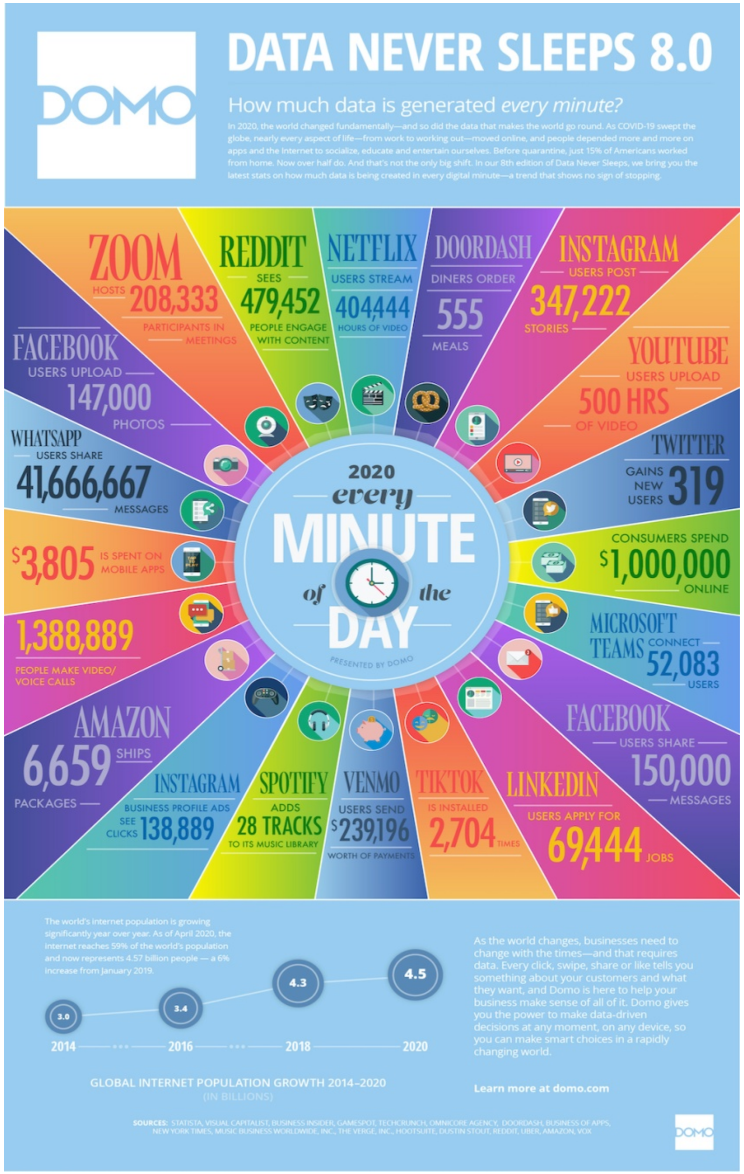
21CSE221T - Big Data Tools and Techniques

Unit I – Introduction to Big Data and Hadoop

Big Data



DATA NEVER SLEEPS 8.0

How much data is generated *every minute?*

In 2020, the world changed fundamentally—and so did the data that makes the world go round. As COVID-19 swept the globe, nearly every aspect of life—from work to working out—moved online, and people depended more and more on apps and the Internet to socialize, educate and entertain ourselves. Before quarantine, just 15% of Americans worked from home. Now over half do. And that's not the only big shift. In our 8th edition of Data Never Sleeps, we bring you the latest stats on how much data is being created in every digital minute—a trend that shows no sign of stopping.

The world's internet population is growing significantly year over year. As of April 2020, the internet reaches 59% of the world's population and now represents 4.57 billion people — a 6% increase from January 2019.

GLOBAL INTERNET POPULATION GROWTH 2014–2020
(IN BILLIONS)

As the world changes, businesses need to change with the times—and that requires data. Every click, swipe, share or like tells you something about your customers and what they want, and Domo is here to help your business make sense of all of it. Domo gives you the power to make data-driven decisions at any moment, on any device, so you can make smart choices in a rapidly changing world.

Learn more at domo.com

SOURCES: STATISTA, VISUAL CAPITALIST, BUSINESS INSIDER, GAMESPOT, TECHCRUNCH, OMNICORE AGENCY, DOORDASH, BUSINESS OF APPS, NEW YORK TIMES, MUSIC BUSINESS WORLDWIDE, INC., THE VERGE, INC., HOOTSUITE, DUSTIN STOUT, REDDIT, UBER, AMAZON, VOX

Infographic courtesy: DOM Inc, 2020

Sources of Big Data

What's Driving Data Deluge?

Mobile Sensors

Social Media

Video Surveillance

Video Rendering

Smart Grids

Geophysical Exploration

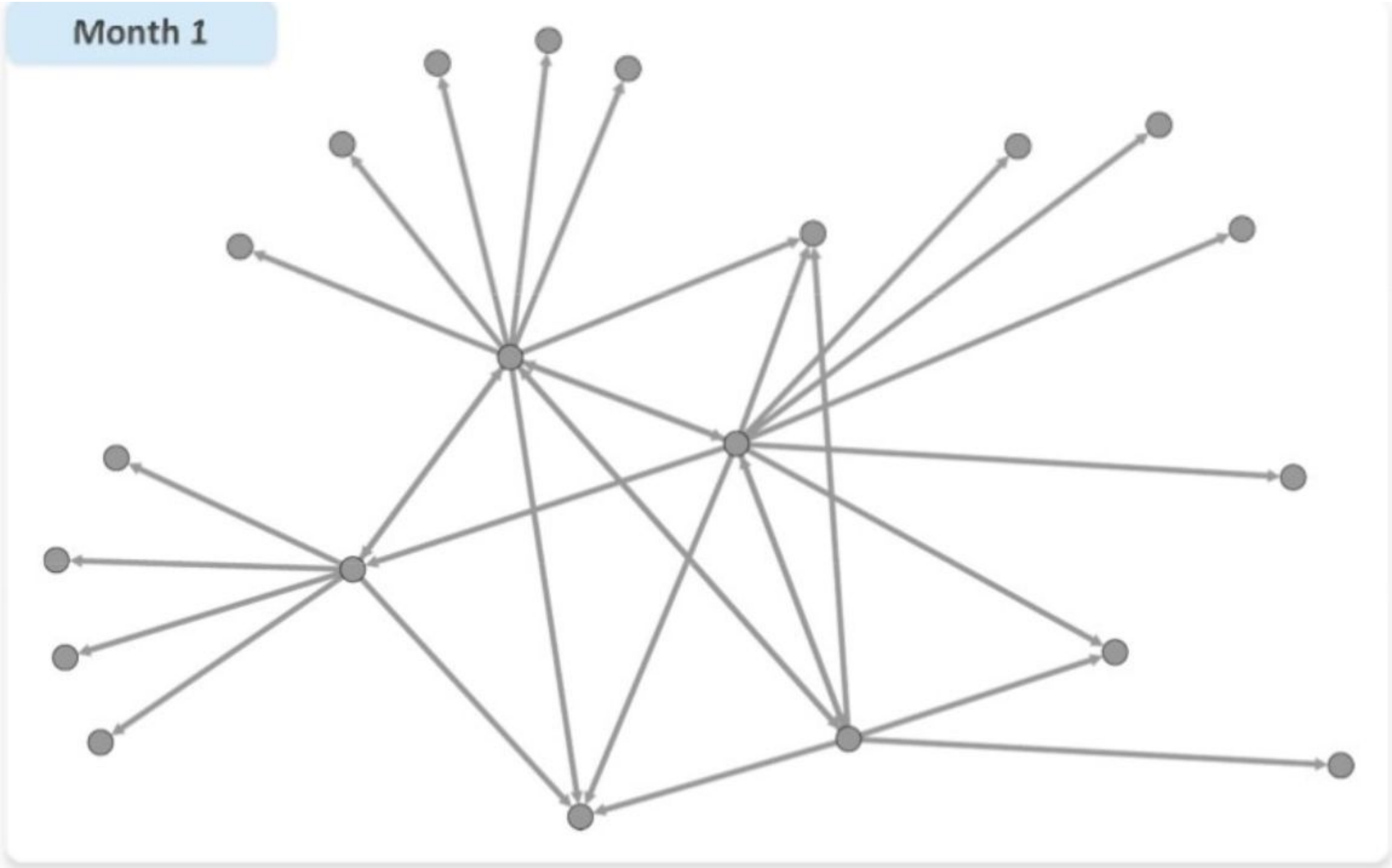
Medical Imaging

Gene Sequencing

Big Data Analytics

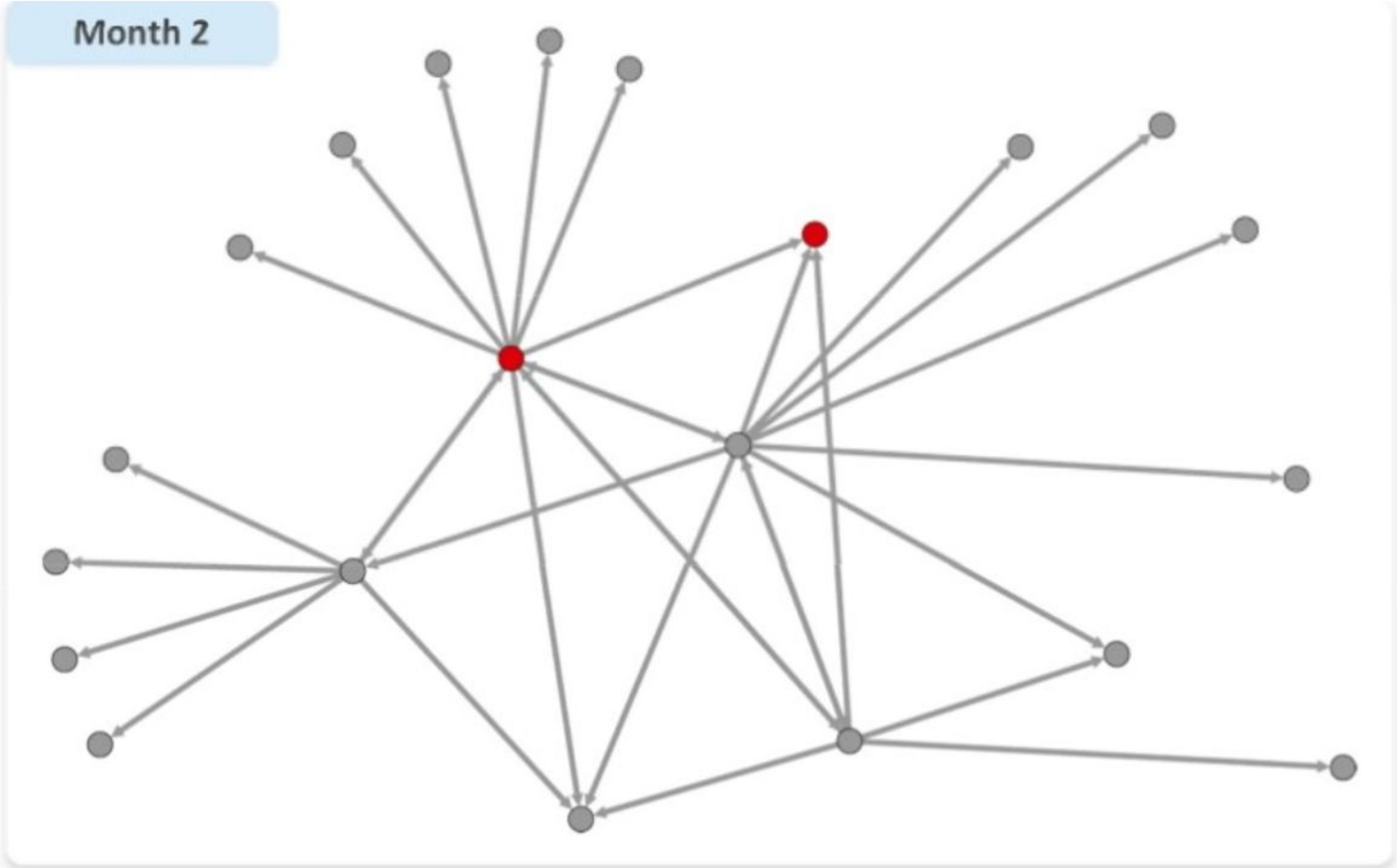
- Performing operations like data mining, machine learning or deep learning to large scale datasets.
- Operations can be as simple as counting or as complicated as large scale complex neural network analysis.

