Fundamentals of Data Engineering

Trainer: Nilesh Ghule

Introduction

- **Big Data Fundamentals**
  - Evolution of Data Engg
  - V's: Volume, Velocity, Variety, Veracity, Value
  - Type: Structured / Semi-structured / Unstructured
- **Databases**
  - RDBMS - ACID, SQL (basic concept only)
  - NoSQL - BASE, CAP theorem
- **Data warehouse - OLAP vs OLTP**
  - Data cleansing, Data transformations and Data modelling
  - Data warehouse vs Data mart vs Data lake
- **Data Engineering Life Cycle**
  - Source → Ingestion → Storage → Transformation → Serving
  - Ingestion: ETL vs ELT
  - Storage: Distributed storage, Storage services
  - Processing: Batch vs Stream
- **Big Data Technologies**
  - Frameworks: Hadoop, Hive, Spark, Kafka
  - Programming Languages: Python, Java, Scala
  - Job profiles: Data engineer, Data architects, Database/DWH engineer.
  - Applications: Retail, Healthcare, Telecom, Finance, Media, etc.

History of Big Data / Data Engg

Database
&
Warehouse

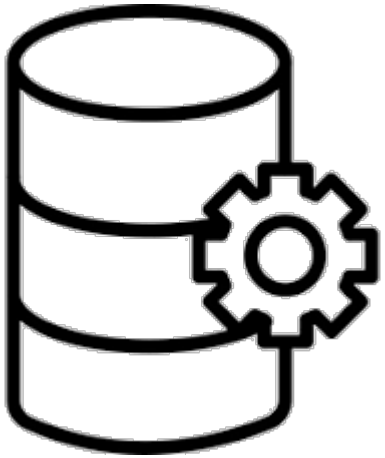
1970
⋮
1990 +

Internet
&
DotCom

1990
⋮
2000

NoSQL
Database

1998
⋮
2004 +



MPP &
Big Data
Tech

Massive Parallel Processing
+ Distributed Computing

2002
⋮
2012 +

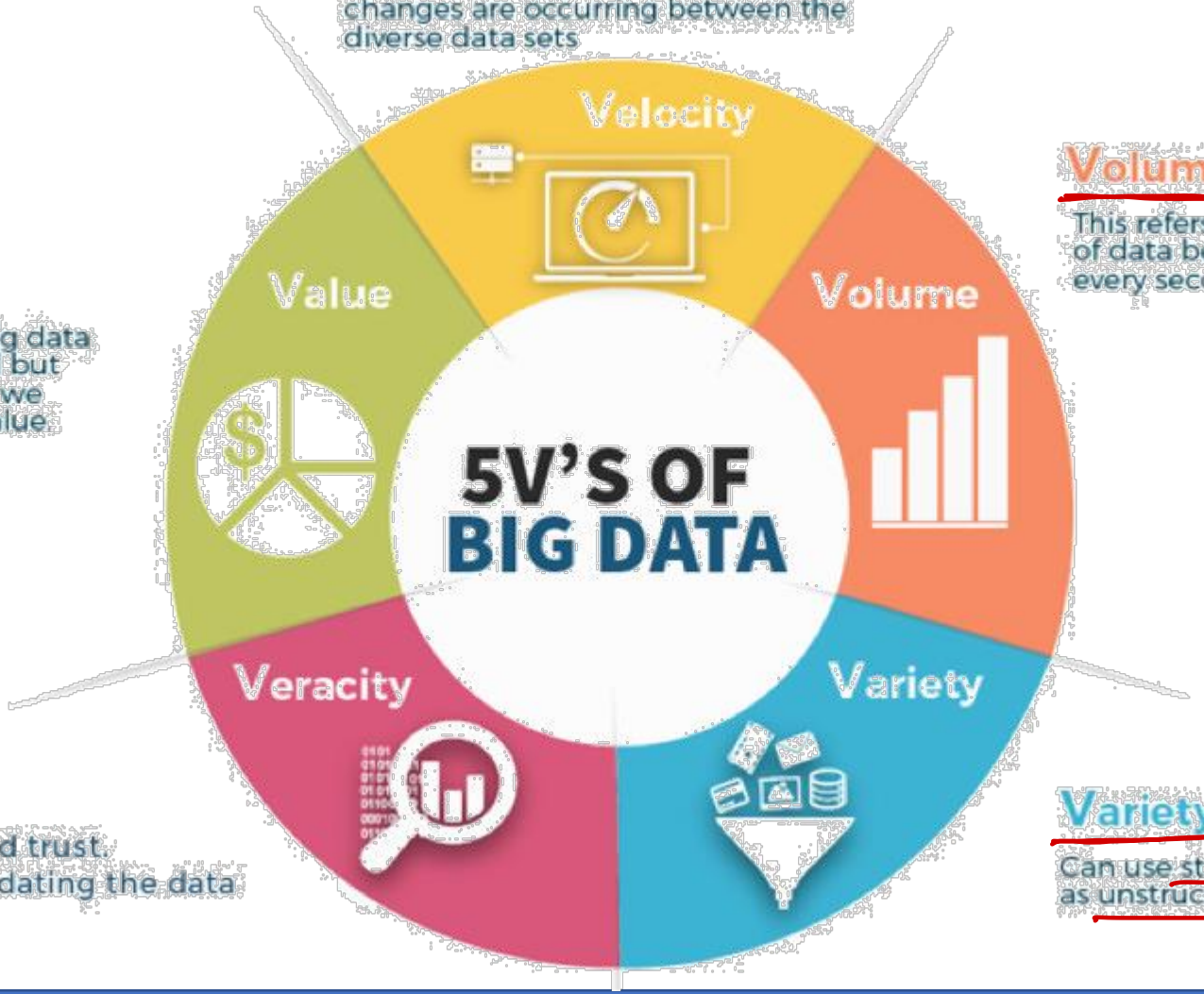
Cloud
Computing

2006 +
⋮
'infrastructure

Big Data characteristics

Velocity

Speed at which data is emanating and changes are occurring between the diverse data sets



Volume TB+

This refers to the sheer volume of data being generated every second

Value

Having access to big data is all well and good but that's only useful if we can turn it into a value

Veracity

Data reliability and trust
Verifying and validating the data

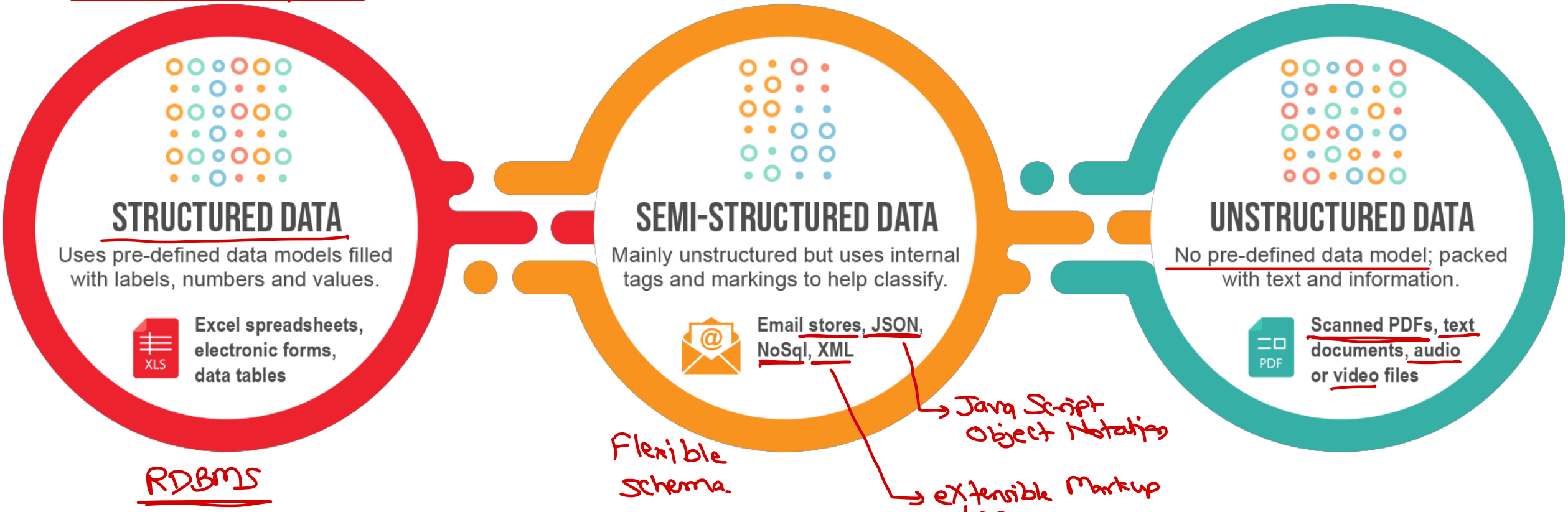
Variety

Can use structured as well as unstructured data

Types of Data

roll	name	marks	std
1	~	~	~
2	~	~	~
3	~	~	~

Product specification
- mobile, tv, computers
- grocery, cloths, ...



