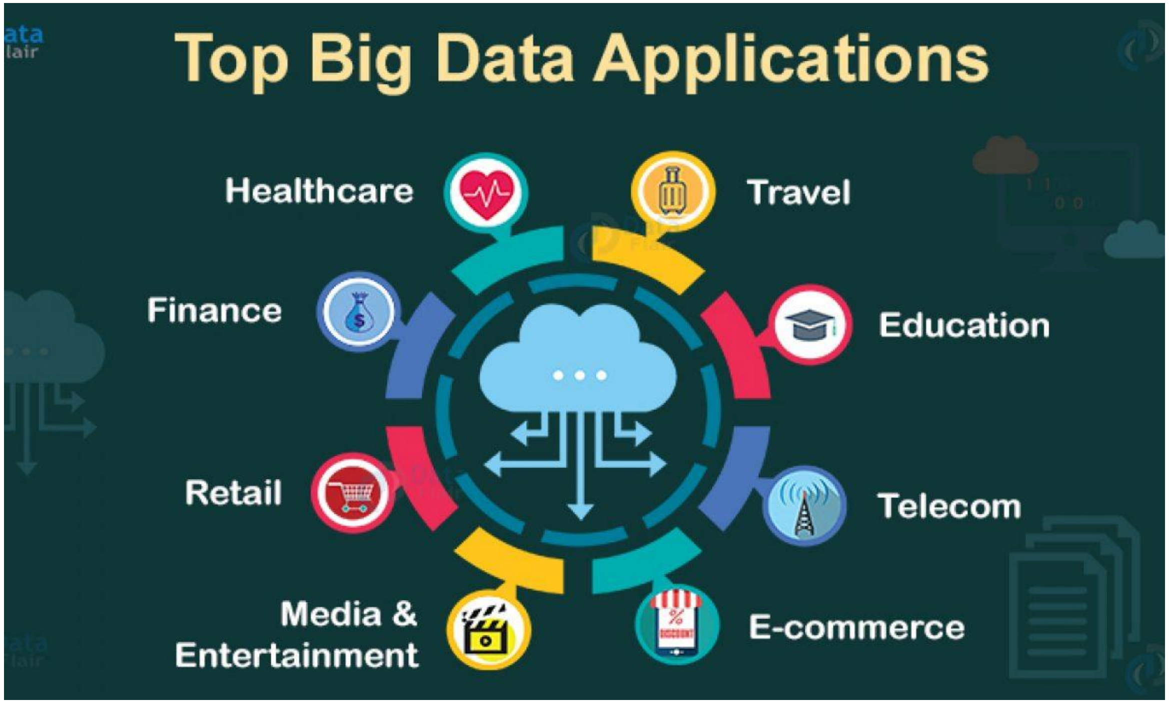
(IaaS)

Infrastructure as a Service

CC-by-SA 3.0 by SD Habeger

Big Data Applications

- Bank and security
- Communications
- Healthcare
- Education
- Industry
- Energy



Big Data Applications

[NASA Turbofan Jet Engine Data Set]

[NASA Bearing Dataset]

Hands-on (30 min)

- Based on the exploratory analysis presented by Kaggle in https://www.kaggle.com/code/brjapon/nasa-turbofan-exploratoryanalysis

