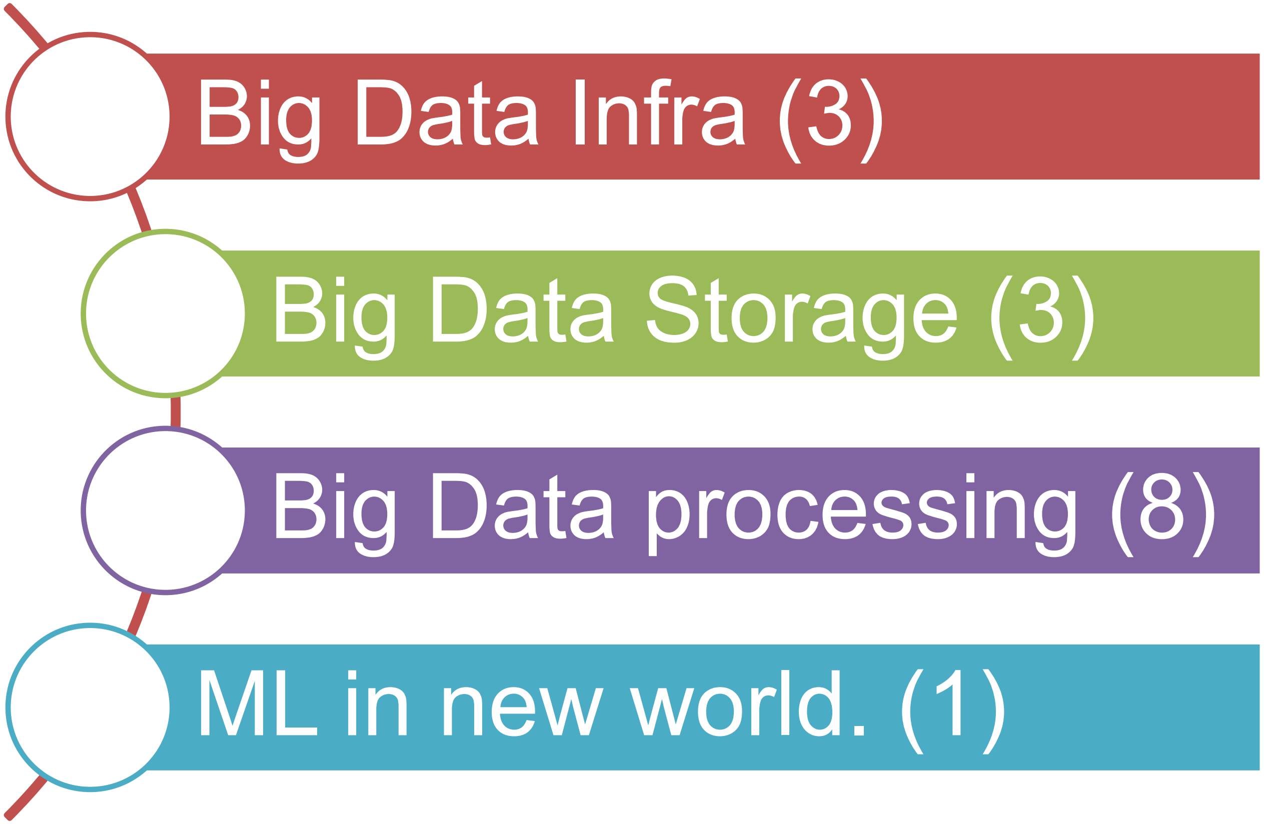
UNIVERSITAT DE BARCELONA

Master in Fundamental Principles of Data Science

Dr Rohit Kumar

Course Outline



Big Data Infra (3)

Big Data Storage (3)

Big Data processing (8)

ML in new world. (1)

15 Sessions in total

Big Data Infra

1. Introduction to Big Data. (1 session) – T
2. Introduction to Cloud Infrastructure (0.5 session) – TP
3. Introduction to Docker(1.5 session) - TP

Big Data Storage

1. No SQL (2 sessión) - TP
    1. MongoDB
2. HDFS/S3/Azure Blob (1 sessión) TP

Big Data processing

1. Data Pipelines using Airflow (2 sessions) - P
2. PySpark (6 sessions) - TP

ML in the new world

- Data Science Life cycle Managment (1 sessión) - TP
  - ML Training in Cloud
  - ML Production deployment

Grading Policy

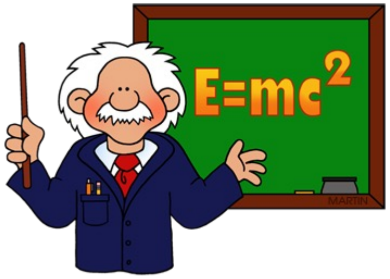
- 2 Assisgnments (30% each)
  - After 7th Session
  - After 11th Session
- 1 Final exam (40%)

Today’s Objective

- Lets get to know each other.
- Introduction to Big Data.
- An example architecture of Big data

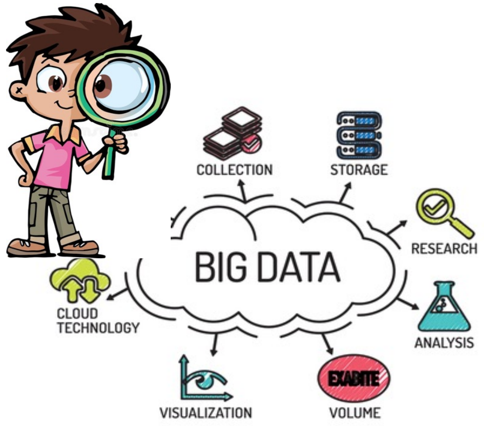
Hello!

Who Am I



Physicist

Java Developer/
System Architect (6+ years)



Big Data Researcher
(5+ years)

I am not a ML expert or a Data Scientist!!
I am just an engineer who can follow complex maths… ☺

Evolution of Infrastructure

What is an Infrastructure?

Lets talk about a city

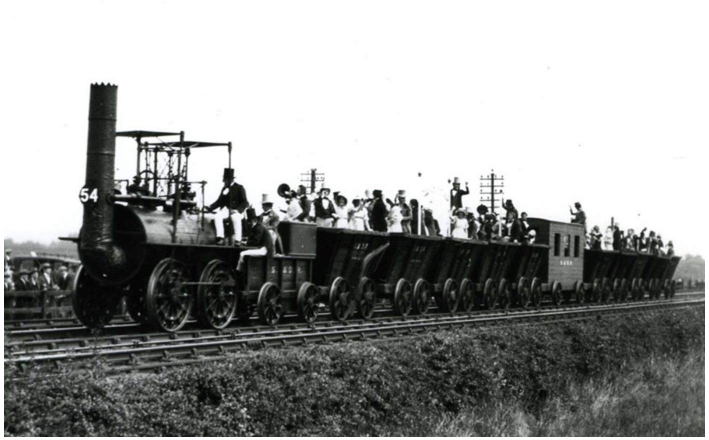
- What are different components in a city infrastrucutre?
  - Roads
  - Bridges
  - Flyovers
  - Underpasses
  - Railroads
  - Buses
  - Buildings
  - Park
  - Shops
  - ……

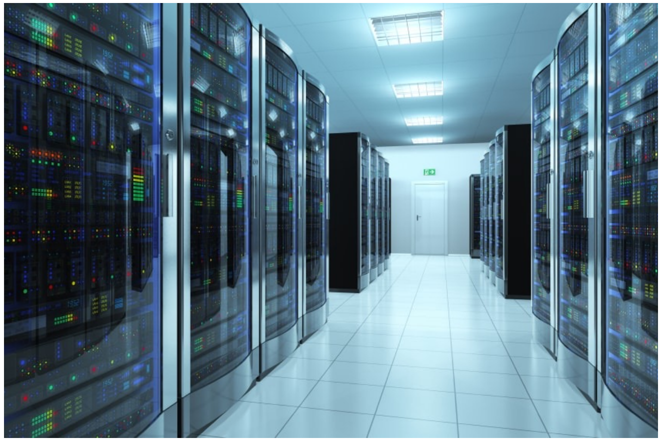
Digital infrastructre

- **Hardware**: Servers, switches, routers, PC, storage device, Cooling systems
- **O/S**: Most used Linux/Unix (Redhat, Solaris, HP-UX, AIX, Debian, Arch, Darwin(OSX)…) and Windows, Z/OS IBM…
- **Application**: Software (front-end or back-end), Processing Software(Logtash, Kafka…), Visual Software (Kibana)…
- **Network**: Firewalls, Proxies, subnet …
- **Storage**: RAM, SDD, HDD, Distributed, tapes, punch cards…