RDBMS

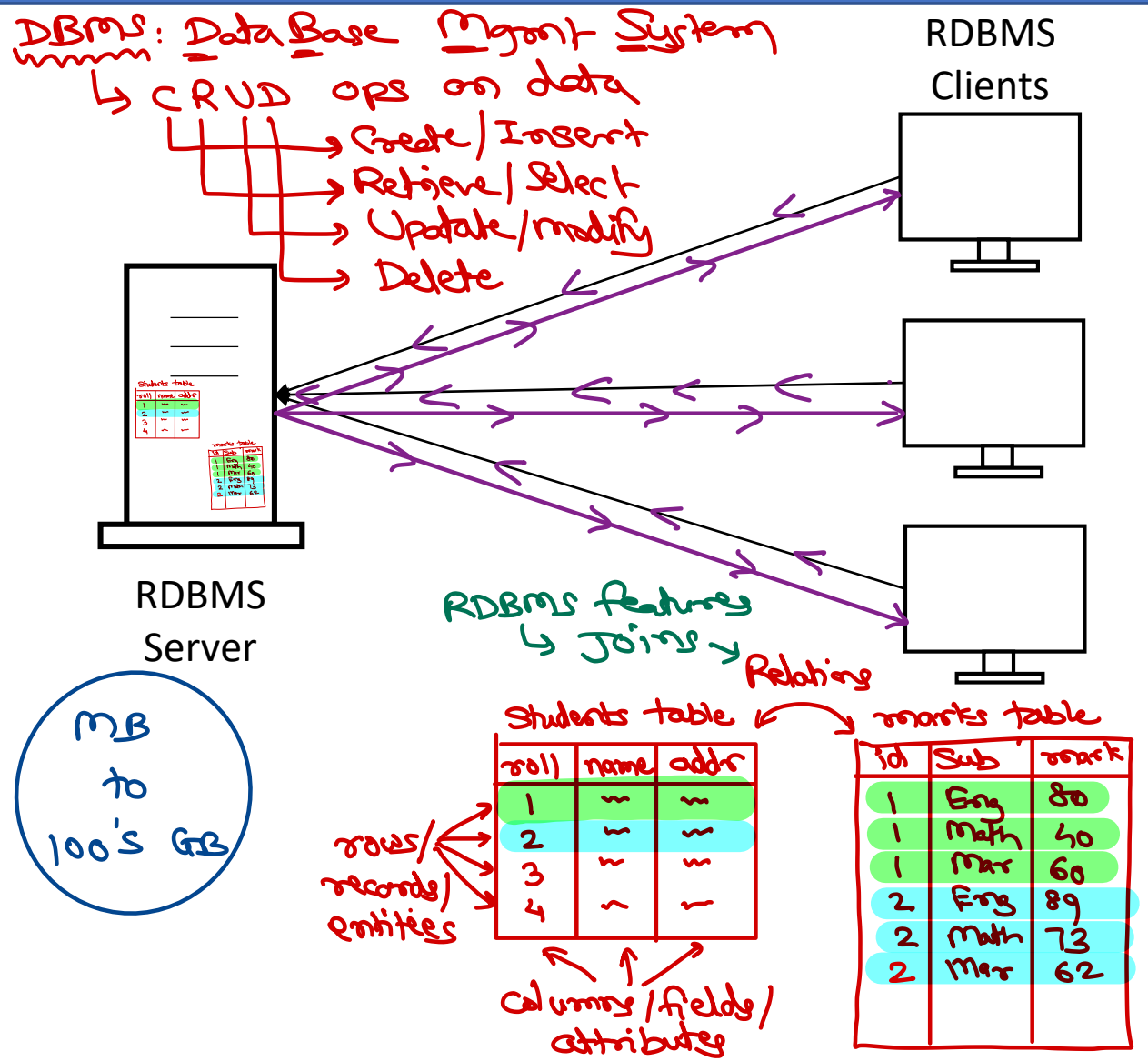
Flexible schema.

Java Script Object Notation

eXtensible Markup Language

RDBMS

- Every enterprise application need to manage data.
- RDBMS is relational DBMS than manages structured data.
- Data is organized into tables, rows and columns. Tables are related to each other.
- All enterprise RDBMS follow server-client architecture, have built-in relational capabilities, fully ACID transactions, based on Codd's rules.
- DB2, Oracle, MS-SQL, MySQL, Postgre-SQL, MS-Access, SQLite, etc.



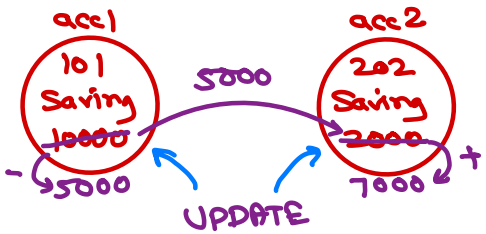
SQL – Structured Query language

IBM

- RDBMS data is processed with SQL queries.
- Originally it was named as RQBE (Relational Query By Example).
- ANSI standardised in 1987.
- Five major categories:
  - DDL: Data Definition Language e.g. CREATE, ALTER, DROP, RENAME.
    - CREATE TABLE people(id INT, name CHAR(40), birth DATE);
  - DML: Data Manipulation Language e.g. INSERT, UPDATE, DELETE.
    - INSERT INTO people VALUES(1, 'Nilesh', '1983-09-28');
    - UPDATE people SET name='NILESH' WHERE id=1;
    - DELETE FROM people WHERE id=1;
  - DQL: Data Query Language e.g. SELECT.
    - SELECT * FROM people;
  - DCL: Data Control Language e.g. CREATE USER, GRANT, REVOKE.
  - TCL: Transaction Control Language e.g. SAVEPOINT, COMMIT, ROLLBACK.

Transactions

↳ Funds transfer



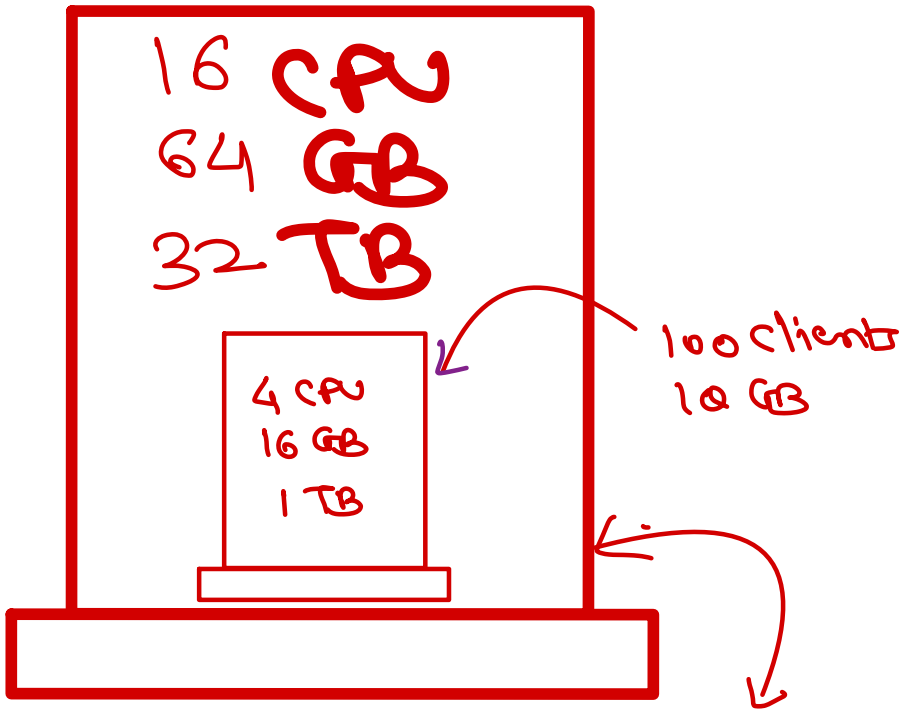
Tx is set of DML queries executed as a single unit. Either all will succeed (commit) or all will discard (rollback).

Atomic: Full complete/ not partial.

Consistent: Same result for all clients

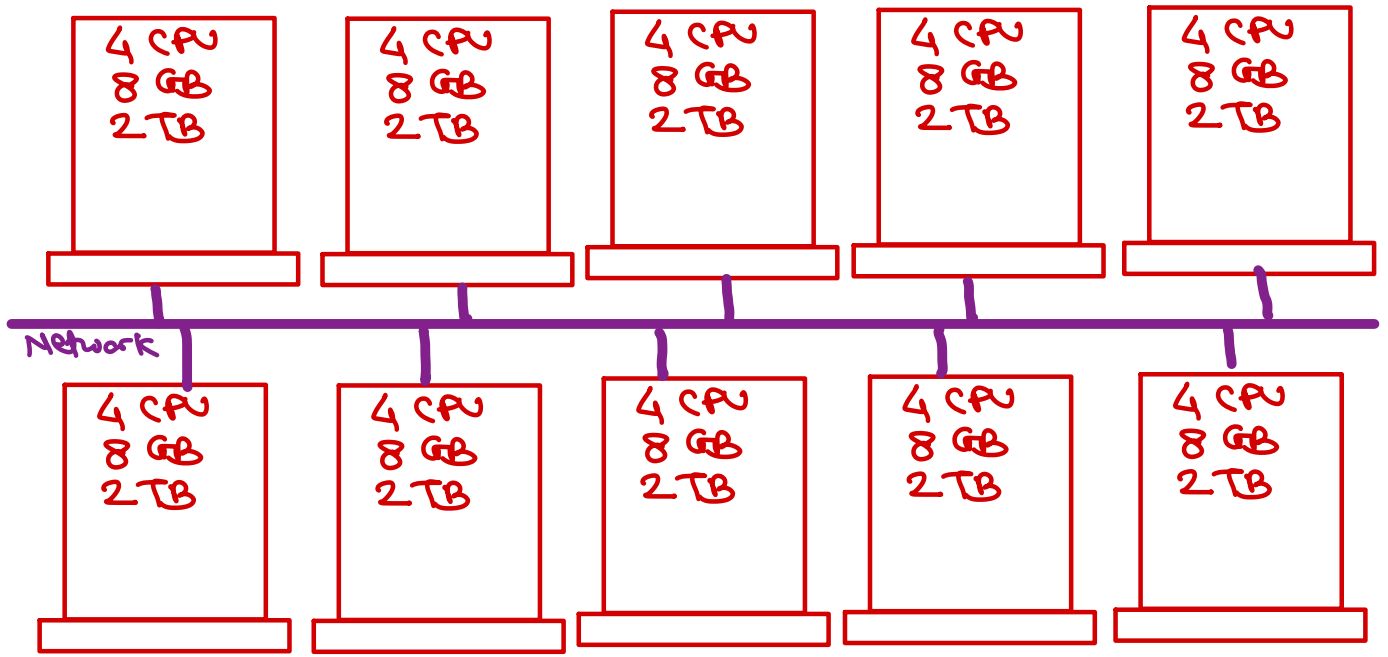
Isolation: multiple tx run independent.

