Introduction to Apache Hive

TIMESPRO

Transactional Systems vs. Analytical Systems

Feature	Transactional Systems	Analytical Systems
Type of Data Handled	Day-to-day transactional data	Historical data
Operations Performed	Insert, Delete, Update	Majorly Read operations to analyze large volumes of data
Example	ATM transactions, e-commerce transactions	Analyzing data of a sales campaign
Best Suited Systems	RDBMS (Databases: Ex - Oracle, MySQL, etc.), Monolithic Systems	Data Warehouses (Ex - Teradata, etc.), Distributed Systems

Key Takeaway:

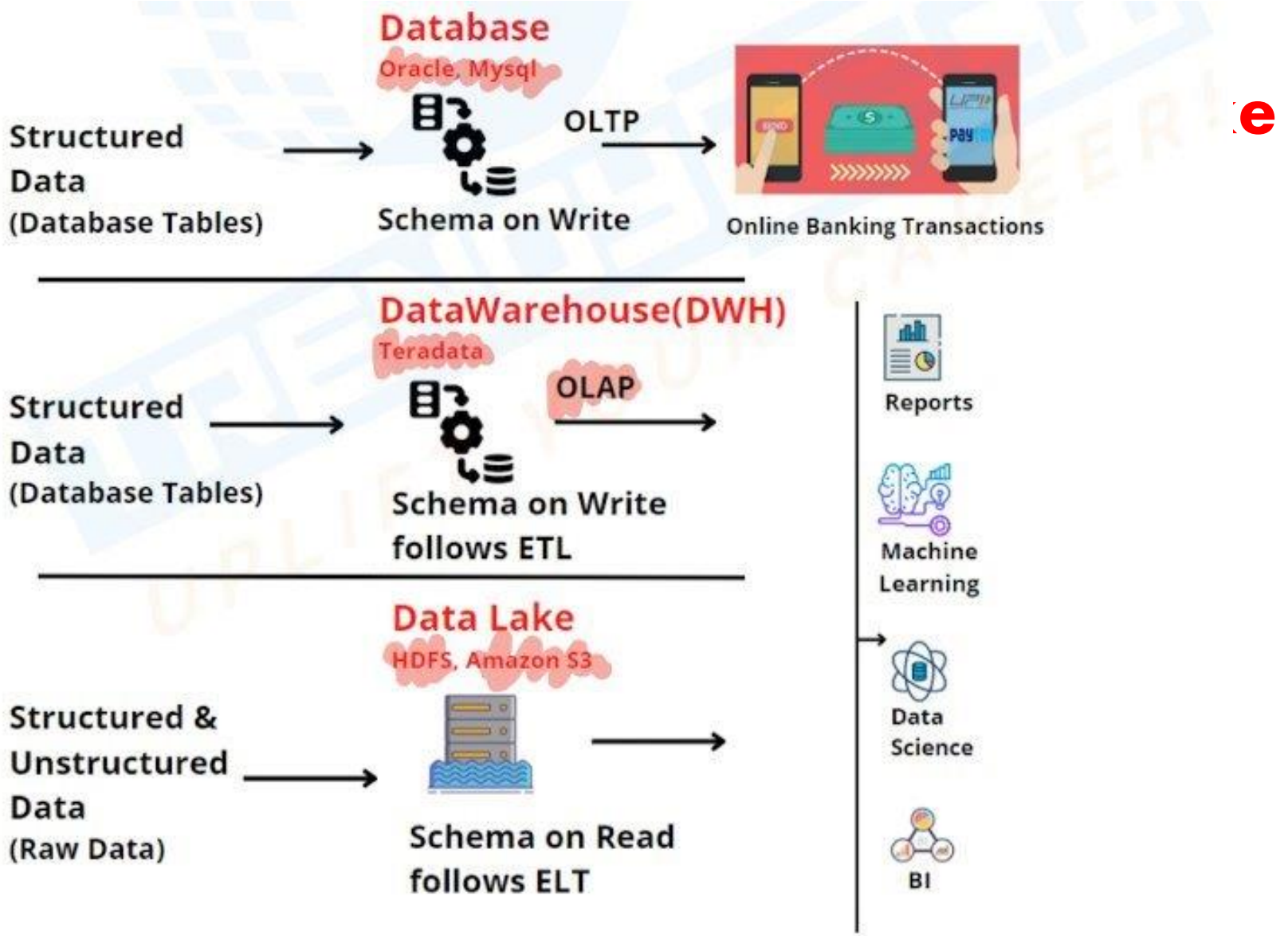
- Transactional systems capture and process day 2 day transactions, while
- Analytical systems focus on analyzing historical data to gain insights.

