BIG DATA ANALYTICS (DSE 3264)

Faculty Name:

Shavantrevva S Bilakeri (SSB)

Assistant Professor, Dept of DSCA, MIT.

Phone No: +91 8217228519 E-mail: ss.bilakeri@manipal.edu

Course Objectives

- To be familiar with overview of **Apache Hadoop Ecosystem**
- Understand the **storage mechanism**, architecture, features and execution modes of big data tools (**Pig, Hive, HBase, Spark**)
- To be proficient in data analysis and its implications on **structured, unstructured, and semi- structured data.**
- To be proficient with **Big Data framework and use cases**.

Course Outcome

- Identify Big Data and its business inferences.
- Explore, Manage and **Analyze Job executions** in local & cluster-based Hadoop environment.
- Apply and Perform machine learning techniques using Scala & python.

Syllabus

Introduction to Big Data: evolution, structuring elements, big data analytics, distributed and parallel computing for big data, Life cycle of Big data, Cloud computing and big data, in-memory computing technology for big data, Big Data Stack, Layer Structure, Big Data Layout.

Hadoop: ecosystem, Hadoop Distributed File System (HDFS),

MapReduce: MapReduce Framework, optimizing MapReduce jobs, MapReduce Applications, Understanding YARN architecture.

Big Data Tools: **"PIG":** History, Features, Architecture, Components, Data Models, Operators, Running & Executing Modes, Analysing data with Pig, Pig Libraries, Processing Structured Data using Pig.

Syllabus

Big Data Tools: “HBASE”: History, Characteristics, Features, Architecture, Storage Mechanism, HDFS Versus HBASE, HBase Query writing.

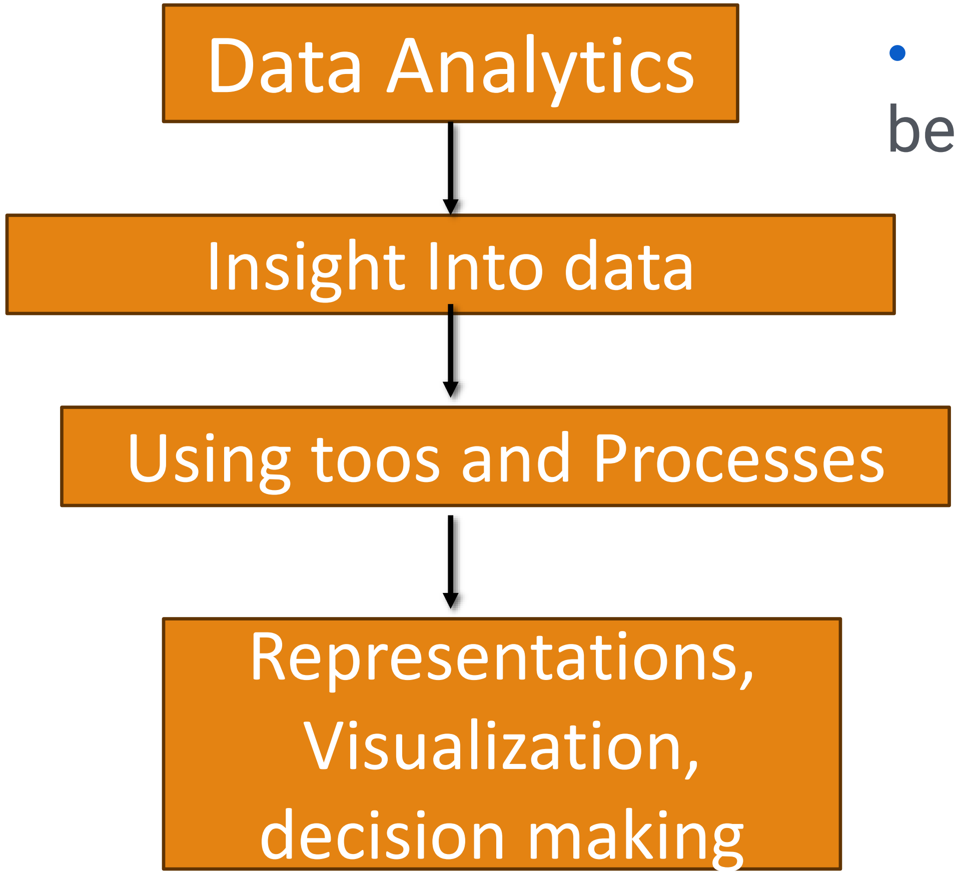
Big Data Tools: “Hive”: Brief History of Hive, Data Types In Hive, Executing Modes, Writing & Executing Hive queries.

Big Data tools: “Apache Spark”: Spark Architecture, Components, Features, Spark vs Hadoop, RDD, Need for RDD, Spark memory management & Fault tolerance, Spark’s Python and Scala shells, Programming with RDD: RDD Operations, Passing Functions to Spark, Common Transformations and Actions,

Contents

- Introduction to Big Data
- Types of Big Data
- Big Data characteristics
- Challenges
- Data Generators (Fields of Big Data)
- Traditional Vs Big Data approach
- Life Cycle
- Case Study.

Big Data Analytics



Data Analytics

Insight Into data

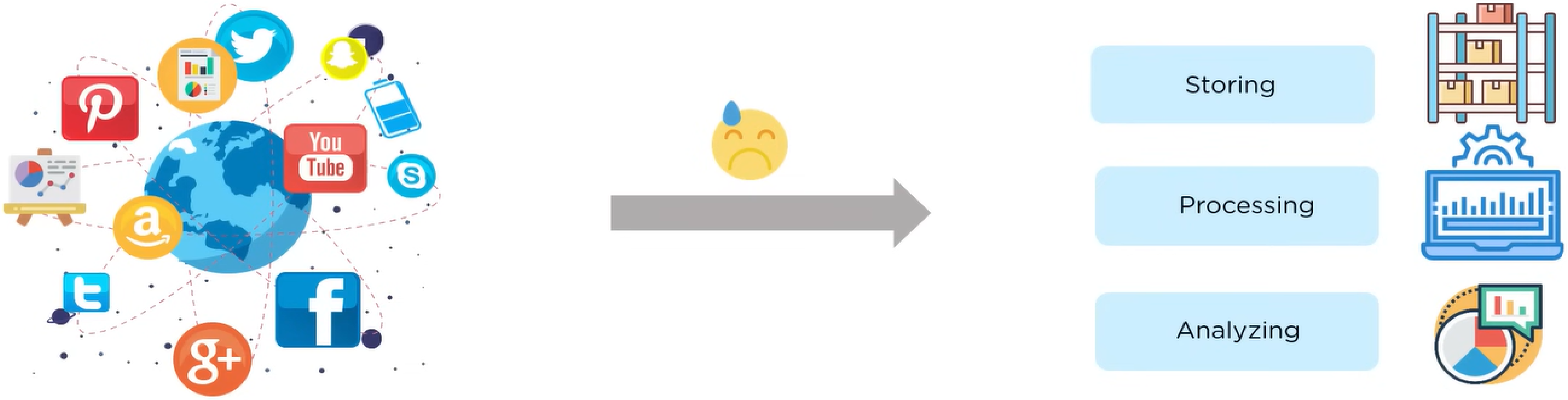
Using toos and Processes

Representations, Visualization, decision making

- Big Data is a massive amount of datasets that **cannot** be **stored, processed, or analyzed using traditional tools**.
  - The term *big data* was first used to refer to increasing data volumes in the mid-1990s.
  - In 2001, Doug Laney, then an analyst at consultancy Meta Group Inc., expanded the definition of big data.
    - ***Volume*** of data being stored and used by organizations.
    - ***Variety*** of data being generated by organizations.
    - ***Velocity***, or speed, in which that data was being created and updated.

Why Big Data Analytics

Massive amount of data which cannot be stored, processed and analyzed using traditional tools is known as big data



Storing

Processing

Analyzing

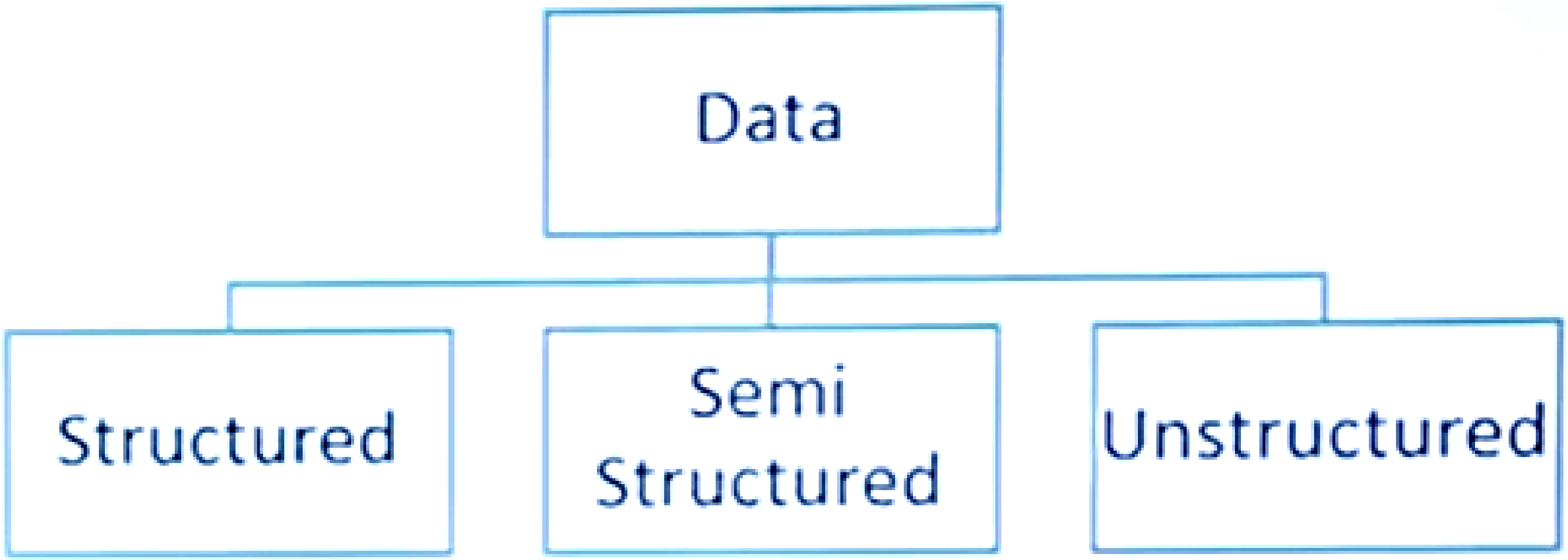
Big data

Storing, processing and analyzing big data became difficult using traditional methods

Small Data VS Big Data

SL.NO	Small Data	Big Data
1	Structured	Structured /Unstructured/ semistructured/
2	**Megabyte (MB)**: 1 MB is equivalent to 1,024 kilobytes or approximately 1 million bytes. **Gigabyte (GB)**: 1 GB equals 1,024 megabytes or approximately 1 billion bytes. **Terabyte (TB)**: 1 TB is equal to 1,024 gigabytes or approximately 1 trillion bytes.	**Petabyte (PB)**: 1 PB is equivalent to 1,024 terabytes or 1 million gigabytes. **Exabyte (EB)**: 1 EB equals 1,024 petabytes or 1 billion gigabytes. **Zettabyte (ZB)**: 1 ZB is equal to 1,024 exabytes or 1 trillion gigabytes. **Yottabyte (YB)**: 1 YB represents 1,024 zettabytes or 1 quadrillion gigabytes.
3	Gradually Increases (Slow)	Exponentially/ Rapidly Increases
4	Locally Present	Globally Present
5	Centralized	Distributed
6	Frameworks:Oracle, SQL Server	Hadoop, Spark, Cassandra
7	Single Node	Multiple Node

TYPES OF DATA



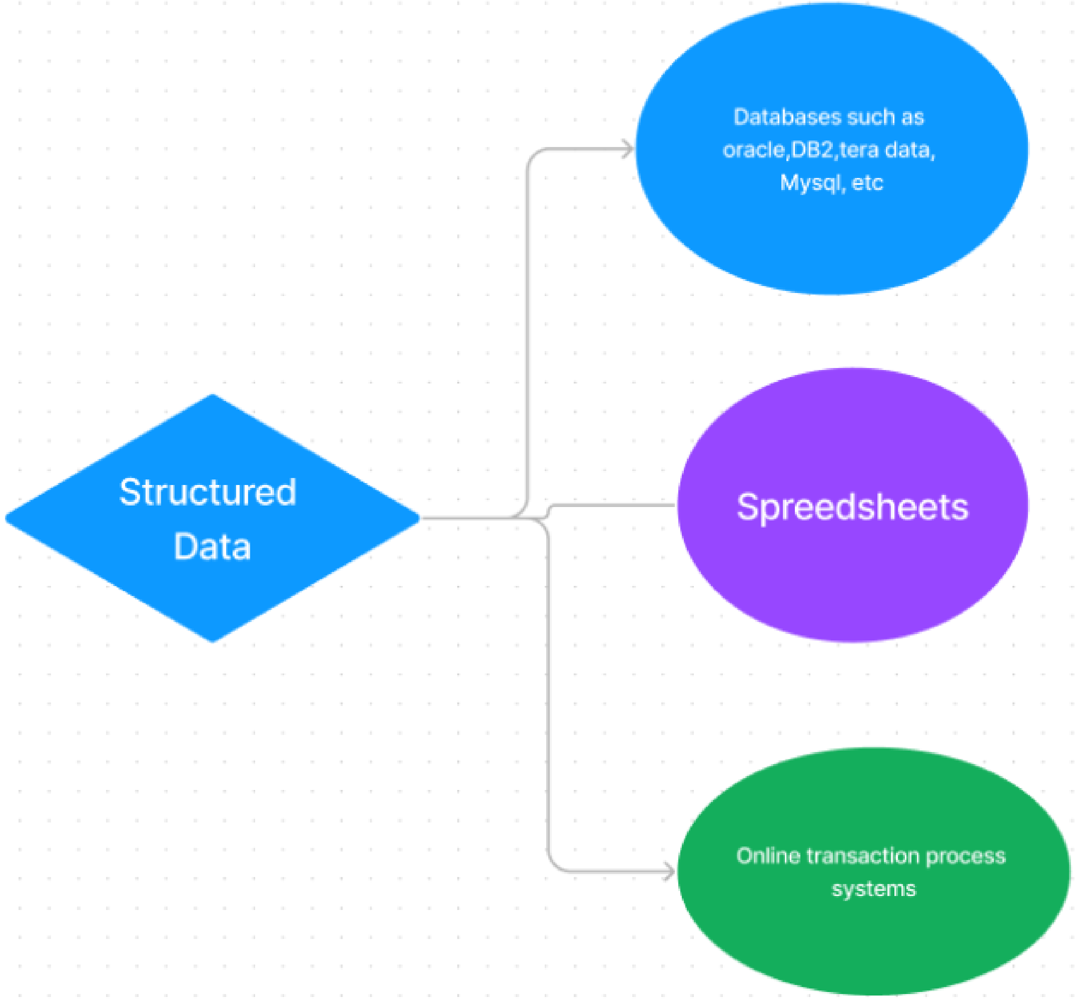
Data

Structured

Semi Structured

Unstructured

Structured


Databases such as oracle,DB2,tera data, Mysql, etc
Structured Data
Spreedsheets
Online transaction process systems

•Data that resides in a fixed field within a record.

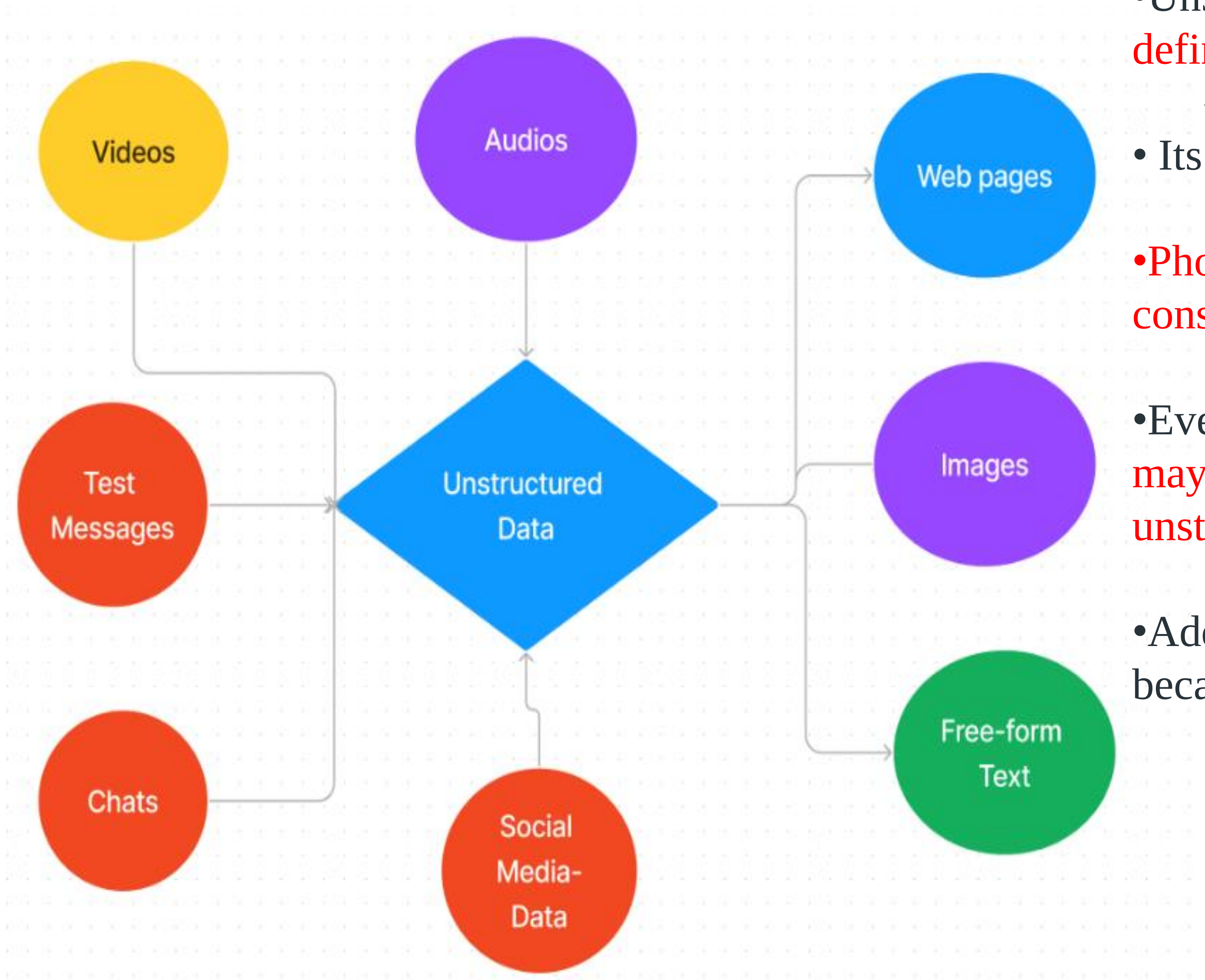
•It is type of data most familiar to our everyday lives. Ex: birthday, address

• A certain schema binds it, so all the data has the same set of properties. Structured data is also called relational data.