Durable: All changes must be saved.



Vertical scaling

↳ followed by traditional RDBMS



Cluster (Set of computers in a network for a dedicated fn)

Horizontal Scaling (Distributed)

↳ followed by NoSQL, Big data

NoSQL Databases

- Stands for Not Only SQL (Beyond the SQL)
- Manages structured and semi-structured data.
- Prioritizes high performance, high availability and scalability
- Designed for Horizontal scaling. Reliable, fault tolerant, Better performance/Speed.
- No declarative query language
- Use if: Huge data (TBs), Many Read/Write ops, Scalable, Flexible schema.
- Don't use if: Need ACID transactions, Need high consistency, Multiple relations
- BASE transactions
- Based on CAP Theorem

10000's rd/wr ops/sec

24x7

1998 - Carlo Strozzi - NoSQL

(100 GB to 100's TB+)
num of clients.

each db have different language.

joins

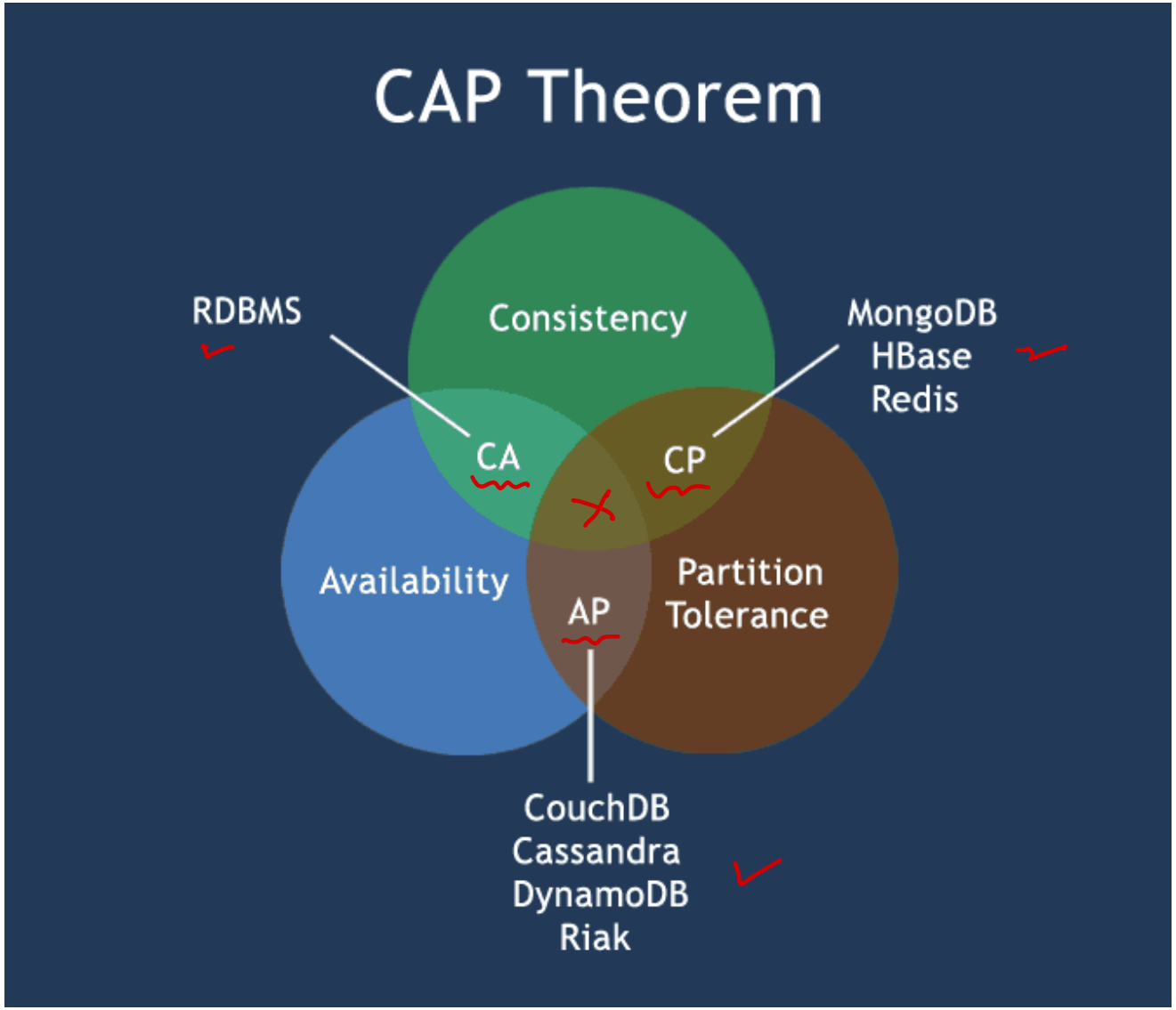
BA = Basically Available = 24x7

S = Soft State = Data moved in cluster without user interaction.

E = Eventual Consistency = all clients see same data eventually (after time)

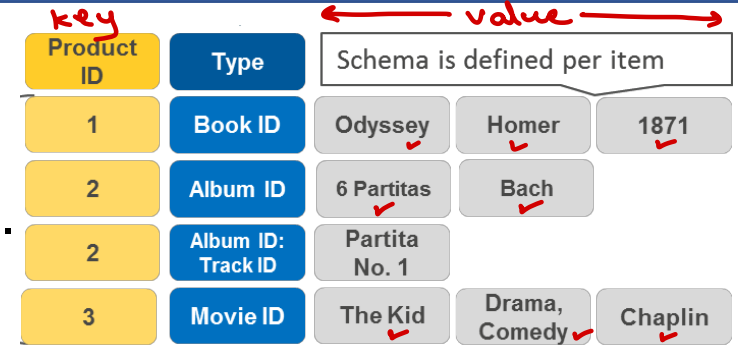
CAP Theorem

- Consistency - Data is consistent after operation. After an update operation, all clients see the same data.
- Availability - System is always on (i.e. service guarantee), no downtime.
- Partition Tolerance - System continues to function even the communication among the servers is unreliable.

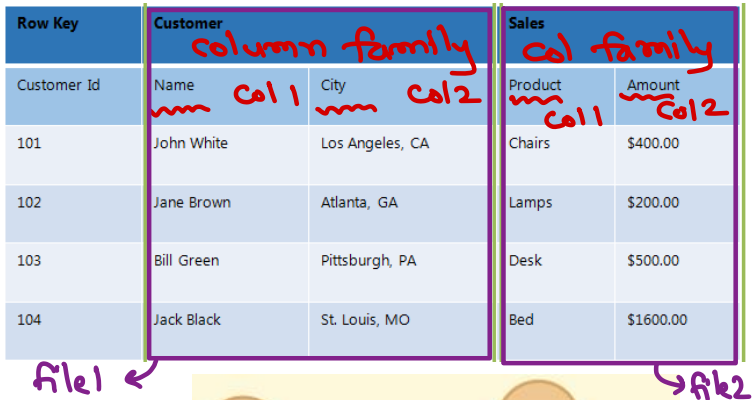


NoSQL Databases

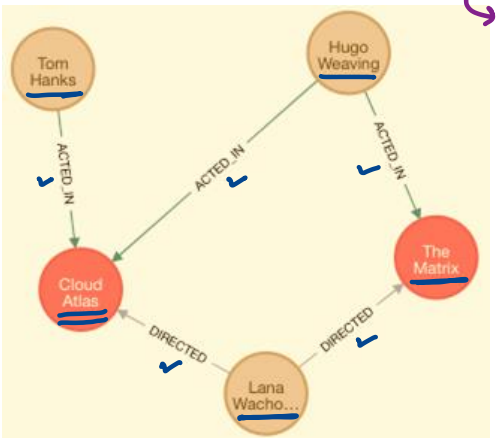
- Key-value databases - e.g. redis, dynamodb, riak, ...
  - Based on Amazon's Dynamo database.
  - Keys are unique and values can be of any type i.e. JSON, BLOB, etc.
  - Implemented as big distributed hash-table for fast searching.



- Wide Column databases - e.g. hbase, cassandra, bigtable, .
  - Values of columns are stored contiguously.
  - Better performance while accessing few columns and aggregations.
  - Good for data-warehousing, business intelligence, CRM, ...



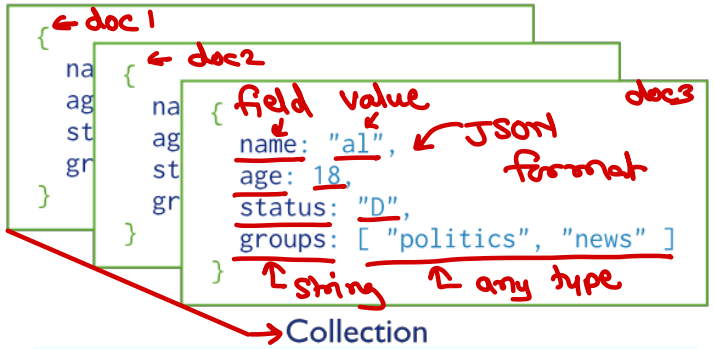
- Graph databases - e.g. Neo4J, Titan, …
  - Graph is collection of vertices and edges.
  - Excellent performance, while dealing with all relations of an entity (irrespective of size of data).