TIMESPRO

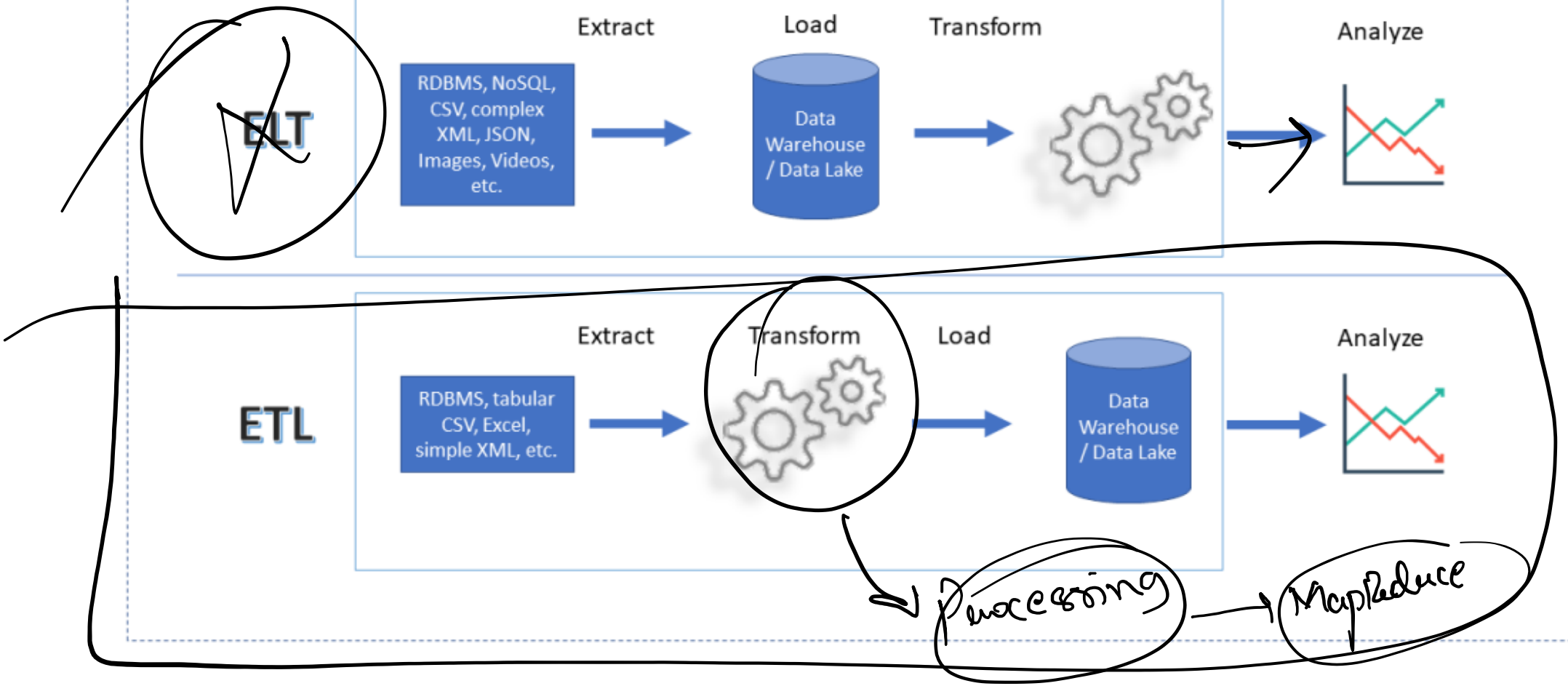
Database Vs Data Warehouse Vs Data Lake

Feature	Database	Data Warehouse (DWH)	Data Lake
Purpose	OLTP (Transactional Processing)	OLAP (Analytical Processing)	Insights from large volumes of data
Data Structure	Structured (Rows/ Columns)	Structured	Raw (Structured & Unstructured)
Data Scope	Recent data for performance	Historical data	Both recent & historical data
Examples	Oracle, MySQL	Teradata	HDFS, Amazon S3
Schema Approach	Schema on Write	Schema on Write	Schema on Read
Process	-	ETL (Extract, Transform, Load)	ELT (Extract, Load, Transform)