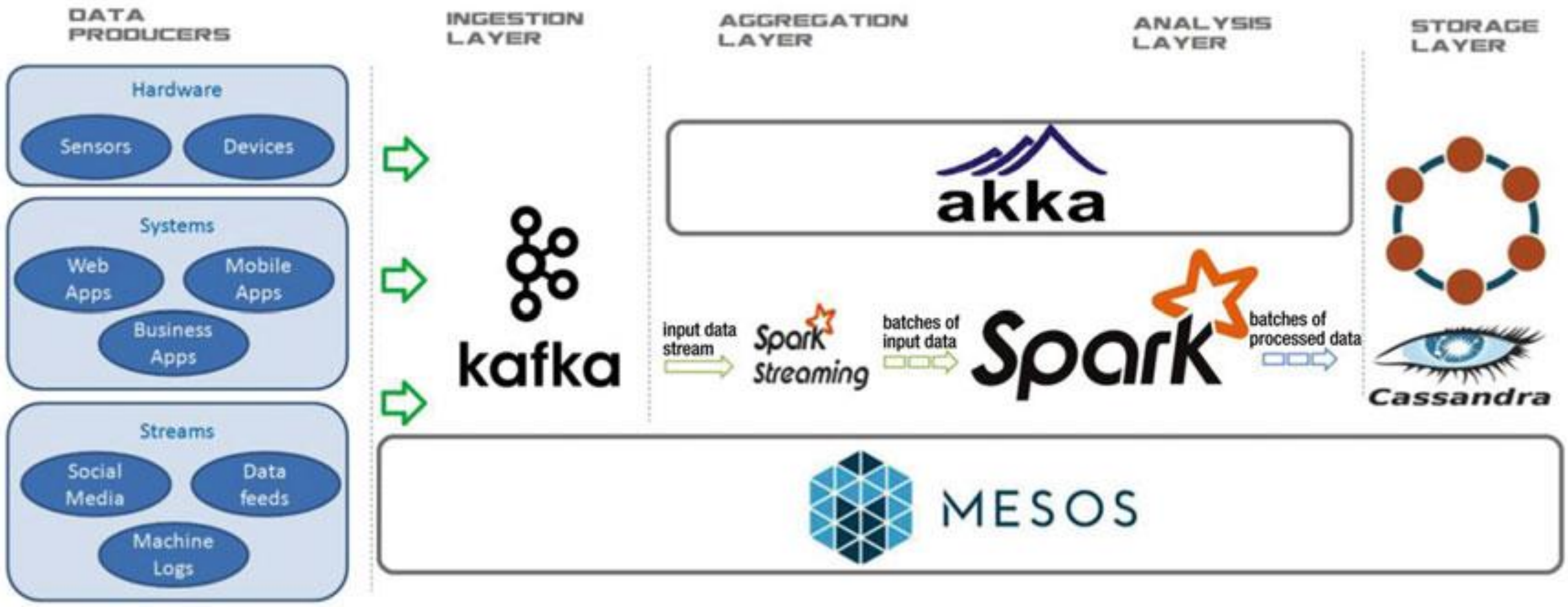
Do an exploratory analysis of both datasets first using just pandas:

- Files
- Variables
- Correlation
- Graphs
- Correlations
- Which ML model would you apply?

NASA Turbofan Jet Engine Data Set

- Then let to apply spark

Big Data Solutions



Spark Installation

- Mac https://sparkbyexamples.com/pyspark/how-to-install-pyspark-on-mac/
- Windows

Spark Installation - Windows

- **Installation of the Java Development Kit 1.8**

https://www.oracle.com/java/technologies/javase/javase-jdk8-downloads.html

- Environment variables Definition
  - set JAVA_HOME = path of JDK

- Installation of Anaconda

https://www.anaconda.com/products/individual

Spark Installation - Windows

- **Create Virtual Environment Anaconda**
  - conda create -n pyspark_env
  - conda activate pyspark_env
  - conda install pip
  - pip install pyspark
  - pip install numpy
  - pip install pandas

- Download Apache Spark with Apach Hadoop 3.2.2

https://archive.apache.org/dist/spark/spark-3.2.2/spark-3.2.2-bin-hadoop3.2.tgz

Spark Installation - Windows

- Installation Folder – Copy the Apache Spark unzip folder to ”C:/Spark/”
- Download Winutils – https://github.com/cdarlint/winutils
- Installation Folder – Copy the Winutils.exe to ”C:/Hadoop/bin/”
- Environment variables Definition
  - HADOOP_HOME = path of Hadoop
  - SPARK_HOME = path of Spark
  - PYSPARK_DRIVER_PYTHON = path pyspark_env/python.exe
  - PYSPARK_PYTHON = path pyspark_env/python.exe