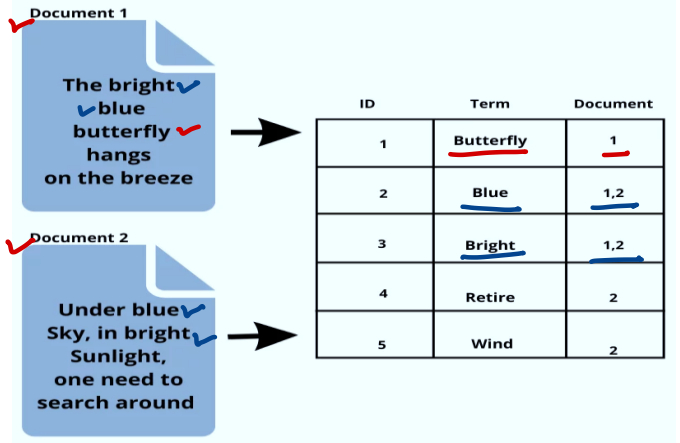
NoSQL Databases

- Document oriented databases - e.g. MongoDb, CouchDb, …
  - Document contains data as key-value pair as JSON or XML.
  - Document schema is flexible & are added in collection for processing.

RDBMS: Row / Table → Fixed schema
Mongo : Doc / Collection → Flexible schema



- Search databases – e.g. Elasticsearch, Solr, Lucene, …
  - For faster search – Text search, Log analysis.
  - Indexed, Exact/Fuzzy matches, Anomaly detection, Analytics.

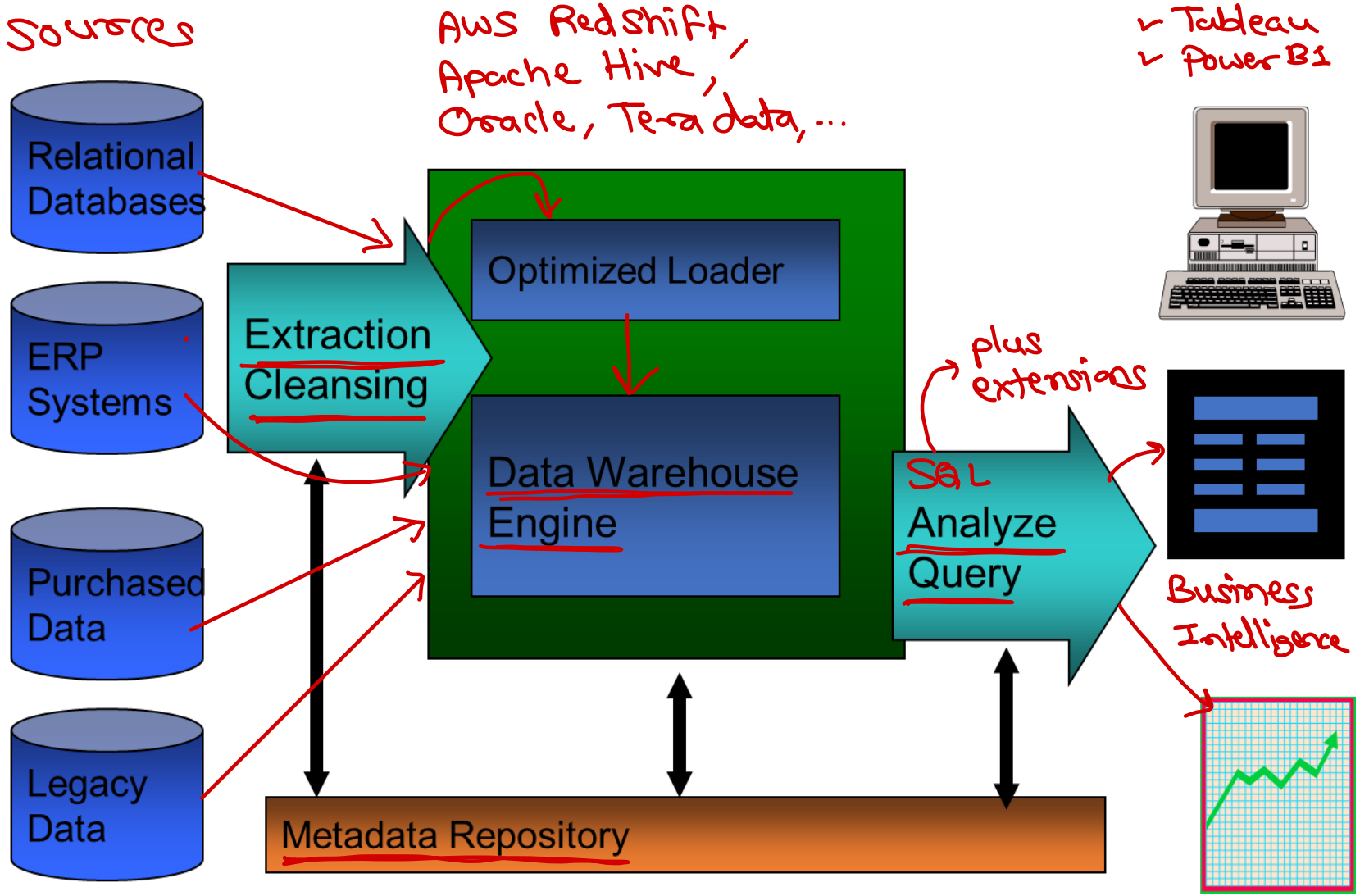


- Time series databases – e.g. Influx, Druid, …
  - Values organized by time like stock market, weather, …
  - Optimized for retrieval, statistical processing, …
  - Used for measurement data (weather, …) and event-based data (accidents, …)

Date	Ozone ($\mu g/m^3$)	Temperature (°C)	Relative humidity (%)	*n* deaths
1 Jan 2002	4.59	−0.2	75.7	199
2 Jan 2002	4.88	0.1	77.5	231
3 Jan 2002	4.71	0.9	81.3	210
4 Jan 2002	4.14	0.5	85.4	203
5 Jan 2002	2.01	4.3	93.5	224
6 Jan 2002	2.4	7.1	96.4	198

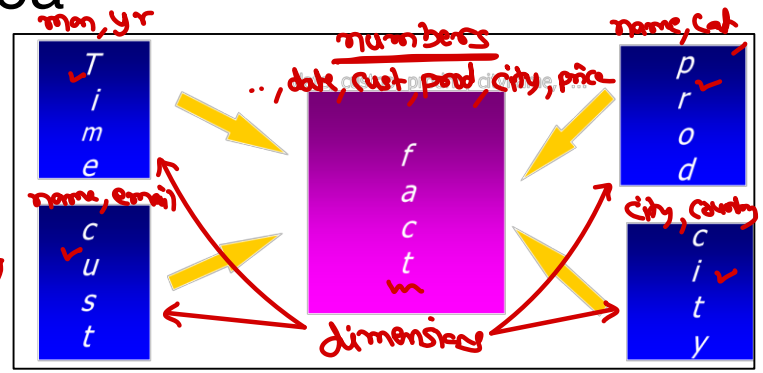
Data warehousing

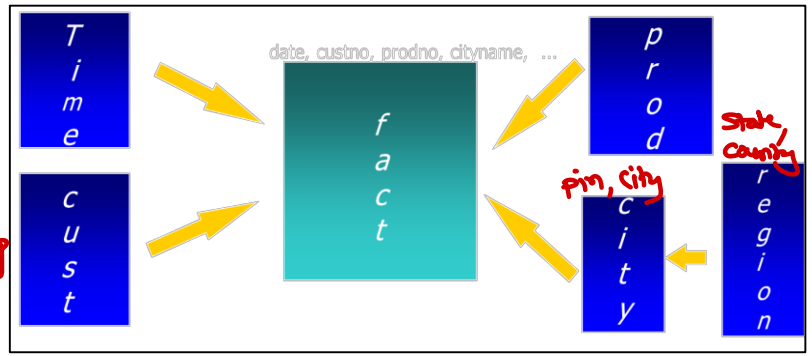
- Data warehouse is a single, complete and consistent store of data obtained from a variety of different sources made available to end users in a what they can understand and use in a business context.
- Data warehousing is a process of transforming data into information and making it available to users in a timely enough manner to make a difference.