•It is split into multiple tables to enhance the integrity of the data by creating a single record to depict an entity.

•Relationships are enforced by the application of table constraints.

Unstructured



- Unstructured data is the kind of data that doesn't adhere to any definite schema or set of rules.

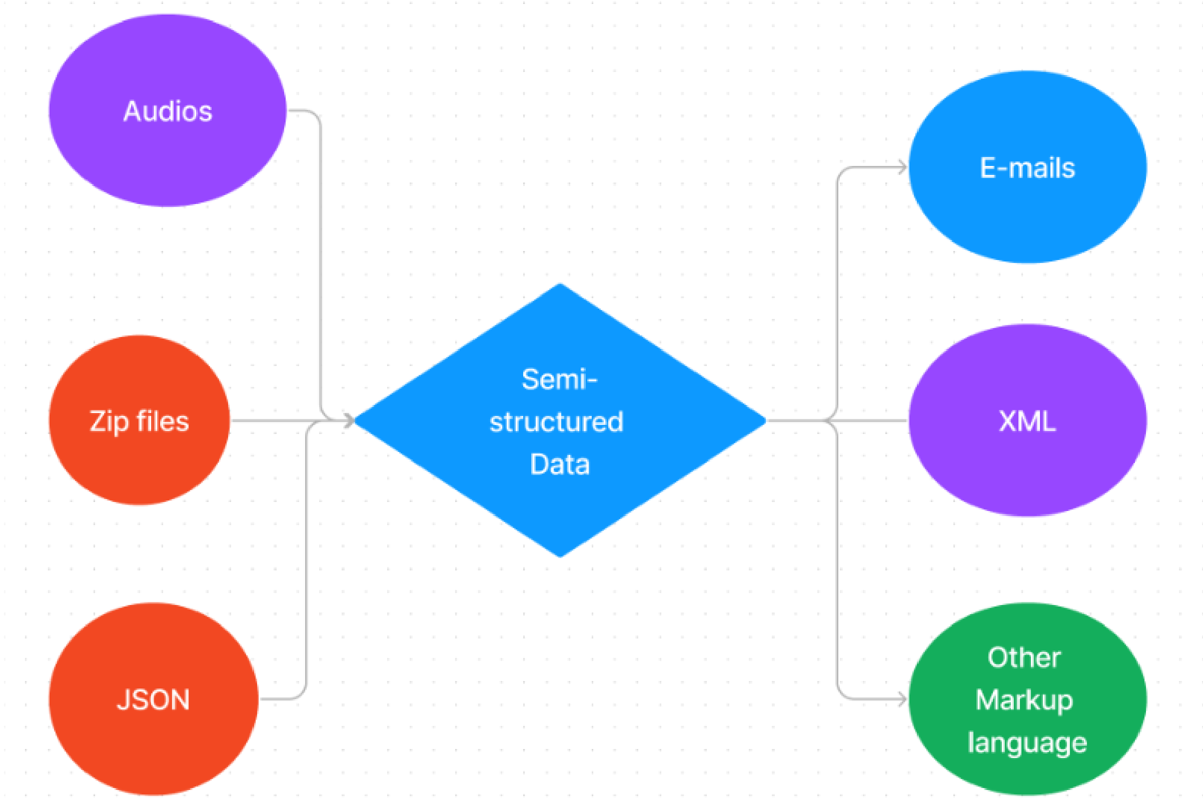
- Its arrangement is unplanned and haphazard.

- Photos, videos, text documents, and log files can be generally considered unstructured data.

- Even though the metadata accompanying an image or a video may be semi-structured, the actual data being dealt with is unstructured.

- Additionally, Unstructured data is also known as "dark data" because it cannot be analyzed without the proper software tools.

Semi- Structured



•Semi-structured data is not bound by any rigid schema for data storage and handling.

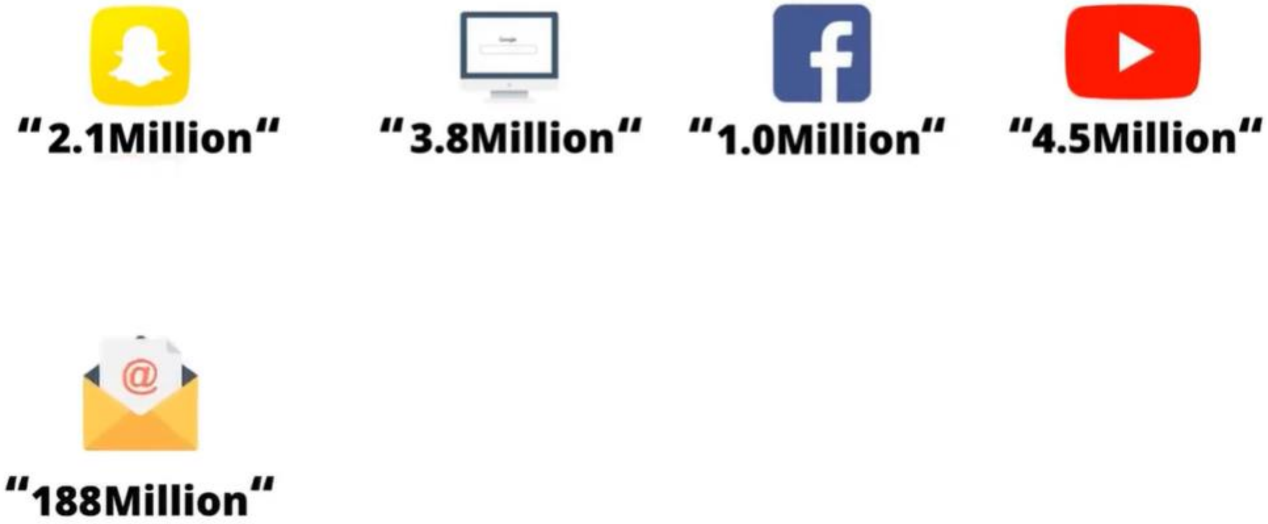
•The data is not in the relational format and is not neatly organized into rows and columns like that in a spreadsheet.

•However, there are some features like key-value pairs that help in discerning the different entities from each other.

•Since semi-structured data doesn’t need a structured query language, it is commonly called *NoSQL data*.

•This type of information typically comes from external sources such as social media platforms or other web-based data feeds.

Data Generated On internet – Per minute



Examples of Big Data

- Big data is a clustered management of different forms of data
- Generated by various devices (Android, iOS, etc.),
- Applications (music apps, web apps, game apps, etc.),
- Actions (searching through SE, navigating through similar types of web pages, etc.).

Challenges of Big Data

- **Rapid Data Growth:** The growth velocity at such a high rate creates a problem to look for insights using it. **There is no 100% efficient way to filter out relevant data.**
- **Storage:** The generation of such a massive amount of data needs space for storage, and organizations face challenges to **handle such extensive data without suitable tools and technologies.**
- **Unreliable Data:** It cannot be guaranteed that the big data collected and analyzed are totally (100%) accurate. **Redundant data, contradicting data, or incomplete** data are challenges that remain within it.
- **Data Security:** Firms and organizations storing such massive data (of users) can be a target of cybercriminals, and there is a risk of data getting stolen. Hence, encrypting such colossal data is also a challenge for firms and organizations.

Fields of data that come under the umbrella of Big Data: (Generators of Data)

- Black Box Data: Black box data is a type of data that is collected from private and government helicopters, airplanes, and jets. This data includes the capture of Flight Crew Sounds, separate recording of the microphone as well as earphones, etc.

- Stock Exchange Data: Stock exchange data includes various data prepared about 'purchase' and 'selling' of different raw and well-made decisions.

- Social Media Data: This type of data contains information about social media activities that include posts submitted by millions of people worldwide.

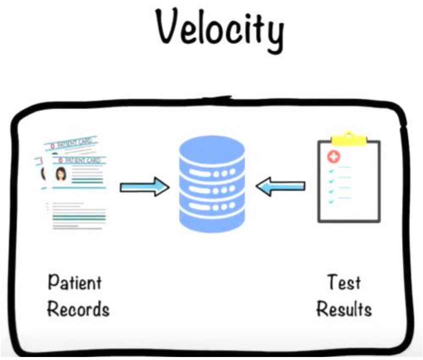
- Transport Data: Transport data includes vehicle models, capacity, distance (from source to destination), and the availability of different vehicles.

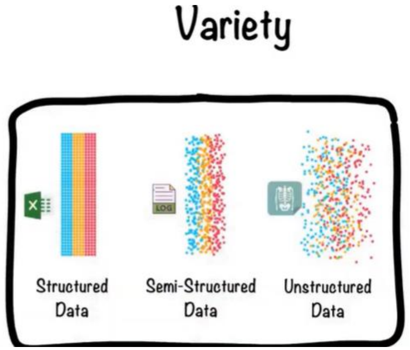
- Search Engine Data: Retrieve a wide variety of unprocessed information that is stored in SE databases.

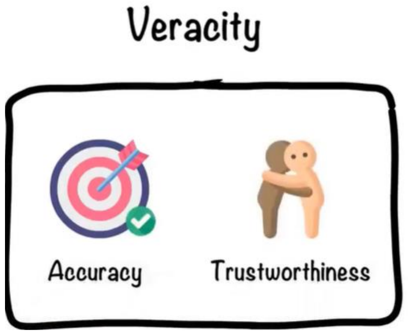
Classify any data as Big Data ?

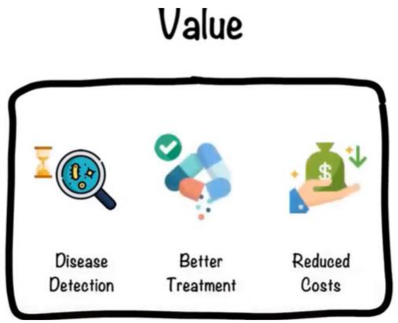
- Based on 5V’s.