Spark Installation - Windows

- Environment variables Definition
  - add to PATH
  - %SPARK_HOME%/bin
  - %HADOOP_HOME%/bin
  - %JAVA_HOME%/bin
  - path to pyspark_env folder

- Launch Spark
  - open a conda command line terminal
  - %SPARK_HOME%/bin/pyspark

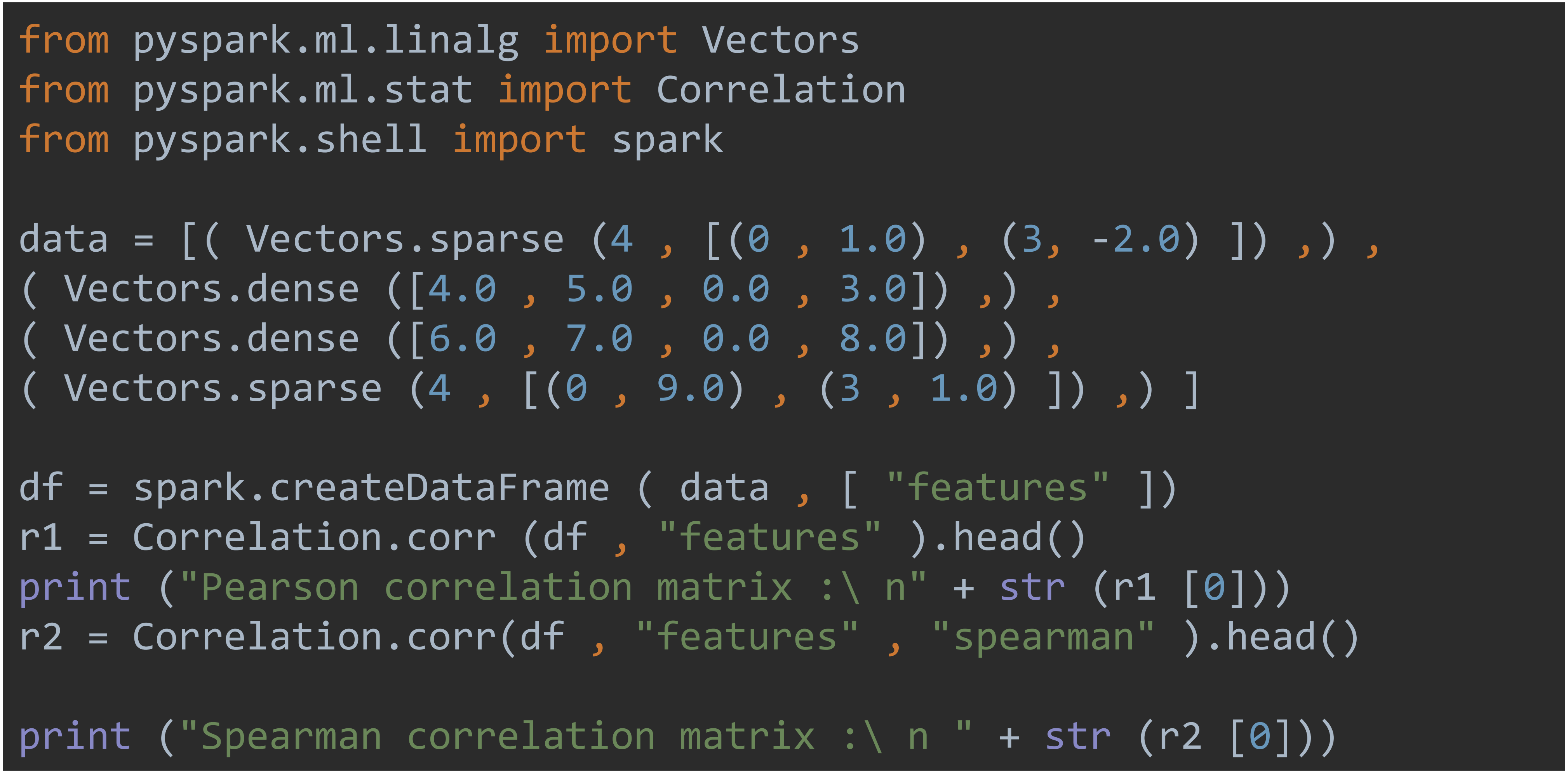
Spark Example terminal line

```
> from pyspark.ml.linalg import Vectors
> from pyspark.ml.stat import Correlation

> data = [( Vectors.sparse (4 , [(0 , 1.0) , (3, -2.0) ]) ,) ,
( Vectors.dense ([4.0 , 5.0 , 0.0 , 3.0]) ,) ,
( Vectors.dense ([6.0 , 7.0 , 0.0 , 8.0]) ,) ,
( Vectors.sparse (4 , [(0 , 9.0) , (3 , 1.0) ]) ,) ]

> df = spark.createDataFrame ( datr1 = Correlation.corr (df ,
"features" ).head()
a , [ "features" ])
> > print ("Pearson correlation matrix :\ n" + str (r1 [0]))
> r2 = Correlation.corr(df , "features" , "spearman" ).head()

> print ("Spearman correlation matrix :\ n " + str (r2 [0]))
```