Example – Mobile SP Churn Prediction



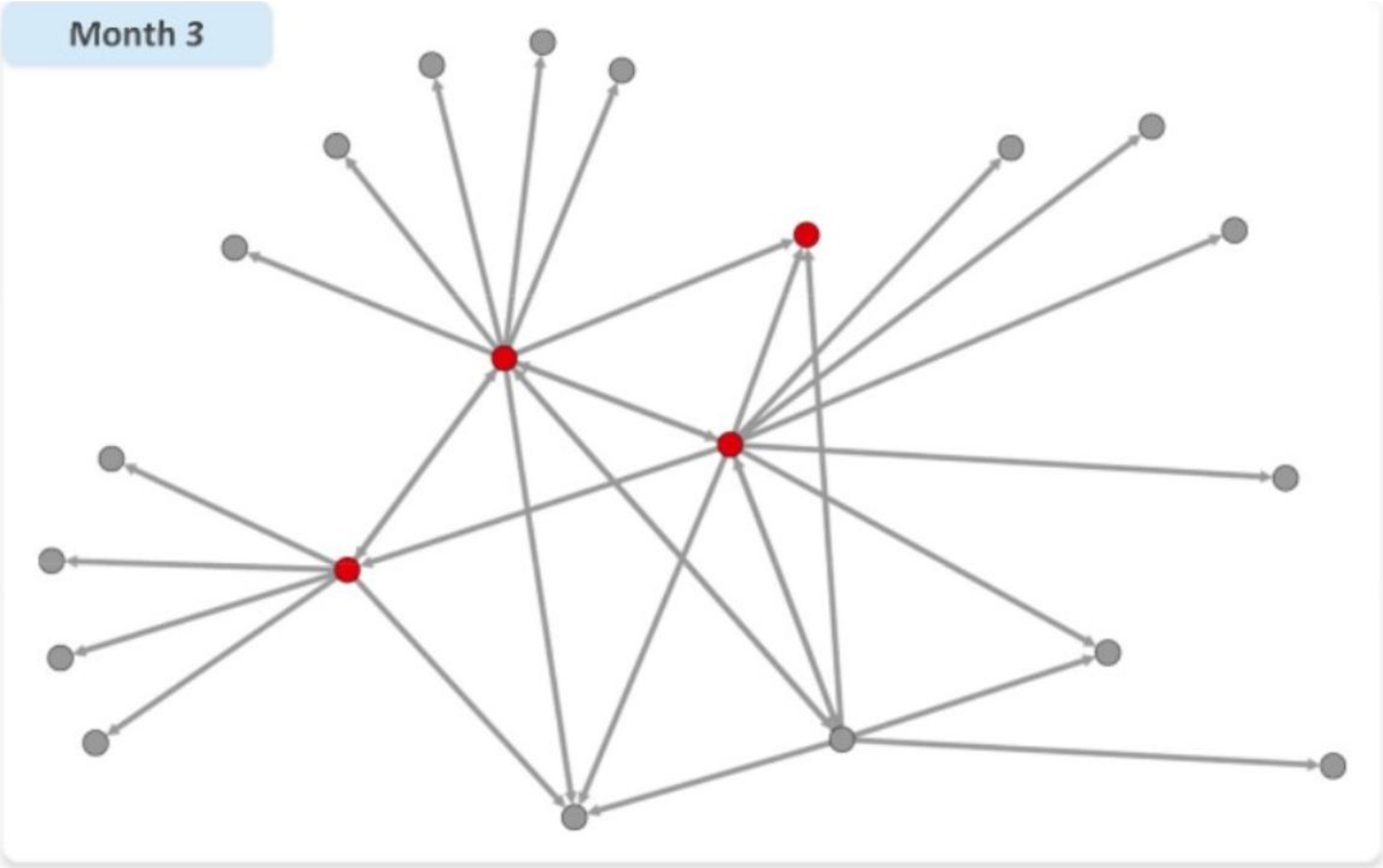
Courtesy: EMC

Example – Mobile SP Churn Prediction



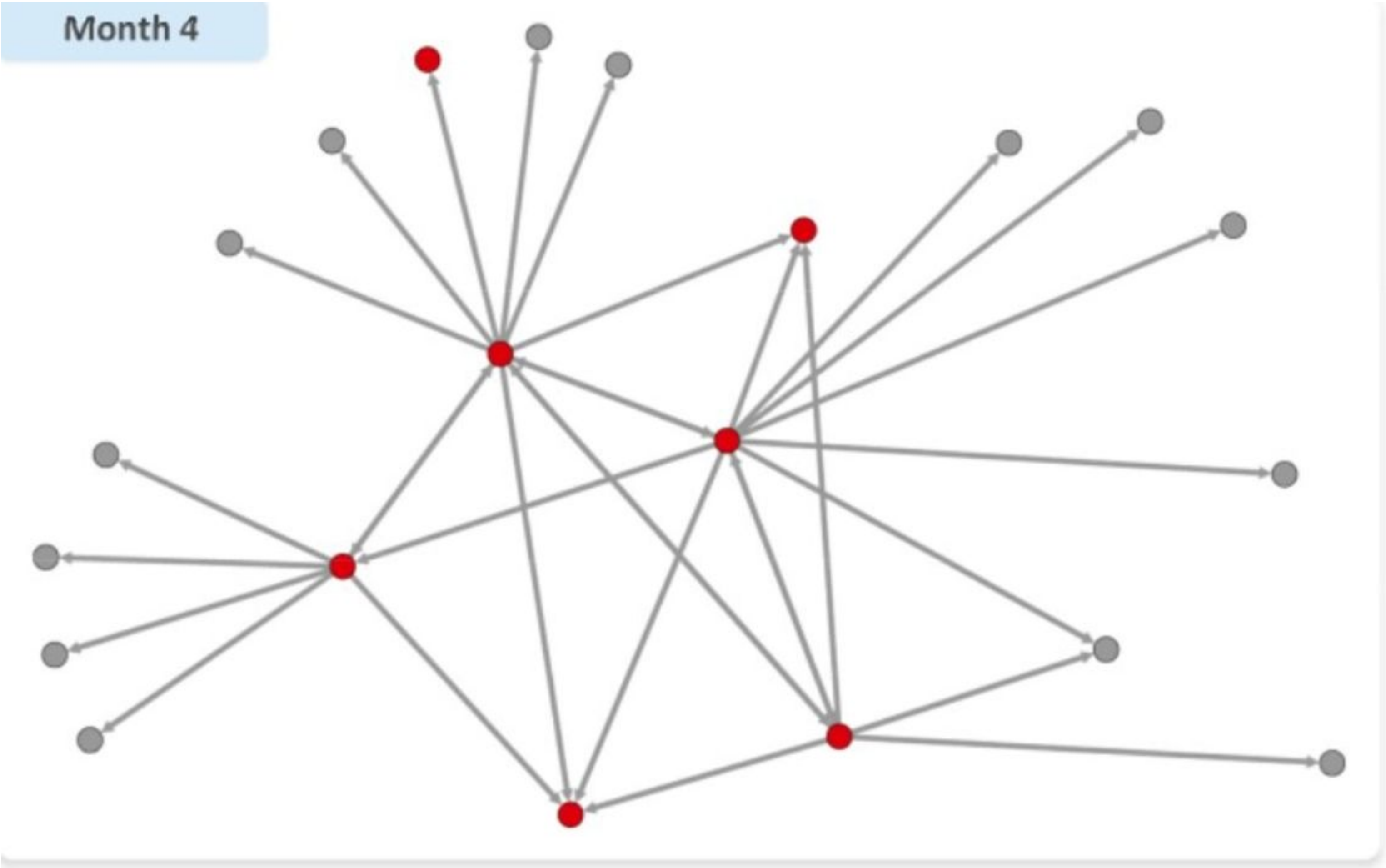
Courtesy: EMC

Example – Mobile SP Churn Prediction



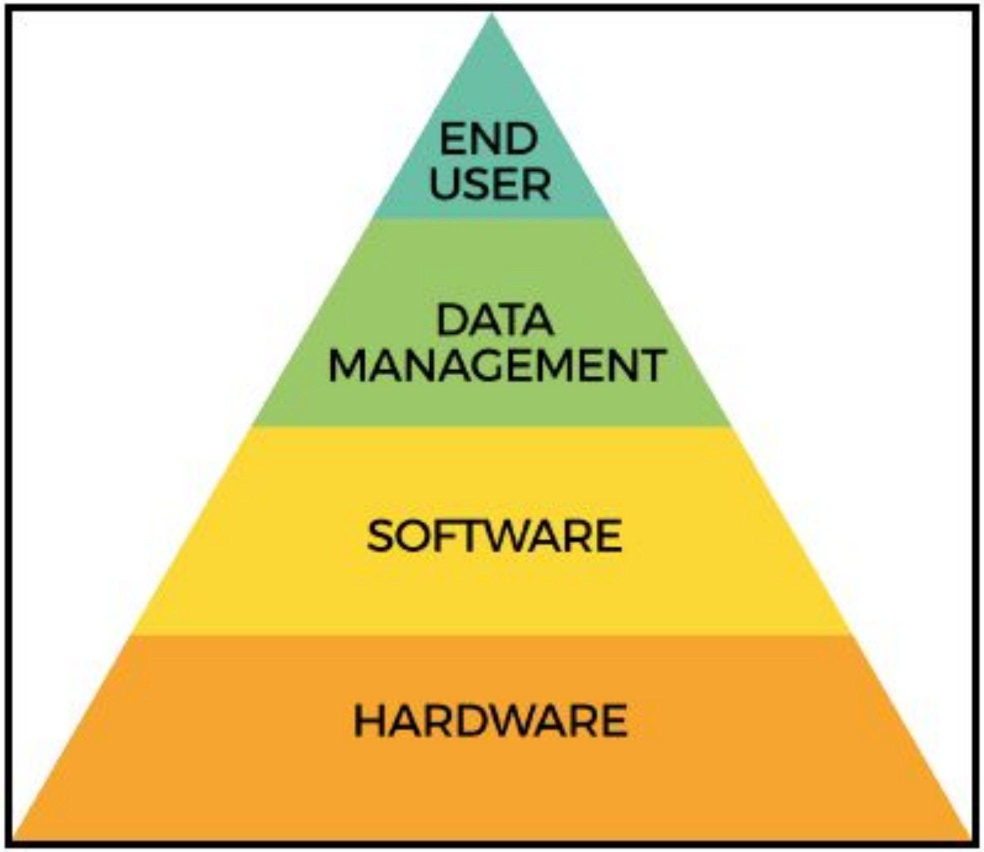
Courtesy: EMC

Example – Mobile SP Churn Prediction



Courtesy: EMC

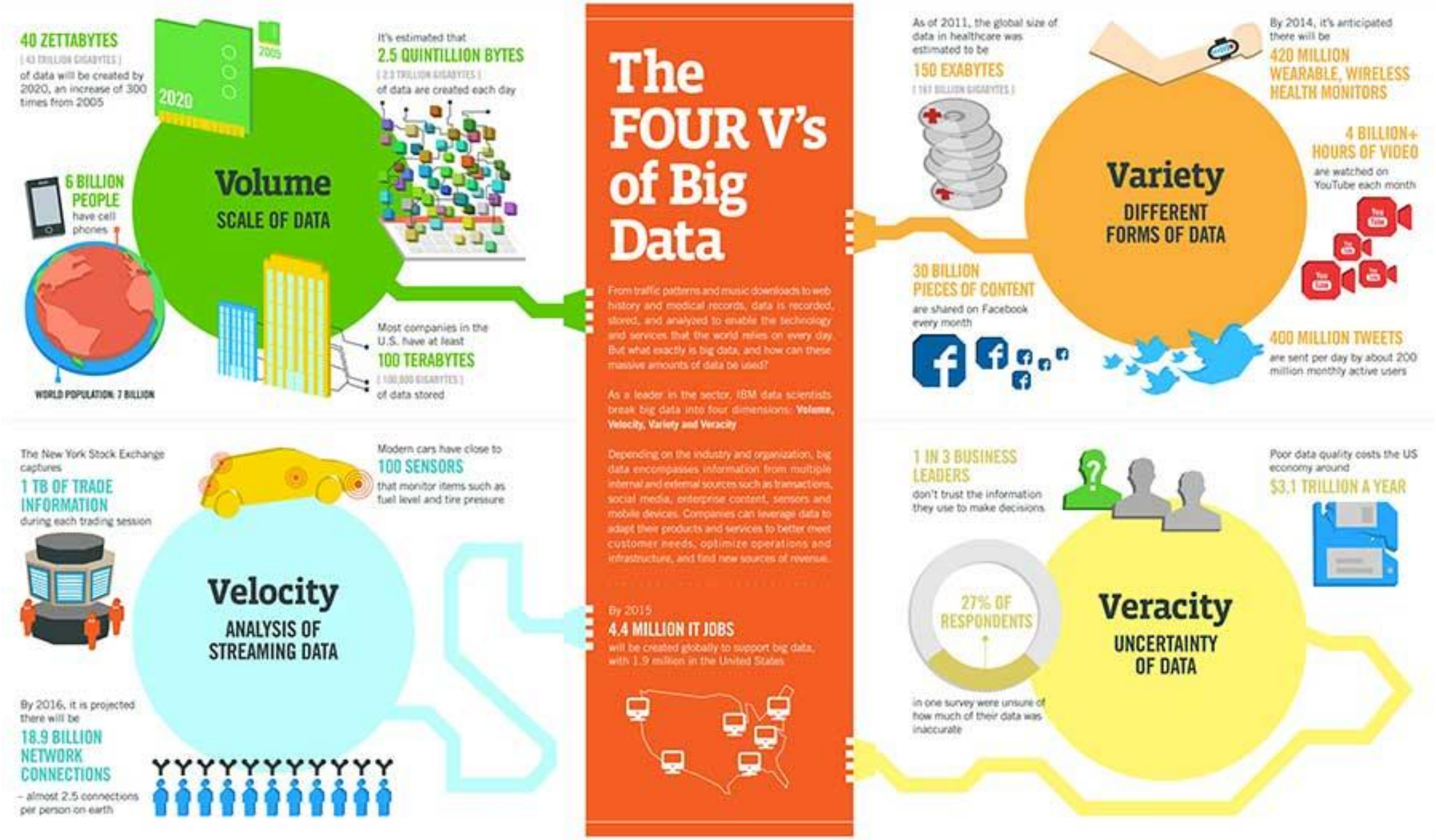
Building Blocks of Big Data Analytics



Characteristics of Big Data

- Volume
- Variety
- Velocity
- Veracity

Characteristics of Big Data



The FOUR V's of Big Data

From traffic patterns and music downloads to web history and medical records, data is recorded, stored, and analyzed to enable the technology and services that the world relies on every day. But what exactly is big data, and how can these massive amounts of data be used?

As a leader in the sector, IBM data scientists break big data into four dimensions: **Volume, Velocity, Variety and Veracity**

Depending on the industry and organization, big data encompasses information from multiple internal and external sources such as transactions, social media, enterprise content, sensors and mobile devices. Companies can leverage data to adapt their products and services to better meet customer needs, optimize operations and infrastructure, and find new sources of revenue.