Digital Infrastructure

The Evaluation





Big Data world

Scale of Data

- 40ZB of data will be created by 2020, 300% increase since 2005
- 6 billion people have cell phones
- World population 7 billion

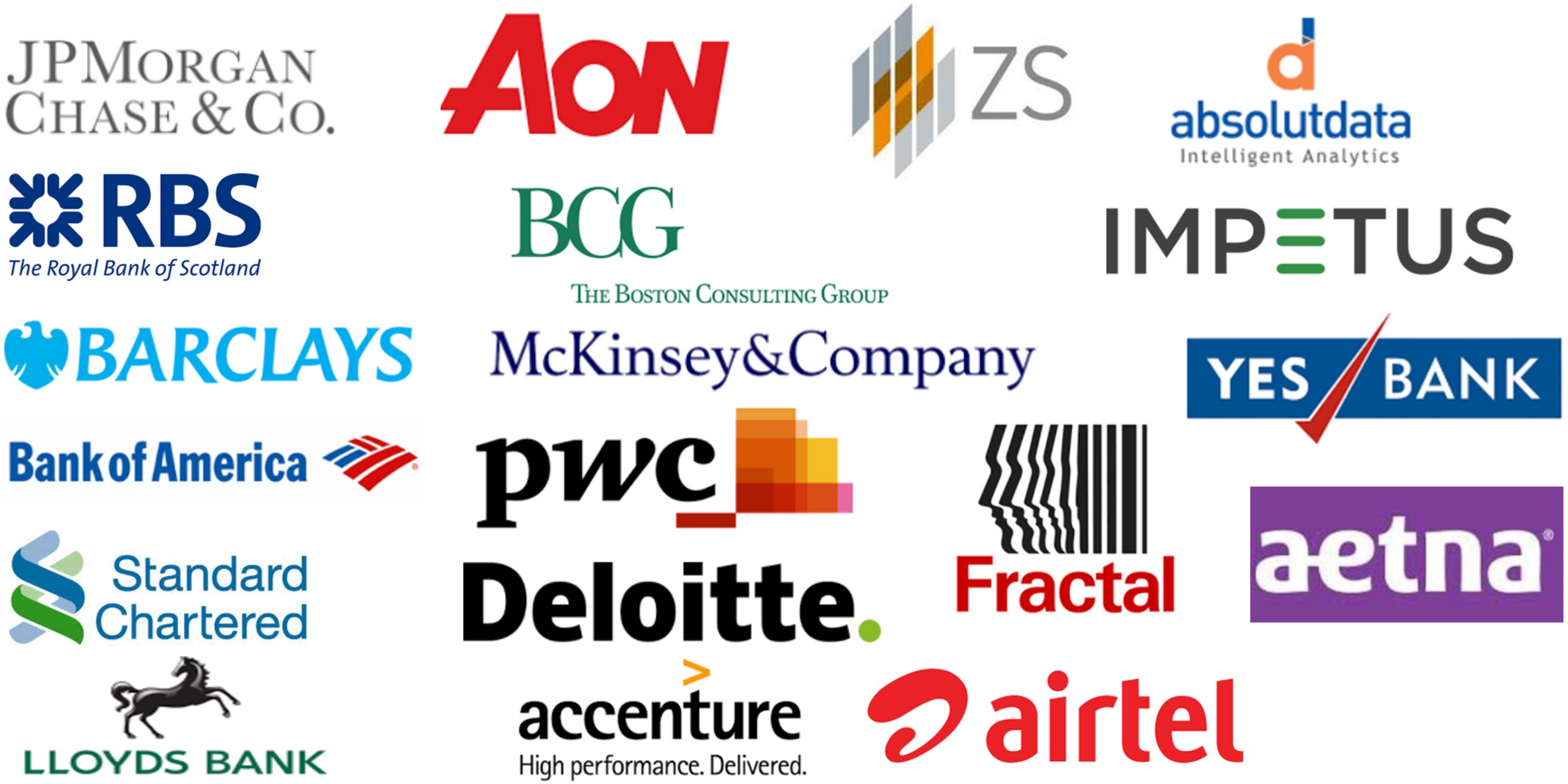
Analysis of streaming data

- 1TB of NYSE trade infomation during each trading session
- A jet engine can generate 10TB of data in 30min, 25k flights per day worldwide
- Sensors everywhere - IOT data

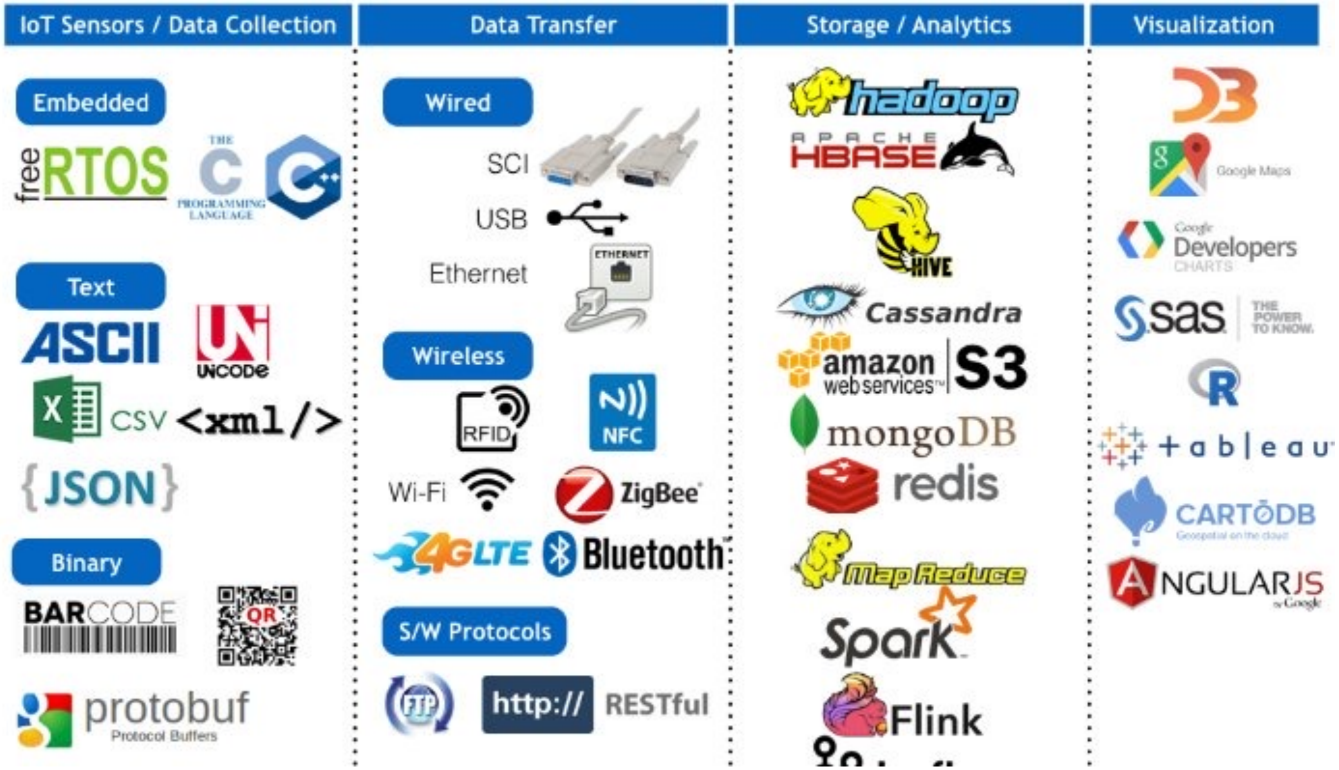
Different form of data

- 150EB of data by 2011 generated by Healthcare domain
- 30 billion content shared on FB every month. 208,300 photos every minute.
- 400 million tweets sent per day by 200 million active users
- 4 billion hours of video is watched on Youtube each month