Spark Example Jupiter lab or notebook

- In conda terminal
  - conda activate pyspark_env
  - pip install jupyterlab
  - jupyter-lab

Spark Example Pycharm



```
from pyspark.ml.linalg import Vectors
from pyspark.ml.stat import Correlation
from pyspark.shell import spark

data = [( Vectors.sparse (4 , [(0 , 1.0) , (3, -2.0) ]) ,) ,
( Vectors.dense ([4.0 , 5.0 , 0.0 , 3.0]) ,) ,
( Vectors.dense ([6.0 , 7.0 , 0.0 , 8.0]) ,) ,
( Vectors.sparse (4 , [(0 , 9.0) , (3 , 1.0) ]) ,) ]

df = spark.createDataFrame ( data , [ "features" ])
r1 = Correlation.corr (df , "features" ).head()
print ("Pearson correlation matrix :\ n" + str (r1 [0]))
r2 = Correlation.corr(df , "features" , "spearman" ).head()

print ("Spearman correlation matrix :\ n " + str (r2 [0]))
```

Spark Hands-on

NASA Turbofan Jet Engine Data Set

- Based on the exploratory analysis presented by Kaggle in https://www.kaggle.com/code/brjapon/nasa-turbofan-exploratoryanalysis

Do an exploratory analysis of both datasets first using just pandas:

  - Files
  - Variables
  - Correlation
  - Which ML model would you apply?