By 2015 **4.4 MILLION IT JOBS** will be created globally to support big data, with 1.9 million in the United States

Volume

SCALE OF DATA

40 ZETTABYTES [43 TRILLION GIGABYTES] of data will be created by 2020, an increase of 300 times from 2005

It's estimated that **2.5 QUINTILLION BYTES** [2.3 TRILLION GIGABYTES] of data are created each day

6 BILLION PEOPLE have cell phones

Most companies in the U.S. have at least **100 TERABYTES** [100,000 GIGABYTES] of data stored

Velocity

ANALYSIS OF STREAMING DATA

The New York Stock Exchange captures **1 TB OF TRADE INFORMATION** during each trading session

Modern cars have close to **100 SENSORS** that monitor items such as fuel level and tire pressure

By 2016, it is projected there will be **18.9 BILLION NETWORK CONNECTIONS** – almost 2.5 connections per person on earth

Variety

DIFFERENT FORMS OF DATA

As of 2011, the global size of data in healthcare was estimated to be **150 EXABYTES** [161 BILLION GIGABYTES]

By 2014, it's anticipated there will be **420 MILLION WEARABLE, WIRELESS HEALTH MONITORS**

4 BILLION+ HOURS OF VIDEO are watched on YouTube each month

30 BILLION PIECES OF CONTENT are shared on Facebook every month

400 MILLION TWEETS are sent per day by about 200 million monthly active users

Veracity

UNCERTAINTY OF DATA

1 IN 3 BUSINESS LEADERS don't trust the information they use to make decisions

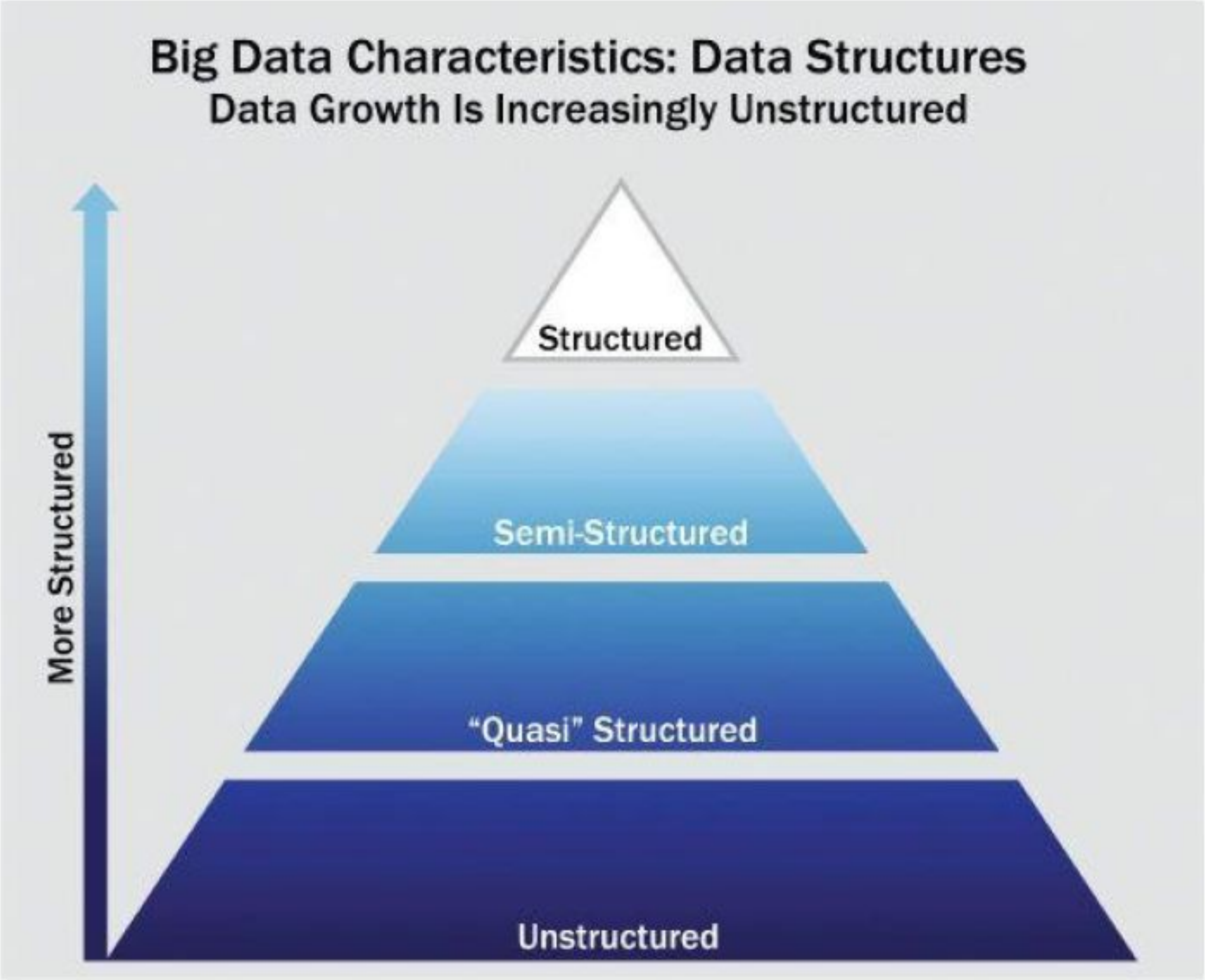
27% OF RESPONDENTS in one survey were unsure of how much of their data was inaccurate

Poor data quality costs the US economy around **$3.1 TRILLION A YEAR**

Sources: McKinsey Global Institute, Twitter, Cisco, Gartner, EMC, SAS, IBM, MEPTEC, QAS

Courtesy: IBM

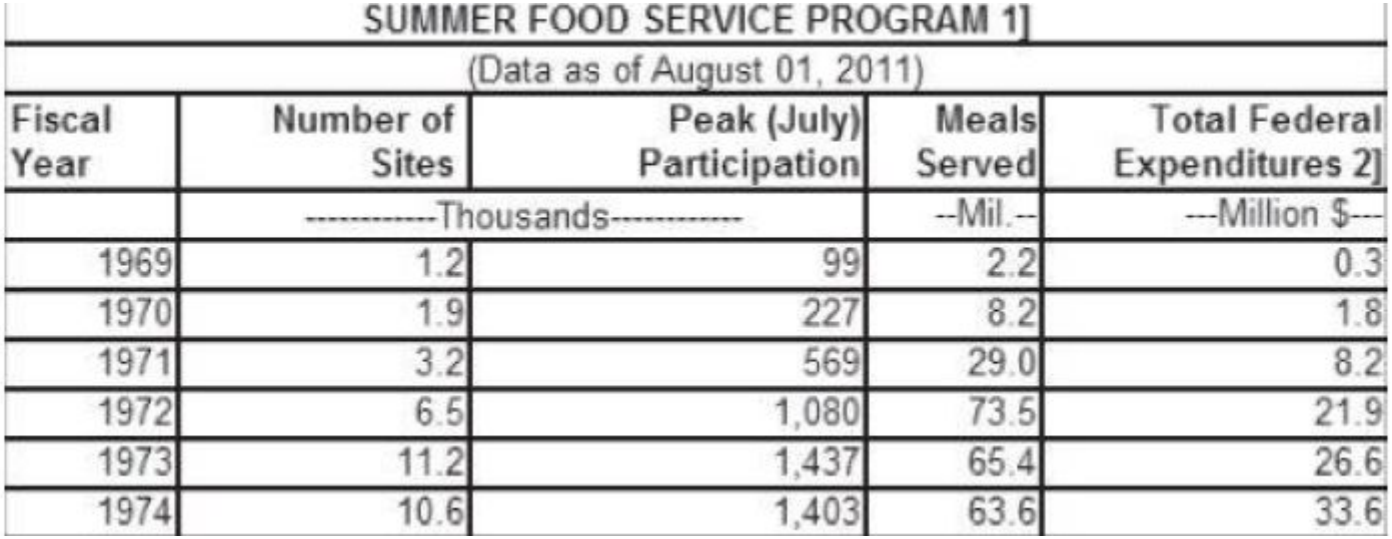
Types of data


Big Data Characteristics: Data Structures
Data Growth Is Increasingly Unstructured
More Structured
Structured
Semi-Structured
"Quasi" Structured
Unstructured

Courtesy: EMC

Structured data

- Data having defined organizational structure
- Uses a representable schema
- All possible data types
- Store and query



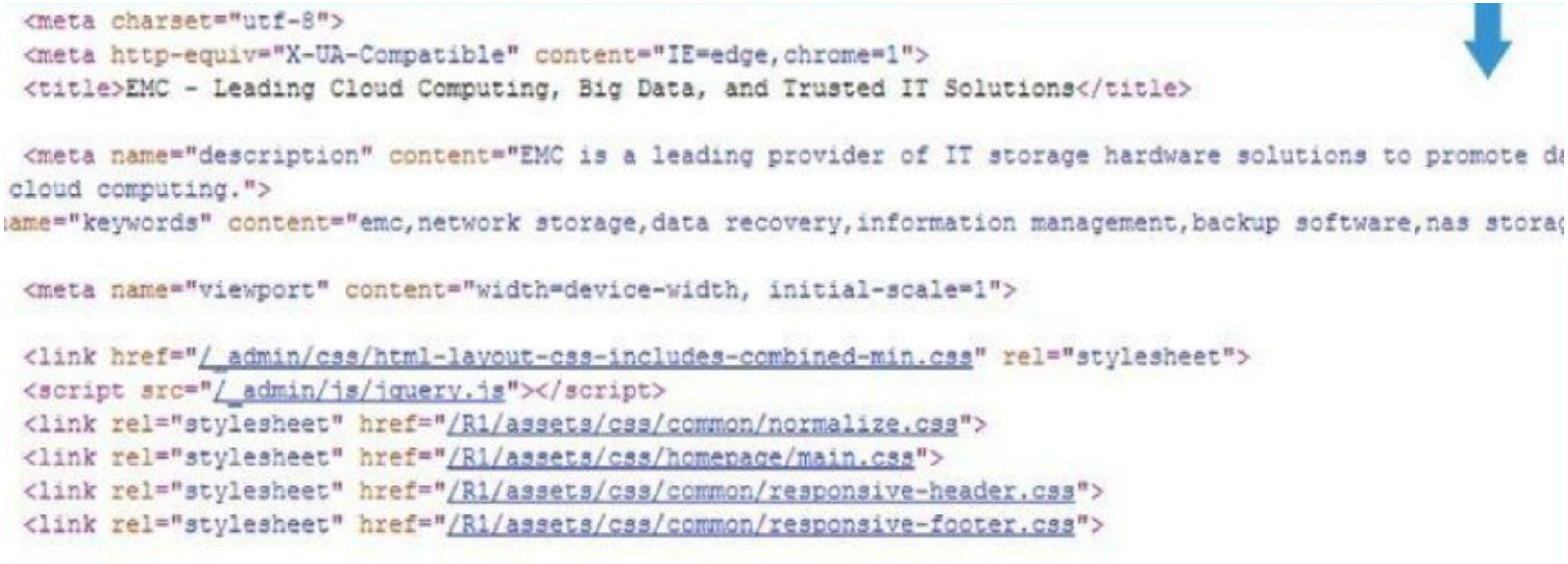
SUMMER FOOD SERVICE PROGRAM 1]

(Data as of August 01, 2011)

Fiscal Year	Number of Sites	Peak (July) Participation	Meals Served	Total Federal Expenditures 2]
	------------Thousands------------		--Mil.--	---Million $---
1969	1.2	99	2.2	0.3
1970	1.9	227	8.2	1.8
1971	3.2	569	29.0	8.2
1972	6.5	1,080	73.5	21.9
1973	11.2	1,437	65.4	26.6
1974	10.6	1,403	63.6	33.6