Adoption of Big Data Technology



Big data technology Stack



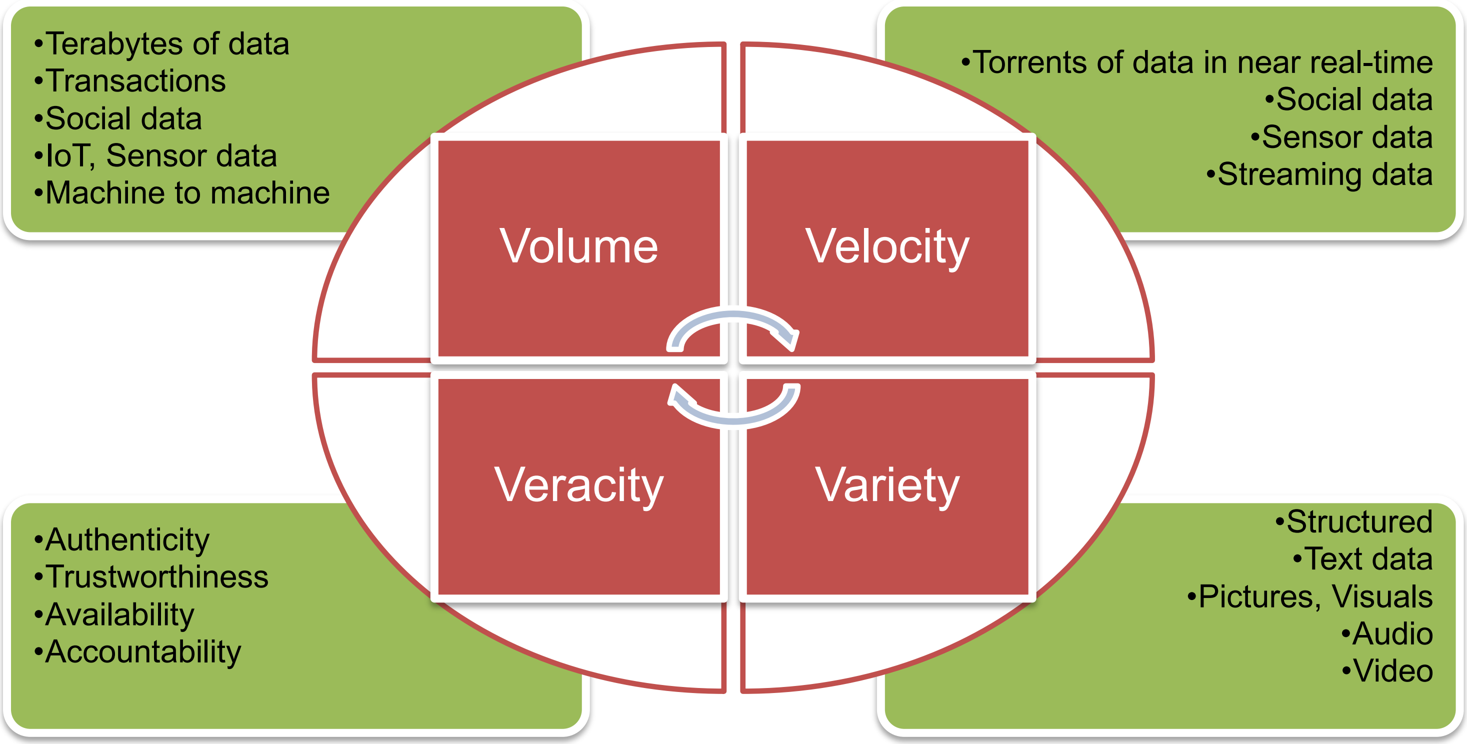
What is Big Data?

- Big data is large collection of data (both structured and unstructured)
  - Structured : Data stored in SQL based systems
  - Unstructured/Semi-Structured: JSON files, XML files, Server logs, Documents, Images, videos audio…

What is Big Data?

“Big data is being generated by everything around us at all times. Every digital process and social media exchange produces it. Systems, sensors and mobile devices transmit it. Big data is arriving from multiple sources at an alarming ***velocity***, ***volume*** and ***variety***.” - **IBM**

The 4 Vs



- Terabytes of data
- Transactions
- Social data
- IoT, Sensor data
- Machine to machine

Volume

- Torrents of data in near real-time
- Social data
- Sensor data
- Streaming data

Velocity

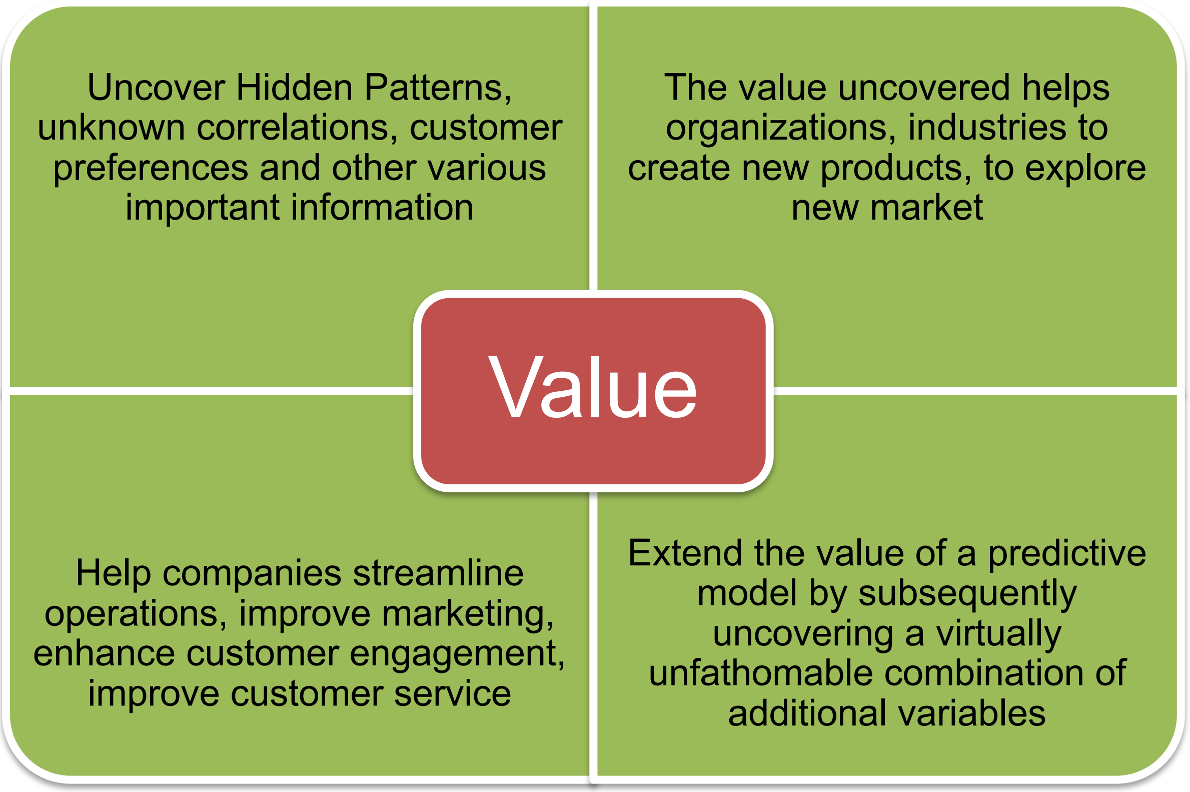
Veracity

- Authenticity
- Trustworthiness
- Availability
- Accountability

Variety

- Structured
- Text data
- Pictures, Visuals
- Audio
- Video

The 5th V



Value

Uncover Hidden Patterns, unknown correlations, customer preferences and other various important information