Volume
2,314
Exabytes

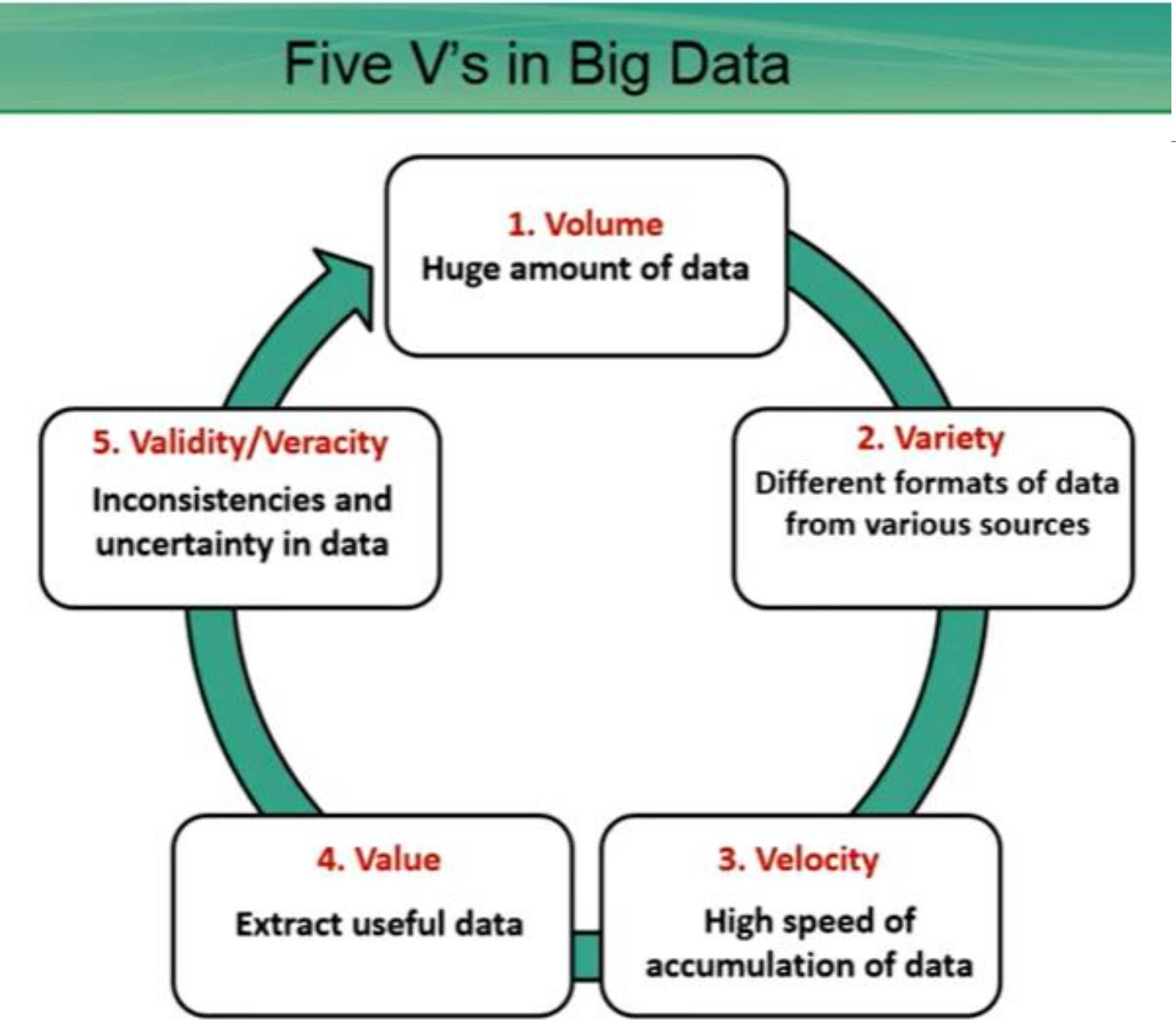




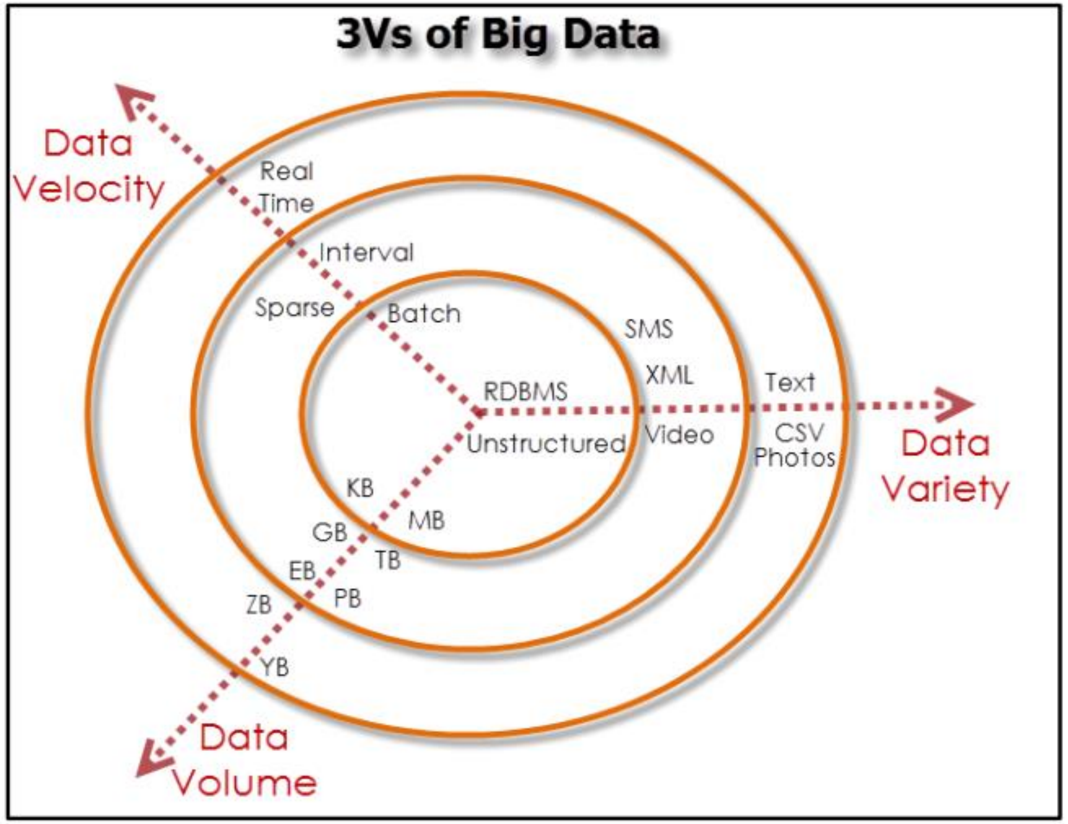




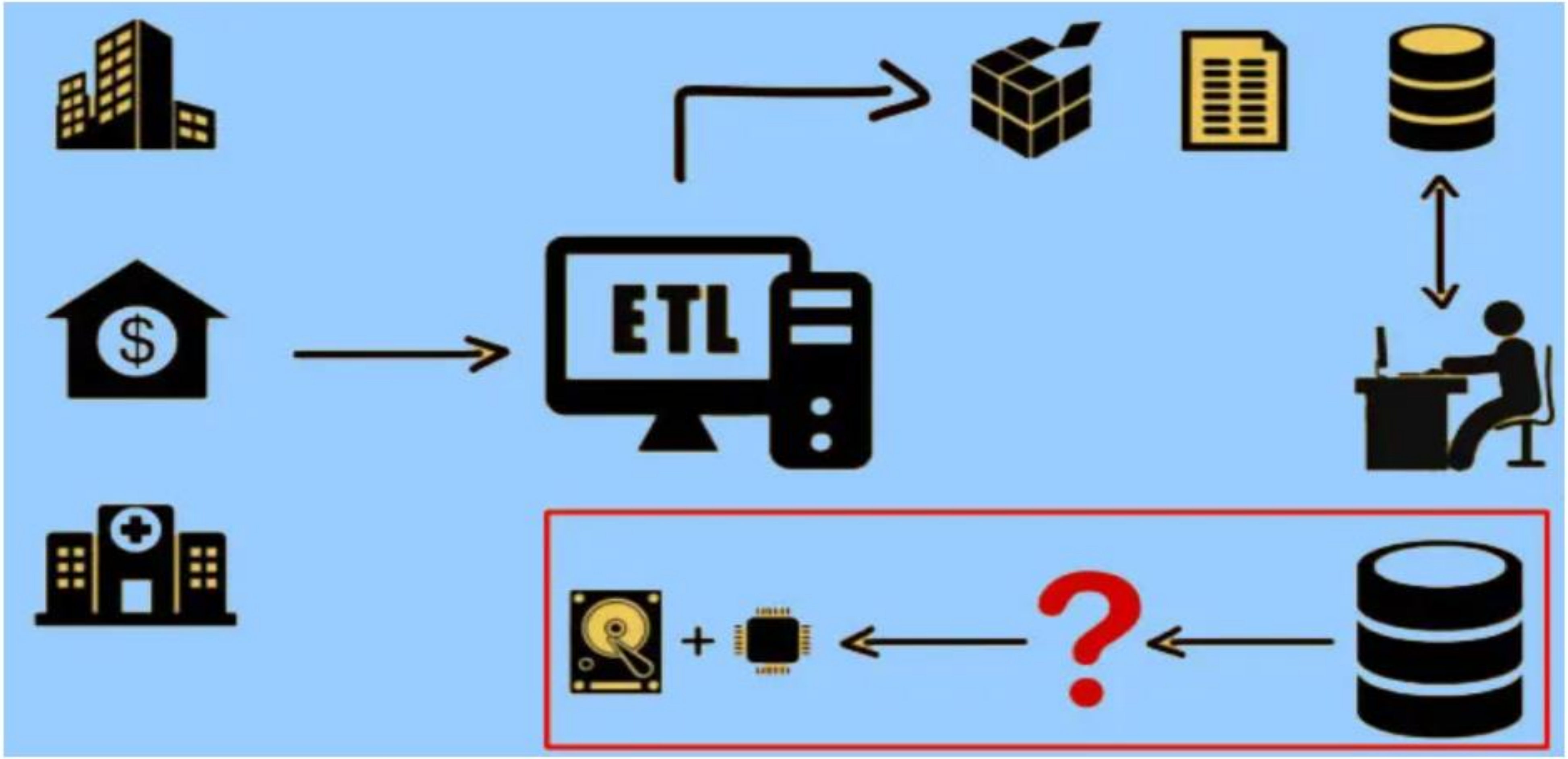
Big Data Characteristics


Five V's in Big Data
1. Volume
Huge amount of data
2. Variety
Different formats of data from various sources
3. Velocity
High speed of accumulation of data
4. Value
Extract useful data
5. Validity/Veracity
Inconsistencies and uncertainty in data

- Volume: Related to size of the data
- Variety: Comprises of a variety of data
- Velocity: Refers to the speed of generation of data.
- Veracity: Quality of data captured, which can vary greatly, affecting its accurate analysis

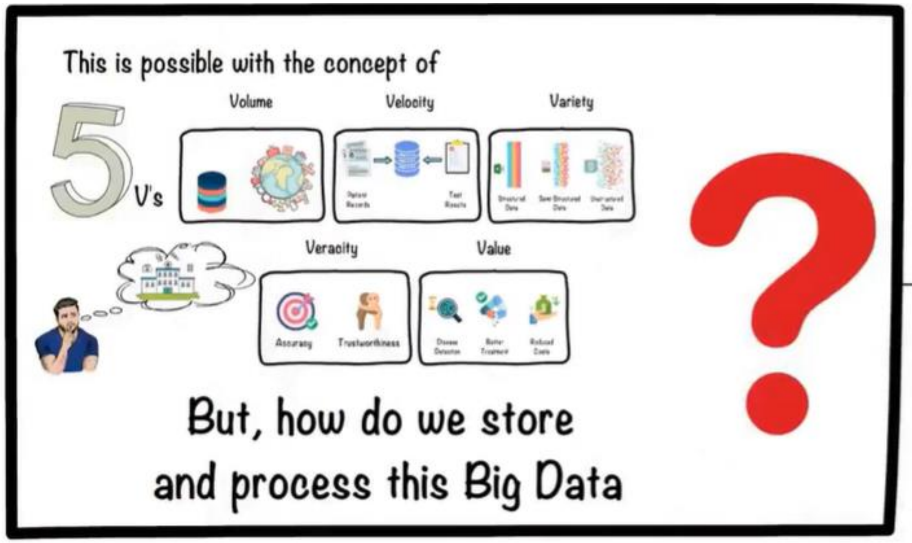


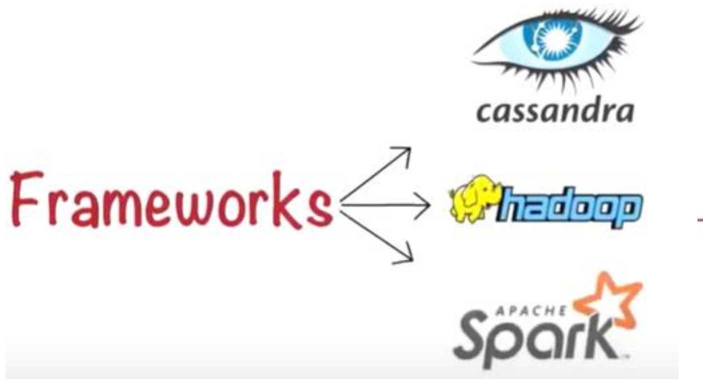
Traditional Approach: Data storage and processing

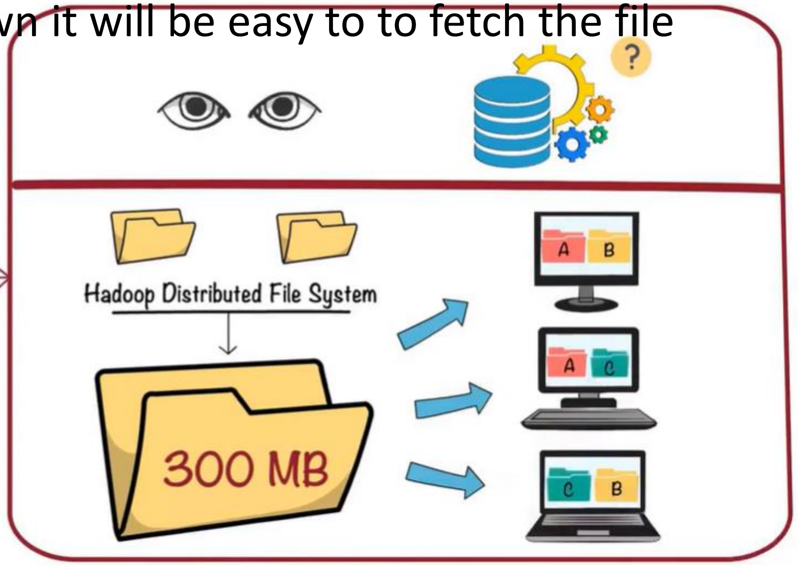


How we store and process this Big Data?

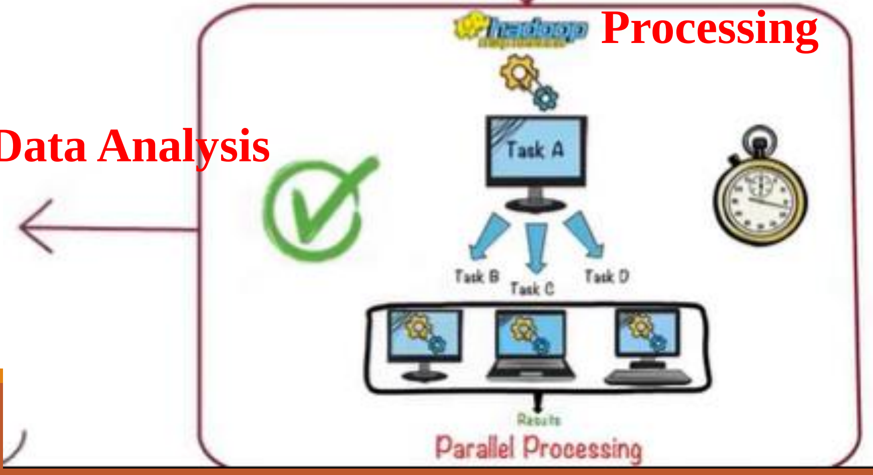
Store – Break file in to small sizes – 300 MB = 128Mb, 128 MB, 44MB
Store them with different nodes – when node breaks down it will be easy to to fetch the file


This is possible with the concept of
5 V's
Volume
Velocity
Variety
Veracity
Value
Accuracy
Trustworthiness
But, how do we store and process this Big Data
?

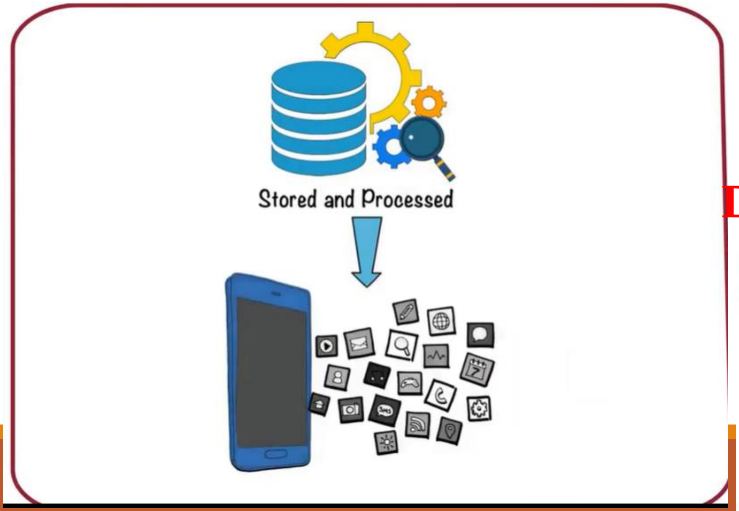




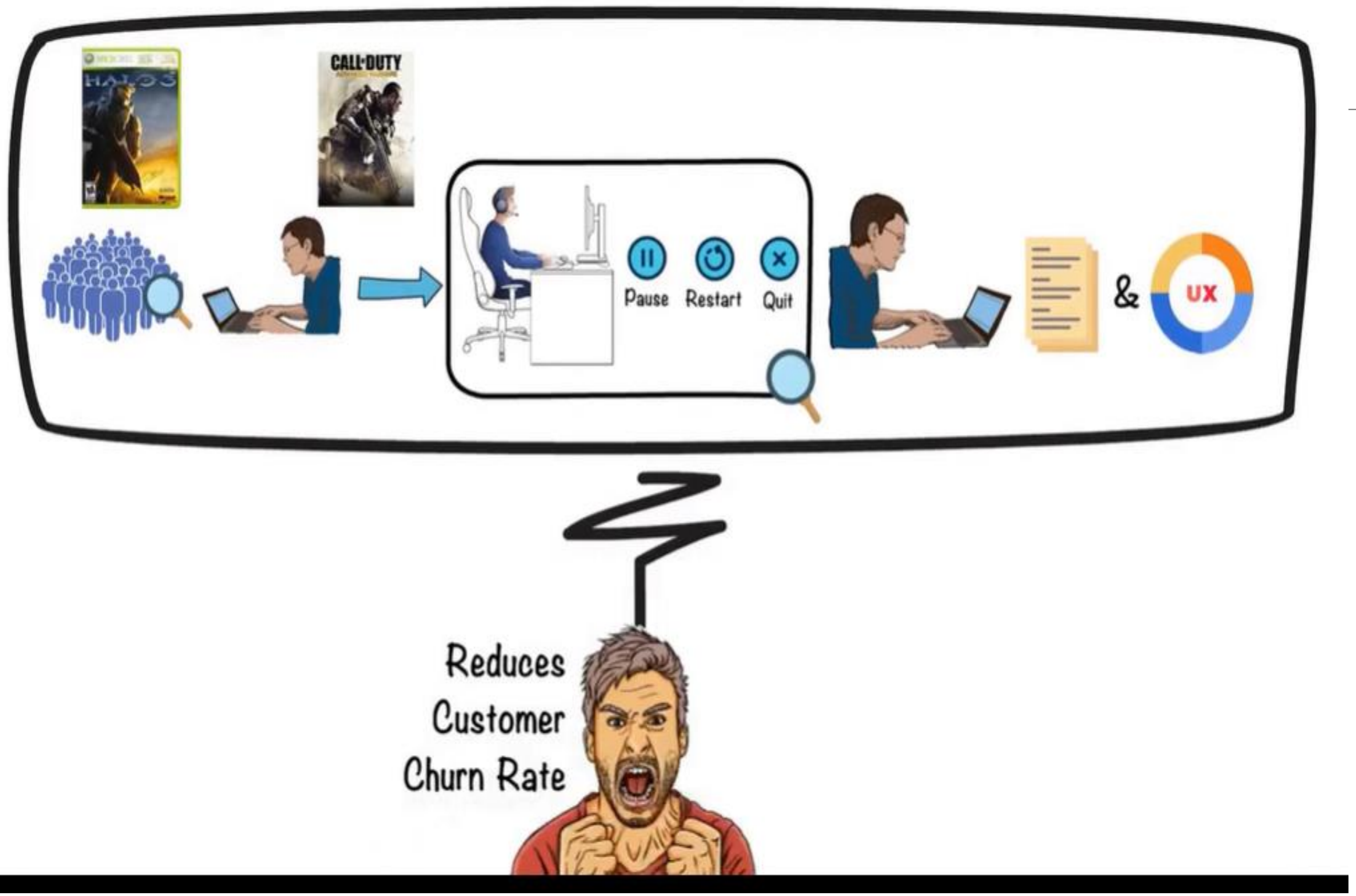
MapReduce/ Processing



Data Analysis



Use case of Big Data Analysis – Gaming



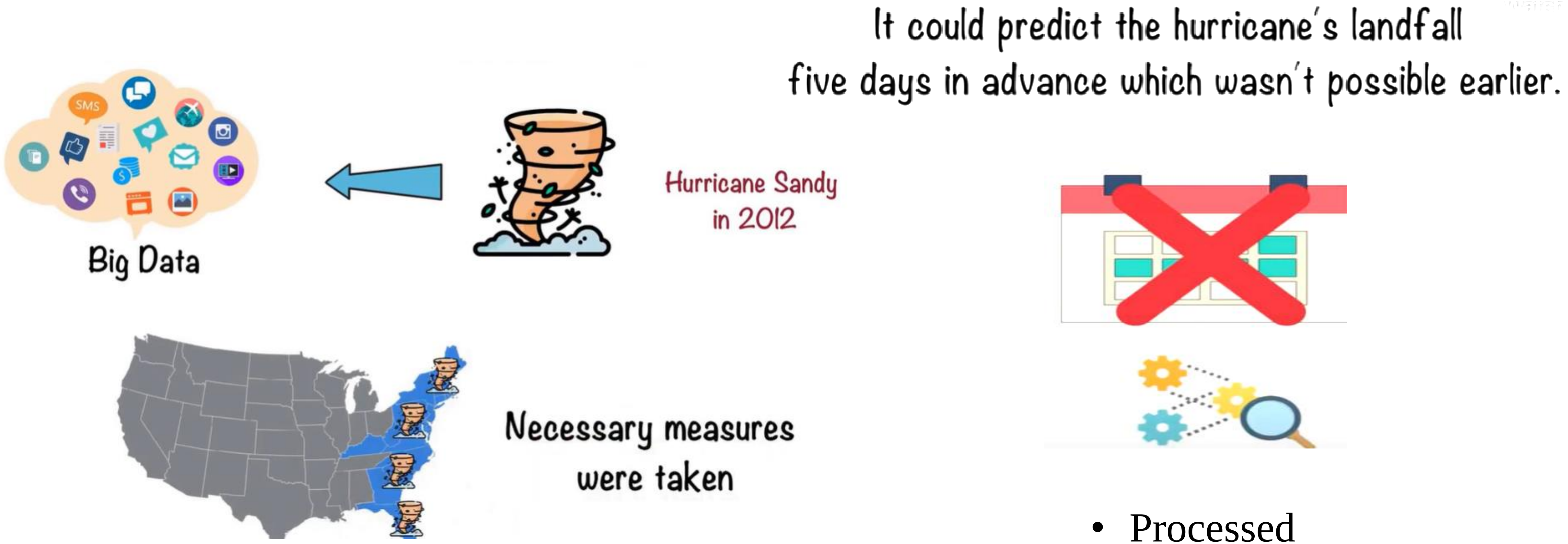
- Designers Analyze gaming data at which stage customers pause , restart , quit etc.

- This analysis will help designers to work on enhancement of story line of games.

- Improve “user Experience”

- Reduce Churn rate.