Semi-structured data

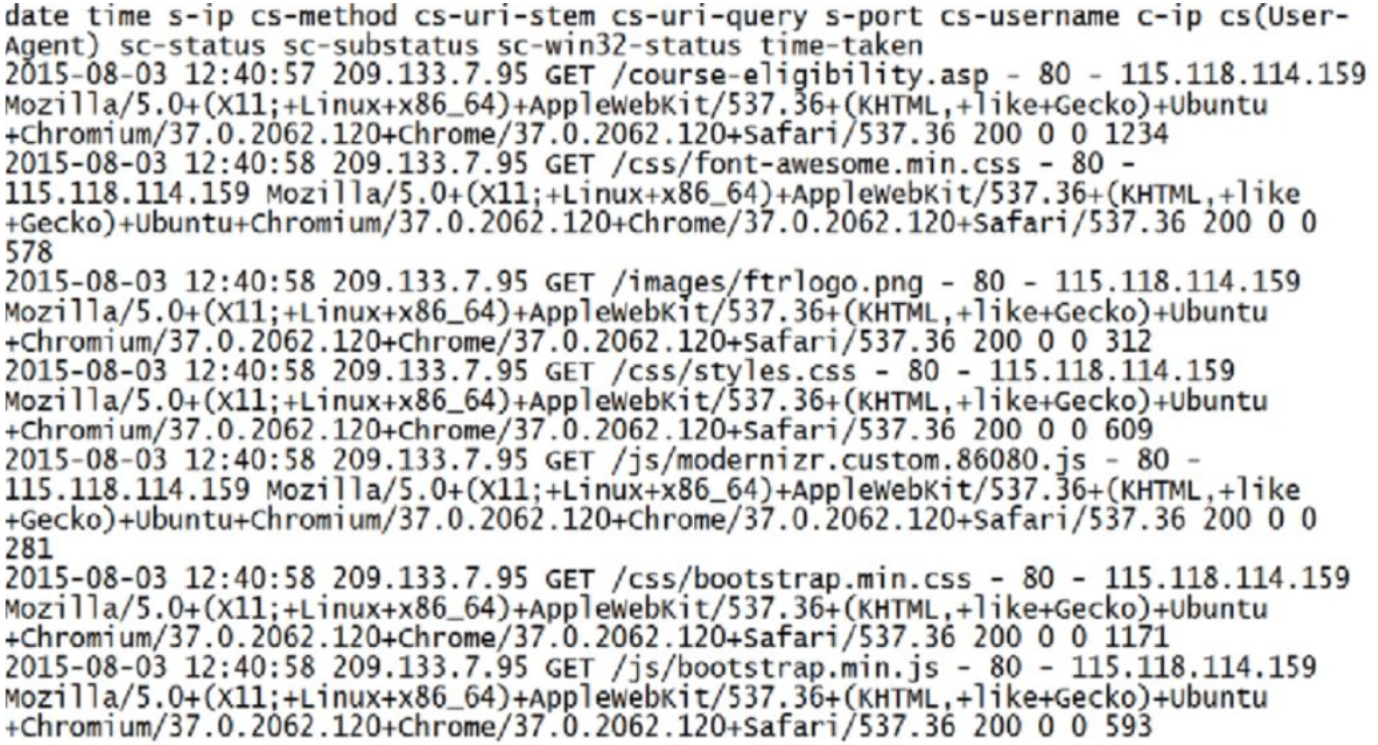
- Textual data type with a discernible pattern
- XML, JSON
- XML or JSON schema
- MongoDB, Couch DB for storage



```
 <meta charset="utf-8">
 <meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1">
 <title>EMC - Leading Cloud Computing, Big Data, and Trusted IT Solutions</title>

 <meta name="description" content="EMC is a leading provider of IT storage hardware solutions to promote da
cloud computing.">
ame="keywords" content="emc,network storage,data recovery,information management,backup software,nas storag

 <meta name="viewport" content="width=device-width, initial-scale=1">

 <link href="/_admin/css/html-layout-css-includes-combined-min.css" rel="stylesheet">
 <script src="/_admin/js/jquery.js"></script>
 <link rel="stylesheet" href="/R1/assets/css/common/normalize.css">
 <link rel="stylesheet" href="/R1/assets/css/homepage/main.css">
 <link rel="stylesheet" href="/R1/assets/css/common/responsive-header.css">
 <link rel="stylesheet" href="/R1/assets/css/common/responsive-footer.css">
```

Quasi structured data

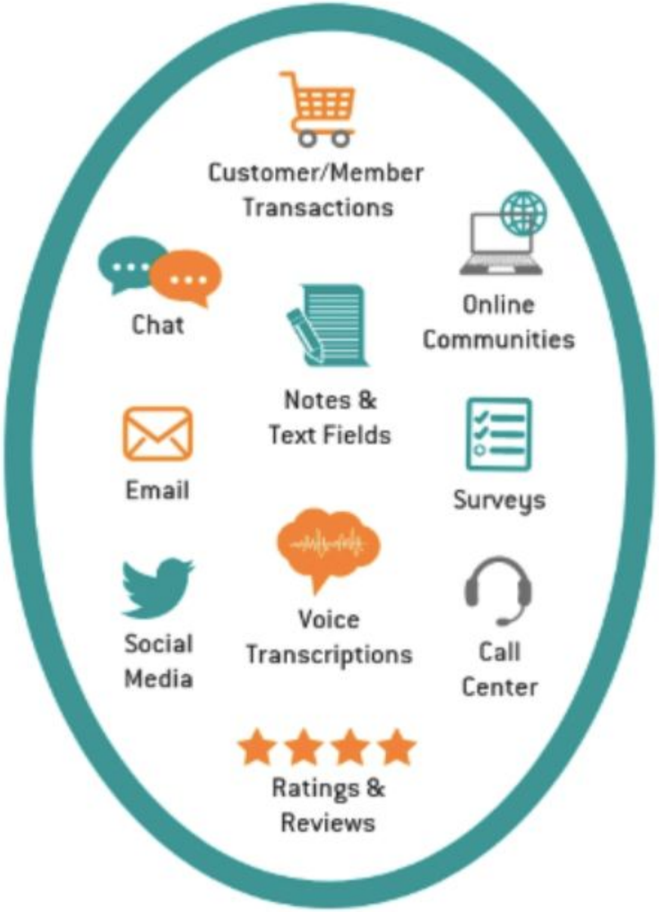
- Textual data but with inconsistencies
- Effort and time to pre process
- E.g Clickstream data, Web server log, Network packet tracing



```
date time s-ip cs-method cs-uri-stem cs-uri-query s-port cs-username c-ip cs(User-
Agent) sc-status sc-substatus sc-win32-status time-taken
2015-08-03 12:40:57 209.133.7.95 GET /course-eligibility.asp - 80 - 115.118.114.159
Mozilla/5.0+(X11;+Linux+x86_64)+AppleWebKit/537.36+(KHTML,+like+Gecko)+Ubuntu
+Chromium/37.0.2062.120+Chrome/37.0.2062.120+Safari/537.36 200 0 0 1234
2015-08-03 12:40:58 209.133.7.95 GET /css/font-awesome.min.css - 80 -
115.118.114.159 Mozilla/5.0+(X11;+Linux+x86_64)+AppleWebKit/537.36+(KHTML,+like
+Gecko)+Ubuntu+Chromium/37.0.2062.120+Chrome/37.0.2062.120+Safari/537.36 200 0 0
578
2015-08-03 12:40:58 209.133.7.95 GET /images/ftrlogo.png - 80 - 115.118.114.159
Mozilla/5.0+(X11;+Linux+x86_64)+AppleWebKit/537.36+(KHTML,+like+Gecko)+Ubuntu
+Chromium/37.0.2062.120+Chrome/37.0.2062.120+Safari/537.36 200 0 0 312
2015-08-03 12:40:58 209.133.7.95 GET /css/styles.css - 80 - 115.118.114.159
Mozilla/5.0+(X11;+Linux+x86_64)+AppleWebKit/537.36+(KHTML,+like+Gecko)+Ubuntu
+Chromium/37.0.2062.120+Chrome/37.0.2062.120+Safari/537.36 200 0 0 609
2015-08-03 12:40:58 209.133.7.95 GET /js/modernizr.custom.86080.js - 80 -
115.118.114.159 Mozilla/5.0+(X11;+Linux+x86_64)+AppleWebKit/537.36+(KHTML,+like
+Gecko)+Ubuntu+Chromium/37.0.2062.120+Chrome/37.0.2062.120+Safari/537.36 200 0 0
281
2015-08-03 12:40:58 209.133.7.95 GET /css/bootstrap.min.css - 80 - 115.118.114.159
Mozilla/5.0+(X11;+Linux+x86_64)+AppleWebKit/537.36+(KHTML,+like+Gecko)+Ubuntu
+Chromium/37.0.2062.120+Chrome/37.0.2062.120+Safari/537.36 200 0 0 1171
2015-08-03 12:40:58 209.133.7.95 GET /js/bootstrap.min.js - 80 - 115.118.114.159
Mozilla/5.0+(X11;+Linux+x86_64)+AppleWebKit/537.36+(KHTML,+like+Gecko)+Ubuntu
+Chromium/37.0.2062.120+Chrome/37.0.2062.120+Safari/537.36 200 0 0 593
```

Unstructured data

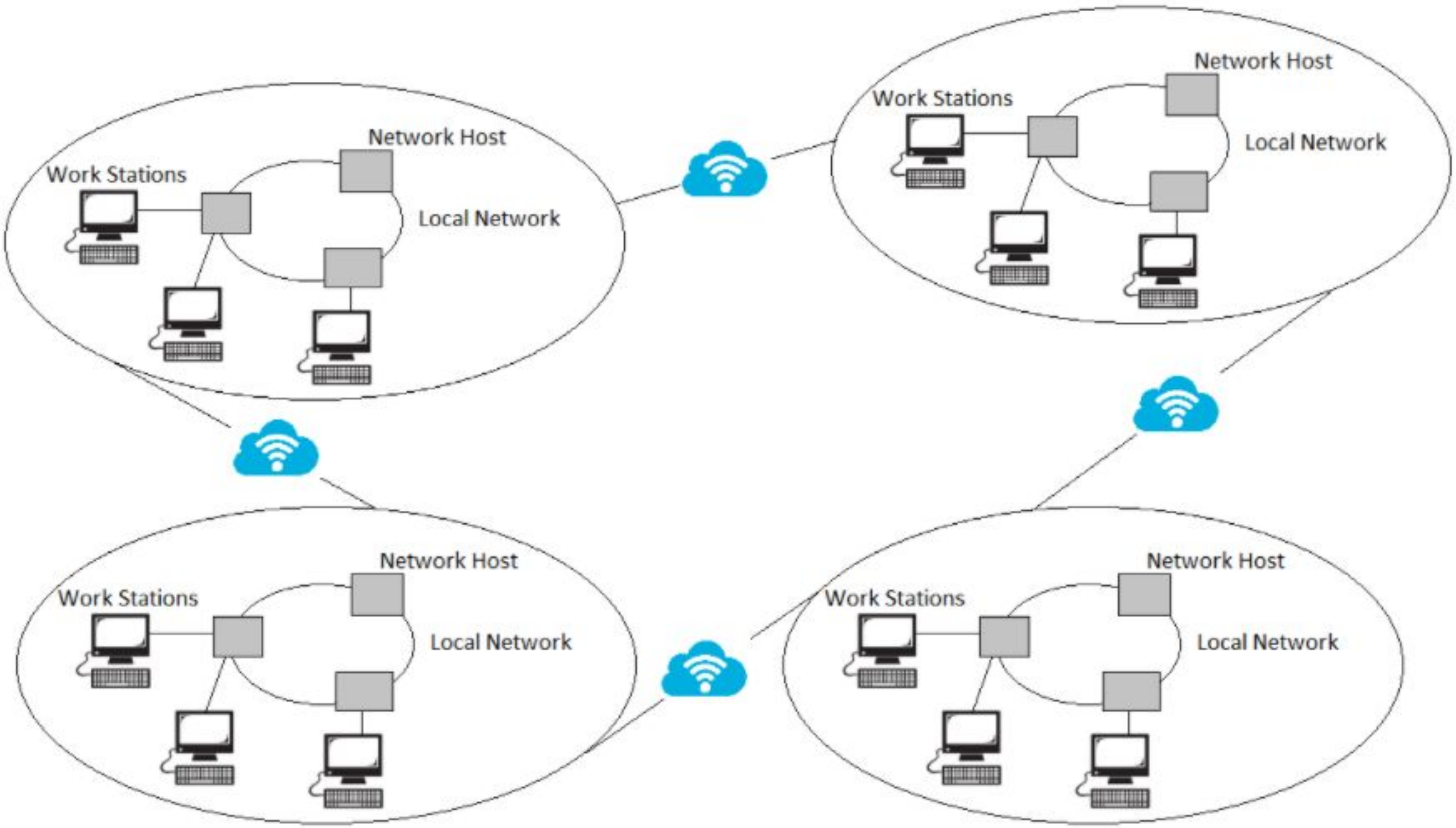
- Data that has no inherent structure
- Videos, Images, Documents
- Text data
  - ElastiSearch, Solr



Examples of unstructured data. Image courtesy: ORI

Distributed Computing for Big Data

- Usage of more than one connected computer
- Examples: NoW(Network of Workstations)



Distributed Computing for Big Data

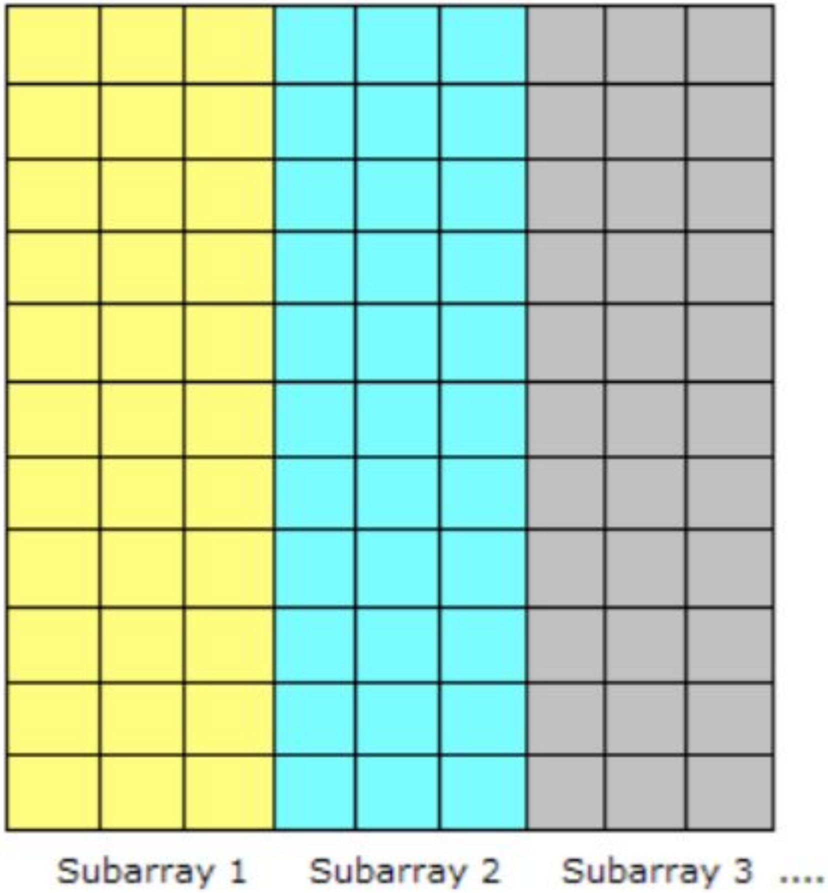
- Treated infrastructure as one big pool of computing, storage and networking resources
- Moves to another resource in the pool in case of failure
- **Fault Tolerance or High Availability**
- Cost of computing and storage resources has decreased considerably
- E.g Distributed File System