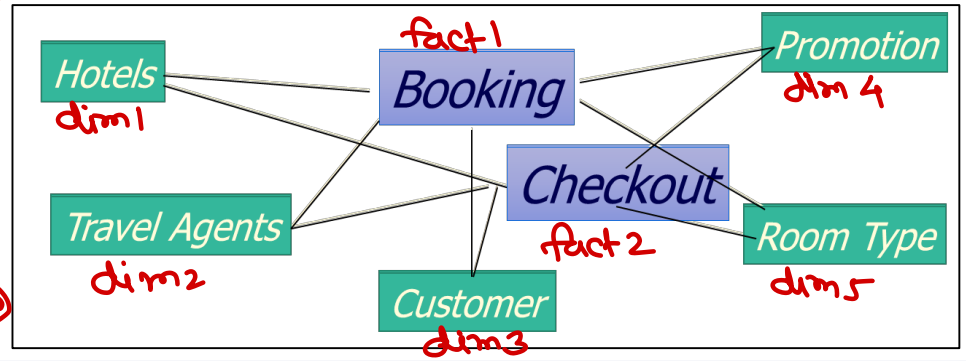


Extract – Transform – Load

- Extracting: Extract data from various sources into staging area
- Conditioning: Conversion of data types to fit warehouse.
- House holding: Grouping similar data
- Enrichment: Add relevant data from external sources
- Scoring: Computation of probability of an event
- Scrubbing: Data cleaning: find duplicate, missing data
- Merging: Merging data from various sources.
- De-normalize: Duplicate data to reduce joins.
- Loading: Load data into warehouse models like Star, Snowflake, Fact constellation.
- Delta Updating: Incremental data uploading
- Partitioning: Dividing data in logical parts to improve performance





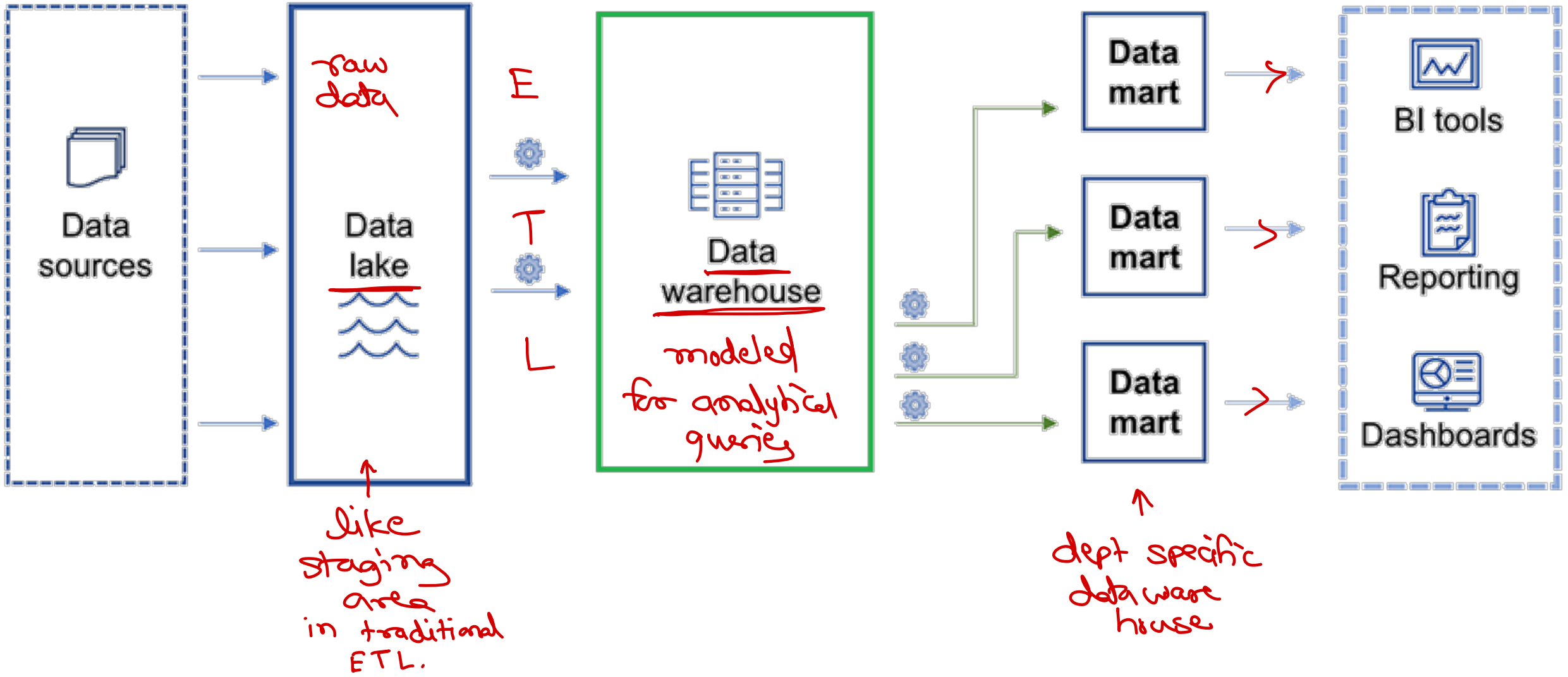


OLTP (Database) vs OLAP (Data warehouse)

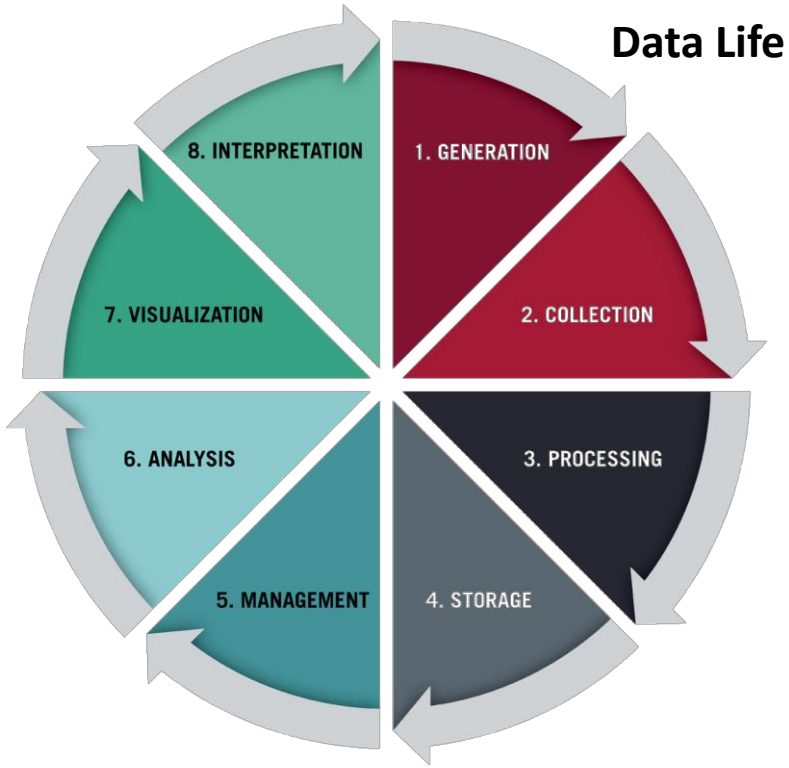
- Online Transaction Processing
  RDBMS / NoSQL
- Modeled to run the business
- Detailed/Transactional normalized real-time data (no duplicates)
- Transaction performance DML / Search
- Read/Write operations
- Isolated data (Application specific) – Limited data (100 MB to 100 GB)

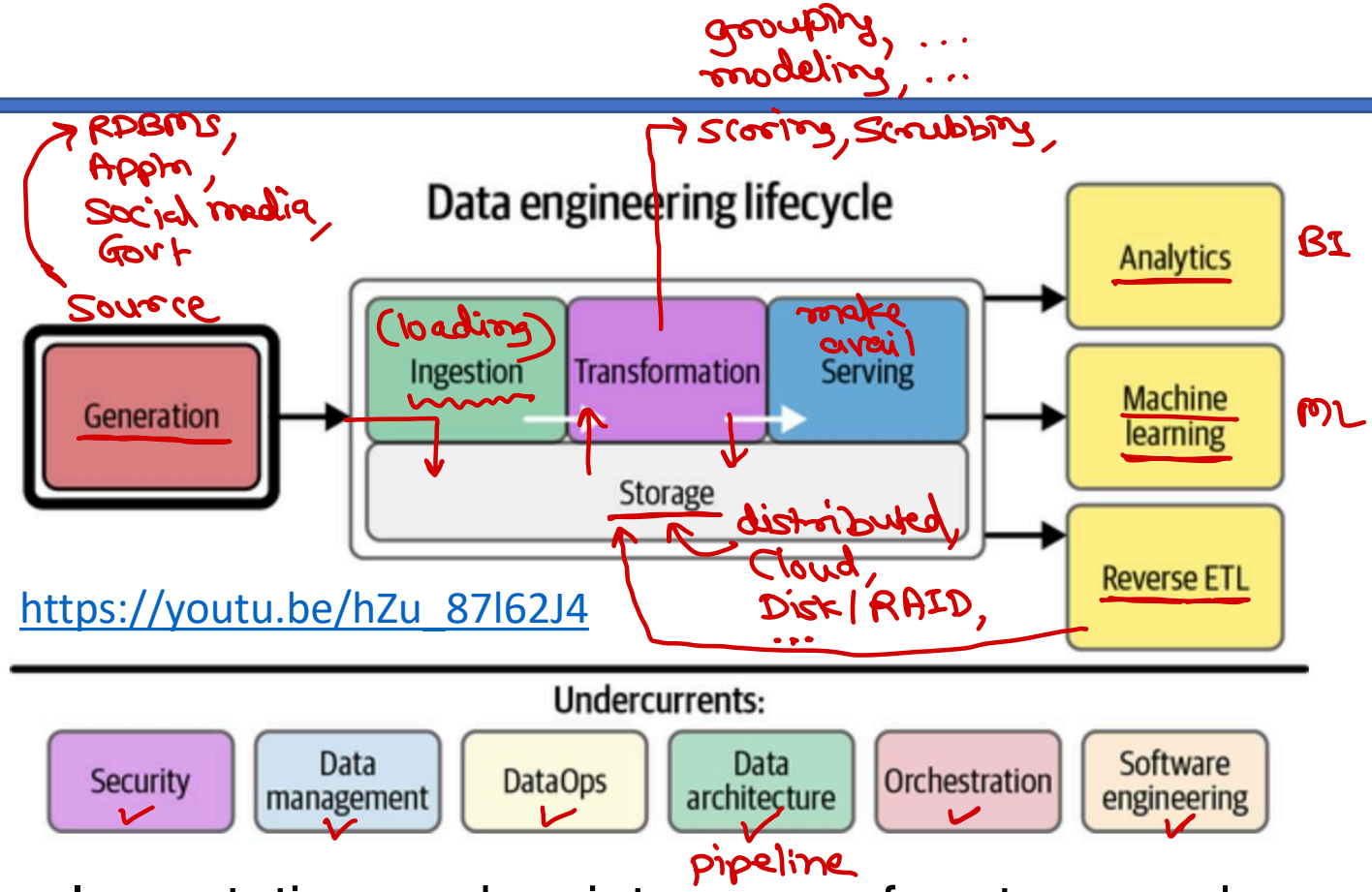
- Online Analytical Processing
  Data warehouse
- Modeled to analyze/optimize business
- Summarized/refined redundant snapshot data
- Analytical query performance
- Mostly Read operations update X delete X
- Integrated data (from all sources) – Huge data (100 GB to Few TB)

Data lake vs Data warehouse vs Data mart



Data engineering





- Data engineering is the development, implementation, and maintenance of systems and processes that take in raw data and produce high-quality, consistent information that supports downstream use cases, such as analysis and machine learning.
- Data engineer manages data engineering lifecycle, beginning with getting data from source systems & ending with serving data for use cases, such as analysis or machine learning.