The value uncovered helps organizations, industries to create new products, to explore new market

Help companies streamline operations, improve marketing, enhance customer engagement, improve customer service

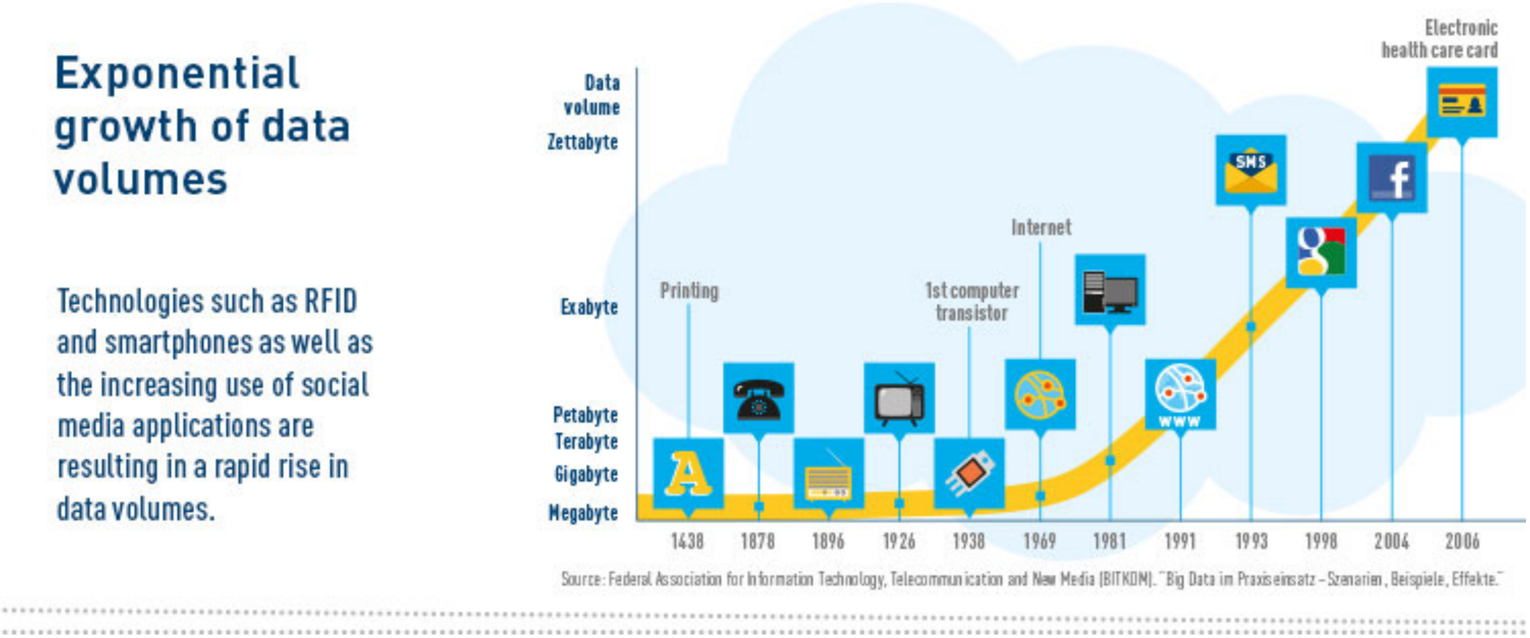
Extend the value of a predictive model by subsequently uncovering a virtually unfathomable combination of additional variables

Where Big data come from?

- Everything we do generates data somewhere:
  - Web Click
  - Online browsing and shopping
  - Server logs
  - Purchase Transactions in super market
  - Network communication between computers
  - Mobile communication
  - RFID data
  - ..

Universitat de Barcelona

Where Big data come from?



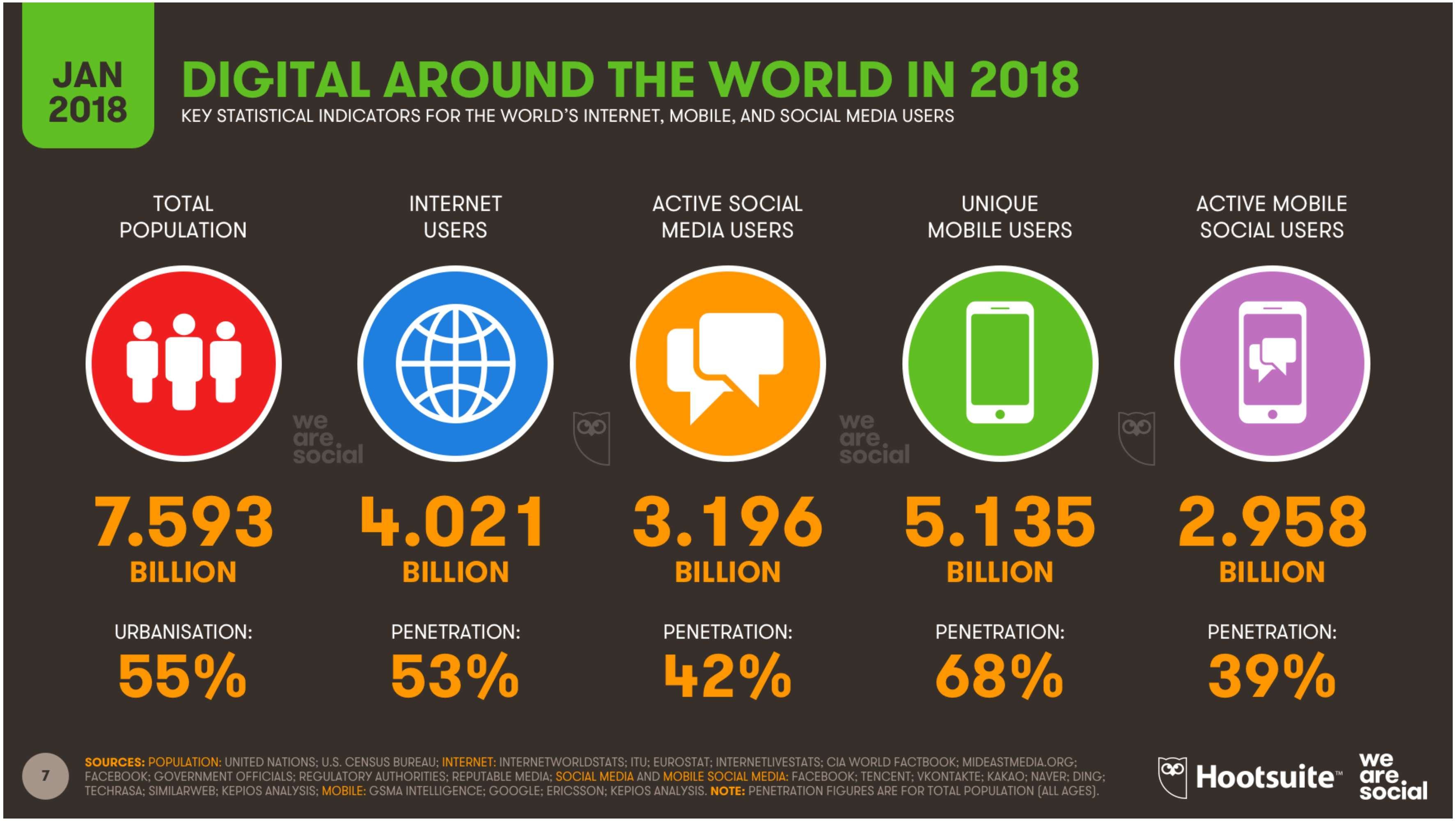
Exponential growth of data volumes

Technologies such as RFID and smartphones as well as the increasing use of social media applications are resulting in a rapid rise in data volumes.