Storage Cost	High	High, but less than Database	Cost-effective
Challenges	-	Complex transformations, rigidity	Flexibility, but requires management

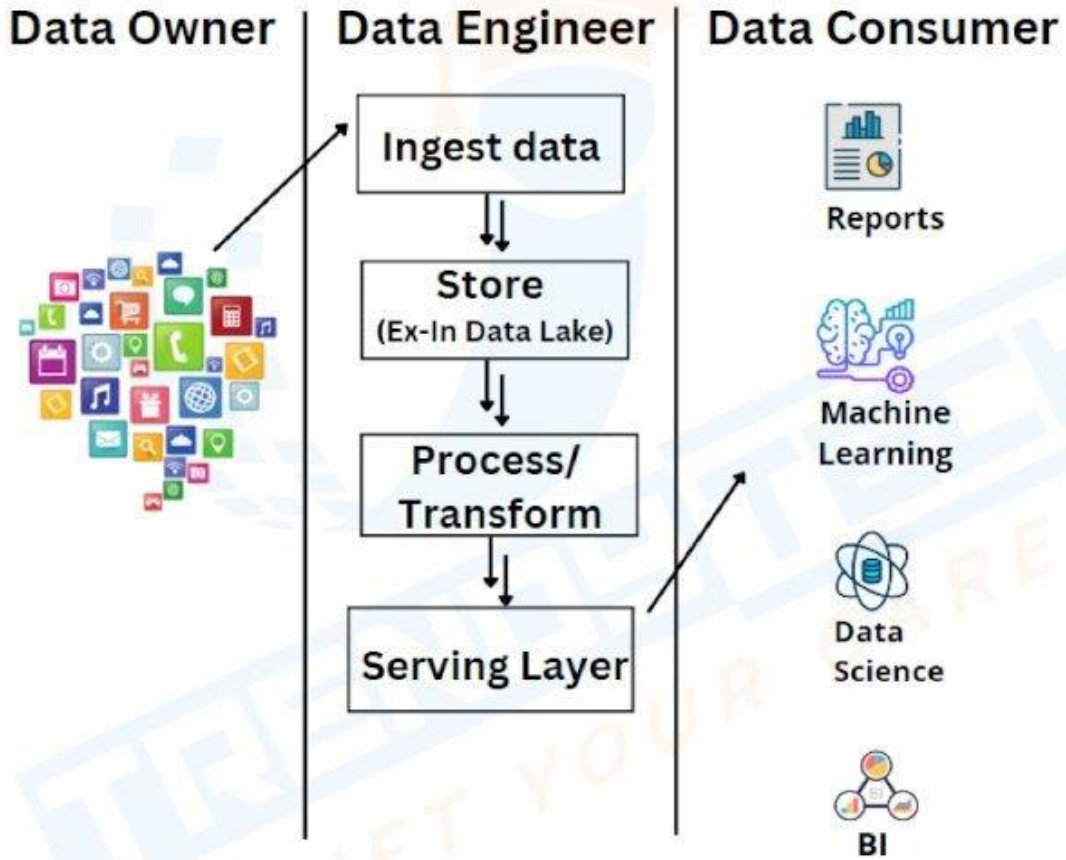
Da e


Database
Oracle, Mysql
Structured Data (Database Tables)
OLTP
Schema on Write
Online Banking Transactions
DataWarehouse(DWH)
Teradata
Structured Data (Database Tables)
OLAP
Schema on Write follows ETL
Reports
Machine Learning
Data Science
BI
Data Lake
HDFS, Amazon S3
Structured & Unstructured Data (Raw Data)
Schema on Read follows ELT