Use case of Big Data Analysis: Predict Hurricane’s landfall

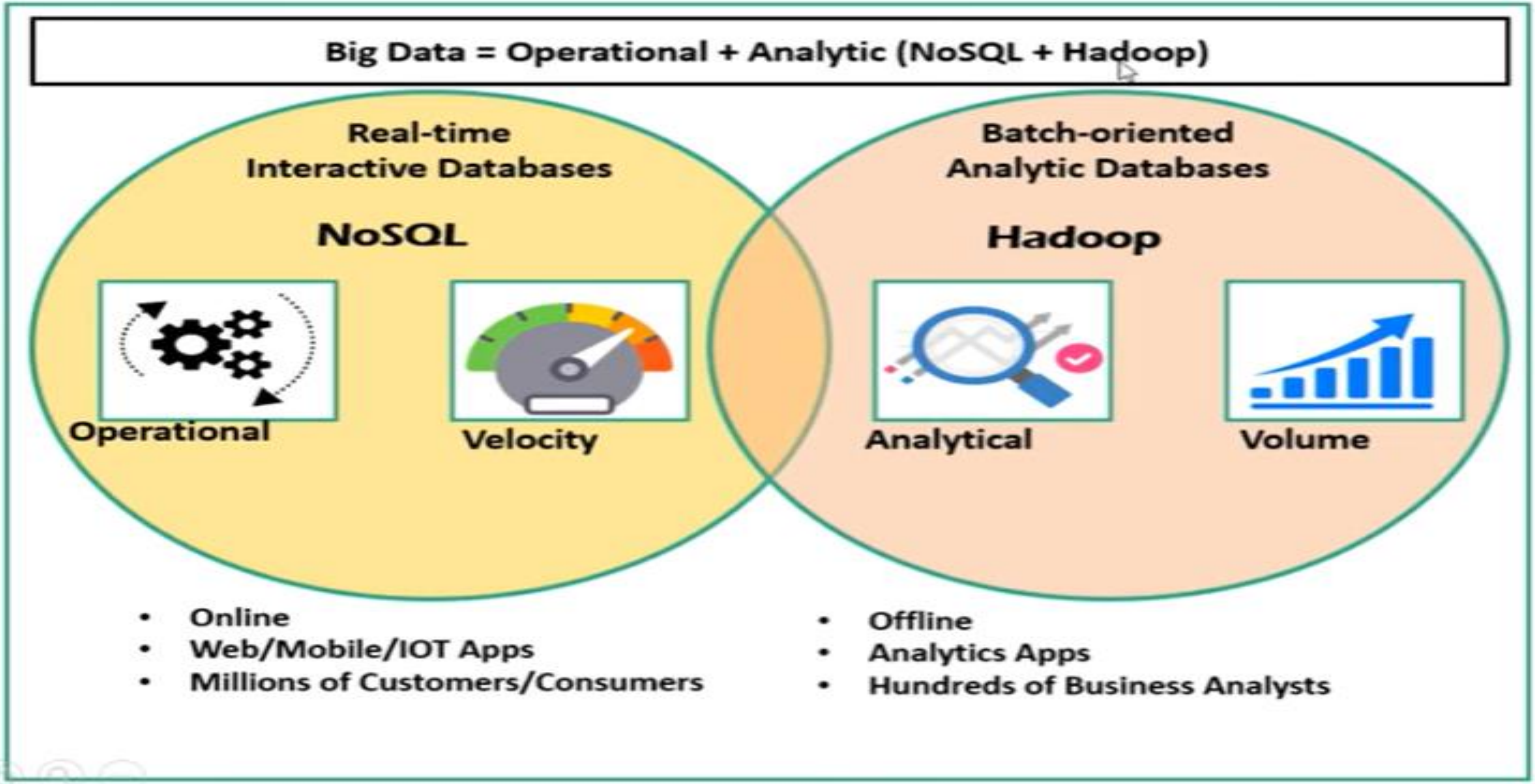


It could predict the hurricane's landfall five days in advance which wasn't possible earlier.

Necessary measures were taken

- Processed
- Analyzed Accurately

Big data Technologies/ Classes



Big Data = Operational + Analytic (NoSQL + Hadoop)

Real-time
Interactive Databases

NoSQL

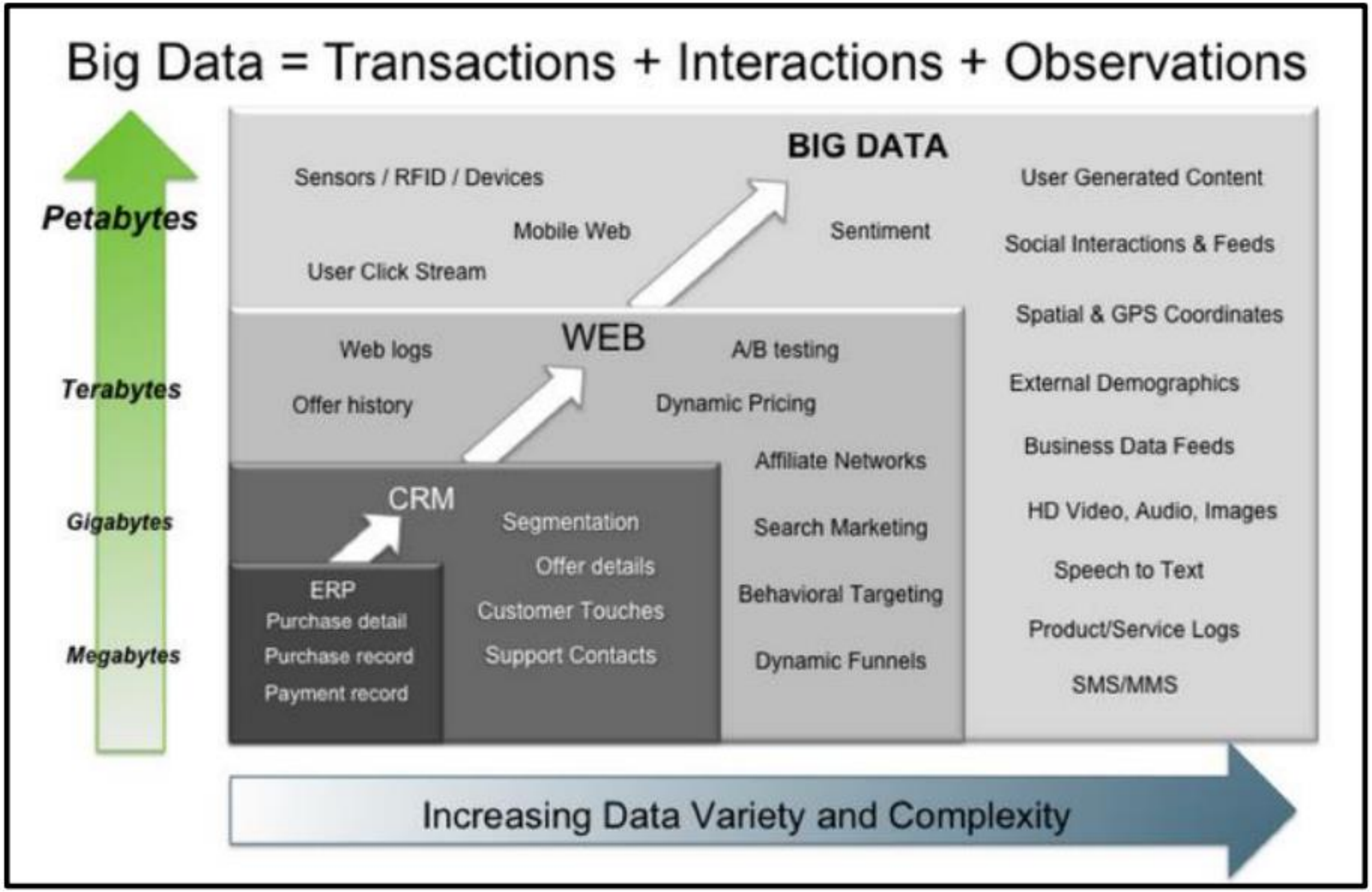
Batch-oriented
Analytic Databases

Hadoop

- Online
- Web/Mobile/IOT Apps
- Millions of Customers/Consumers

- Offline
- Analytics Apps
- Hundreds of Business Analysts

Big Data?


Big Data = Transactions + Interactions + Observations
Petabytes
Terabytes
Gigabytes
Megabytes
BIG DATA
Sensors / RFID / Devices
Mobile Web
User Click Stream
Sentiment
User Generated Content
Social Interactions & Feeds
Spatial & GPS Coordinates
External Demographics
Business Data Feeds
HD Video, Audio, Images
Speech to Text
Product/Service Logs
SMS/MMS
WEB
Web logs
Offer history
A/B testing
Dynamic Pricing
Affiliate Networks
Search Marketing
Behavioral Targeting
Dynamic Funnels
CRM
Segmentation
Offer details
Customer Touches
Support Contacts
ERP
Purchase detail
Purchase record
Payment record
Increasing Data Variety and Complexity

Big Data Layout

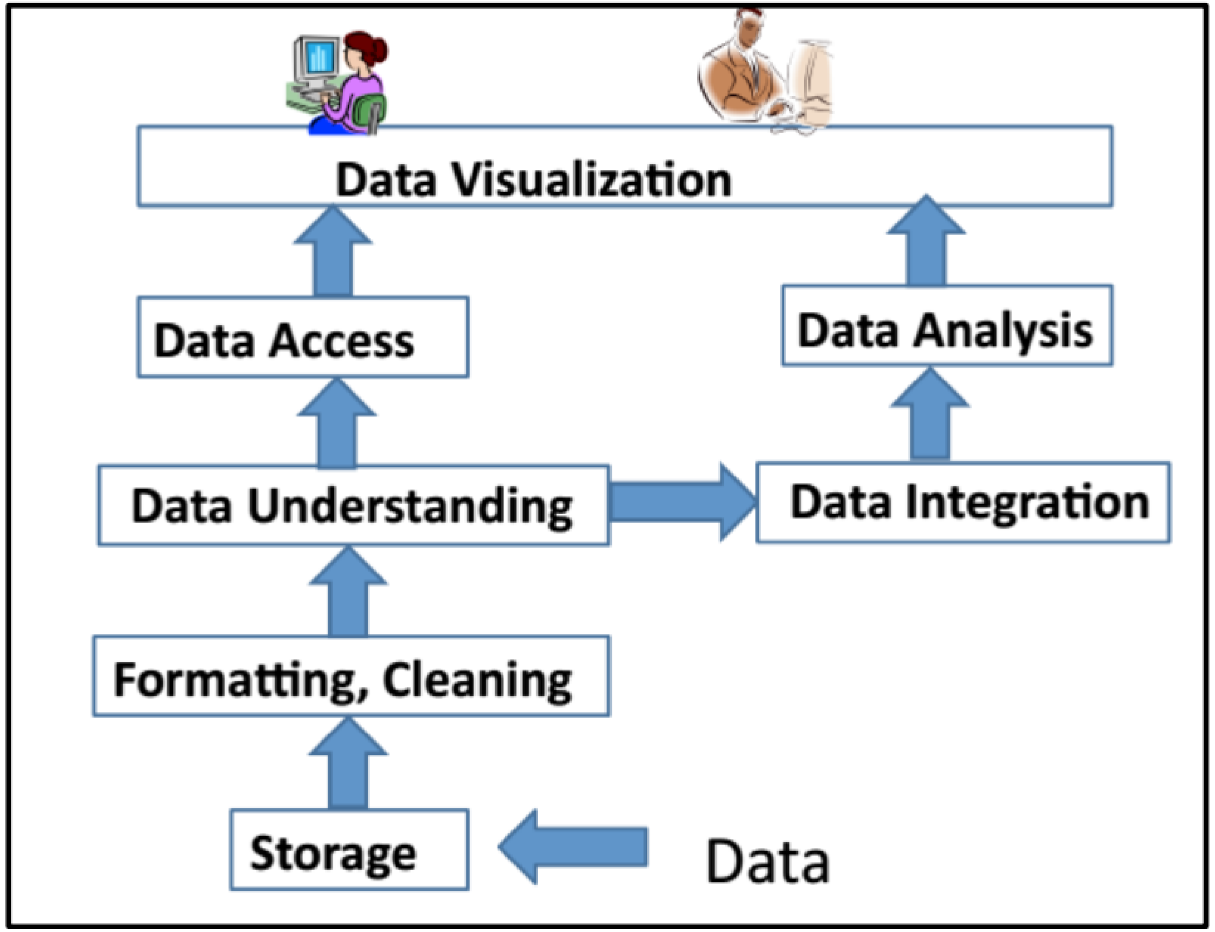


Figure : Big Data layout

- Apache Hadoop- Apache
- MapReduce – Google
- HDFS(Hadoop distributed file system) – Apache
- Hive – Facebook
- Pig - Yahoo

Apache Hadoop

- Apache Hadoop is one of the main supportive element in Big Data technologies.
- It simplifies the processing of large amount of structured or unstructured data in a cheap manner.
- Hadoop is an open source project from Apache that is continuously improving over the years.
- Hadoop is basically a set of software libraries and frameworks to manage and process big amount of data from a single server to thousands of machines.
- It provides an efficient and powerful error detection mechanism based on application layer rather than relying upon hardware."
- In December 2012 Apache releases Hadoop 1.0.0, more information and installation guide can be found at Apache Hadoop Documentation.
- Hadoop is not a single project but includes a number of other technologies in it.

Hadoop

Main Components

1. HDFS (Hadoop Distributed File System)
2. YARN (Yet Another Resource Negotiator)
3. Mapreduce

Includes several additional modules: Hive ,Pig, and **HBase**

key features

- Distributed Storage
- Scalability
- Fault-Tolerance
- Data locality
- High Availability
- Flexible Data Processing
- Data Integrity
- Data Compression

HDFS(Hadoop distributed file system)

- HDFS is a java based file system that is used to store structured or unstructured data over large clusters of distributed servers.
- The data stored in HDFS has no restriction or rule to be applied, the data can be either fully unstructured or purely structured.
- In HDFS the work to make data senseful, is done by developer's code only.
- Hadoop distributed file system provides a highly fault tolerant atmosphere with a deployment on low cost hardware machines.
- HDFS is now a part of Apache Hadoop project, more information and installation guide can be found at Apache HDFS documentation

MapReduce

- MapReduce was introduced by google, which has mapper and reducer modules.
- Helps to analyze web logs to create large amount of web search indexes.
- It is basically a framework to write applications that processes a large amount of structured or unstructured data over the web. (programming model)
- MapReduce takes the query and breaks it into parts to run it on multiple nodes.
- By distributed query processing it makes it easy to maintain large amount of data by dividing the data into several different machines.
- Hadoop MapReduce is a software framework for easily writing applications to manage large amount of data sets with a highly fault tolerant manner.
- More tutorials and getting started guide can be found at Apache Documentation.

HIVE

- Hive was originally developed by Facebook, now it is made open source.
- Hive works something like a bridge in between SQL and Hadoop, it is basically used to make SQL queries on Hadoop clusters.
- Apache Hive is basically a data warehouse that provides ad-hoc queries, data summarization and analysis of huge data sets stored in Hadoop compatible file systems.
- Hive provides a SQL like called HiveQL query based implementation of huge amount of data stored in Hadoop clusters.
- In January 2013 apache releases Hive 0.10.0

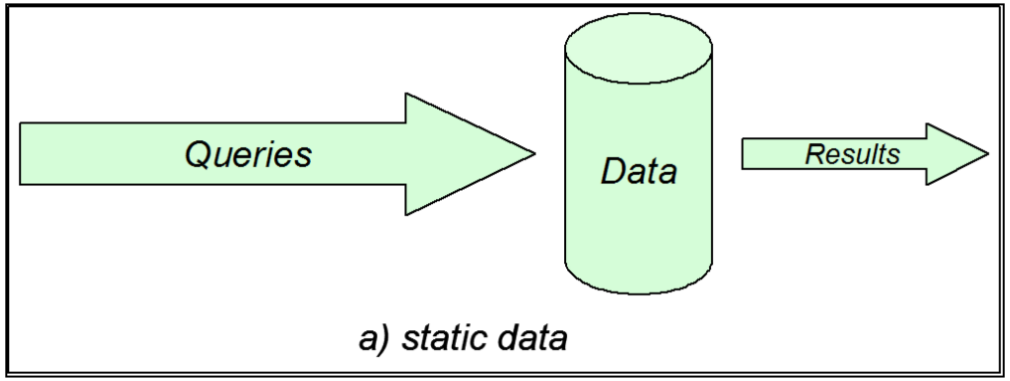
Pig

- Pig was introduced by yahoo and later on it was made fully open source.
- It also provides a bridge to query data over Hadoop clusters but unlike hive, it implements a script implementation to make Hadoop data access able by developers and business persons.
- Apache pig provides a high level programming platform for developers to process and analyses Big Data using user defined functions and programming efforts.
- In January 2013 Apache released Pig 0.10.1 which is defined for use with Hadoop 0.10.1 or later releases.