Source: Federal Association for Information Technology, Telecommunication and New Media (BITKOM). "Big Data im Praxiseinsatz – Szenarien, Beispiele, Effekte."

* Image Source: http://www.metro-handelslexikon.de/en/special-topics/big-data/1/

Where Big data come from?



JAN 2018

DIGITAL AROUND THE WORLD IN 2018

KEY STATISTICAL INDICATORS FOR THE WORLD'S INTERNET, MOBILE, AND SOCIAL MEDIA USERS

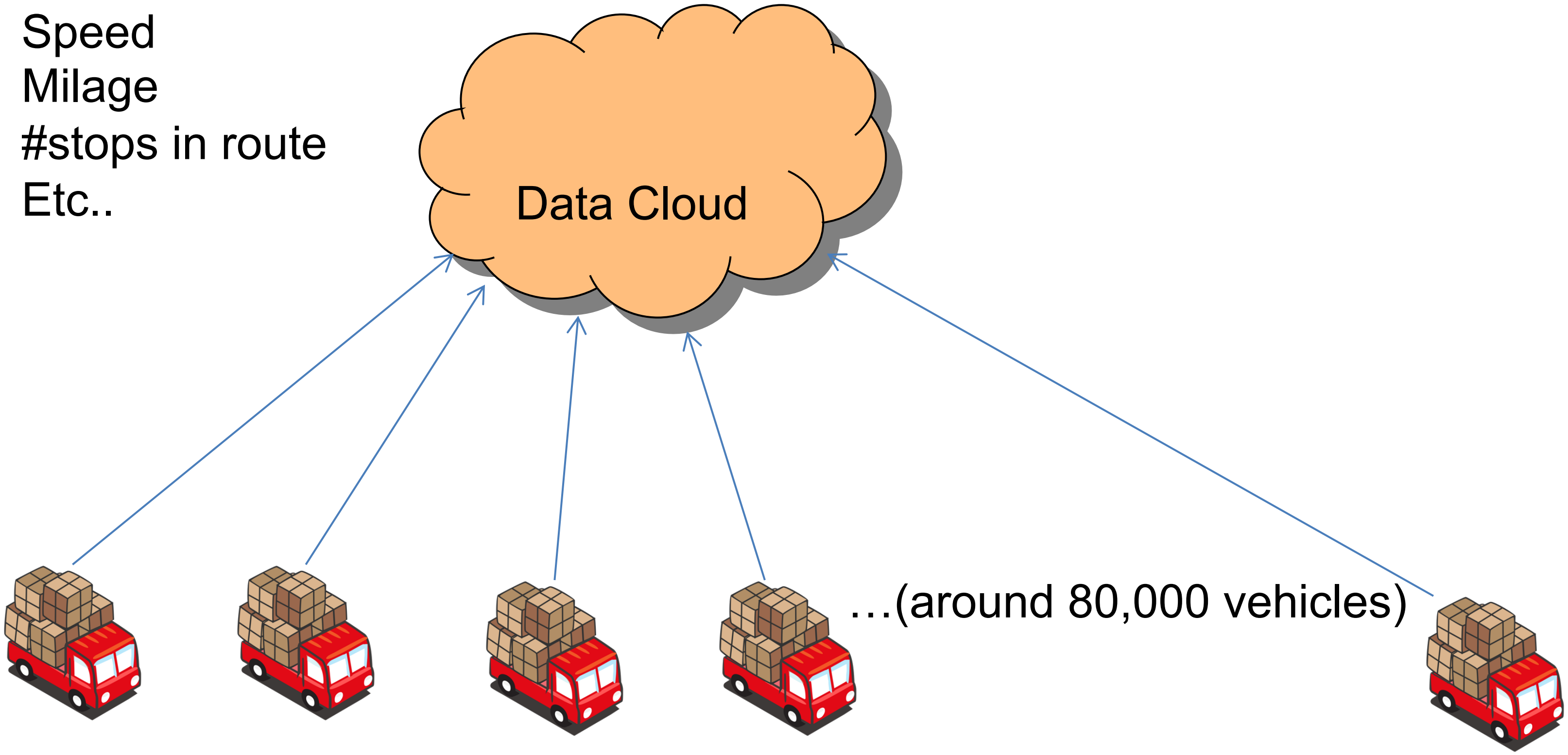
TOTAL POPULATION	INTERNET USERS	ACTIVE SOCIAL MEDIA USERS	UNIQUE MOBILE USERS	ACTIVE MOBILE SOCIAL USERS
7.593 BILLION	4.021 BILLION	3.196 BILLION	5.135 BILLION	2.958 BILLION
URBANISATION: 55%	PENETRATION: 53%	PENETRATION: 42%	PENETRATION: 68%	PENETRATION: 39%

SOURCES: POPULATION: UNITED NATIONS; U.S. CENSUS BUREAU; INTERNET: INTERNETWORLDSTATS; ITU; EUROSTAT; INTERNETLIVESTATS; CIA WORLD FACTBOOK; MIDEASTMEDIA.ORG; FACEBOOK; GOVERNMENT OFFICIALS; REGULATORY AUTHORITIES; REPUTABLE MEDIA; SOCIAL MEDIA AND MOBILE SOCIAL MEDIA: FACEBOOK; TENCENT; VKONTAKTE; KAKAO; NAVER; DING; TECHRASA; SIMILARWEB; KEPIOS ANALYSIS; MOBILE: GSMA INTELLIGENCE; GOOGLE; ERICSSON; KEPIOS ANALYSIS. **NOTE:** PENETRATION FIGURES ARE FOR TOTAL POPULATION (ALL AGES).

UPS: IoT example

200 data points per vehicle every day from Various sensors in the truck:

1) Speed
2) Milage
3) #stops in route
4) Etc..

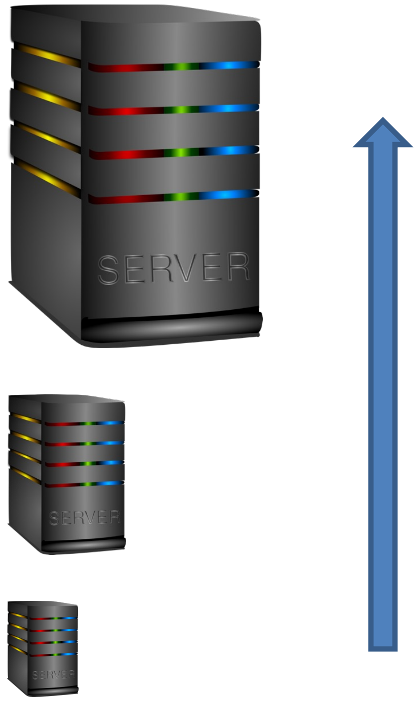


Intro to Big data Architecture

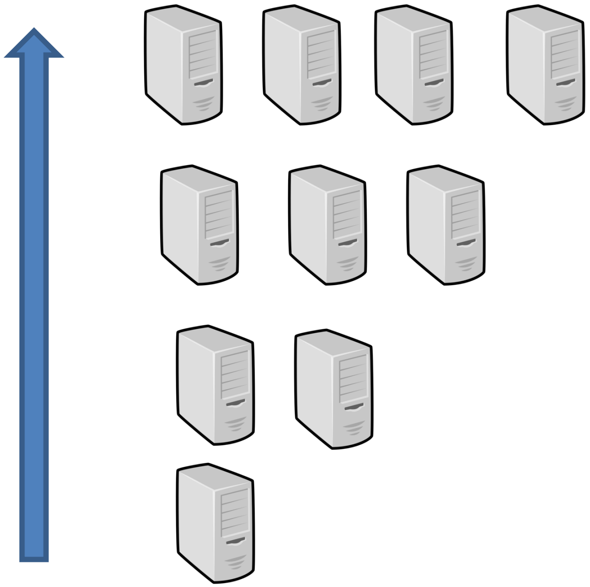
Horizontal vs Vertical Scaling!