Parallel Computing for Big Data

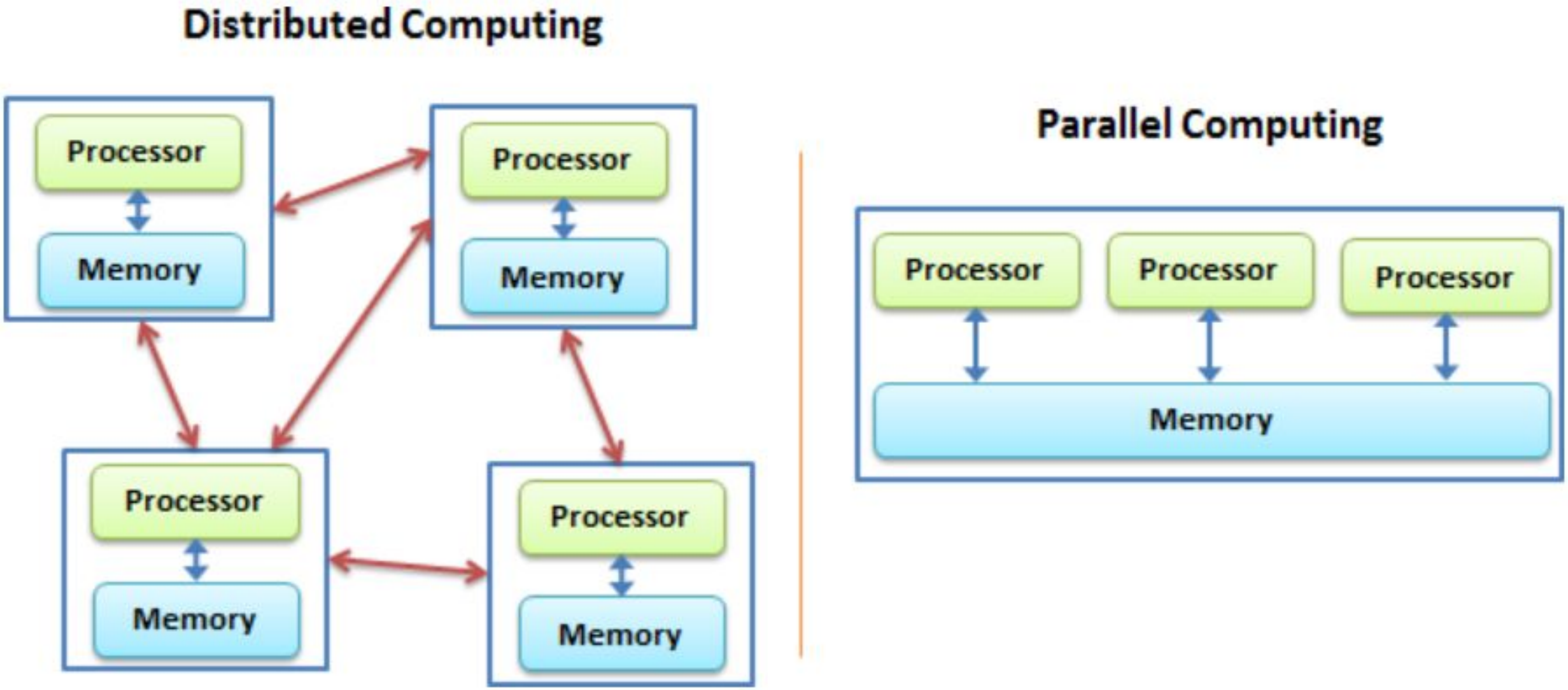
- Serial – Sequence of instructions one by one
- Start to Finish on a single processor
- Parallel Programming
  - Processing broken into multiple parts
  - Each part executed concurrently
  - Different CPUs on a single or different machines

Parallel Computing for Big Data

- First step
  - Identify set of tasks that could be run concurrently
  - Data partitioned without dependencies



Distributed, Parallel Computing



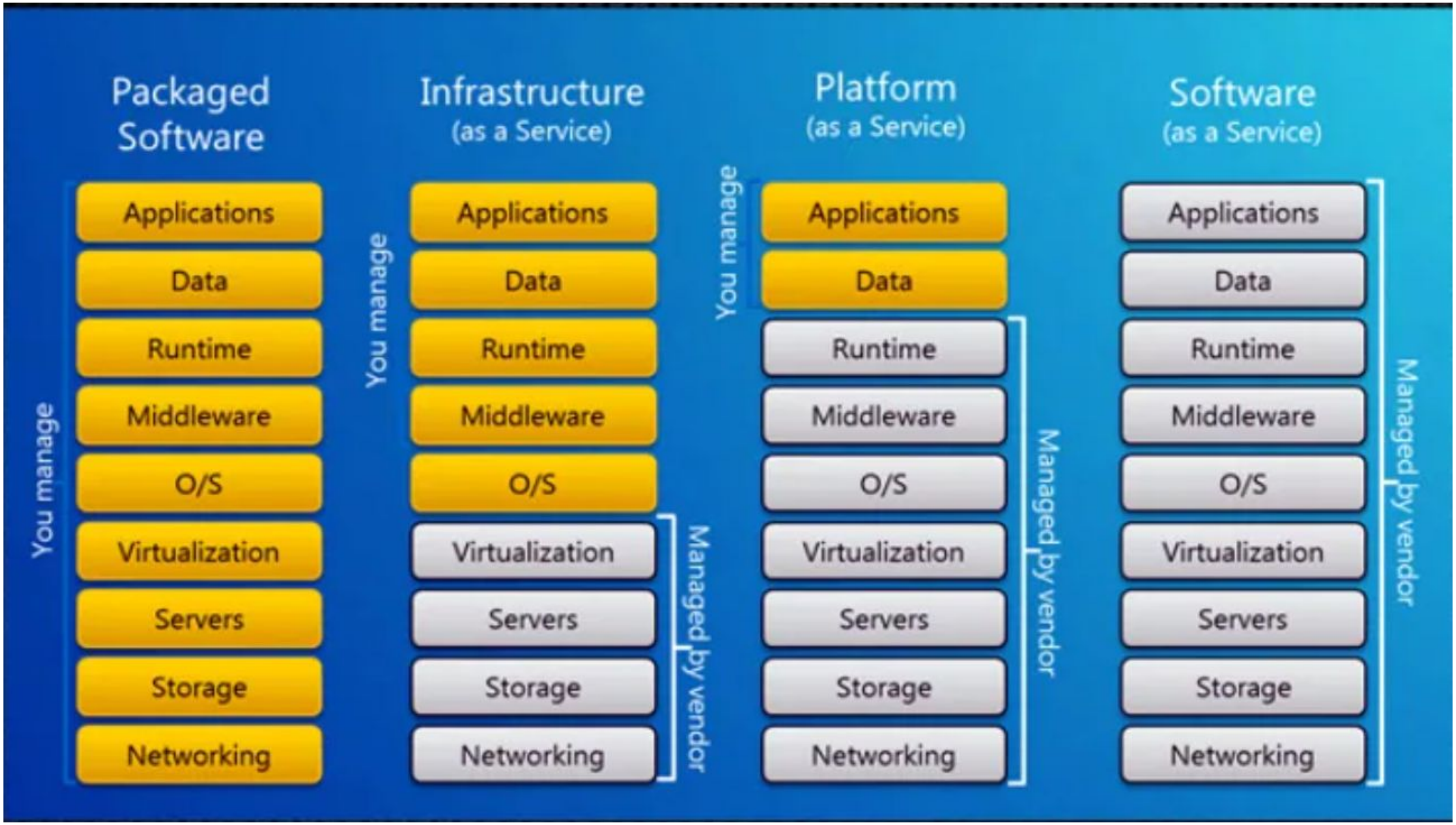
Distributed Computing

Parallel Computing

Cloud Computing for big data

- Set of high powered servers
- Pay as you use
- Memory or Storage intensive
- View and query large data sets quickly
- Infrastructure as a Service
- Platform as a Service
- Software as a Service

Cloud Computing for big data



Packaged Software

You manage

- Applications
- Data
- Runtime
- Middleware
- O/S
- Virtualization
- Servers
- Storage
- Networking

Infrastructure (as a Service)

You manage

- Applications
- Data
- Runtime
- Middleware
- O/S

Managed by vendor

- Virtualization
- Servers
- Storage
- Networking

Platform (as a Service)

You manage

- Applications
- Data

Managed by vendor

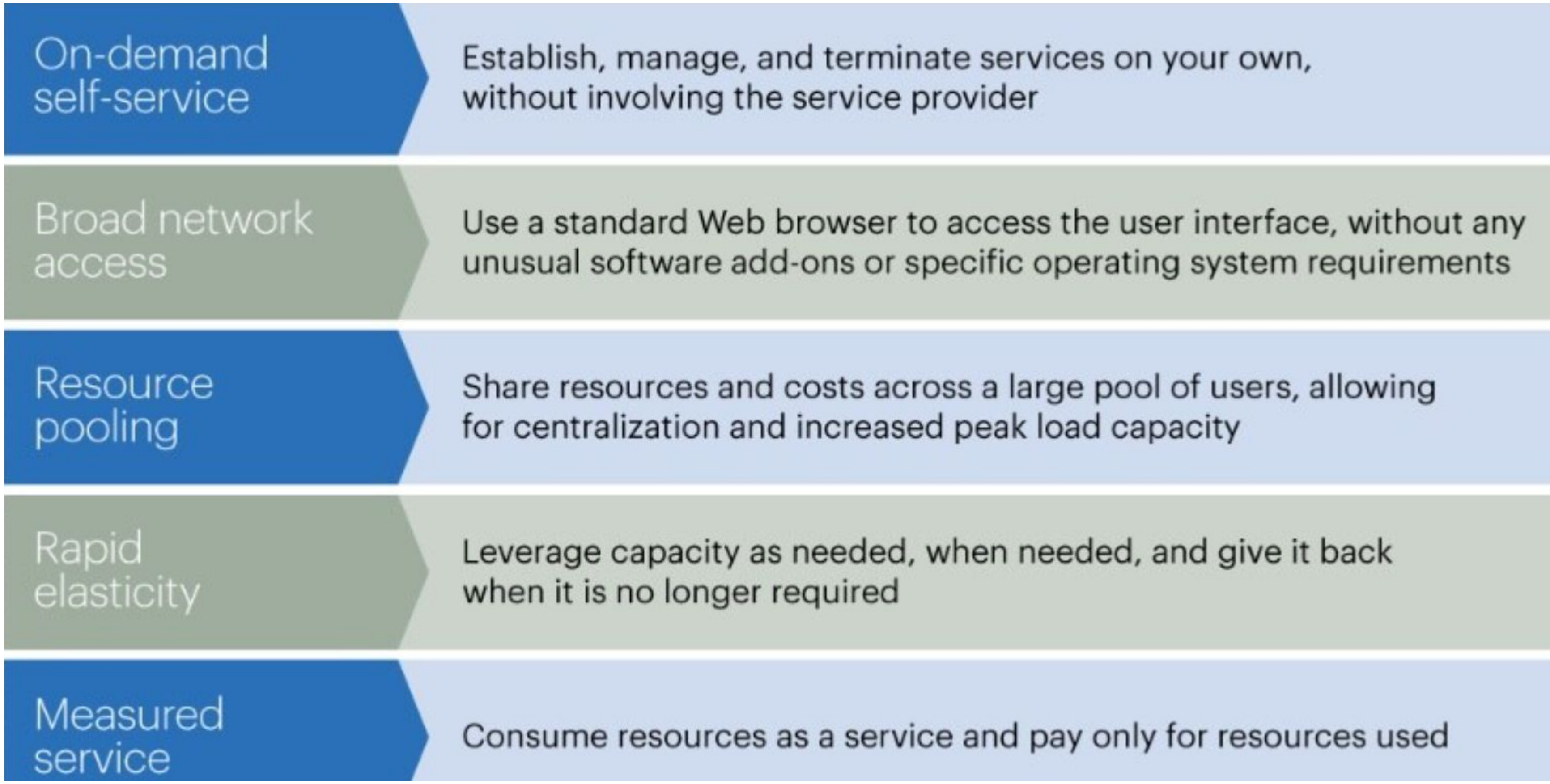
- Runtime
- Middleware
- O/S
- Virtualization
- Servers
- Storage
- Networking

Software (as a Service)

Managed by vendor

- Applications
- Data
- Runtime
- Middleware
- O/S
- Virtualization
- Servers
- Storage
- Networking

Drivers of Cloud Computing for big data



On-demand self-service	Establish, manage, and terminate services on your own, without involving the service provider
Broad network access	Use a standard Web browser to access the user interface, without any unusual software add-ons or specific operating system requirements
Resource pooling	Share resources and costs across a large pool of users, allowing for centralization and increased peak load capacity
Rapid elasticity	Leverage capacity as needed, when needed, and give it back when it is no longer required
Measured service	Consume resources as a service and pay only for resources used

Cloud Computing for big data

- Need not ship terabytes of data around
- Store in cloud and **move analytics close to data**
- Store insensitive data in cloud
- Sensitive data – Private Cloud/Clusters
- Advantages
  - Scalability
  - Cost reduction
  - Faster time to market

Cloud Computing for big data

- Limitations
  - Latency
  - Multi-tenancy overhead
  - Data privacy
  - Data Security
  - Costs are charged for static storage

In-memory computing for big data

- Computations are **latency sensitive**
- Have to be done **in memory**
- Disk reads and writes would cause delays
- For compute-intensive applications
  - Real time stream processing
  - Apache Spark

Big Data Mining

- Two broad categories of BDA
  - Big Data Mining
  - Predictive Analytics
- Big Data Mining
  - Refers to the entire life cycle of processing large data sets
  - From procurement of data to implementation of respective tools to analyze data