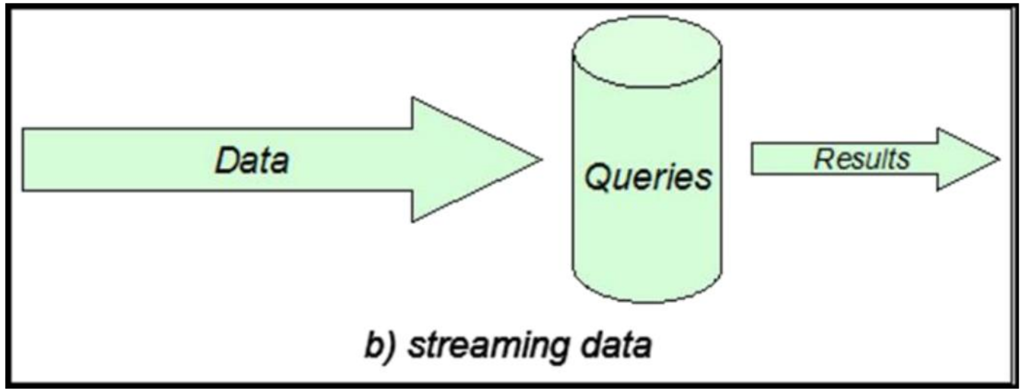
Traditional vs Big Data

Schema Hard Code/SQL/Small Data

- online transactions and quick updates.
- Schema Based DB (hard code schema attachment)
- Structured
- Uses SQL for Data processing
- Maintains relationship between elements



a) static data



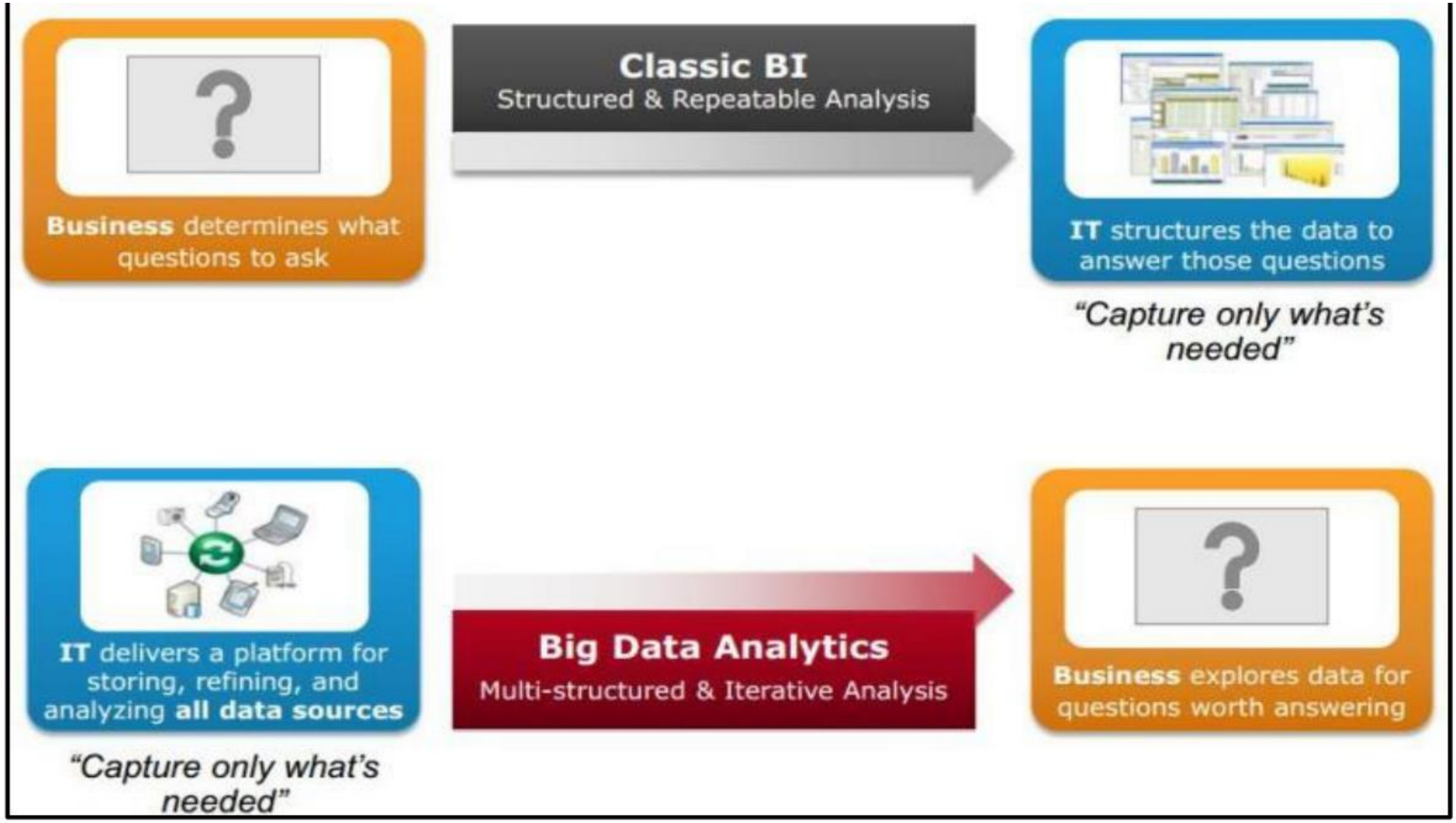
b) streaming data

Schema Less /NoSQL/Big Data

- Migration Easy
- Schema Less Based DB
- store unstructured, semi structured or even fully structured data
- Store a huge amount of data and not to maintain relationship between elements.

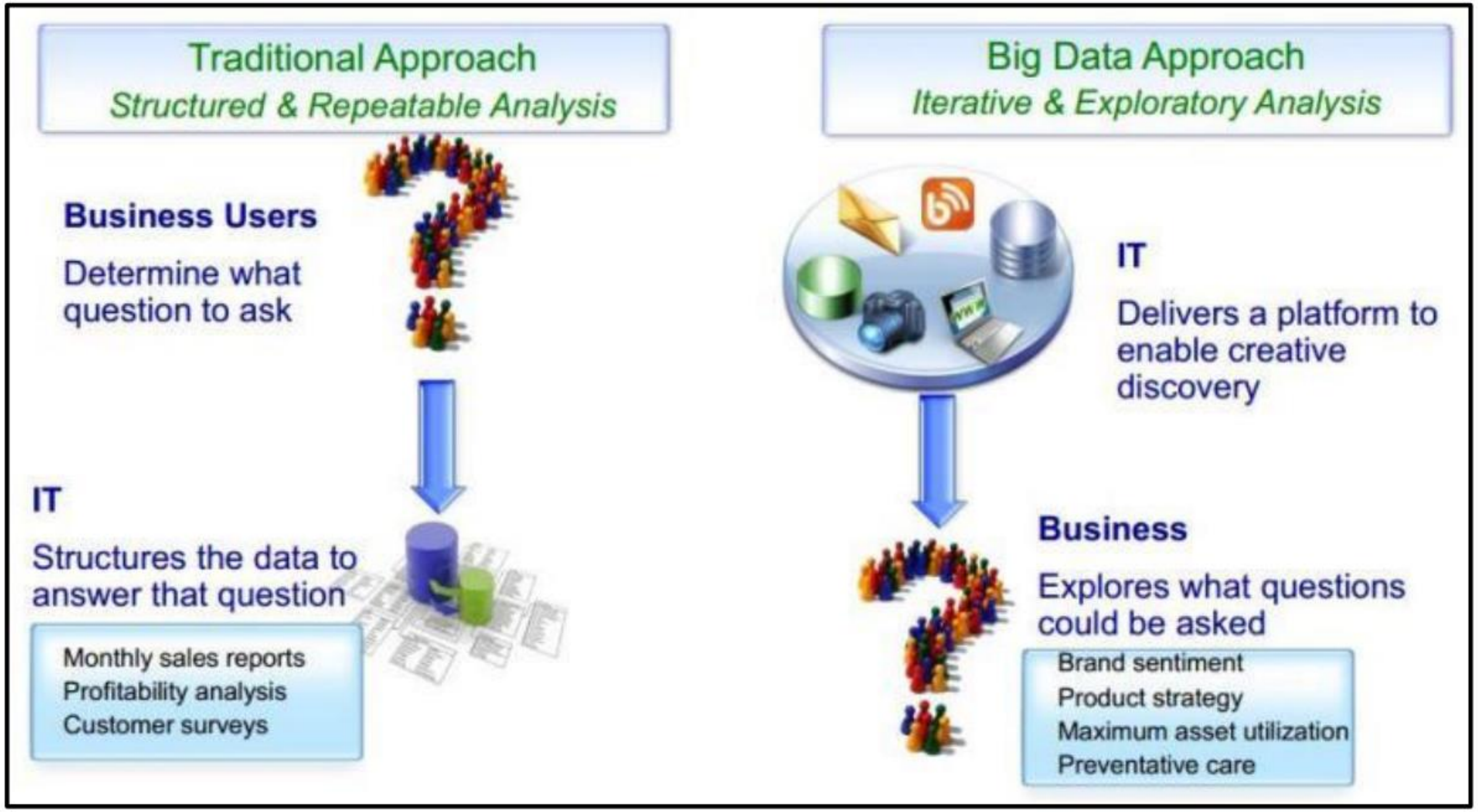
Traditional BI Vs. Big Data Analytics

- Working on the live coming data, which can be an input from the ever-changing scenario cannot be dealt in the traditional approach.


Business determines what questions to ask
Classic BI
Structured & Repeatable Analysis
IT structures the data to answer those questions
"Capture only what's needed"
IT delivers a platform for storing, refining, and analyzing all data sources
"Capture only what's needed"
Big Data Analytics
Multi-structured & Iterative Analysis
Business explores data for questions worth answering

- The live flow of data is captured and the analysis is done on it.
- Efficiency increases when the data to be analyzed is large

Traditional vs Big data Approaches



Traditional Approach

Structured & Repeatable Analysis

Business Users

Determine what question to ask

IT

Structures the data to answer that question

- Monthly sales reports
- Profitability analysis
- Customer surveys

Big Data Approach

Iterative & Exploratory Analysis

IT

Delivers a platform to enable creative discovery

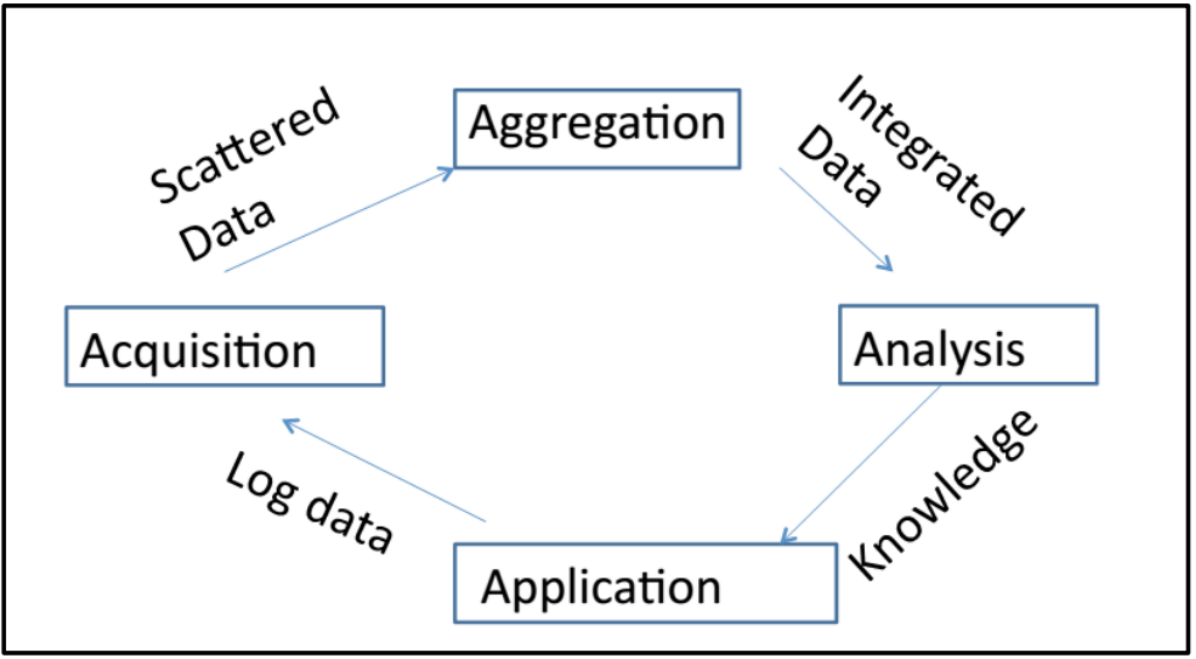
Business

Explores what questions could be asked

- Brand sentiment
- Product strategy
- Maximum asset utilization
- Preventative care

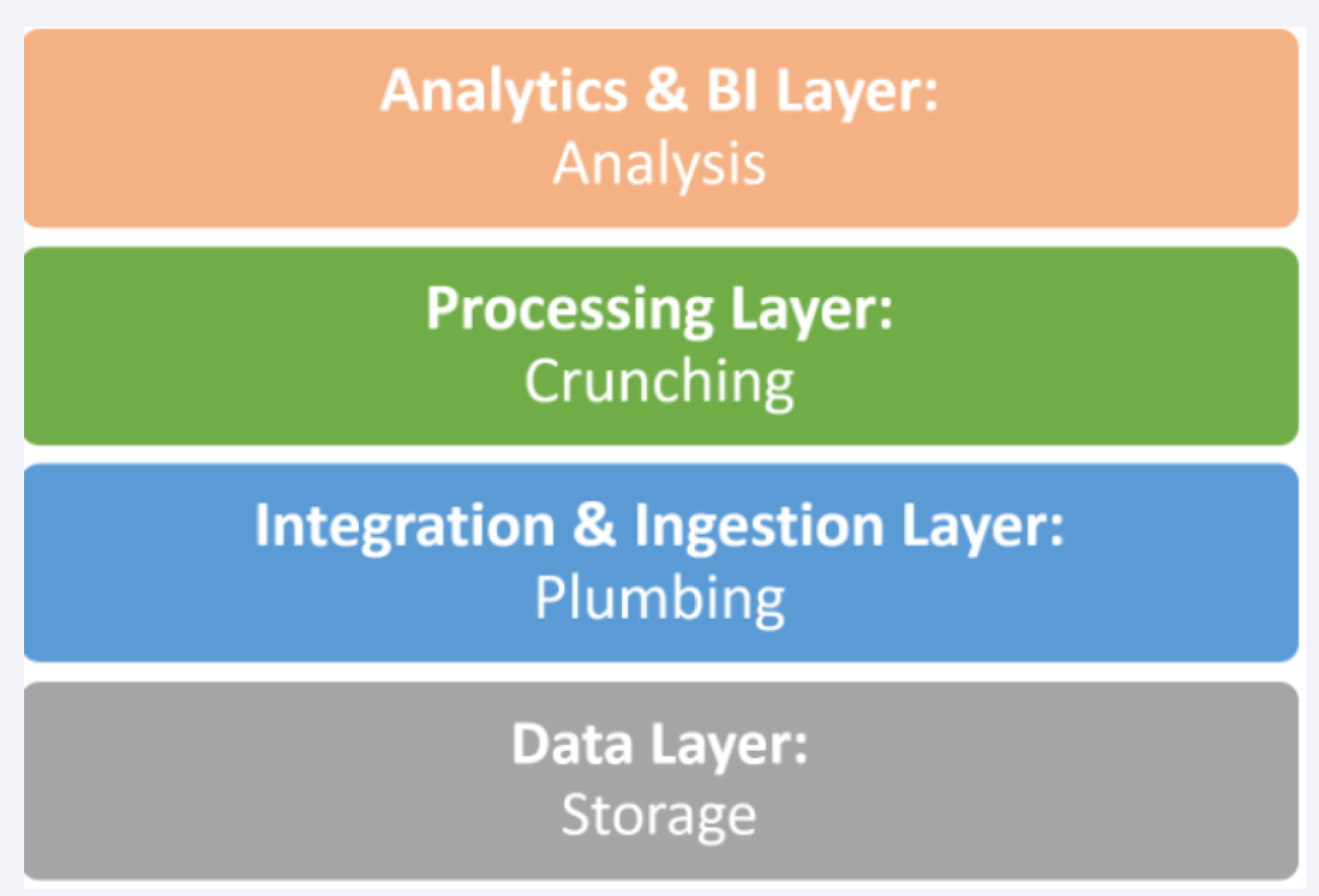
What is Changing in the Realms
of
Big Data ??????

Life Cycle of Data



1) The analysis of data is done from the knowledge experts and the expertise is applied for the development of an application.
2) The streaming of data after the analysis and the application, the data log is created for the acquisition of data.
3) The data id mapped and clustered together on the data log.
4) The clustered data from the data acquisition is then aggregated by applying various aggregation algorithms.
5) The integrated data again goes for an analysis.
6) The complete steps are repeated till the desired, and expected output is produced