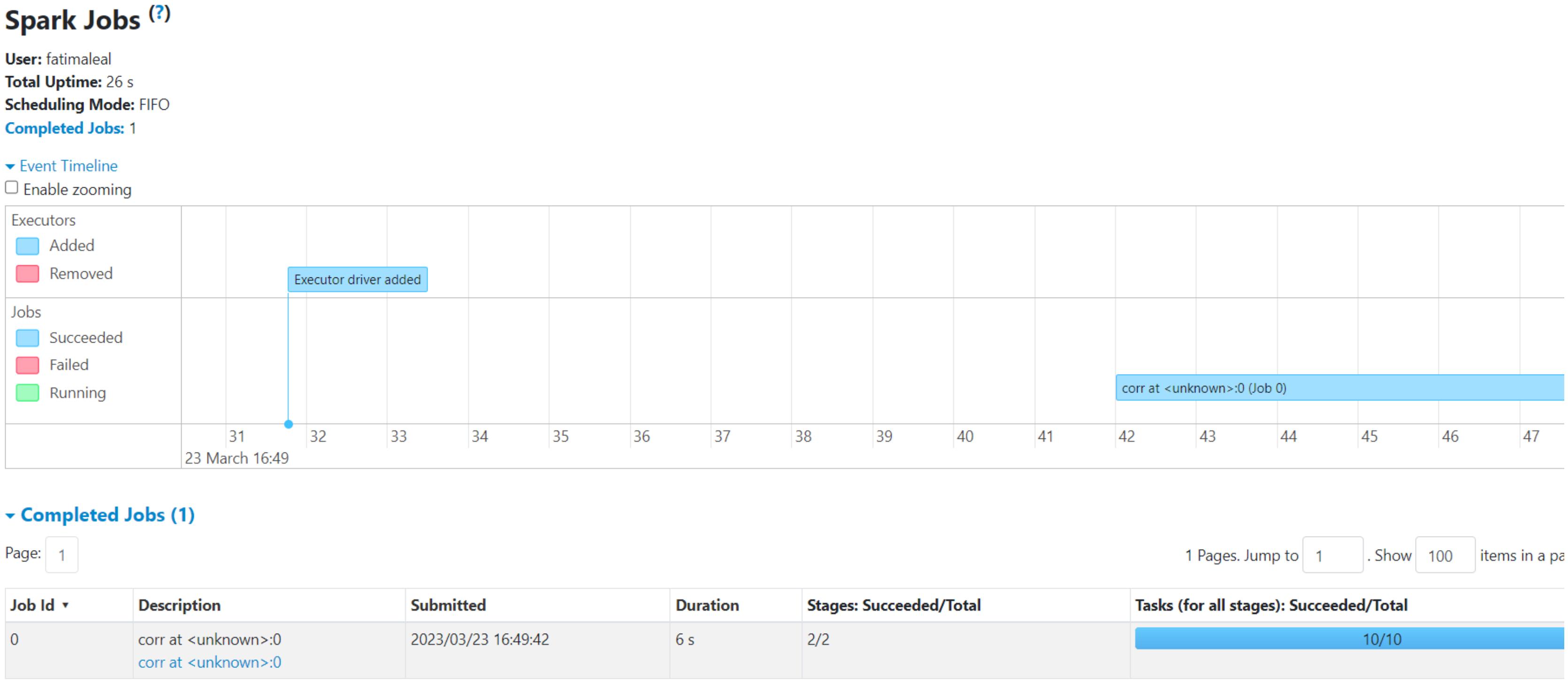
- Then try to apply spark

Spark Hands-on

```
import pandas as pd
from pyspark.shell import spark

train_data_no_name = pd.read_csv("train_FD001.txt", sep =
"\s+", header = None)

columns = ['engineNumber', 'cycleNumber', 'opSetting1',
'opSetting2', 'opSetting3', 'sensor1', 'sensor2',
           'sensor3', 'sensor4', 'sensor5', 'sensor6',
'sensor7', 'sensor8', 'sensor9', 'sensor10',
           'sensor11', 'sensor12', 'sensor13', 'sensor14',
'sensor15', 'sensor16',
            'sensor17', 'sensor18', 'sensor19', 'sensor20',
'sensor21']

df = spark.createDataFrame(train_data_no_name, columns)

print(df.corr('engineNumber', 'sensor1'))
```

Spark Hands-on

- In the User interface of Spark analyse the execution of the jobs



Spark Jobs (?)