ELT vs ETL


ELT
Extract
Load
Transform
Analyze
RDBMS, NoSQL, CSV, complex XML, JSON, Images, Videos, etc.
Data Warehouse / Data Lake
ETL
Extract
Transform
Load
Analyze
RDBMS, tabular CSV, Excel, simple XML, etc.
Data Warehouse / Data Lake
Processing
MapReduce

Role of Data Engineers



History

- Facebook started generating large volumes of data at a very high pace
- These data had to be analyzed and reports had to be generated to get some meaningful inferences.
- Used a Datawarehouse on a commercial RDBMS.
- **Problems**:
  - Handling large volumes
  - scalability

TIMESPRO

History

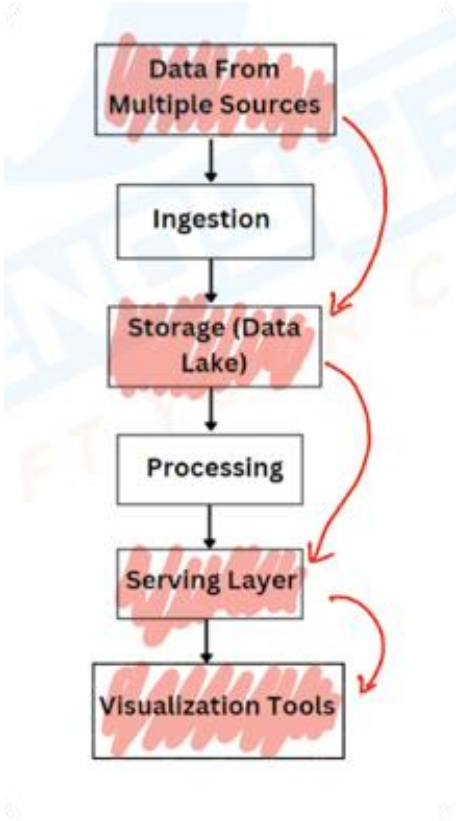
- **Problems with using hadoop:**
  - Required to write map reduce program
  - SQL developers who were earlier getting the reports had to be reskilled/upskilled in writing java/python programs
  - Writing map reduce programs took time
- **Solution**
  - Started using hadoop
  - Data retrieval Jobs gave faster results
  - Face book decided to improve the capabilities of hadoop, so that their developers were allowed to use SQL on top of hadoop
  - Hive was born in 2007 and was open soured in 2008

What is Hive?