Big Data Technology Stack



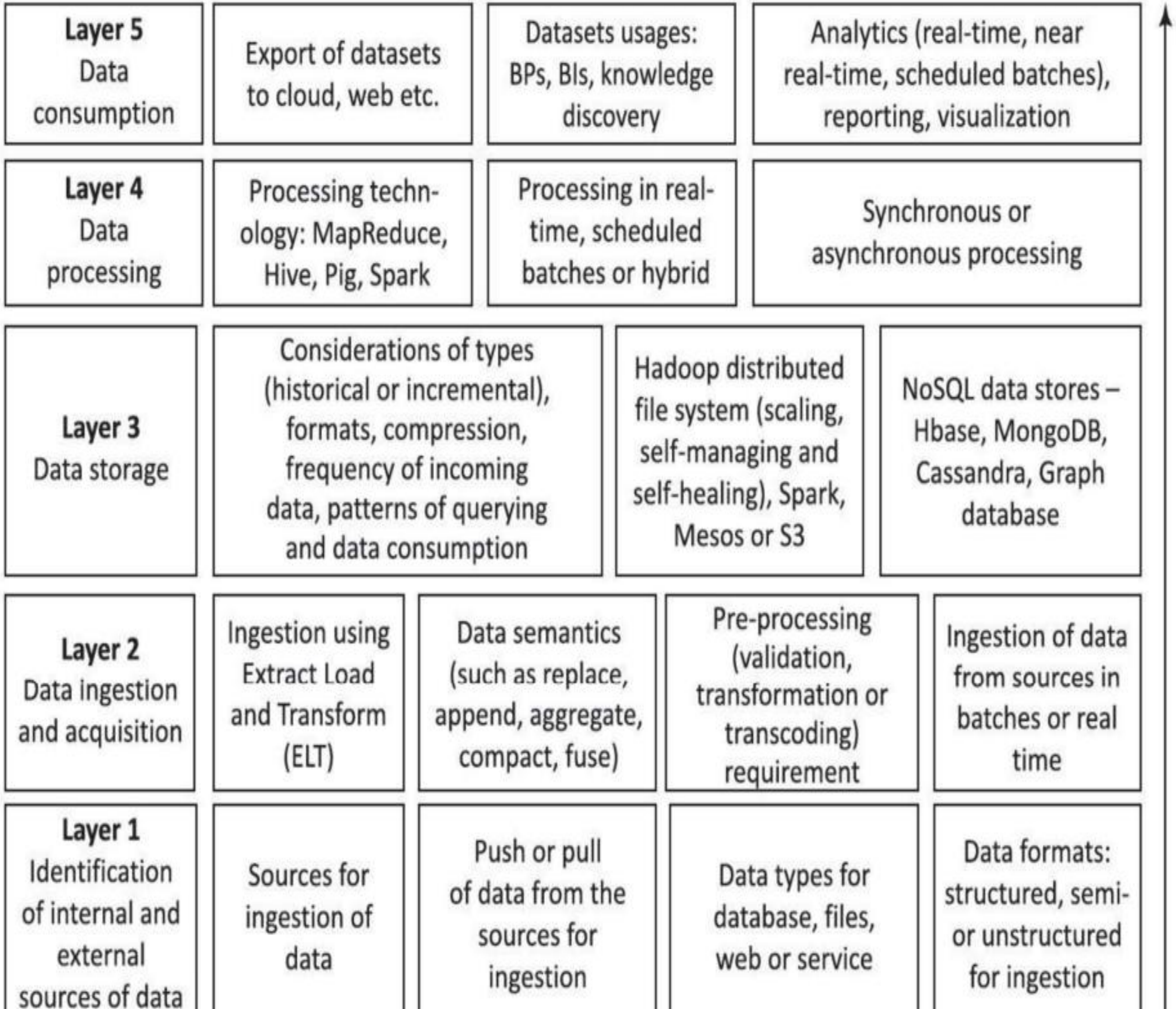
Analytics & BI Layer:
Analysis

Processing Layer:
Crunching

Integration & Ingestion Layer:
Plumbing

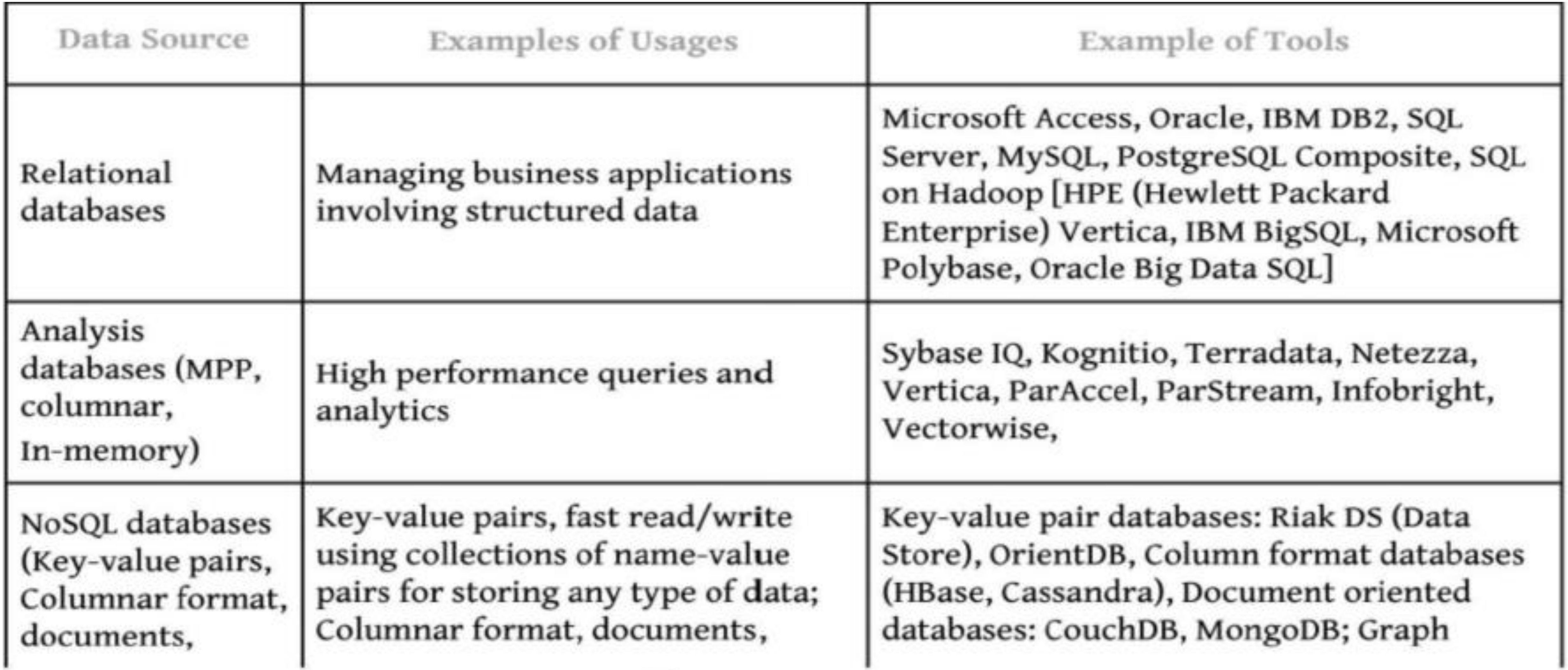
Data Layer:
Storage

Design of logical layers in a data processing



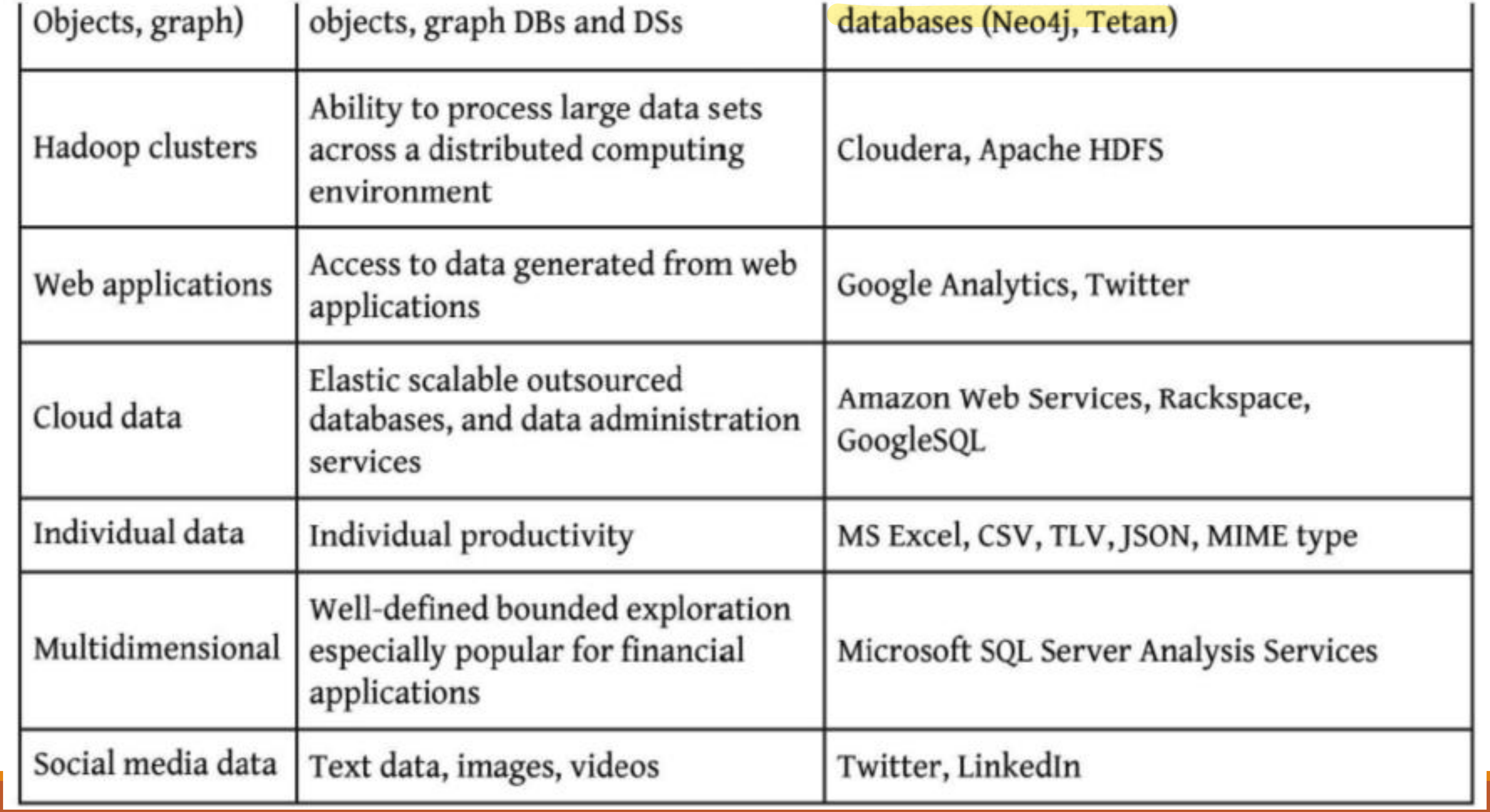
Layer 5 Data consumption	Export of datasets to cloud, web etc.	Datasets usages: BPs, BIs, knowledge discovery	Analytics (real-time, near real-time, scheduled batches), reporting, visualization	
Layer 4 Data processing	Processing techn-ology: MapReduce, Hive, Pig, Spark	Processing in real-time, scheduled batches or hybrid	Synchronous or asynchronous processing	
Layer 3 Data storage	Considerations of types (historical or incremental), formats, compression, frequency of incoming data, patterns of querying and data consumption	Hadoop distributed file system (scaling, self-managing and self-healing), Spark, Mesos or S3	NoSQL data stores – Hbase, MongoDB, Cassandra, Graph database	
Layer 2 Data ingestion and acquisition	Ingestion using Extract Load and Transform (ELT)	Data semantics (such as replace, append, aggregate, compact, fuse)	Pre-processing (validation, transformation or transcoding) requirement	Ingestion of data from sources in batches or real time
Layer 1 Identification of internal and external sources of data	Sources for ingestion of data	Push or pull of data from the sources for ingestion	Data types for database, files, web or service	Data formats: structured, semi- or unstructured for ingestion

Various Data Storage and Usage, Tools



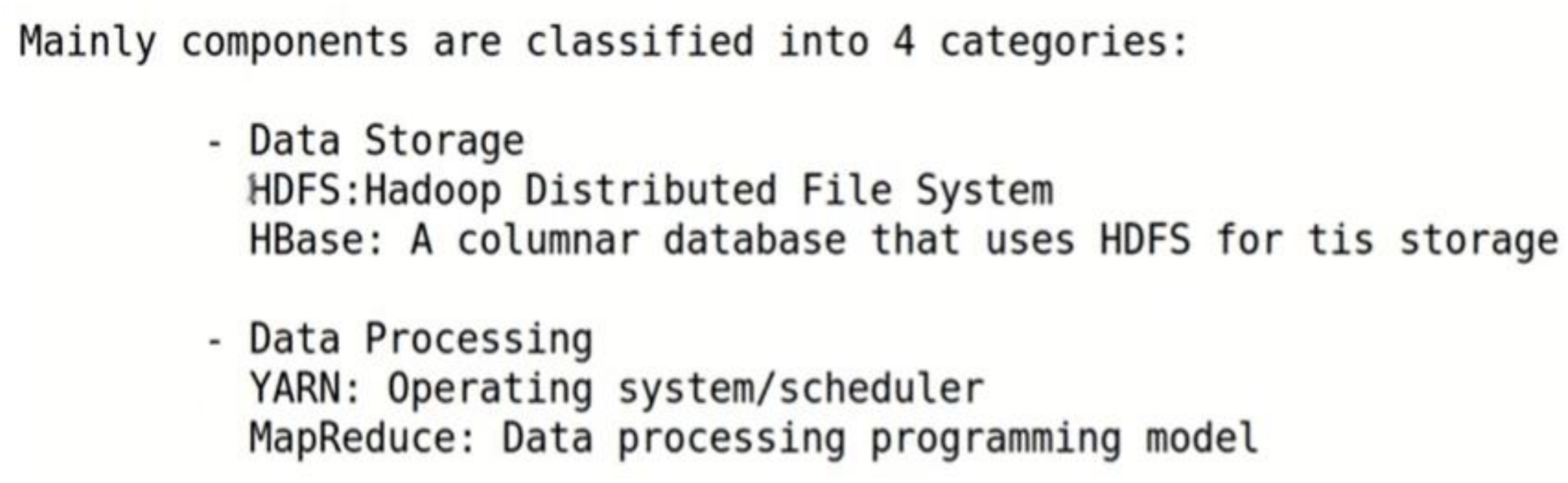
Data Source	Examples of Usages	Example of Tools
Relational databases	Managing business applications involving structured data	Microsoft Access, Oracle, IBM DB2, SQL Server, MySQL, PostgreSQL Composite, SQL on Hadoop [HPE (Hewlett Packard Enterprise) Vertica, IBM BigSQL, Microsoft Polybase, Oracle Big Data SQL]
Analysis databases (MPP, columnar, In-memory)	High performance queries and analytics	Sybase IQ, Kognitio, Terradata, Netezza, Vertica, ParAccel, ParStream, Infobright, Vectorwise,
NoSQL databases (Key-value pairs, Columnar format, documents,	Key-value pairs, fast read/write using collections of name-value pairs for storing any type of data; Columnar format, documents,	Key-value pair databases: Riak DS (Data Store), OrientDB, Column format databases (HBase, Cassandra), Document oriented databases: CouchDB, MongoDB; Graph

Various Data Storage and Usage, Tools



Objects, graph)	objects, graph DBs and DSs	databases (Neo4j, Tetan)
Hadoop clusters	Ability to process large data sets across a distributed computing environment	Cloudera, Apache HDFS
Web applications	Access to data generated from web applications	Google Analytics, Twitter
Cloud data	Elastic scalable outsourced databases, and data administration services	Amazon Web Services, Rackspace, GoogleSQL
Individual data	Individual productivity	MS Excel, CSV, TLV, JSON, MIME type
Multidimensional	Well-defined bounded exploration especially popular for financial applications	Microsoft SQL Server Analysis Services
Social media data	Text data, images, videos	Twitter, LinkedIn

Components Classification in Hadoop

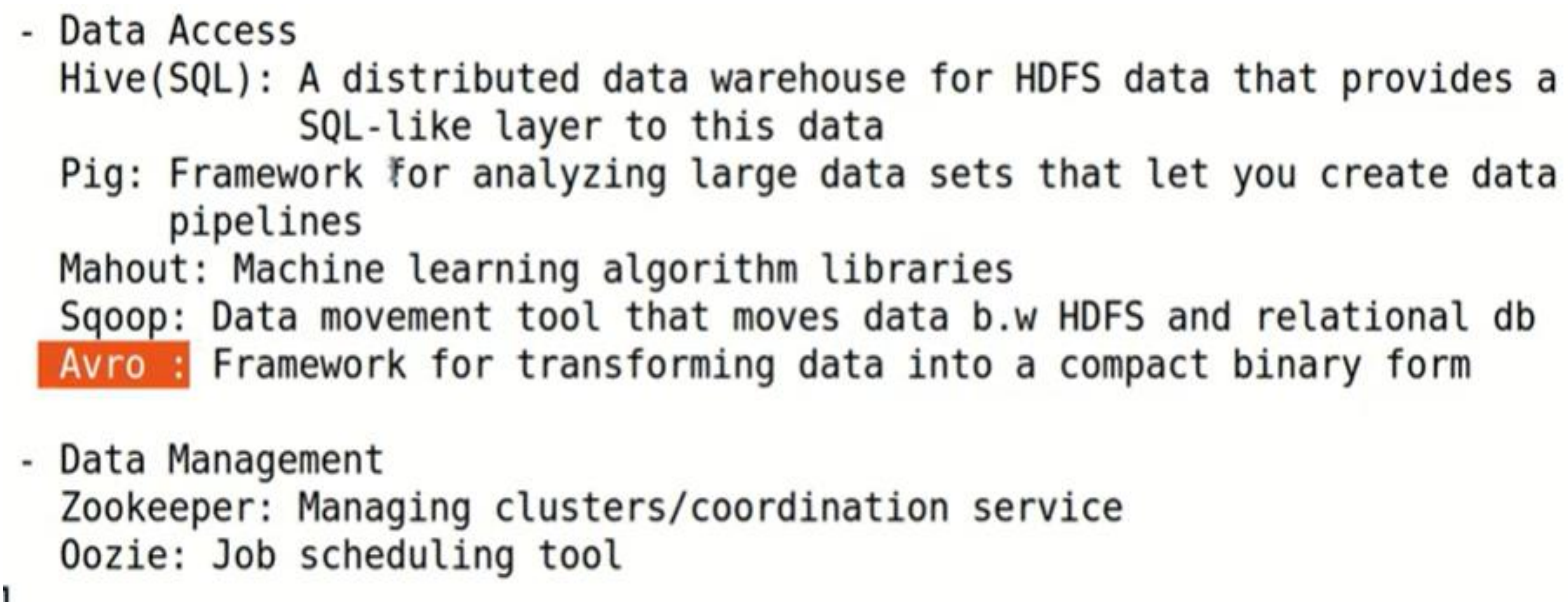


Mainly components are classified into 4 categories:

- Data Storage
  HDFS:Hadoop Distributed File System
  HBase: A columnar database that uses HDFS for tis storage

- Data Processing
  YARN: Operating system/scheduler
  MapReduce: Data processing programming model