Big Data Mining – Building Corporate Big Data strategy

- Determine use cases
- Then fix the platforms

Steps

1. Who needs big data mining?
   - Which business groups will benefit significantly?
   - Revenue impact
   - E.g Pharmaceutical company – Commercial Research, Epidemiology, Health Economics

Big Data Mining – Building Corporate Big Data strategy

2. Determining use cases
   - Impactful analysis by working closer with practitioner (analyst) and stakeholder (business end user)
   - Analyst will work on operational challenges that might be faced in implementing the BD solution
   - Business user will highlight the aspects of business that will benefit from the BD solution
   - An optimal outcome consolidating both

Big Data Mining – Building Corporate Big Data strategy

3. Stakeholder’s buy in
   - Budget decisions
   - Establish a baseline
   - Can be leveraged on successful BD solution implementation
4. Early wins and effort-to-reward ratio
   - Relatively small use case
   - Implemented in short time, smaller budget
   - Optimizes a business critical function
   - Early win

$$\boxed{\text{E-R Ratio} = \frac{\text{Time} + \text{Cost} + \text{Number of Resources} + \text{Criticality of Use Case}}{\textit{Business Value}}}$$

Big Data Mining – Building Corporate Big Data strategy

5. Leveraging the early wins
   - Ground to develop bigger strategy
   - Establishing value of big data to an audience
   - Single department to broader organizational impact

Big Data Mining - Stakeholders

- Business Sponsor
  - Funding department for the project
  - Benefits from the solution
- Implementation group
  - Analysts, Data Scientist, Field workers
- IT procurement
  - Licensing cost
  - Software, Cloud Renting
- Legal
  - Licensing
  - Open source or in-house

Technical elements of BD platform

- Selection of hardware stack
- Selection of software stack

Selection of hardware stack

- Proper choice depends on key metrics
  - Type of data
  - Size of data
  - Frequency of data updates
- Three broad models of hardware architecture
  - Multi-node architecture
  - Single node architecture
  - Cloud based architecture

Selection of hardware stack

- Multi-node architecture
  - Multiple nodes or servers connected
  - Distributed computing
  - Commodity Servers (low end machines that work in tandem to provide large scale mining and analytics capabilities)
  - Data range TBs and above

Selection of hardware stack

- Single node architecture
  - Single server
  - Structured text data
  - 1-5TB
  - Not commonly used
  - Restricted environments (mobile)

Selection of hardware stack

- Cloud architecture
  - Most efficient
  - Reduced costs in procuring, maintaining physical hardware and hosting software
  - Elastic, provisions on demand
  - Amazon AWS, Microsoft Azure, Google Compute
  - IBM Cloud Brokerage – Select and manage multiple cloud based solutions

Selection of software stack

- Popular options
  - Hadoop Ecosystem
    - Multiple projects under the Apache Software Foundation.
    - Core Components:
      - Hadoop Common: Shared utilities and libraries.
      - Hadoop MapReduce: Batch processing engine.
      - Hadoop Distributed File System (HDFS): Reliable distributed storage.
      - Hadoop YARN: Resource management and job scheduling.

Selection of software stack

- Popular options
  - Apache Spark
    - Developed at UC Berkeley AMPLab.
    - Designed to overcome Hadoop's limitations.
    - Key Features:
      - Support for multiple programming languages, including Python and Scala.
      - Utilizes Resilient Distributed Datasets (RDDs).
    - Compatibility:
      - Integrates seamlessly with existing cluster managers like Mesos and YARN.
      - Provides distributed storage and high-speed processing.

Selection of software stack

- Popular options
  - NoSQL Storage
    - Key Value (Redis, Riak)
    - In-memory (Redis, KDB+)
    - Columnar (Cassandra, Google BigTable)
    - Document-based (MongoDB)
  - Cloud based solutions
    - SaaS, AaaS
    - Data ware houses or storage in Cloud
    - Routine tasks like backup are taken care by the vendor

Try answering!

1. Which of the below is an example for quasi structured data?
    a. Employee information in spreadsheets
    b. Images
    c. Audio files
    d. Clickstream data

Try answering!

2. Breaking the process into multiple parts and executing each part concurrently is ----------- paradigm

    a. Distributed Computing

    b. Parallel Processing

    c. Cloud Computing

    d. In-memory computing

Try answering!

3. Which of the below statements is related to the attribute “veracity” of big data
    a. 420 million wearable wireless health monitoring sensors
    b. 2.5 quintillion bytes of data generated everyday
    c. Poor data quality costs aroung $1 trillion very year to US

Try answering!

4. Which of the below is not a Big Data Processing Engine/Framework
    a. Hadoop
    b. Spark
    c. YARN
    d. Flink

Try answering!

5. Which of the below is a NoSQL Storage platform that stores as documents
    a. Redis
    b. MongoDB
    c. Cassandra
    d. IBM DB2

History of Hadoop

- Inspired by GFS paper in 2003
- Doug Cutting was part of Nutch and Lucene
- NDFS – Nutch Distributed File System
- Was moved as a separate project Hadoop

Why Hadoop

- Failure of nodes
- Scalability of nodes
- Petabytes of data to be processed in parallel
- Cost effective
- Efficient
- Data Locality
- Minimal impact to program logic

Apache Hadoop

- Open source framework
- Distributed storage and processing
  - Of large scale data sets
  - On commodity hardware
- Quickly gain insights from structured and unstructured data

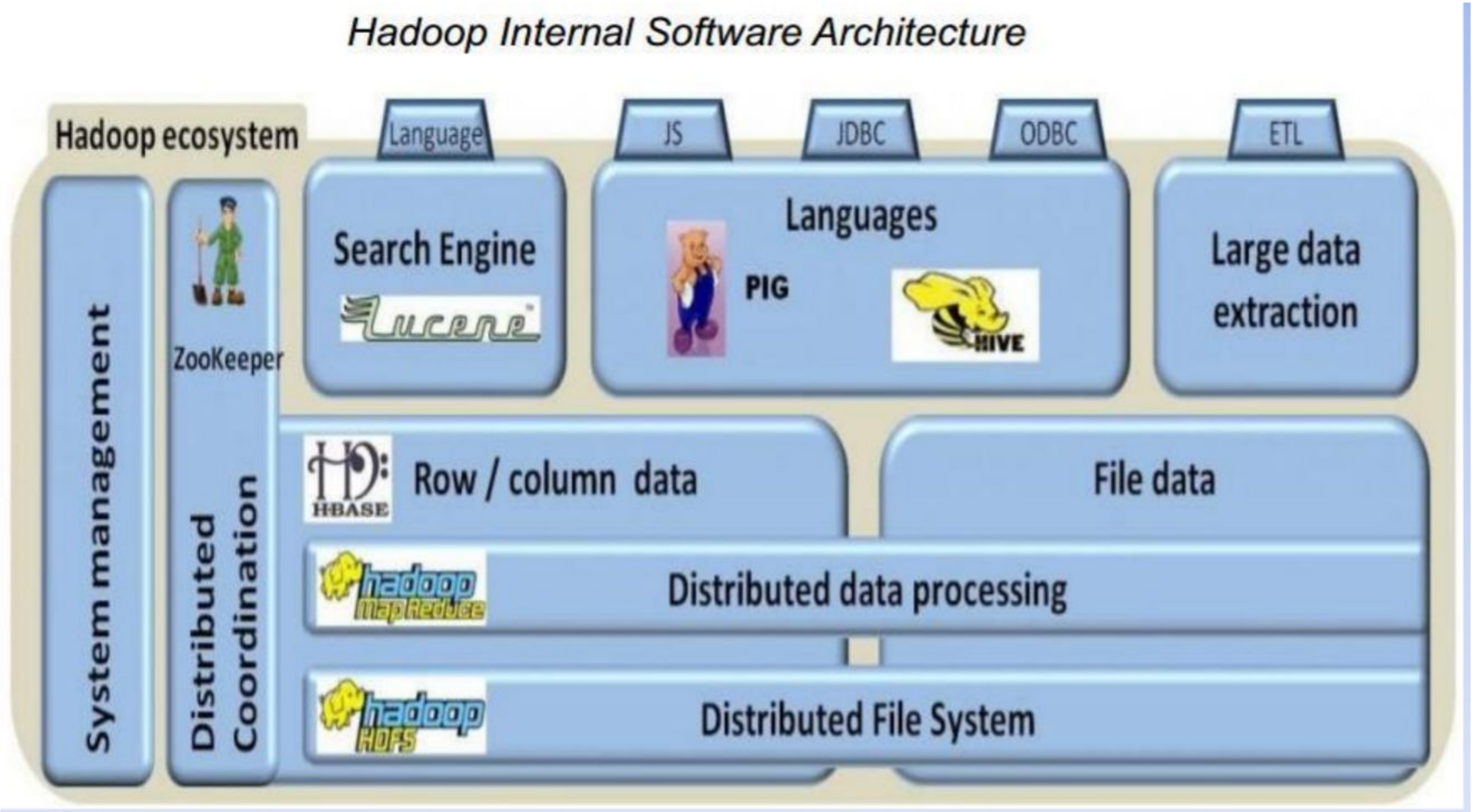
Apache Hadoop - Core components

- Hadoop Common
  - Has libraries and utilities required by other modules
- Hadoop Distributed File System (HDFS)
  - Distributed FS
  - Stores data on commodity machines
  - Very high aggregate bandwidth across clusters

Apache Hadoop - Core components

- Hadoop Mapreduce
  - A programming model for large scale data processing
- Hadoop YARN
  - Resource management platform
  - Manages computing resources in clusters
  - Schedules user applications on computing resources

Hadoop Core Components



Key Components

- **Hadoop Ecosystem & System Management**
- **ZooKeeper**: Manages distributed coordination among Hadoop components, ensuring reliability and fault tolerance.
- **System Management**: Involves tools for monitoring, managing, and optimizing Hadoop clusters.

Key Components

- **Storage Layer**
- **HDFS (Hadoop Distributed File System)**:
  - Serves as the foundational **storage system** in Hadoop.
  - Stores **large-scale file data** across multiple machines.
  - Provides **fault tolerance** by replicating data across nodes.

Key Components

- **Data Processing Layer**
- **MapReduce**:
  - A distributed processing model that **splits tasks into parallel jobs**.
  - Executes computations efficiently on massive datasets.
  - Works with HDFS to process data stored in a distributed manner.

Key Components

- **Data Storage & Querying**
- **HBase (Row/Column Data Storage)**:
  - A **NoSQL database** that provides real-time read/write access.
  - Stores data in a **key-value** format.
  - Works efficiently with MapReduce and HDFS.
- **File Data**:
  - Structured and unstructured file data stored in **HDFS**.

Key Components

- **Query & Processing Languages**
- **PIG** (Scripting Language for Big Data Processing):
  - A high-level scripting language designed for **data transformation**.
  - Converts scripts into **MapReduce** jobs.
  - Handles **semi-structured and structured** data efficiently.
- **Hive** (SQL-on-Hadoop for Querying Large Datasets):
  - Provides an **SQL-like** interface for querying big data.
  - Translates **HiveQL (SQL-like queries) into MapReduce jobs**.
  - Works well with **structured and semi-structured data**.

Key Components

- **Search & Indexing**
- **Lucene**:
    - A **search engine** used for indexing and searching large datasets.
    - Works with Hadoop to enable **text-based retrieval**.

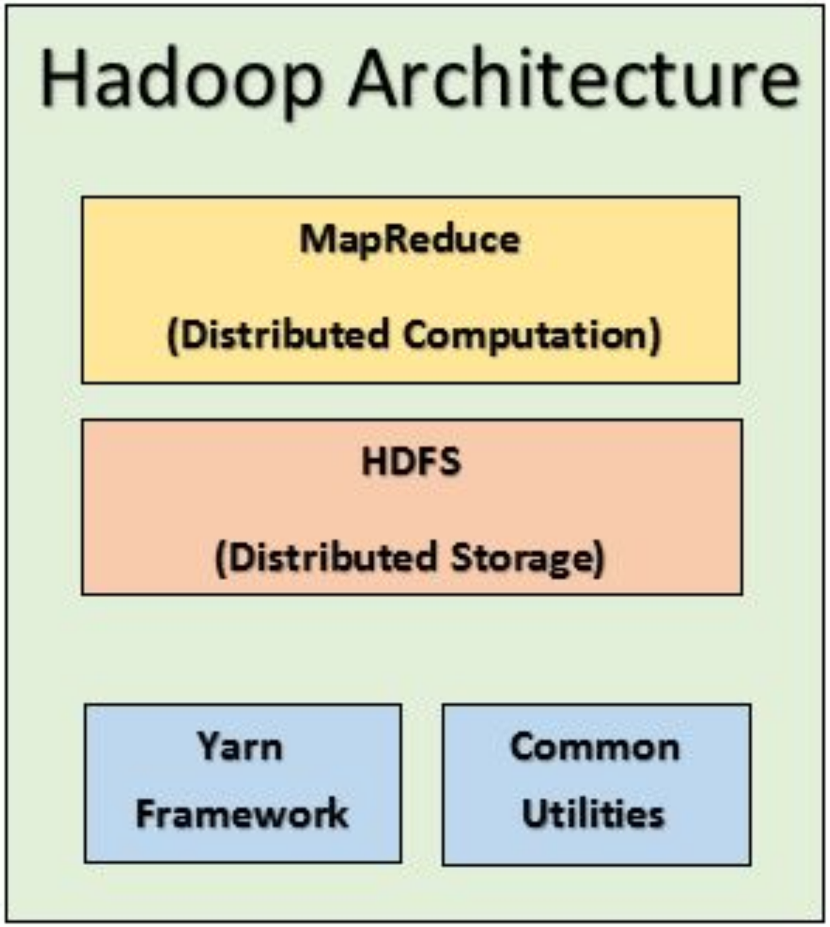
Key Components

- **Data Extraction & Connectivity**
- **Large Data Extraction (ETL - Extract, Transform, Load)**:
  - Involves processing large datasets before storing them in Hadoop.
  - Supports **integration with external databases**.
- **Connectivity Options (JS, JDBC, ODBC)**:
  - **JS (JavaScript)**, **JDBC (Java Database Connectivity)**, and **ODBC (Open Database Connectivity)** allow users to access Hadoop data from external applications.