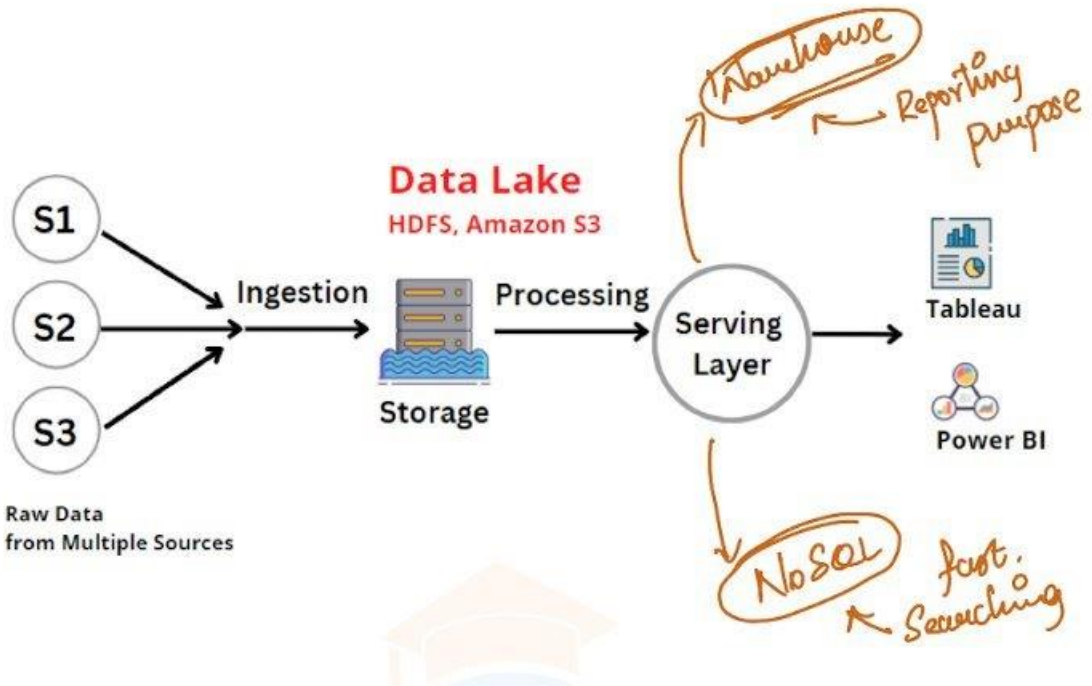
- It is a **data warehouse** in Hadoop with fault tolerant
- Facilitates **querying and managing** Massive petabytes of data residing on distributed storage
- Provides an SQL like interface called **HiveQL** which translates a query into java map reduce program and runs the same on a Hadoop cluster
- Developed by **Facebook** and open sourced to **apache**

Data engineering Flow

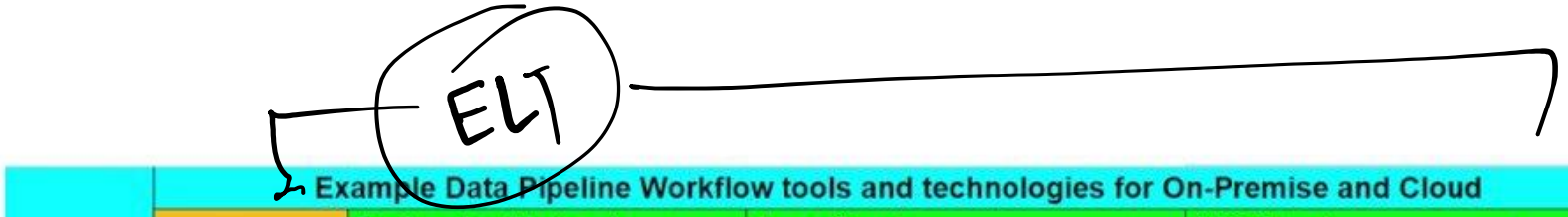
- **Data Collection**: Gather data from various sources into a **Data Lake** for centralized storage.
- **Data Ingestion**: Use an **Ingestion Framework** to move data from different sources into the Data Lake.
- **Data Processing**: Follow the **ELT process**—load data into the Data Lake, then transform it (e.g., cleaning, aggregation, joins) as needed.
- **Data Serving**: Store processed data in the **Serving Layer** for visualization tools like Tableau and Power BI to display results graphically.



Data engineering Flow



Data engineering Flow



Phases / Stages of Data Pipeline	Example Data Pipeline Workflow tools and technologies for On-Premise and Cloud			
		On-Premise (Hadoop)	Azure Cloud	AWS Cloud
	Source	MySQL Database table	Multiple sources	Multiple sources
	Ingestion	Sqoop	ADF	AWS GLUE
	Storage	HDFS	ADLS Gen2	Amazon S3
	Processing	MapReduce / SPARK	Azure Databricks / Synapse	Athena / Redshift
	Serving layer	HBase	Azure SQL / Cosmos DB	AWS RDS / Dynamo DB

TIMESPRO

Why Hive?