Traditional ETL vs Hadoop ELT

- ETL stands for Extract, Transform and Load.
- The ETL process typically extracts data from the source/transactional systems, transforms it to fit the model of data-warehouse and finally loads it to the data warehouse.
- The transformation process involves cleansing, enriching and applying transformations to create desired output.
- Data is usually dumped to a staging area after extraction.
- ELT stands for Extract, Load and Transform.
- As opposed to loading just the transformed data in the target systems, the ELT process loads the entire data into the data lake. This results in faster load times.
- The load process can also perform some basic validations and data cleansing rules.
- The data is then transformed for analytical reporting as per demand.

Data storage

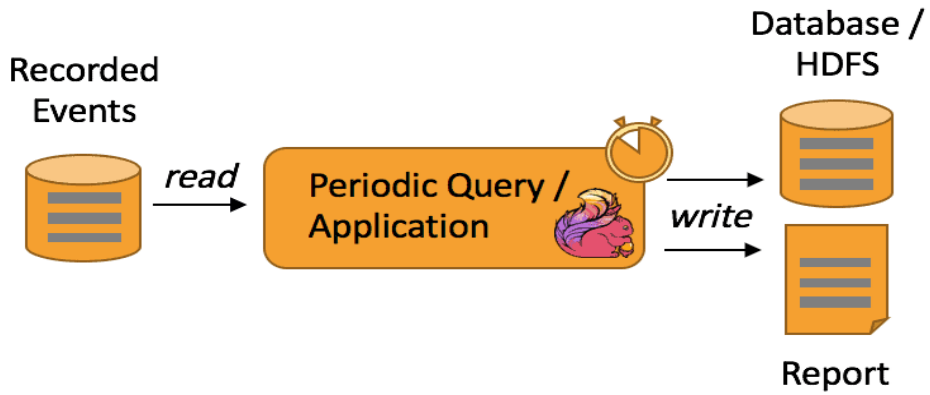
- Data storage is related to multiple stages in data engineering life cycle i.e. ingestion, transformation and serving.
- Storage needs to be selected based on read/write requirement, speed, durability, consistency, availability, scalability, fault tolerance, … factors.
- Storage tradeoffs
  - Local storage vs Distributed storage
  - Strong consistency vs Eventual consistency
- Storage options are: File storage, Local disk storage, Network attached storage (NAS), Cloud file systems (S3/Blob), Block storage, RAID, Storage area network (SAN), Object storage, HDFS, Streaming storage.

Batch processing vs Stream processing

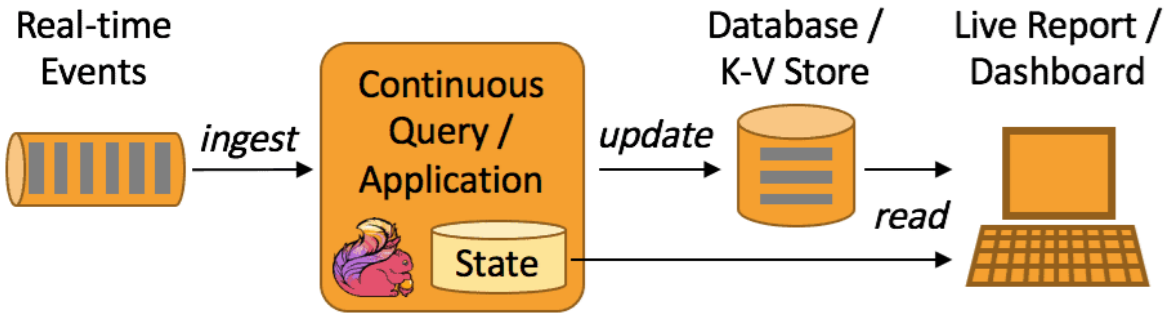
- Processing finite set of data (data at rest).
- Incremental data load is managed by programmer.
- Cluster planned as per data size. High throughput.
- Job run once per batch.

- Processing live stream of data (data in motion).
- Data processing is managed by the framework.
- Less throughput.
- Job is running forever.

Batch Processing



Stream Processing

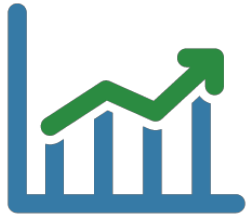


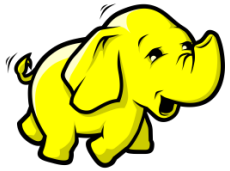
Big Data & Analytics Spectrum

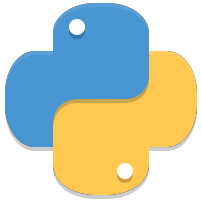
- Data storage
  - RDBMS & NoSQL databases
  - Data warehouse
  - S3, DFS, …
- Data Analysis & visualizations
  - Data Visualizations
  - Business reports
- Artificial Intelligence, Data Science & Data mining
  - Mathematics, Statistics & Computer algorithms
  - Machine learning & Deep learning
  - R Programming, Python
- Data Engineering
  - Hadoop, Hive, Spark, Kafka, BigTable, …
  - Parallel processing
  - Java, Scala, Python.
- Infrastructure
  - Linux, Cloud Computing







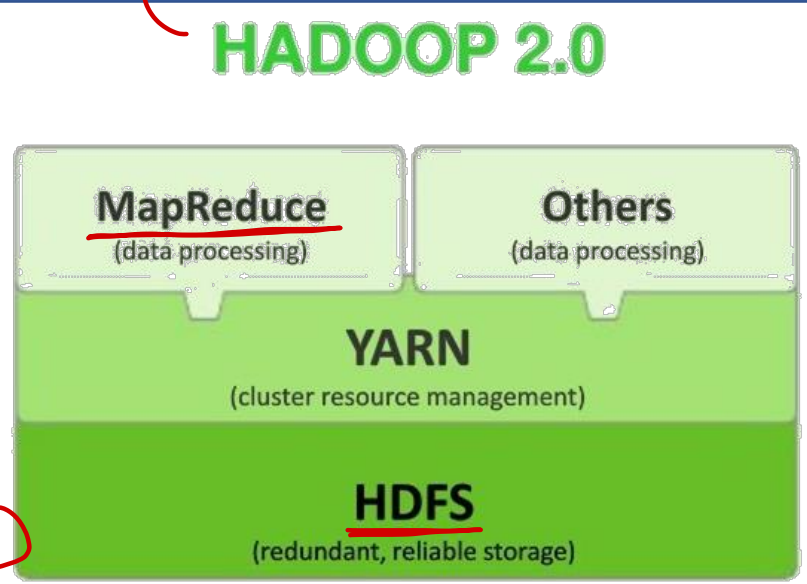


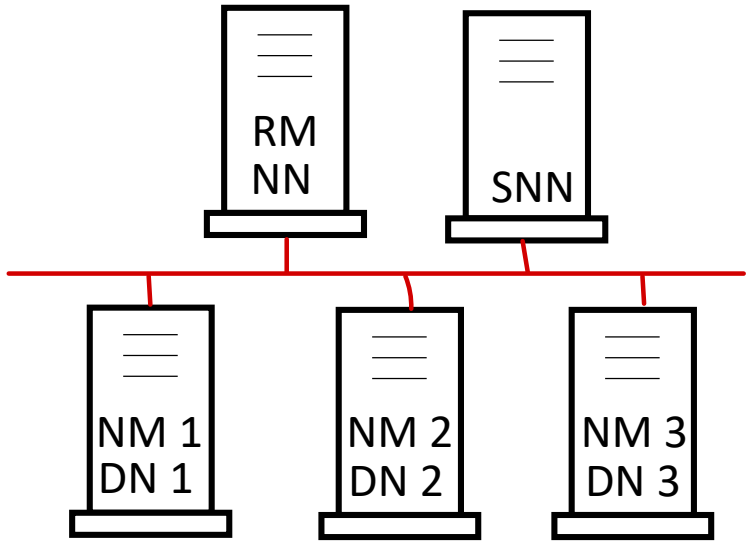
Apache Hadoop

Inspired from Google
GFS → 2003
MR → 2004
Yahoo → Hadoop 2007

- Hadoop is developed by Doug cutting.
  - History
    - Web crawler – Nutch
    - Distributed computing and storage needed to process huge data produced by the crawler.
    - Joined Yahoo. Developed and open sourced under Apache license.
- Hadoop 1.x
  - Distributed storage: HDFS (Hadoop Distributed File System)
  - Distributed computing Map-reduce
- Hadoop 2.x 3.x
  - Distributed storage: HDFS
  - Distributed computing Map-reduce
  - Cluster manager: YARN (Yet Another Resource Negotiator)
- Hadoop is like a Kernel/Platform on which many different applications are built (eco-systems).