Hadoop Core Components Summary

- Four major components
  - Storage --> HDFS
  - Processing --> MR
  - Scheduling --> YARN
  - Common utilities
- Mostly written in Java

Hadoop Core Components



Hadoop Architecture

MapReduce

(Distributed Computation)

HDFS

(Distributed Storage)

Yarn Framework

Common Utilities

File System

- Methods and data structures that an OS uses to keep track of files on a disk or partition
- Allocates space
- Responsible for organizing files and directories
- Unused space/Slack Space – Allocation size

Distributed File System

- File systems that manage storage across a n/w of machines
- Network based - so more complex than regular file systems
- Example: Tolerate node failure without data loss.
- Distribution is hidden and is not visible to the user
- High throughput data access

HDFS

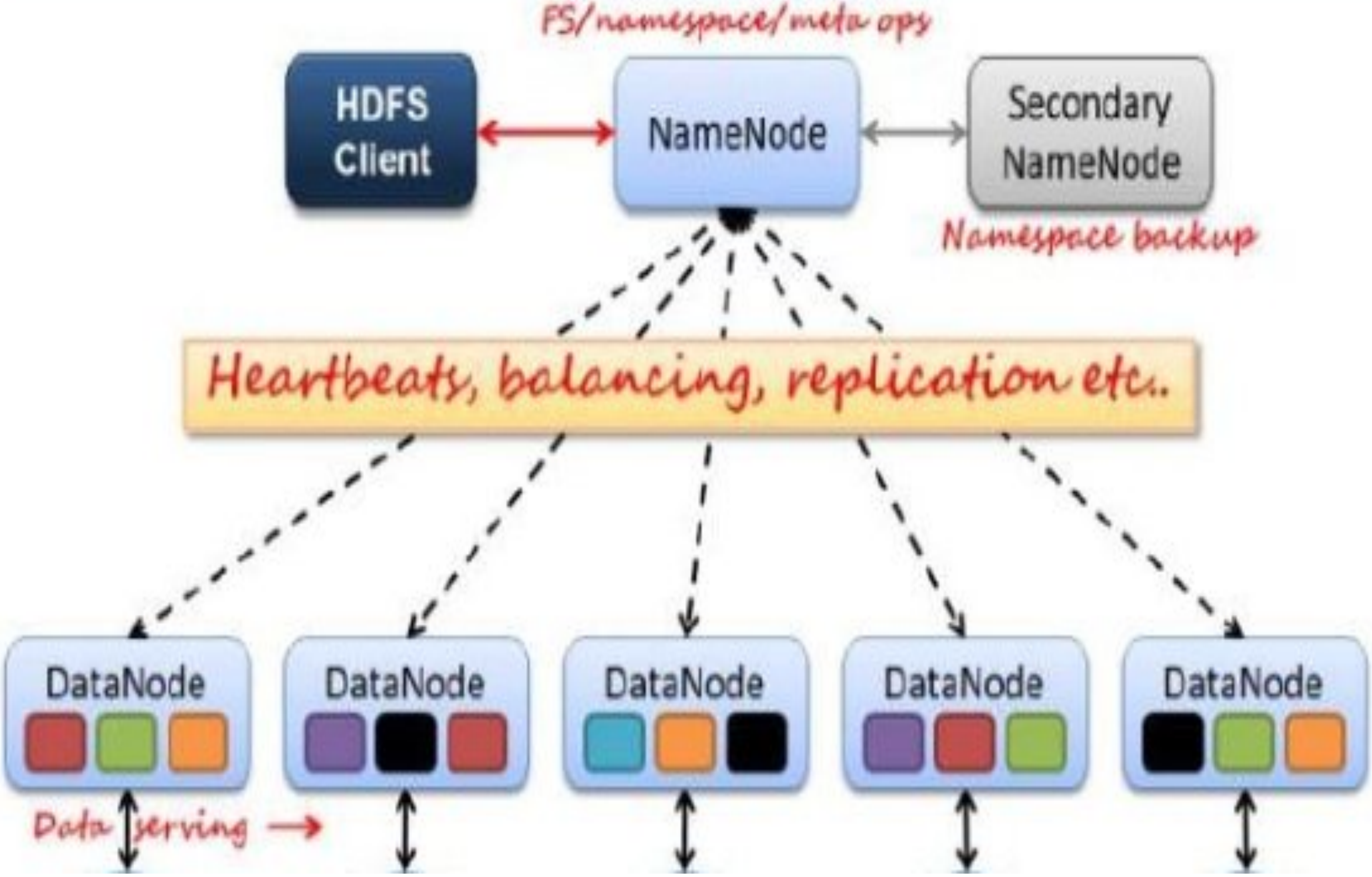
- How HDFS is different?
  - Highly fault tolerant
  - Designed for low cost hardware
  - High throughput access to data

Suitable For	Not Suitable For
Very large files	Low-latency data (Hbase is a better choice)
Streaming data access	Lots of small files
Commodity hardware	Multiple writes, arbitrary access

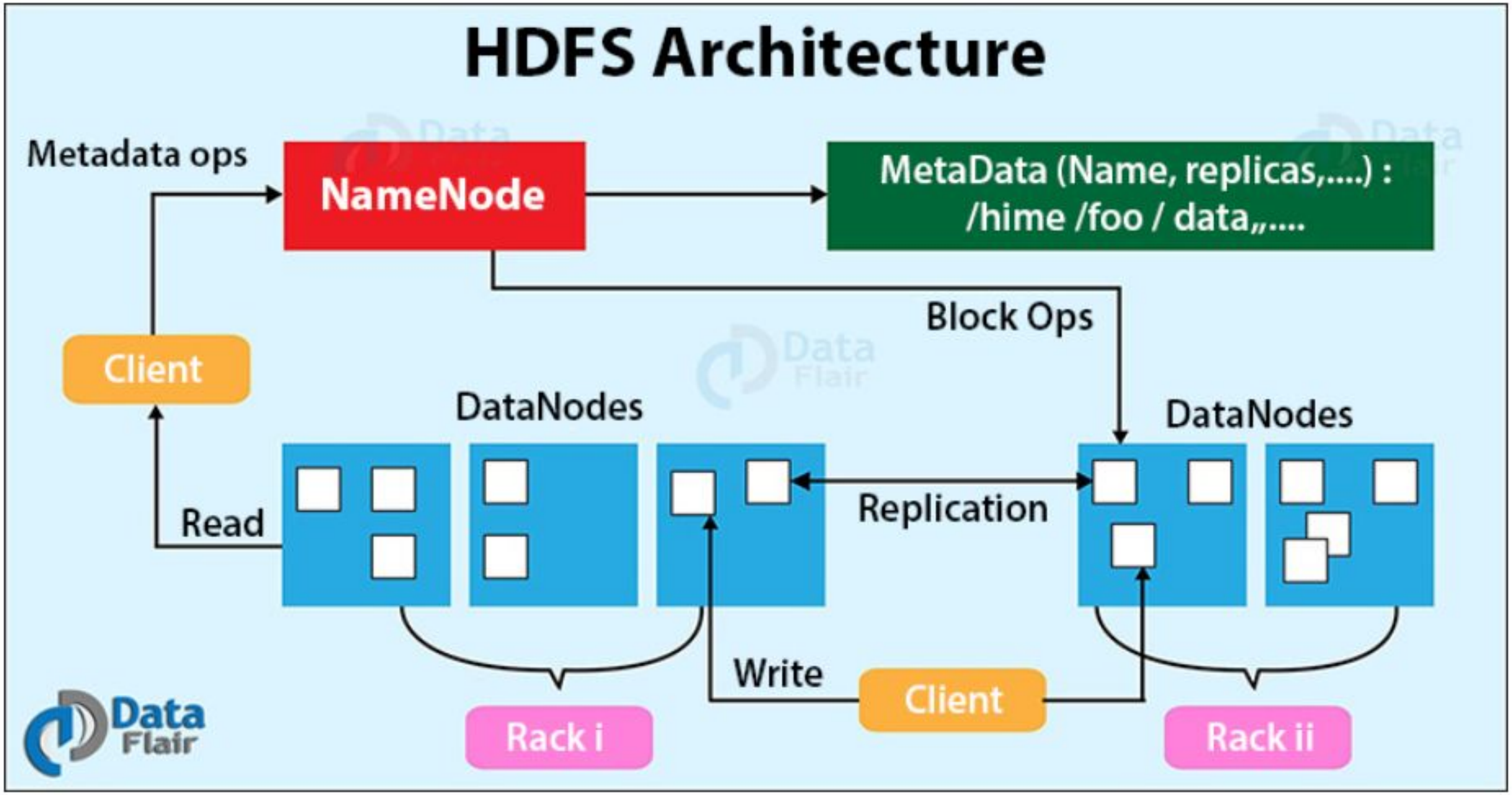
HDFS

- Follows a master slave architecture
- HDFS Cluster (Group of connected machines)
  - Name node
  - Secondary Name node
  - Multiple Data nodes

HDFS


FS/namespace/meta ops
HDFS Client
NameNode
Secondary NameNode
Namespace backup
Heartbeats, balancing, replication etc..
DataNode
DataNode
DataNode
DataNode
DataNode
Data serving

HDFS Architecture



HDFS Concepts

- Blocks
- Namenodes
- DataNodes
- Block Caching
- HDFS Federation
- HDFS High Availability

HDFS Blocks

- Disk Block – 512 bytes
- File System Block – few kBs
- HDFS Block size – 128 MB
- Files in HDFS are broken into block size chunks
- Advantages of block abstraction
  - One file can be larger than a hard disk
  - Blocks from a single file can be stored across multiple disks
  - Simplicity
  - Fit with replication

Name Node

- Master node
- Maps file to blocks and blocks to nodes
- Maintains file system tree and metadata for all files
- Two files
  - FSImage
  - Edit log
- Maintains status of data node
  - Heartbeats
  - Blockreport
- Manages replication
  - Block corruption
  - Data node failure
  - Disk failure

Name Node

- Data Integrity
  - Checksum for each block
- Balancing
  - Addition of new nodes
  - Decommisioning
  - Deletion of failed nodes
  - Maintain metadata

Name Node

- Without name node, HDFS cannot be used
- FS does not know to construct a file
- Should be resilient to failure
  - Metadata is persisted with back up
  - Secondary name node

Secondary Name node

- Present on a separate physical machine
- Does not act as a name node
- Periodically merges edit log with namespace image.
- Keeps a copy of the merged namespace image and edit log
- Data loss is certain

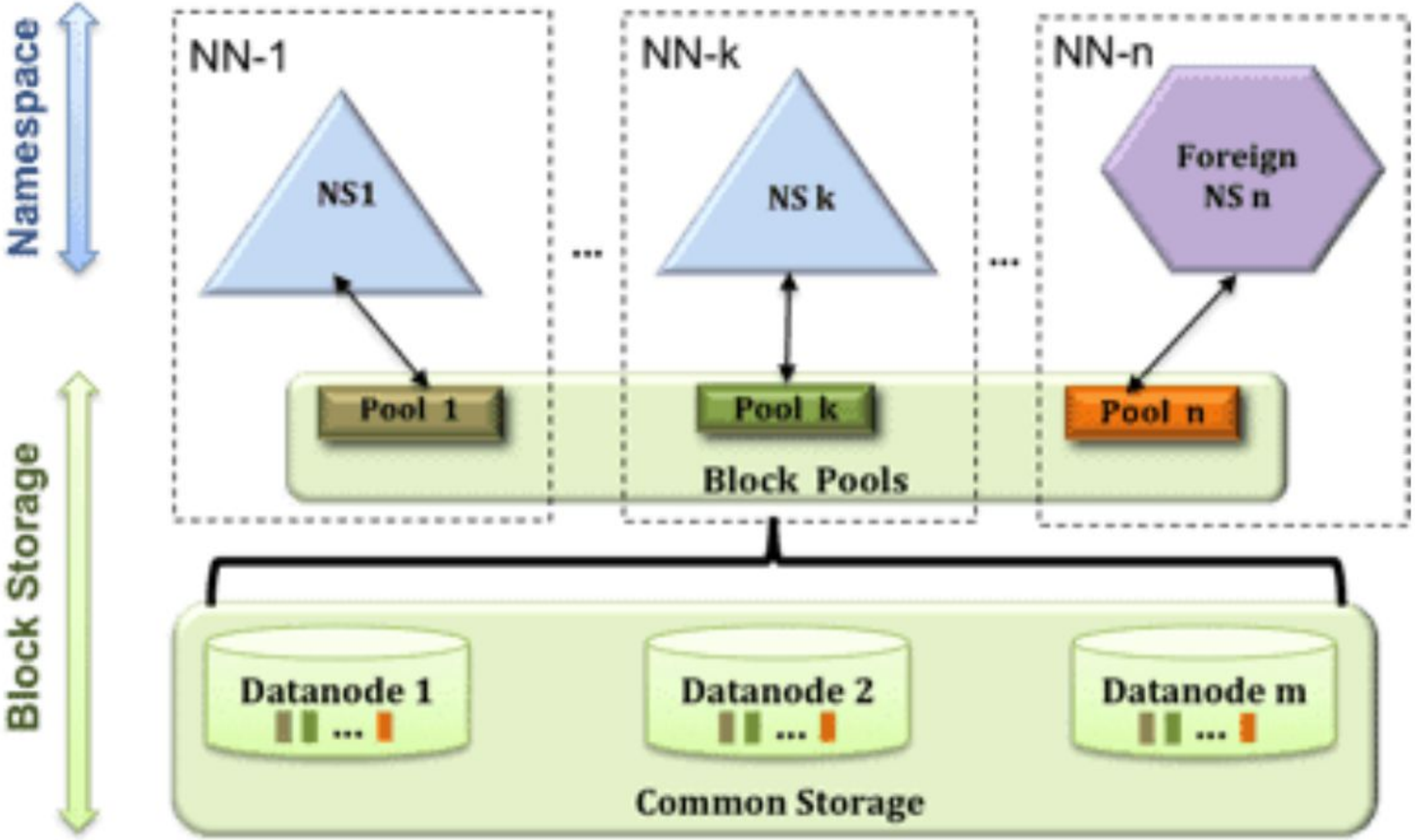
Block Caching

- Data nodes read blocks from disk
- Frequently accessed files – blocks are cached in memory
- Off-heap block cache
- Job schedulers run tasks on data node where data is cached for better performance
- Users or application can configure what to cache and how long