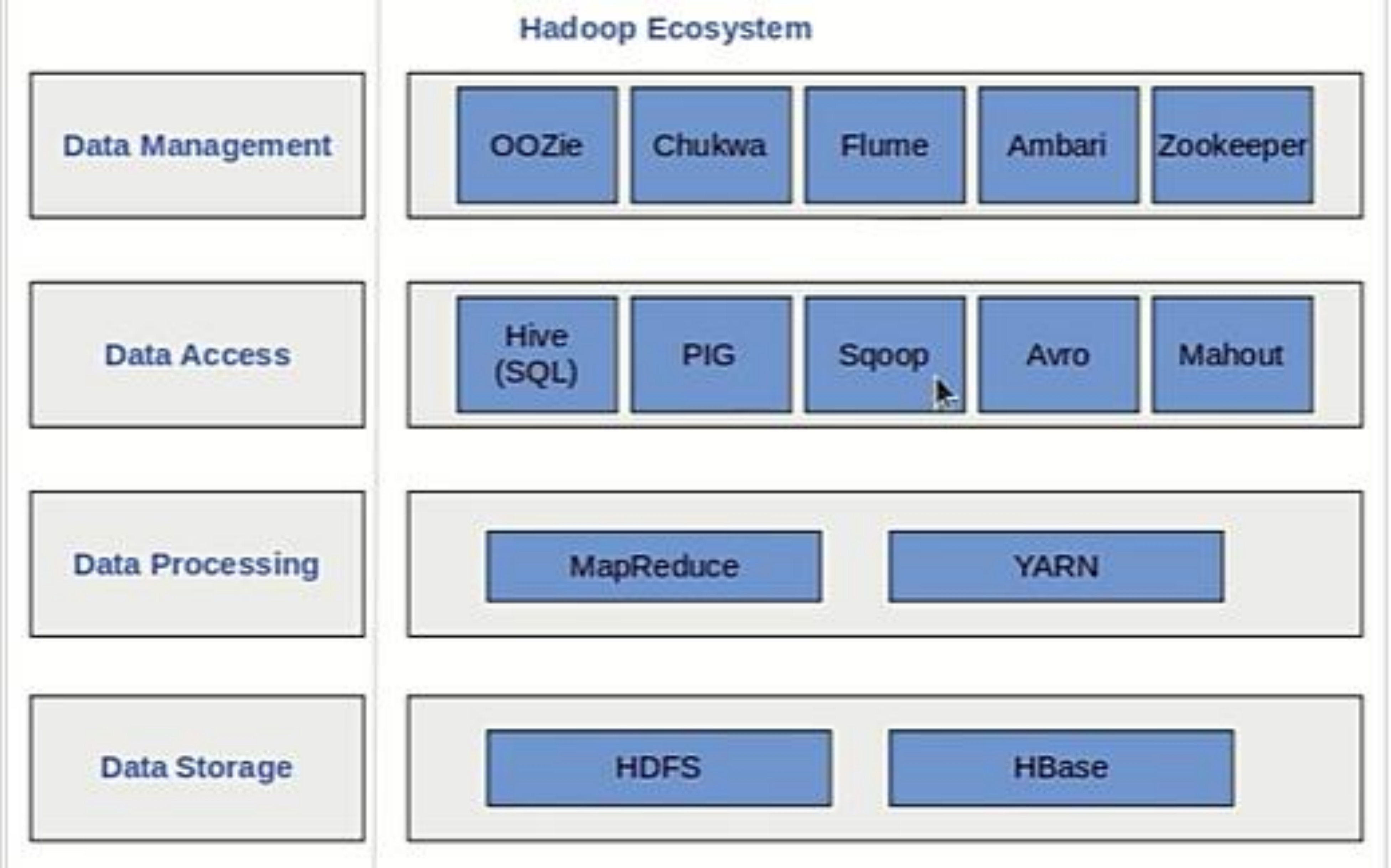
Components Classification



```
- Data Access
  Hive(SQL): A distributed data warehouse for HDFS data that provides a
             SQL-like layer to this data
  Pig: Framework for analyzing large data sets that let you create data
       pipelines
  Mahout: Machine learning algorithm libraries
  Sqoop: Data movement tool that moves data b.w HDFS and relational db
  Avro : Framework for transforming data into a compact binary form

- Data Management
  Zookeeper: Managing clusters/coordination service
  Oozie: Job scheduling tool
```

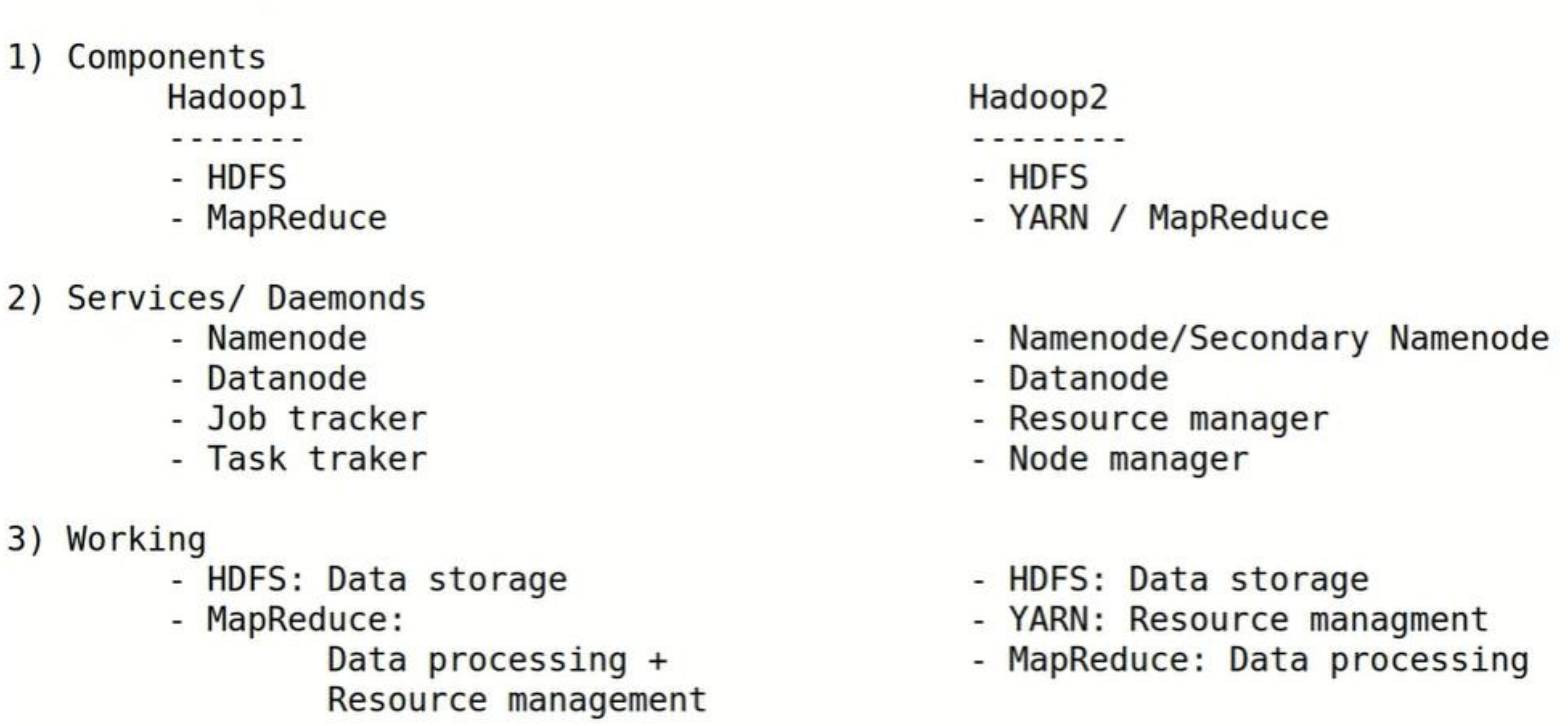
Hadoop Ecosystem



Hadoop Ecosystem

Data Management	OOZie	Chukwa	Flume	Ambari	Zookeeper
Data Access	Hive (SQL)	PIG	Sqoop	Avro	Mahout
Data Processing	MapReduce		YARN		
Data Storage	HDFS		HBase		

Hadoop 1 vs Hadoop 2 [MRV1 vs MRV2]

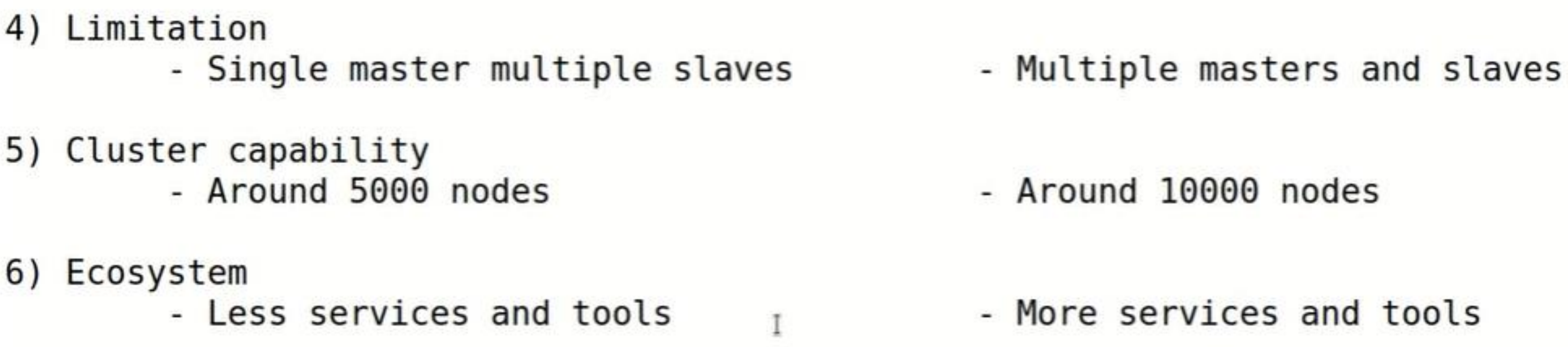


```
1) Components
        Hadoop1                                 Hadoop2
        -------                                 --------
        - HDFS                                  - HDFS
        - MapReduce                             - YARN / MapReduce

2) Services/ Daemonds
        - Namenode                              - Namenode/Secondary Namenode
        - Datanode                              - Datanode
        - Job tracker                           - Resource manager
        - Task traker                           - Node manager

3) Working
        - HDFS: Data storage                    - HDFS: Data storage
        - MapReduce:                            - YARN: Resource managment
                Data processing +               - MapReduce: Data processing
                Resource management
```

Hadoop 1 vs Hadoop 2 [MRV1 vs MRV2]

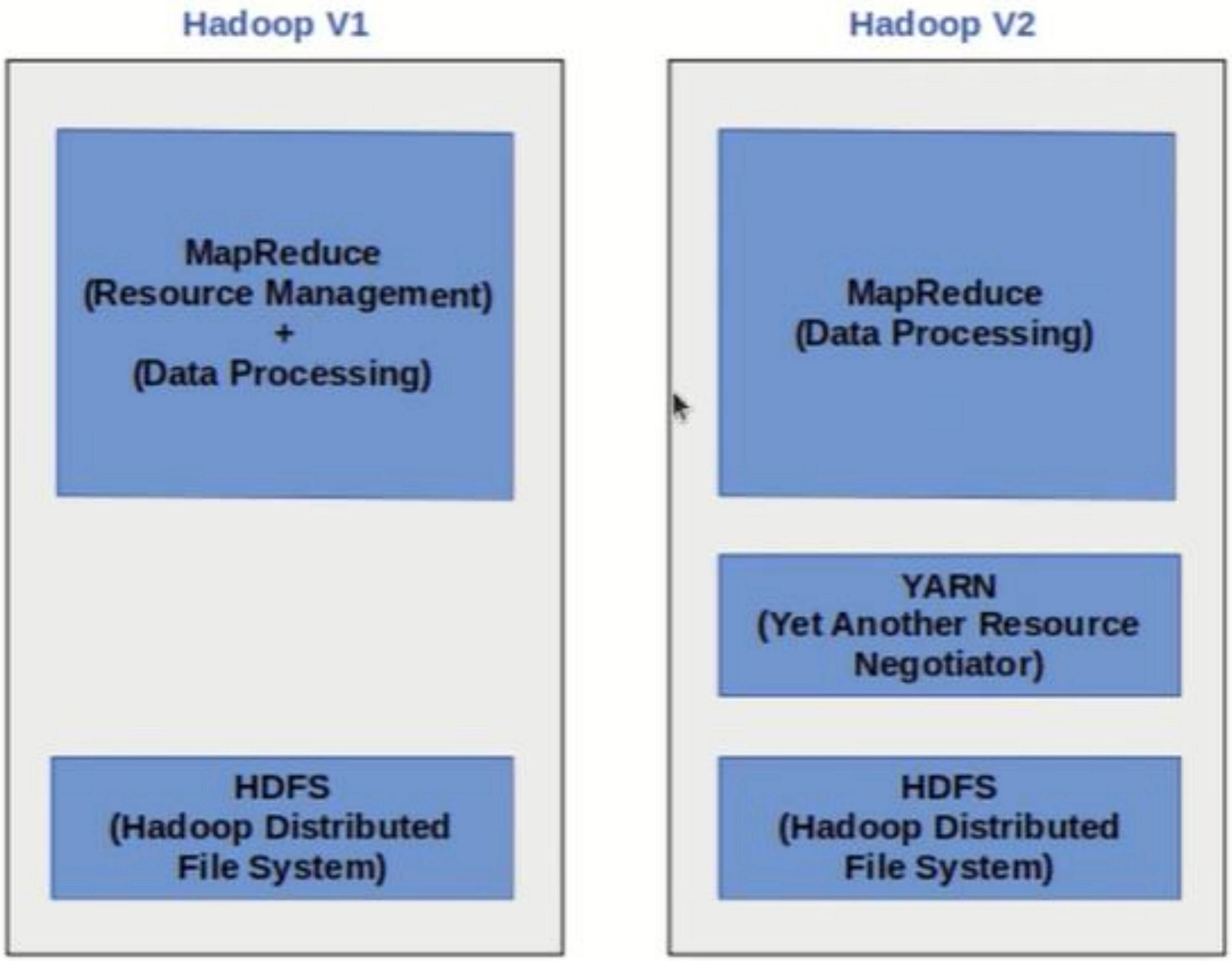


```
4) Limitation
        - Single master multiple slaves          - Multiple masters and slaves

5) Cluster capability
        - Around 5000 nodes                      - Around 10000 nodes

6) Ecosystem
        - Less services and tools                - More services and tools
```

Hadoop V1 vs Hadoop V2



Hadoop V1

MapReduce
(Resource Management)
+
(Data Processing)

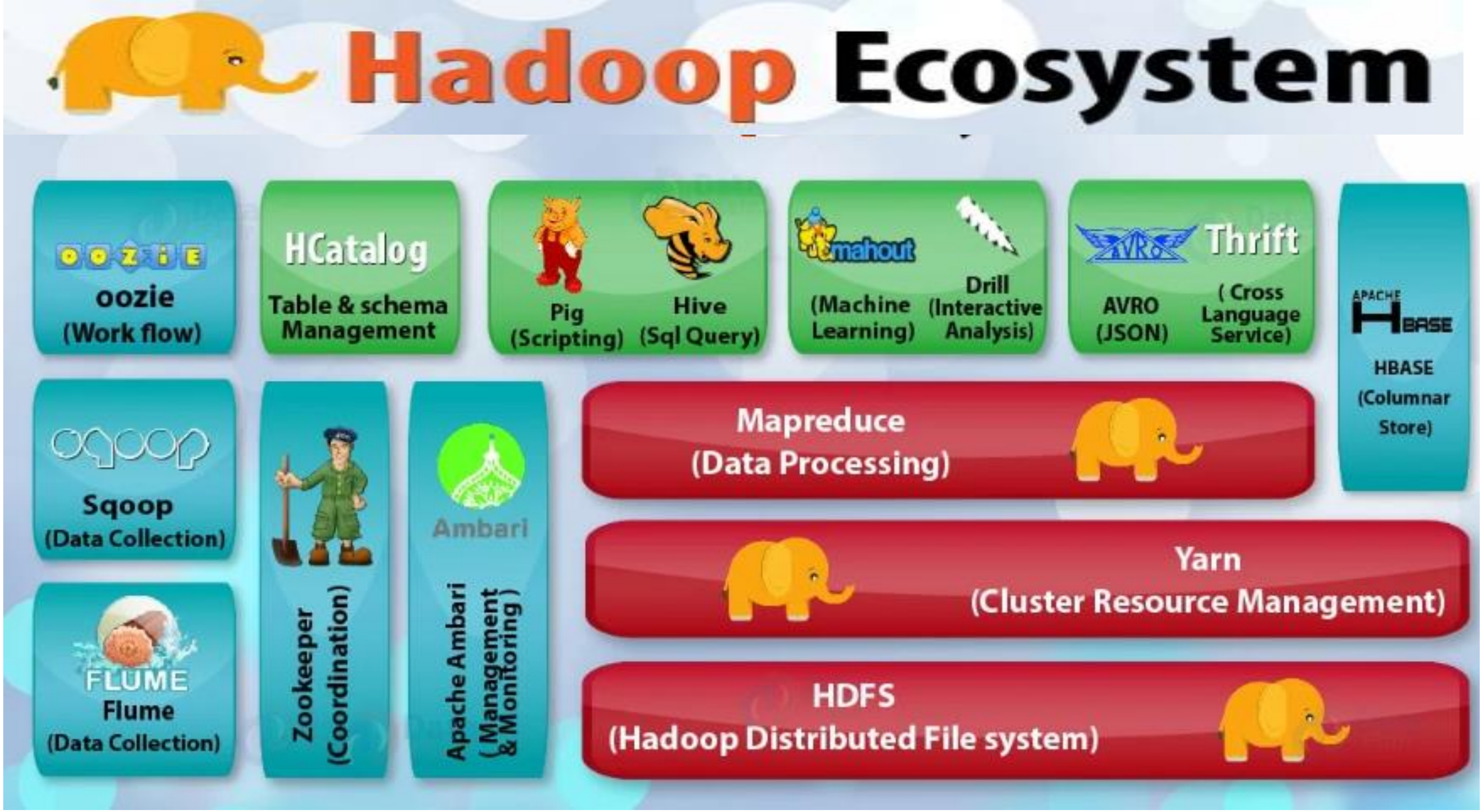
HDFS
(Hadoop Distributed
File System)