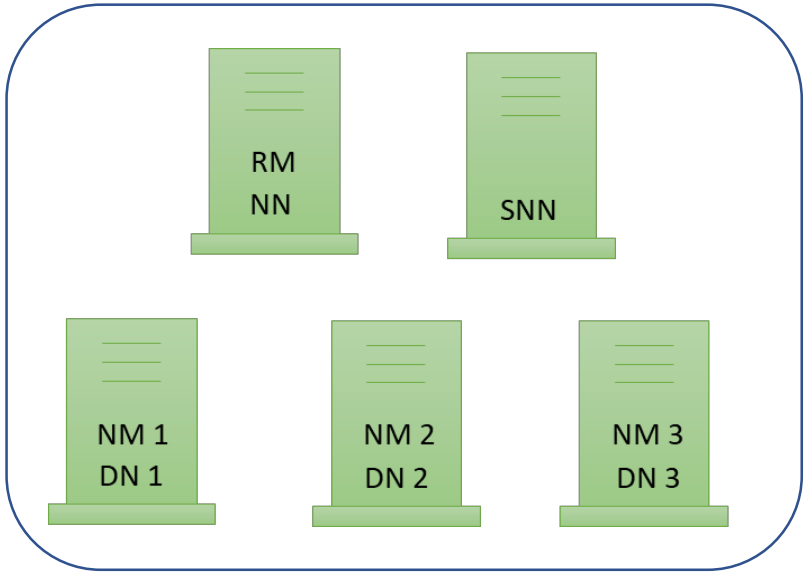
in Java

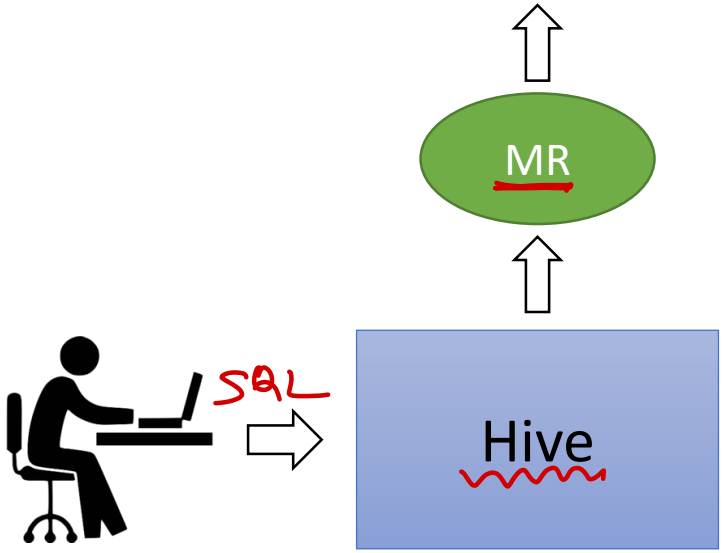




Apache Hive

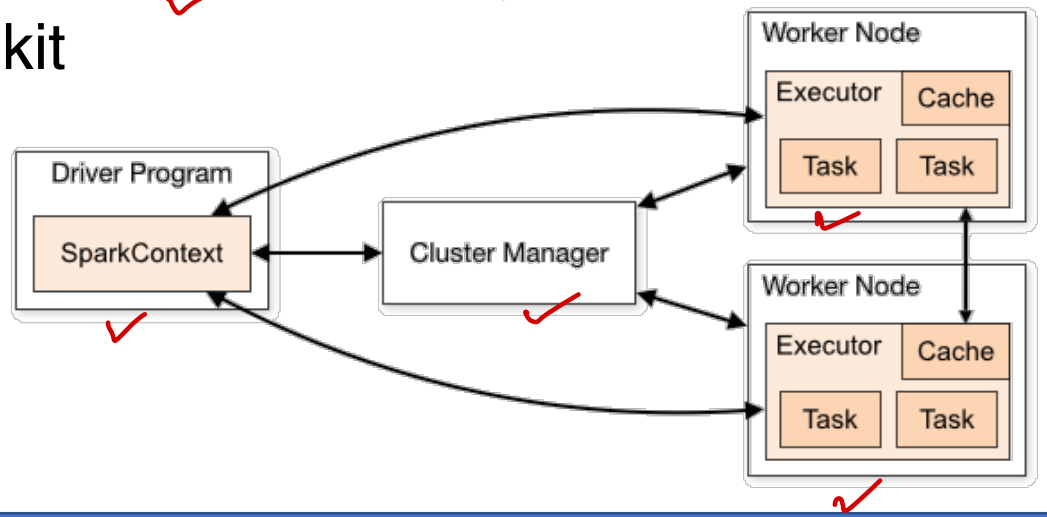
- Developed by Facebook (2007)
- Client software that convert Hive QL queries to MR.
- Hive QL is similar to SQL with many extended features.
- Hive manages structured data.
- Hive is data warehouse (OLAP) built for Hadoop.
  - Data storage = HDFS
  - Metadata = RDBMS
  - Data processing = Map-reduce or Spark or Tez.

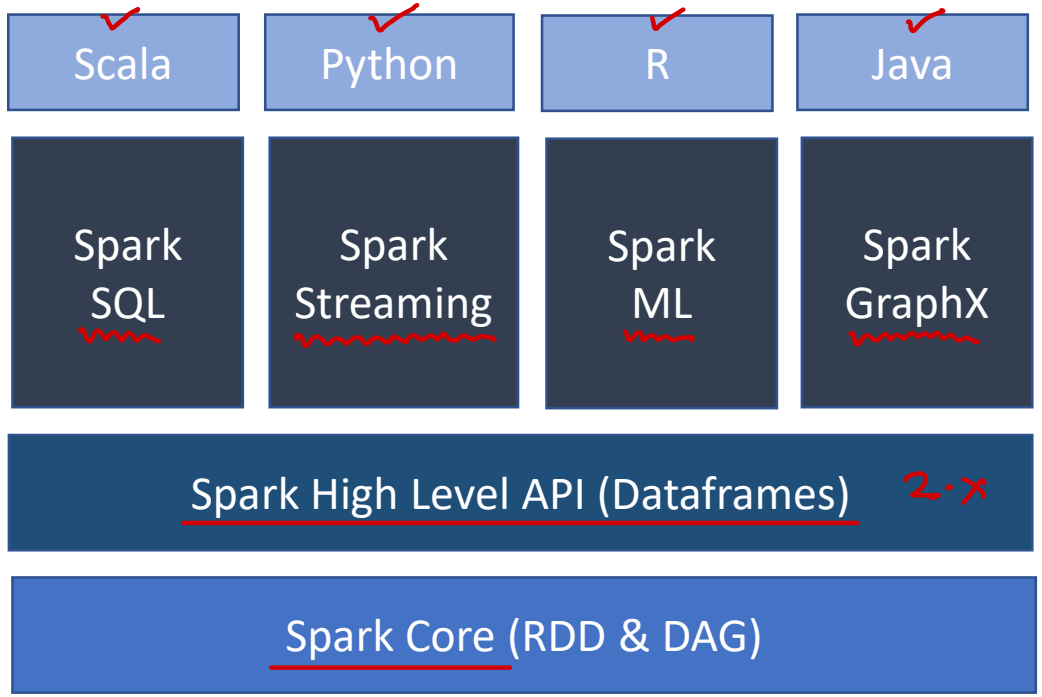




Apache Spark

- Spark is Distributed computing framework, that can process huge amount of data.
- Spark can be used as eco-system of Hadoop or can be used as independent distributed computing framework.
- Developed by UCB AMPlabs division.
- Further developed/maintained by DataBricks.
- Popular Spark vendors
  - DataBricks, AWS EMR, Cloudera, MapR
- Spark Toolkit