- Vertical Scaling:
  - Everything works on one System.
  - Scaling is achieved by adding more RAM or CPU to an existing machine.
  - Very costly
- Horizontal Scaling: (Big Data Infrastructure)
  - Scale by adding more machine in the pool
  - Computation is distributed to different machine
  - Synchronization and Consistency become hard to achieve.
  - Cheaper to scale

Horizontal vs Vertical Scaling!



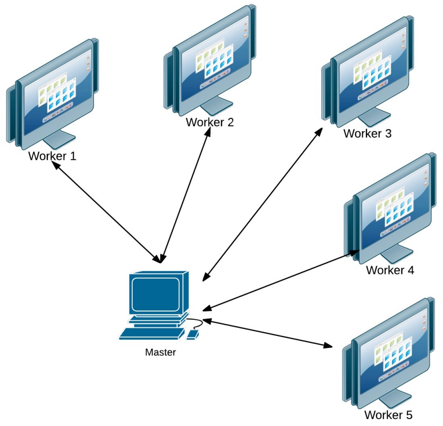
Vertical Scaling



Horizontal Scaling

Distributed System

"A distributed system is a model in which components located on networked computers communicate and coordinate their actions by passing messages" - Wikipedia



Distributed System(continued)

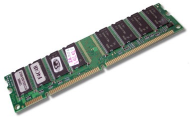
Some important requirements of Distributed File systems:

- Fault tolerance
- Consistency support
- Concurrency support

Distributed System(continued)

From the resource sharing paradigm there are three kinds of distributed architect

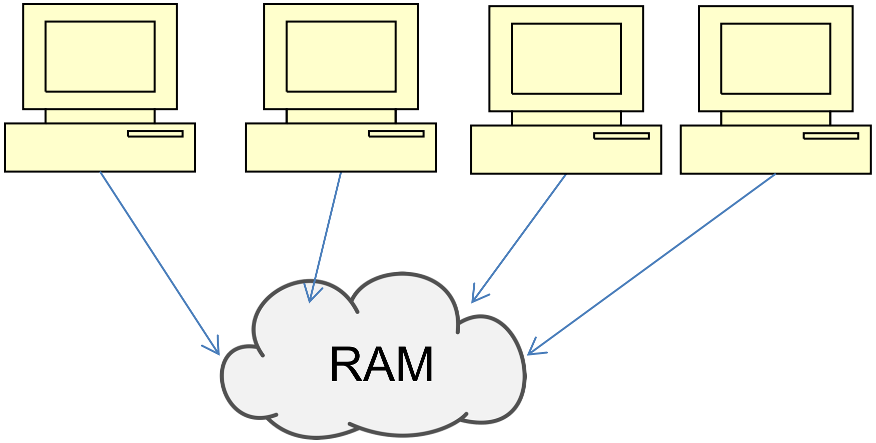
1) Shared memory



2) Shared disk

3) Shared nothing

Shared memory



Shared memory

These system share a common memory space like a distributed cache. For example: Oracle coherence, Hazelcast

Advantage:

High speed access to data.

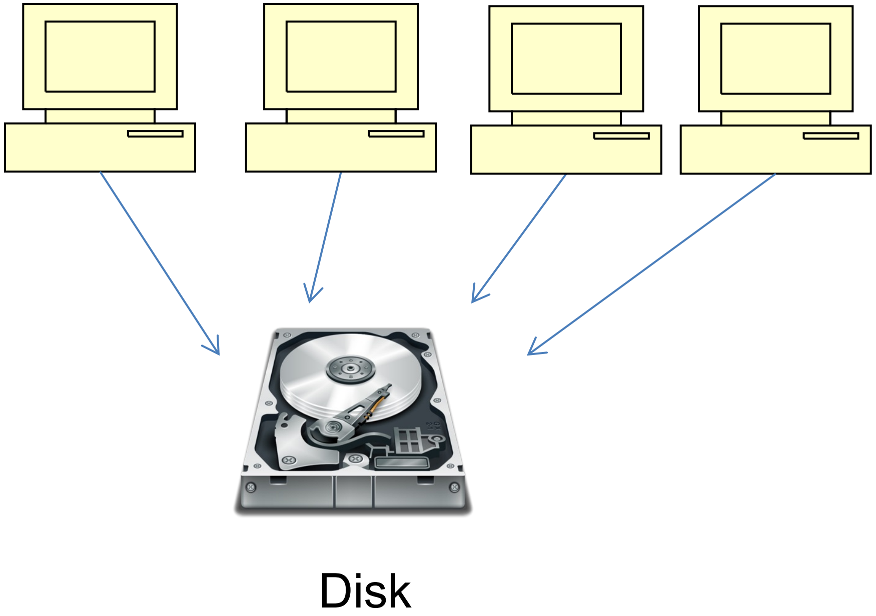
Disadvantage:

Increasing memory is difficult

Maintaining consistency is important and costly

Can be very expensive

Shared disk



Shared disk

These system share a common disk space typically through a LAN. They are also known as ***clustered file system***. For example: NEC ExpressCluster.

Advantage:

Increasing capacity is less costly.

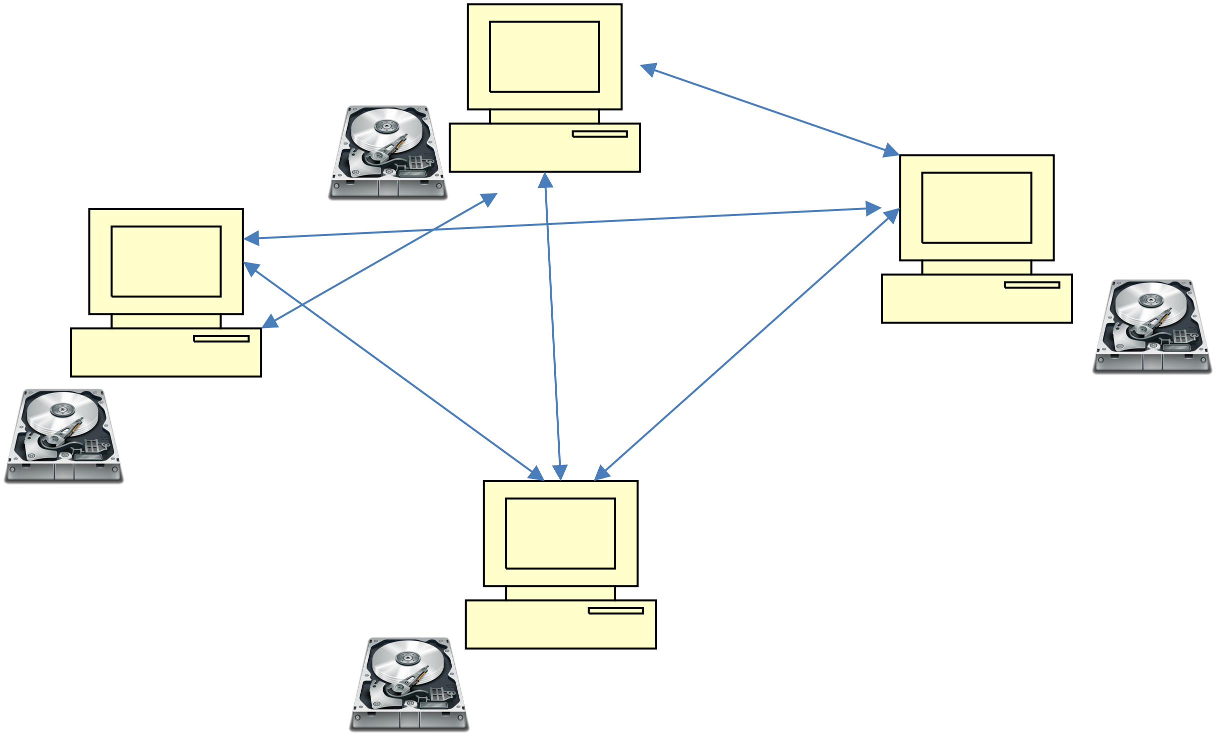
Almost transparent to the application on top.