HDFS Federation

- Single namenode
  - Keeps track of every file and block
  - Very large clusters with many files?
    - Memory becomes a limiting factor for scaling
- Multiple name nodes
  - Each will manage a portion of file system
  - Each namenode manages a name space volume
    - Namespace (Dedicated for each NN – no overlap)
    - Block pool (DNs register with each NN and store blocks from multiple block pools)
  - Name nodes are independent and failure of one does not affect other

HDFS Federation



Courtesy: Cloudera

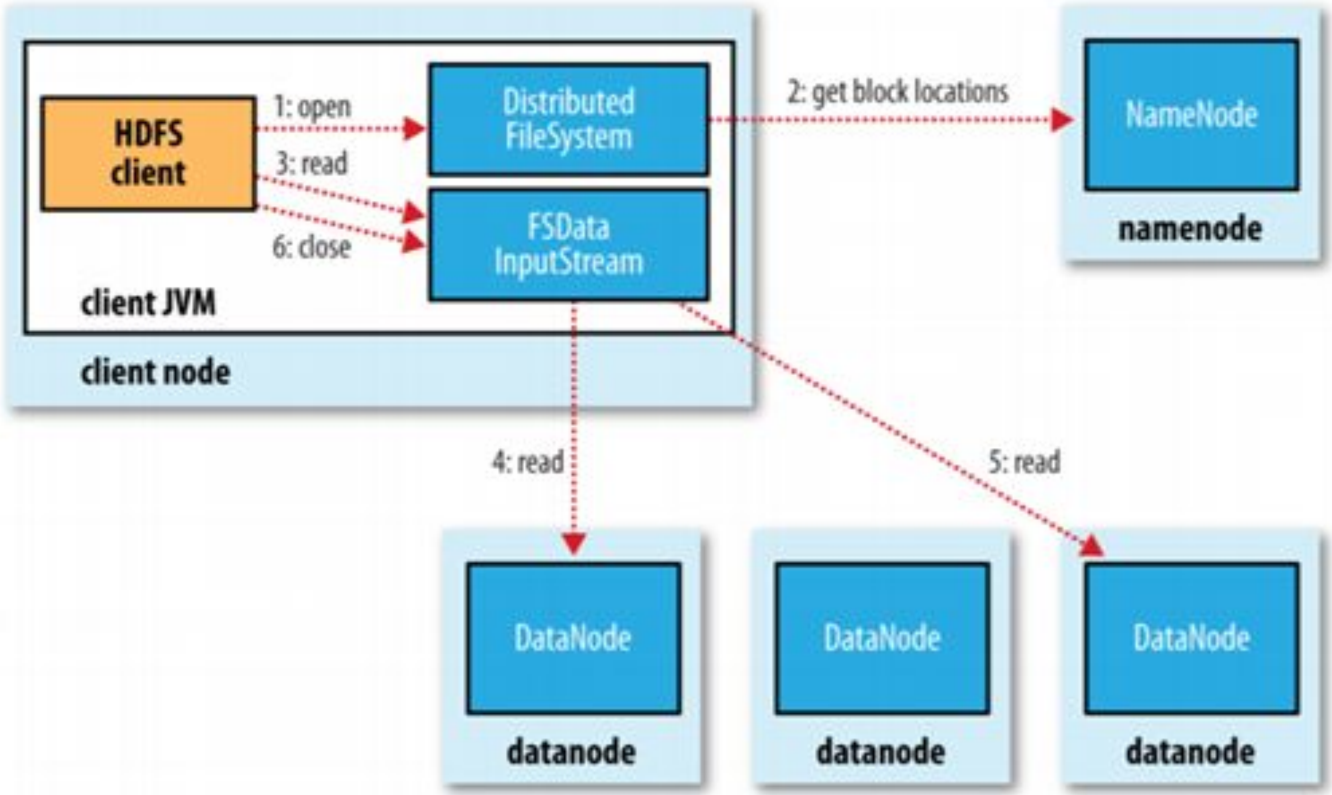
High Availability

- Secondary NN or Backup
- SPOF
- Recovery from a failure
  - Admin starts a new NN with FS replicas
  - Asks the DNs to use the new NN
- New NN is not ready till
  - loaded its namespace image into memory
  - replayed its edit log, and
  - received enough block reports from the datanodes to leave safe mode.

High Availability

- Takes 30 mins for large clusters
- From Hadoop 2 - pair of namenodes in an active-standby configuration
- Architectural changes required:
  - The namenodes must use highly available shared storage to share the edit log.
  - Datanodes must send block reports to both namenodes
  - Clients must be configured to handle namenode failover
  - Includes Secondary NN functionality

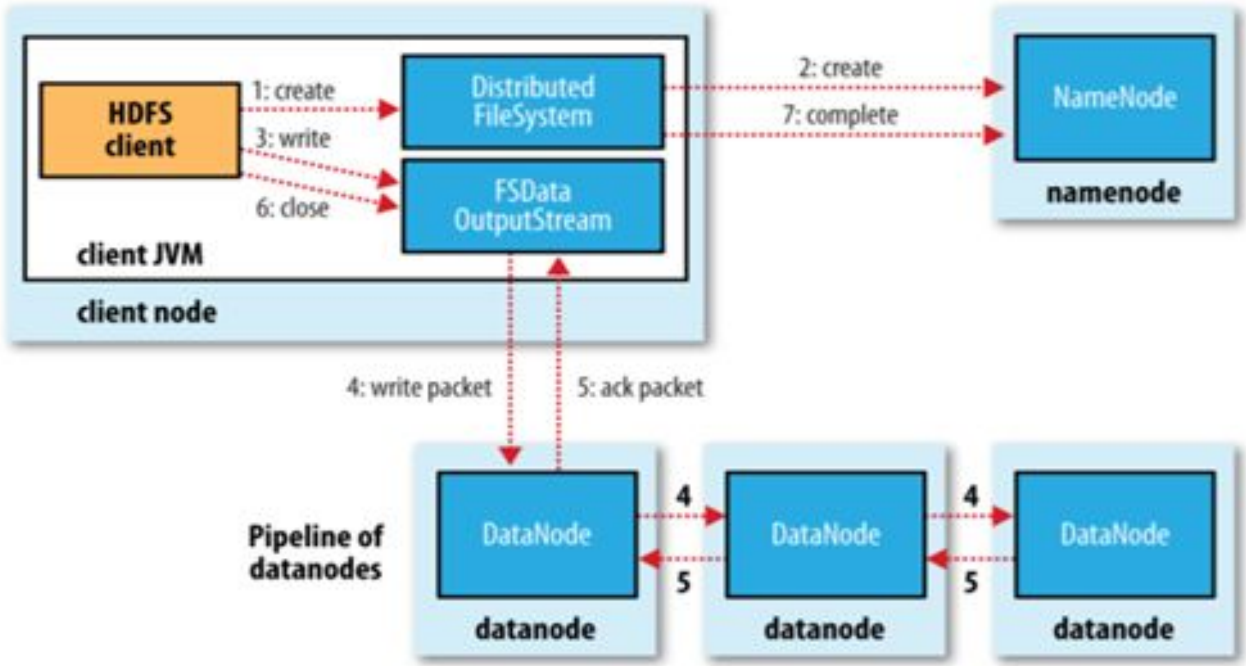
Anatomy of a file read



A client reading data from HDFS

Anatomy of a file write

ANATOMY OF A FILE WRITE



4

A client writing data to HDFS

Basic HDFS Shell Commands

- https://hadoop.apache.org/docs/r1.2.1/file_system_shell.html

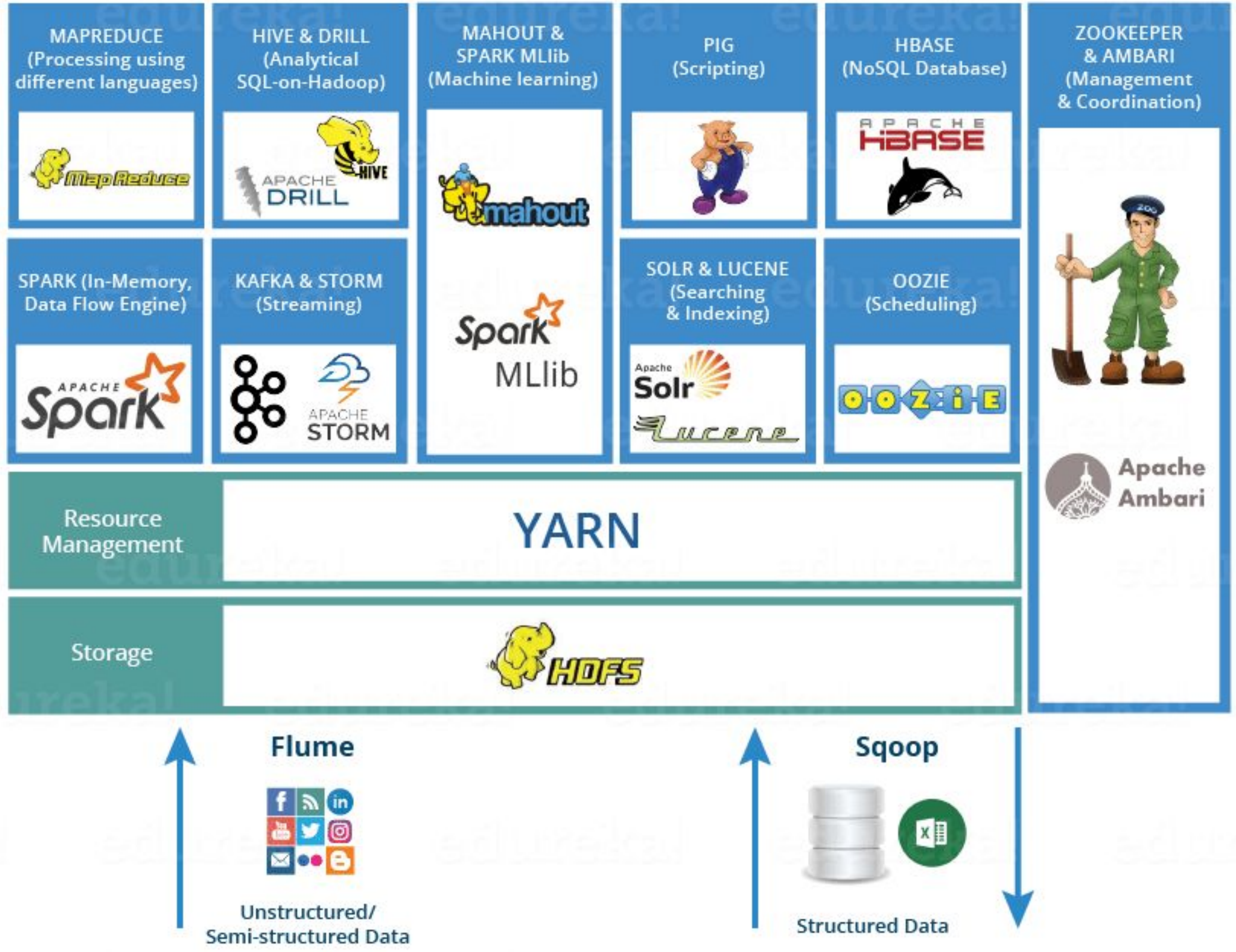
Hadoop Ecosystem

- *Hadoop Ecosystem* is a **platform or a suite** which provides various services to solve the big data problems.
- It includes Apache projects and various commercial tools and solutions.
- All these tools work collectively to provide services such as absorption, analysis, storage and maintenance of data etc.

Hadoop Ecosystem

- **HDFS:** Hadoop Distributed File System
- **YARN:** Yet Another Resource Negotiator
- **MapReduce:** Programming based Data Processing
- **Spark:** In-Memory data processing
- **PIG, HIVE:** Query based processing of data services
- **HBase:** NoSQL Database
- **Mahout, Spark MLLib:** Machine Learning algorithm libraries
- **Solar, Lucene:** Searching and Indexing
- **Zookeeper:** Managing cluster
- **Oozie:** Job Scheduling

Hadoop Ecosystem


MAPREDUCE (Processing using different languages)
MapReduce
HIVE & DRILL (Analytical SQL-on-Hadoop)
HIVE
APACHE DRILL
MAHOUT & SPARK MLlib (Machine learning)
mahout
Spark MLlib
PIG (Scripting)
HBASE (NoSQL Database)
APACHE HBASE
ZOOKEEPER & AMBARI (Management & Coordination)
ZOO
Apache Ambari
SPARK (In-Memory, Data Flow Engine)
APACHE Spark
KAFKA & STORM (Streaming)
APACHE STORM
SOLR & LUCENE (Searching & Indexing)
Apache Solr
Lucene
OOZIE (Scheduling)
OOZIE
Resource Management
YARN
Storage
HDFS
Flume
Unstructured/ Semi-structured Data
Sqoop
Structured Data
edureka!

Courtesy: Cloudera