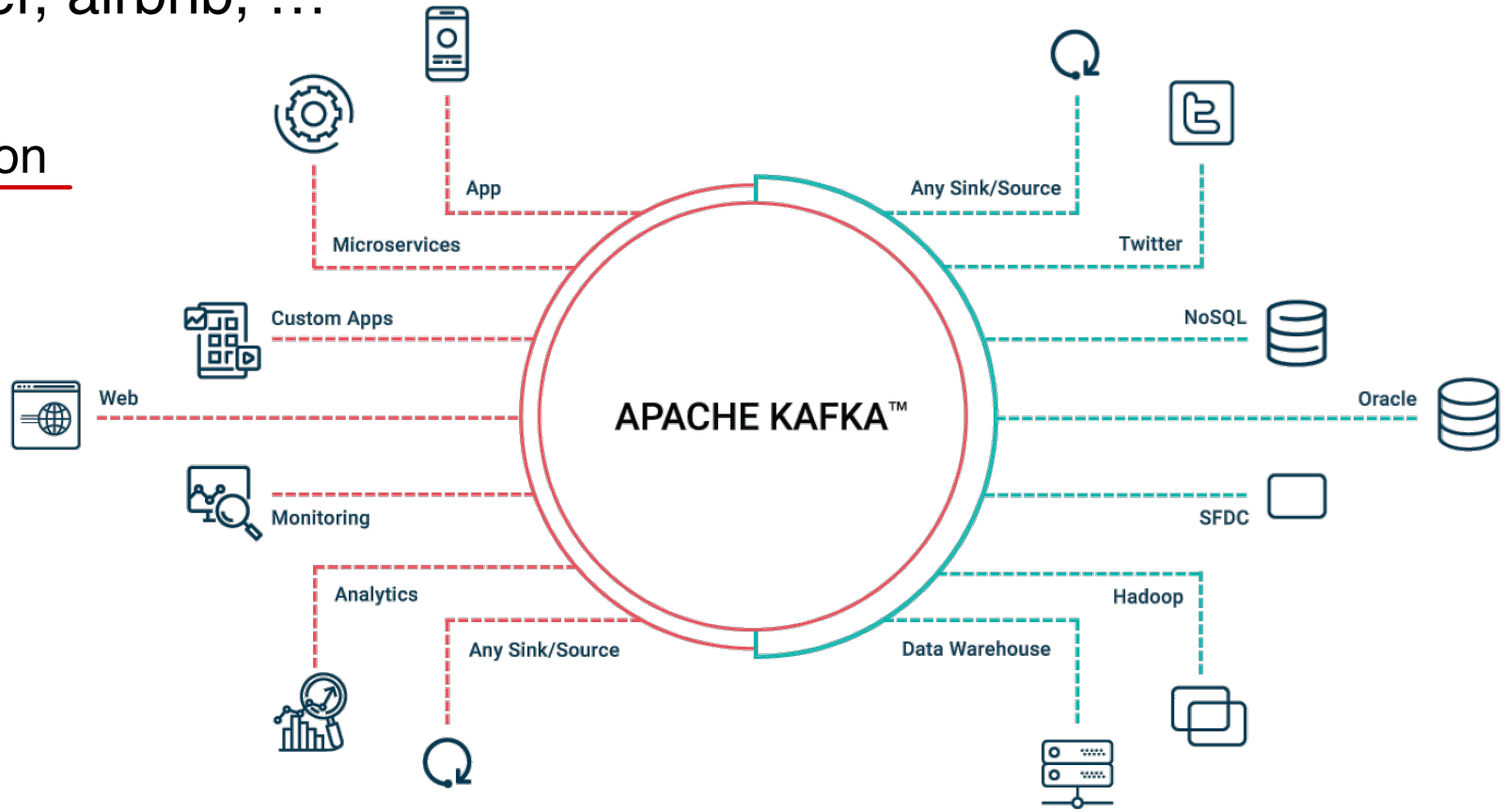




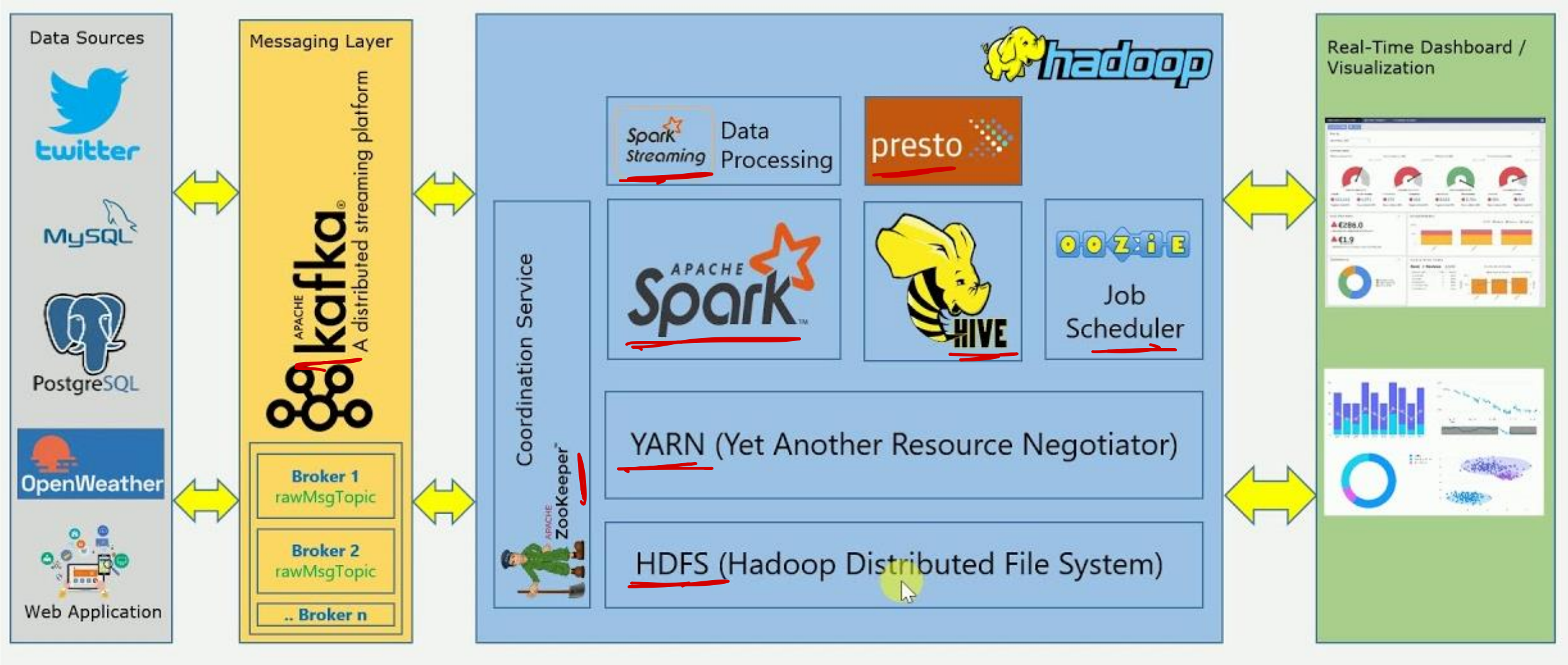
Apache Kafka

- Kafka is a distributed messaging system.
- Developed at LinkedIn and open sourced in 2011.
- Used by LinkedIn, Twitter, Uber, airbnb, …
- Advantages
  - Scalable, Durable, Finite retention
  - Low latency, Strong ordering,
  - Exact once delivery
- Applications
  - Stream processing
  - Notifications.

→ Send messages asynchronously at large scale - Huge data, Many producers/consumers.



Real time dashboard reference architecture

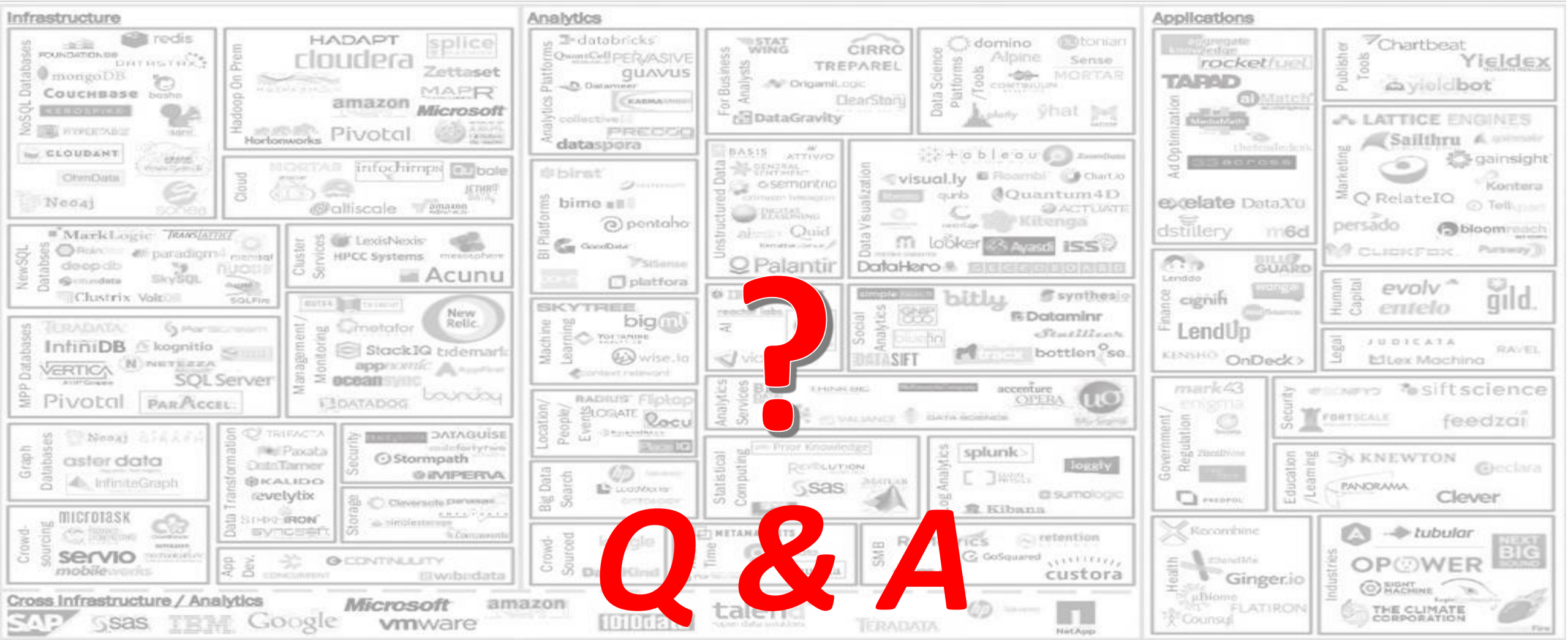


Big Data domains & opportunities

- Domains: Health-care, Retails, Trading/Share market, Finance, Security, Fraud, Search engines, Log Analysis, Telecom, Traffic Control, Manufacturing and lot more.
- Big Data is all about :- Think, Collect, Manage, Analyze, Summarize, Visualize, Discover Knowledge and Take Decisions.
- Job profiles:
  - Business Analyst/Intelligence
  - Database engineer / DWH
  - Big Data engineer
  - Data operations
  - Big Data Architect

- The sexiest job in the 21st century require a mixture of multidisciplinary abilities and suitable candidates must be prepared to learn and develop constantly.

-Ronald Van Loon


Infrastructure
Analytics
Applications
Cross Infrastructure / Analytics
Open Source
Data Sources
Q & A

Thank you!

Nilesh Ghule <nilesh@sunbeaminfo.com>

Also refer: Big data webinar - https://www.youtube.com/live/BxwpqnQ6BgQ