- Data Transformation and ETL
- Data Warehouse Features
- Scalability
- SQL-Like Interface
- Batch Processing
- Schema on Read
- Support for Various Data Formats

TIMESPRO

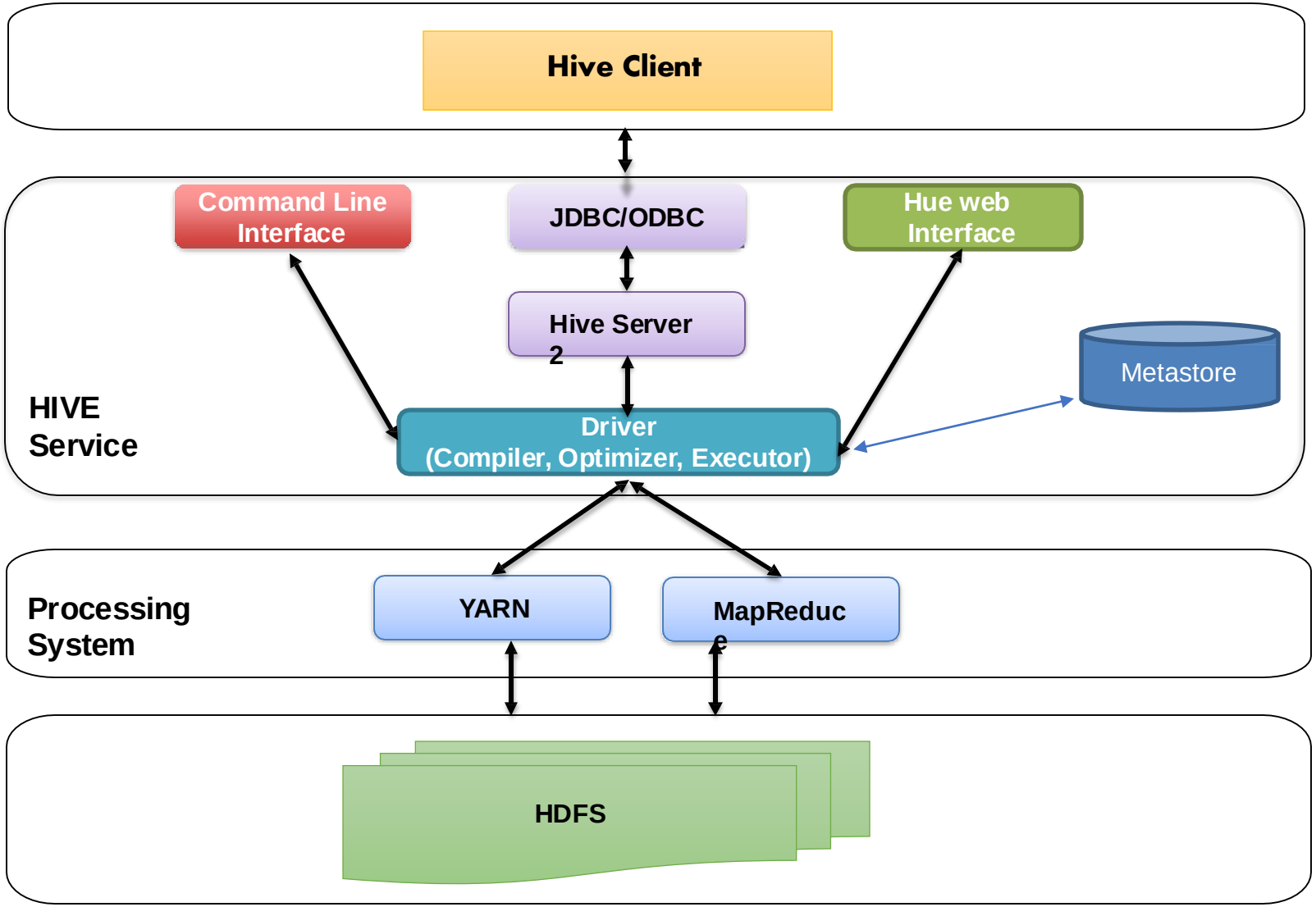
Is hive a database?