Hadoop V2

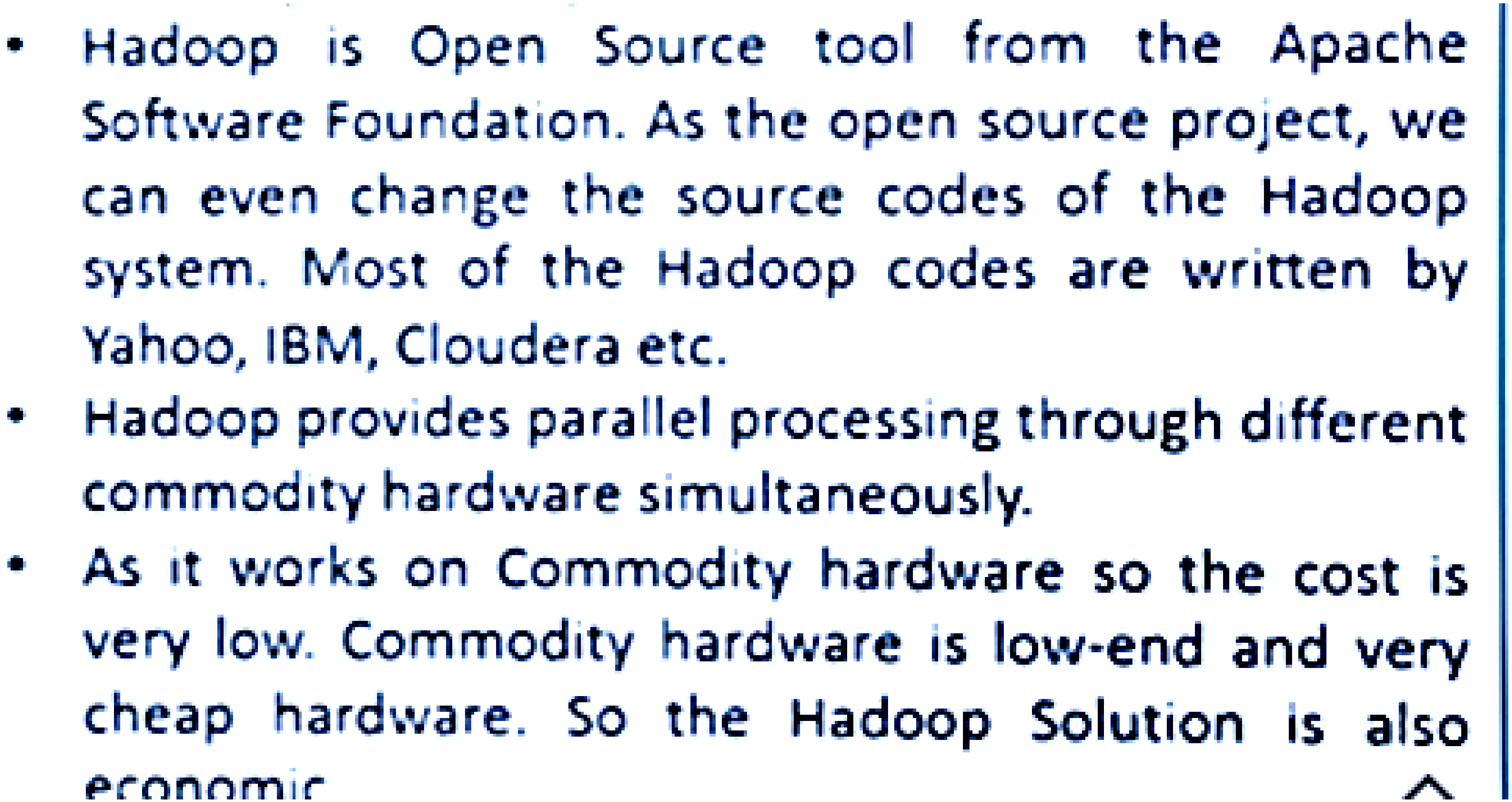
MapReduce
(Data Processing)

YARN
(Yet Another Resource
Negotiator)

HDFS
(Hadoop Distributed
File System)

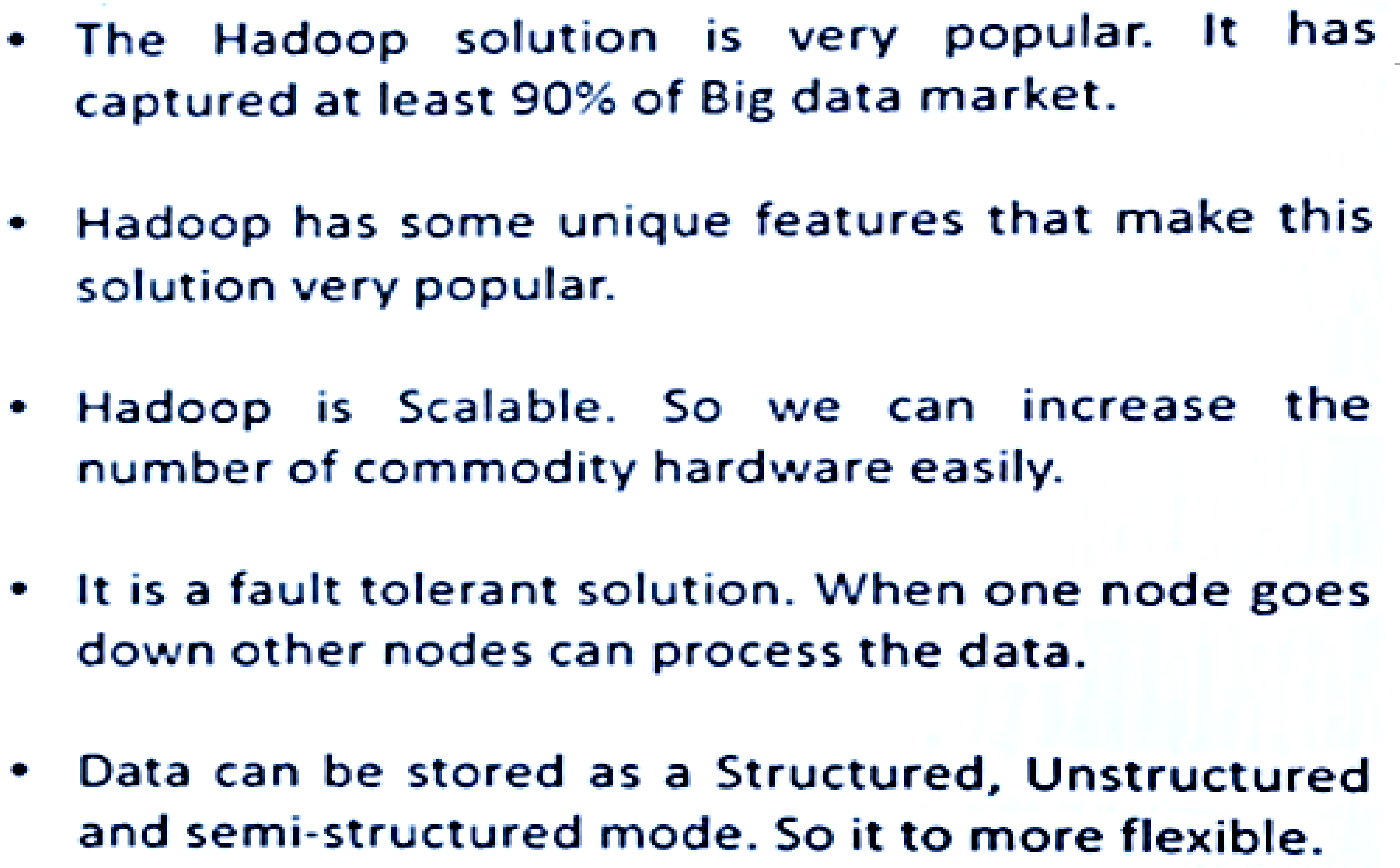

Hadoop Ecosystem
oozie
oozie
(Work flow)
HCatalog
Table & schema
Management
Pig
(Scripting)
Hive
(Sql Query)
mahout
(Machine
Learning)
Drill
(Interactive
Analysis)
AVRO
Thrift
AVRO
(JSON)
(Cross
Language
Service)
APACHE HBASE
HBASE
(Columnar
Store)
sqoop
Sqoop
(Data Collection)
Zookeeper
(Coordination)
Ambari
Apache Ambari
(Management
& Monitoring)
Mapreduce
(Data Processing)
Yarn
(Cluster Resource Management)
FLUME
Flume
(Data Collection)
HDFS
(Hadoop Distributed File system)

Hadoop Technology



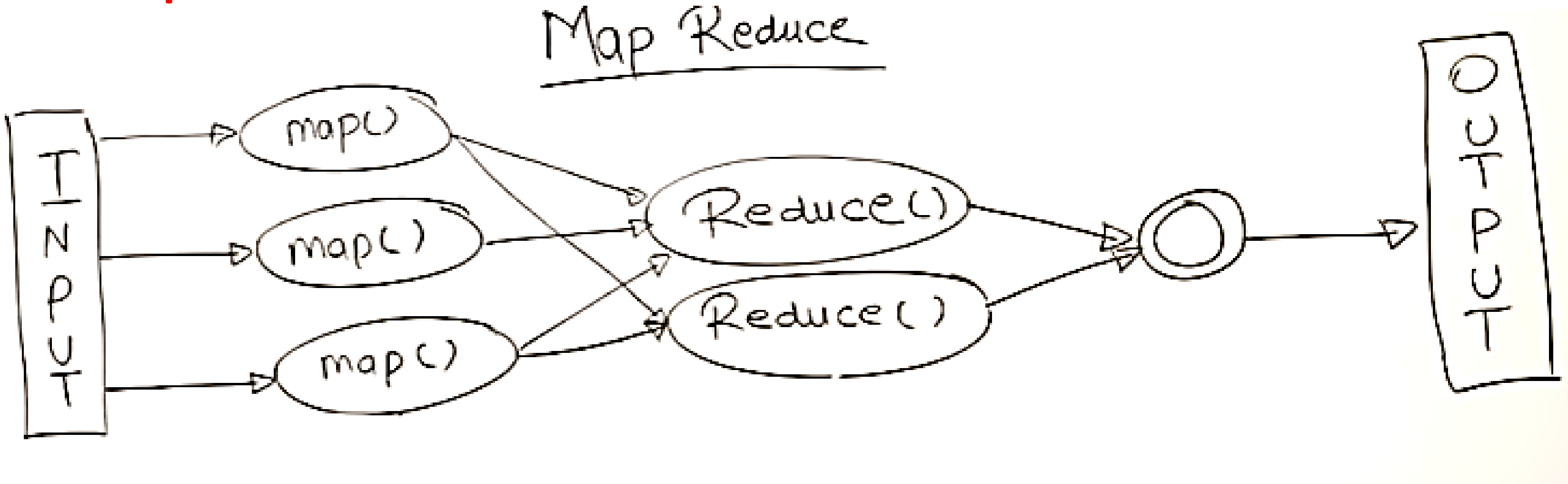
- Hadoop is Open Source tool from the Apache Software Foundation. As the open source project, we can even change the source codes of the Hadoop system. Most of the Hadoop codes are written by Yahoo, IBM, Cloudera etc.
- Hadoop provides parallel processing through different commodity hardware simultaneously.
- As it works on Commodity hardware so the cost is very low. Commodity hardware is low-end and very cheap hardware. So the Hadoop Solution is also economic

Why TO Use Hadoop?

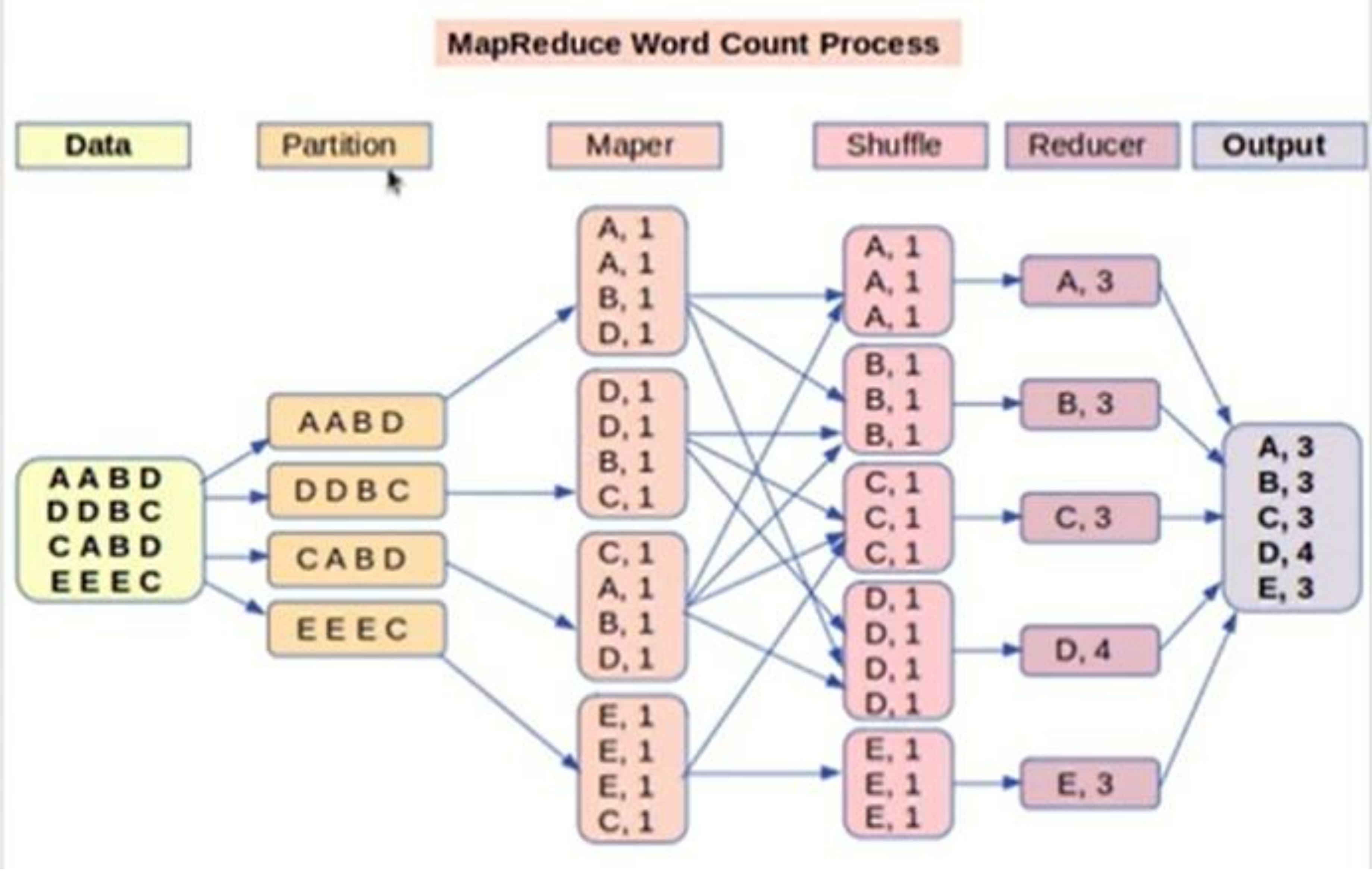


- The Hadoop solution is very popular. It has captured at least 90% of Big data market.
- Hadoop has some unique features that make this solution very popular.
- Hadoop is Scalable. So we can increase the number of commodity hardware easily.
- It is a fault tolerant solution. When one node goes down other nodes can process the data.
- Data can be stored as a Structured, Unstructured and semi-structured mode. So it to more flexible.

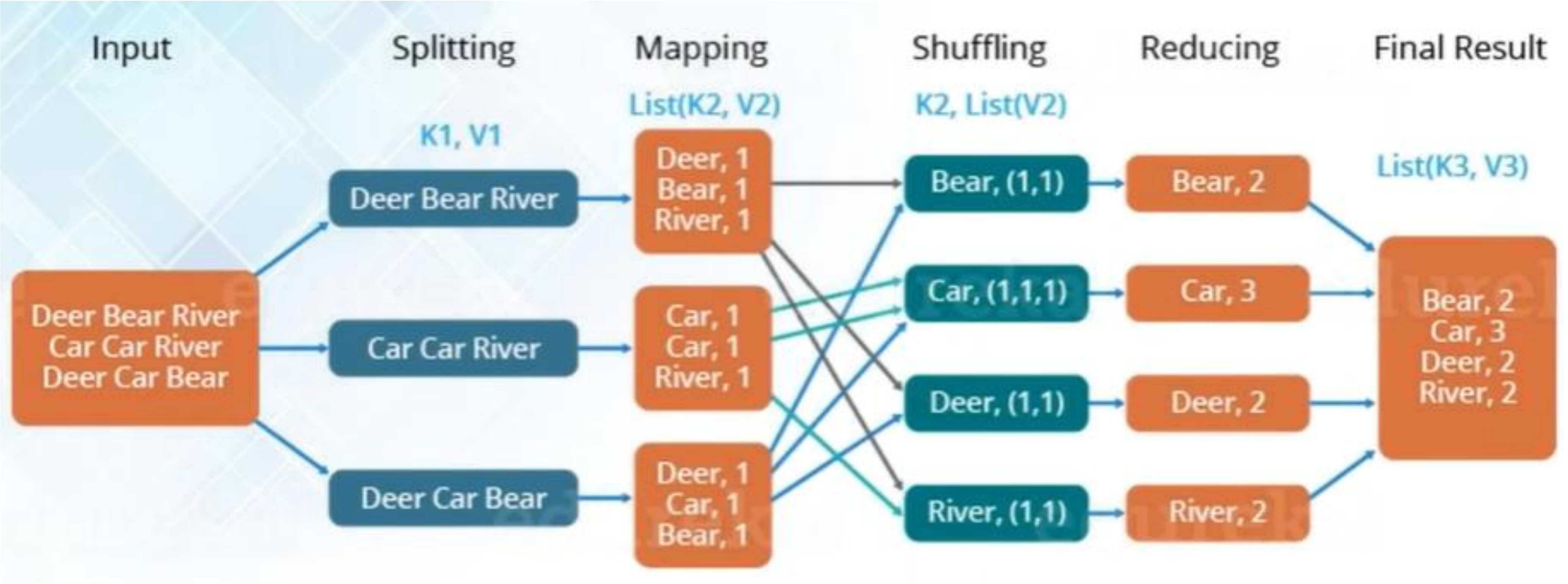
MapReduce:



MapReduce


MapReduce Word Count Process
Data
Partition
Maper
Shuffle
Reducer
Output
A A B D
D D B C
C A B D
E E E C
A A B D
D D B C
C A B D
E E E C
A, 1
A, 1
B, 1
D, 1
D, 1
D, 1
B, 1
C, 1
C, 1
A, 1
B, 1
D, 1
E, 1
E, 1
E, 1
C, 1
A, 1
A, 1
A, 1
B, 1
B, 1
B, 1
C, 1
C, 1
C, 1
D, 1
D, 1
D, 1
D, 1
E, 1
E, 1
E, 1
A, 3
B, 3
C, 3
D, 4
E, 3
A, 3
B, 3
C, 3
D, 4
E, 3

Phases of MapReduce: Word Count



MapReduce : Case Study (Paper Correction and Identifying Topper)

- Map : Parallelism , Reduce : Grouping
- Total = 20 Papers
- Time = 1 min/Paper

Without (MPP):

- 20 Papers Corrections = 20 mins

With (MPP/MapReduce)

- Divide Task – 4 groups / 4 districts (D1, D2, D3, D4) [Divide Tasks – Mapper Class]
- Assign 5 papers- Each district
- To fetch “District Topper” = (D1 =T1, D2= T2, D3 = T3, D4 = T4) = 4 Toppers ; 5 minutes
- To fetch “State Topper” = T1, T2, T3, T4 = Topper; 4 minutes [Aggregation: Reducer Class]
- Total job “To Fetch Topper” = 9 Minutes

- Summary/ Technicality of MapReduce : Time is reduced rapidly in case of MPP application.