User: fatimaleal
Total Uptime: 26 s
Scheduling Mode: FIFO
Completed Jobs: 1

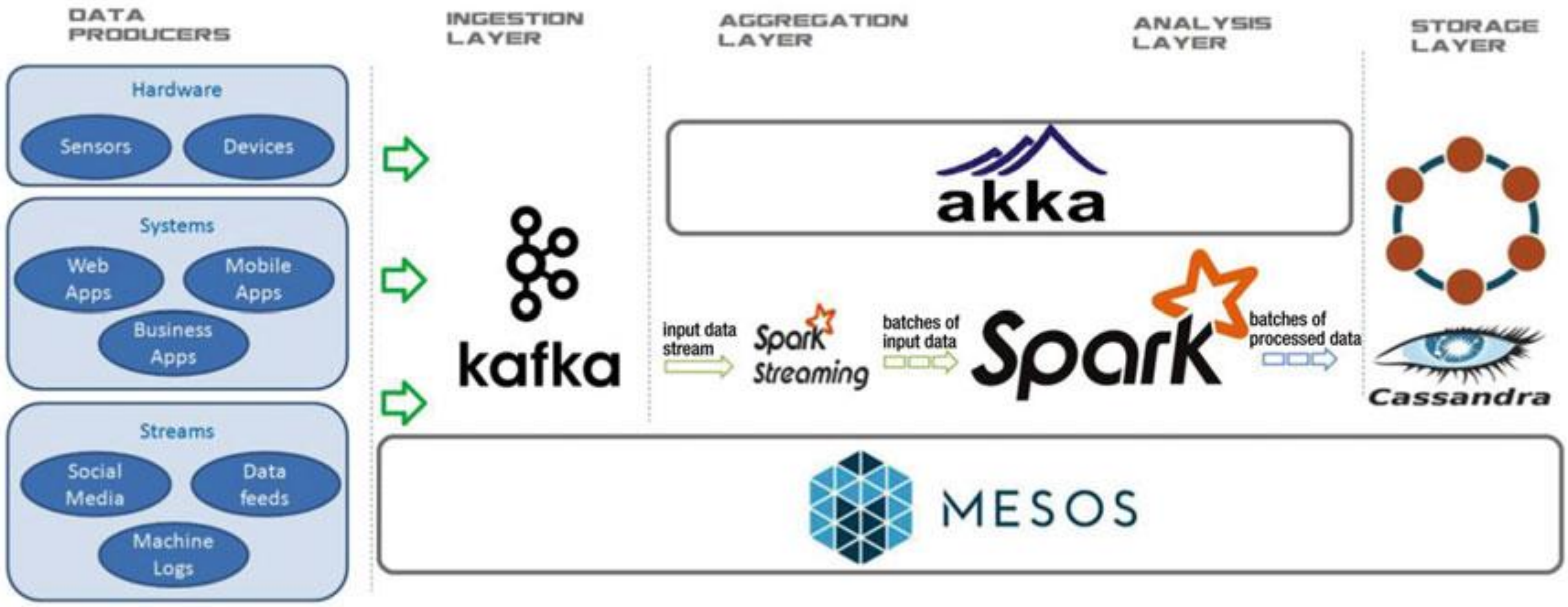
▾ Event Timeline
□ Enable zooming

▾ **Completed Jobs (1)**

Page: 1

1 Pages. Jump to 1 . Show 100 items in a pa

Job Id ▾	Description	Submitted	Duration	Stages: Succeeded/Total	Tasks (for all stages): Succeeded/Total
0	corr at <unknown>:0 corr at <unknown>:0	2023/03/23 16:49:42	6 s	2/2	10/10


DATA PRODUCERS
INGESTION LAYER
AGGREGATION LAYER
ANALYSIS LAYER
STORAGE LAYER
Hardware
Sensors
Devices
Systems
Web Apps
Mobile Apps
Business Apps
Streams
Social Media
Data feeds
Machine Logs
kafka
akka
input data stream
Spark Streaming
batches of input data
Spark
batches of processed data
Cassandra
MESOS

Next Class

- Installation Cassandra

- Installation Kafka

UPT

UNIVERSIDADE
PORTUCALENSE

Do conhecimento à prática.