- **Answer** : No. Many people consider hive as a database management system. But, the truth is different.
- Apache **hive itself is not a database**.
- Consider hive as **logical view** of the underlying data in HDFS.
- It **cannot store any data** of its own. It always uses **HDFS for storing** the processed data.
- The only thing it can do is **enforcing the structure** in which the data can be stored in HDFS.

Hive - Drawbacks

- Is Not an OLTP system
- Is Not a full fledged database – data is stored in **HDFS** as flat files.
- Does not support all the SQL queries especially complicated ones
- Has high latency – since it converts queries to Java MapReduce code and executes

Hive Architecture



Advantages of Hive Against Map Reduce

- 100 lines of code can be easily solved using 2-3 lines of SQL code
- Best suited for SQL developers, who are novice to Java/python programming

```java
package org.myorg;

import java.io.IOException;
import java.util.*;

import org.apache.hadoop.fs.Path;
import org.apache.hadoop.conf.*;
import org.apache.hadoop.io.*;
import org.apache.hadoop.mapreduce.*;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.input.TextInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;
import org.apache.hadoop.mapreduce.lib.output.TextOutputFormat;

public class WordCount {

 public static class Map extends Mapper<LongWritable, Text, Text, IntWritable> {
    private final static IntWritable one = new IntWritable(1);
    private Text word = new Text();

    public void map(LongWritable key, Text value, Context context) throws IOException, InterruptedException {
        String line = value.toString();
        StringTokenizer tokenizer = new StringTokenizer(line);
        while (tokenizer.hasMoreTokens()) {
            word.set(tokenizer.nextToken());
            context.write(word, one);
        }
    }
 }

 public static class Reduce extends Reducer<Text, IntWritable, Text, IntWritable> {

    public void reduce(Text key, Iterable<IntWritable> values, Context context)
      throws IOException, InterruptedException {
        int sum = 0;
        for (IntWritable val : values) {
            sum += val.get();
        }
        context.write(key, new IntWritable(sum));
    }
 }

 public static void main(String[] args) throws Exception {
    Configuration conf = new Configuration();

        Job job = new Job(conf, "wordcount");

    job.setOutputKeyClass(Text.class);
    job.setOutputValueClass(IntWritable.class);

    job.setMapperClass(Map.class);
    job.setReducerClass(Reduce.class);

    job.setInputFormatClass(TextInputFormat.class);
    job.setOutputFormatClass(TextOutputFormat.class);
```

```sql
CREATE TABLE docs (line STRING);

LOAD DATA INPATH 'docs' OVERWRITE INTO TABLE docs;

CREATE TABLE word_counts AS
SELECT word, count(1) AS count FROM
  (SELECT explode(split(line, '\s')) AS word FROM docs) w
GROUP BY word
ORDER BY word;
```

Recap : Apache Hive

- Hive is NOT an RDBMS
- Hive is a SQL for HDFS
- No constraints !!!
- Hive SQL syntax is similar to MySQL's SQL syntax
- Hive is flexible
    - Data can come first, Schema can come later or vice versa
    - Create schema definitions only for the datasets we wish to query
    - Hence its Warehouse.
- Hive's data will be on HDFS, and the schema will be in a metastore DB
- Hive is Schema-On-Read → The structure is validated against the data when we issue a select query

Assignment

- **Based on the extra reads and research, write a LinkedIn Blog(100-words) on following**
  1. **DE workflow(on-premises & cloud) and DE role**
  2. **Data lake vs DW vs DB**

DEMO

Hive

Some Extra Reads

Useful info

GUIDELINES FOR HiveQL STATEMENTS

- Can be Executed by invoking a hive shell or webUI
- Not case sensitive
- Keywords cannot be abbreviated or split across lines
- Clauses (like SELECT, FROM,WHERE etc.) are usually placed on separate lines
- Indents are used to enhance readability
    - In the command line client it is mandatory to terminate each SQL statement end with a semicolon(;)