Disadvantage:

Expensive

Access contention and consistency issue if high number of clients using same data.

Shared Nothing



Shared Nothing

Distributed system where each machine has its own memory space and is agnostic to other machine memory. For example: Spark, Hadoop, Flink etc.

Advantage:

Cheap to scale

Highly adaptive

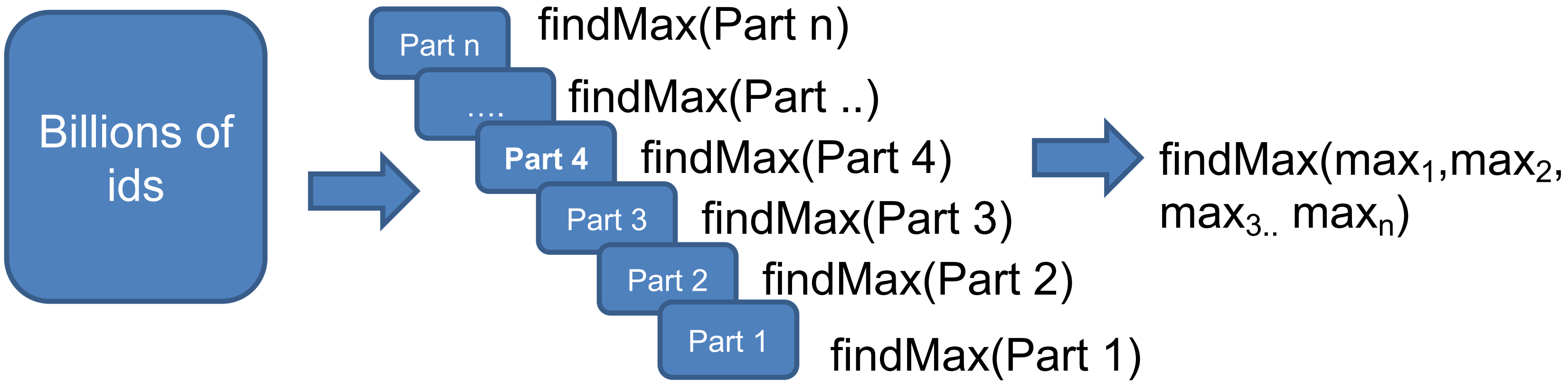
Disadvantage:

New programing abstraction hence applications need to be redesigned.

Shared Nothing (Continued..)

The shared nothing architecture works on an interesting concept called data locality i.e., *sending the code to the data instead of sending data to code.*

Example: findMax(data)=maximum id in the data.



Questions?

- We saw some example of hardware level architecture difference.
- Now lets see an application level architecture example with Lambda architecture.

Lambda Architecture

λ

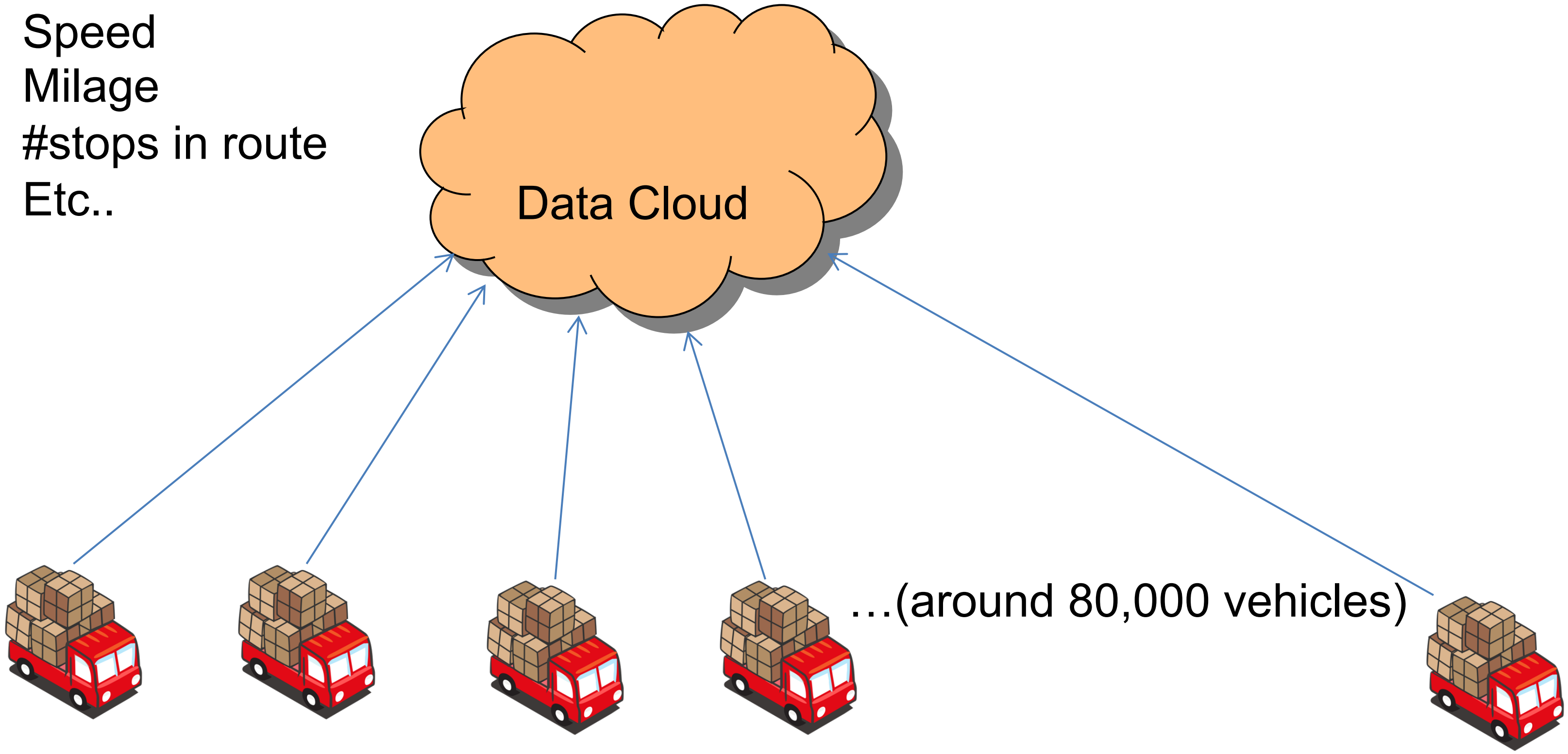
What problem does it solve?

Lets go back to the IoT use case we saw earlier

UPS: IoT example

200 data points per vehicle every day from Various sensors in the truck:

1) Speed
2) Milage
3) #stops in route
4) Etc..



Data Cloud

…(around 80,000 vehicles)

What kind of information you need?

- What is the average speed of the trucks?
- How many trucks have more than 20 stops in the route?
- How many of the trucks are currently running at speed more than 60 Kmph?
- .. Any more suggestions??

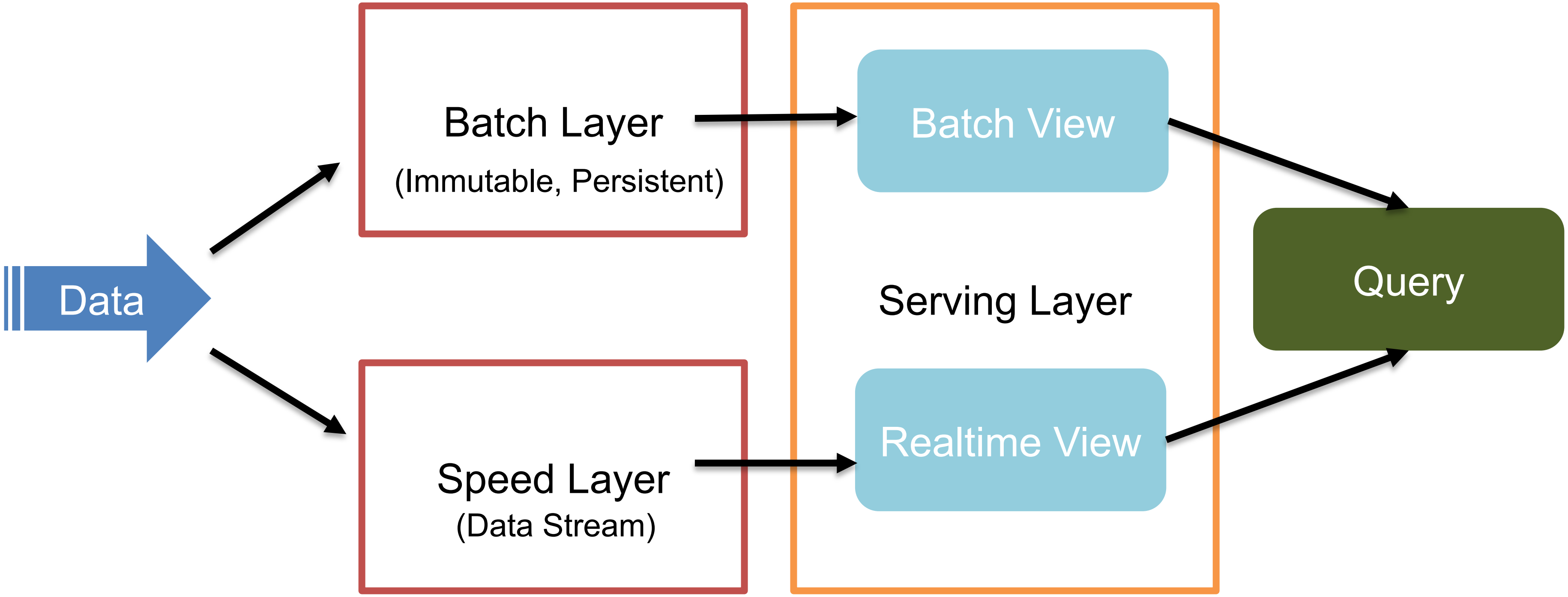
Lambda Architecture

So we see there are two kinds of information we need that will be answered using two different kind of queries

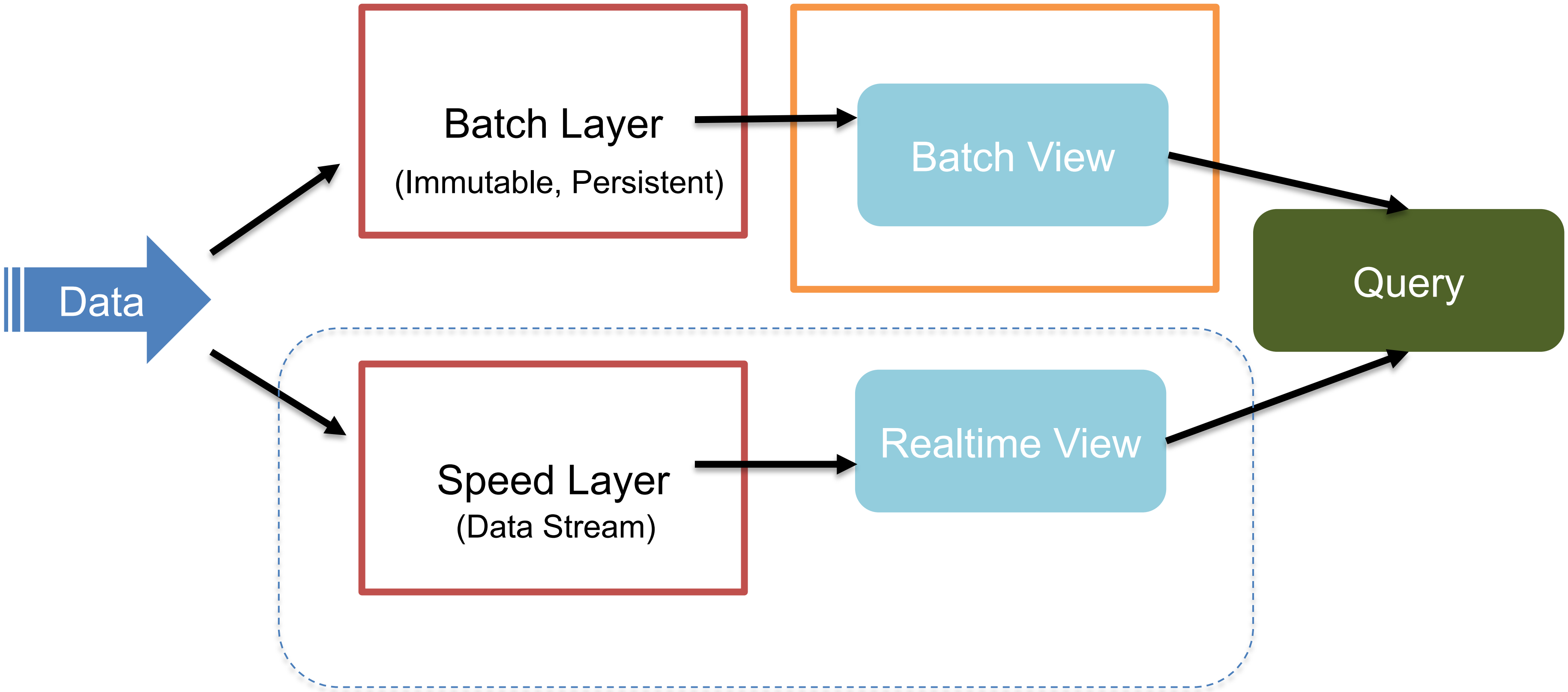
1. Static less changing data points: such as average speed, average route stops etc..
2. Realtime fast changing data points: such as #vehicles running faster than 60 Kmph.

Problem: Data is coming at fast speed and you want to answer some queries very fast without re compute.

Lambda Architecture



Lambda Architecture



In some cases depending on the tools used
the speed layer and real time view could be together

Lambda Architecture

- All the data is sent to both the batch and speed layers.
- Batch layer also acts as the **master data** set and is **immutable.**
- Batch layer pre-computes query functions from scratch in regular

Batch layer

New data comes continuously, as a feed to the data system. It gets fed to the batch layer and the speed layer simultaneously. It looks at all the data at once and eventually corrects the data in the stream layer. Here you can find lots of ETL and a traditional data warehouse. This layer is built using a predefined schedule, usually once or twice a day.

The batch layer has two very important functions:

1. To manage the master dataset
2. To pre-compute the batch views.

Ref: https://databricks.com/glossary/lambda-architecture

Speed Layer (Stream Layer)

This layer handles the data that are not already delivered in the batch view due to the latency of the batch layer. In addition, it only deals with recent data in order to provide a complete view of the data to the user by creating real-time views.

It needs to be really fast in compute so uses incremental algorithms and read/write databases to produce realtime views.

Ref: https://databricks.com/glossary/lambda-architecture

Serving Layer

The outputs from the batch layer in the form of batch views and those coming from the speed layer in the form of near real-time views get forwarded to the serving. This layer indexes the batch views so that they can be queried in low-latency on an ad-hoc basis.

Before Next Class

- Setup free azure account (https://azure.microsoft.com/es-es/offers/ms-azr-0170p/)

No credit card required use UB student emails.

- Please install docker in your laptops you can follow following links.

https://docs.docker.com/docker-for-mac/install/

https://docs.docker.com/install/linux/docker-ce/ubuntu/#install-using-the-repository

https://docs.docker.com/desktop/windows/install/

Thank you!