Master Executive di II Livello

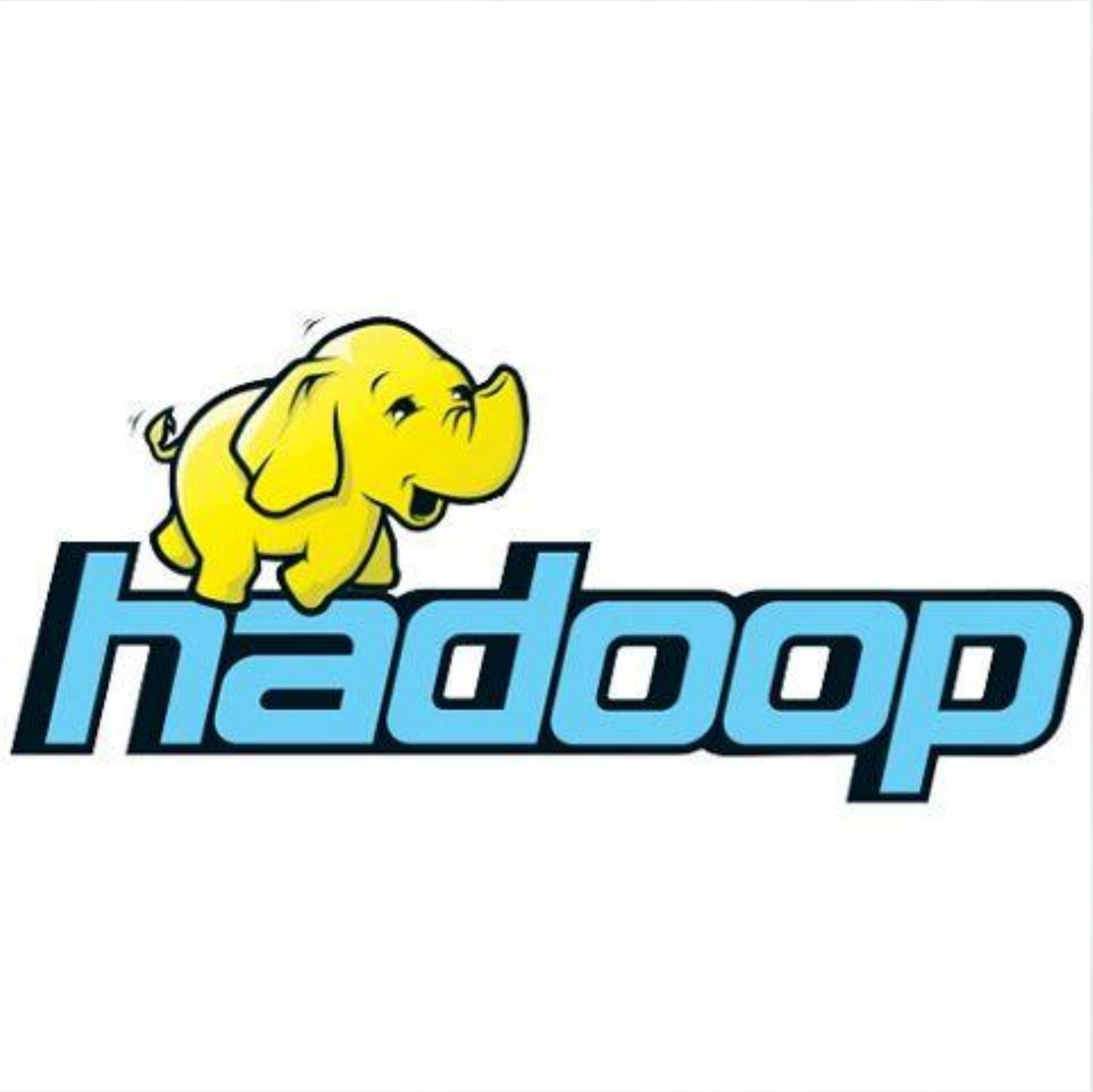
BIG DATA ANALYSIS AND BUSINESS INTELLIGENCE

Introduction to Hadoop Ecosystem

Vamsi Krishna Varma Gunturi
Deep learning researcher at ISTAT
vamsivarmagunturi@gmail.com

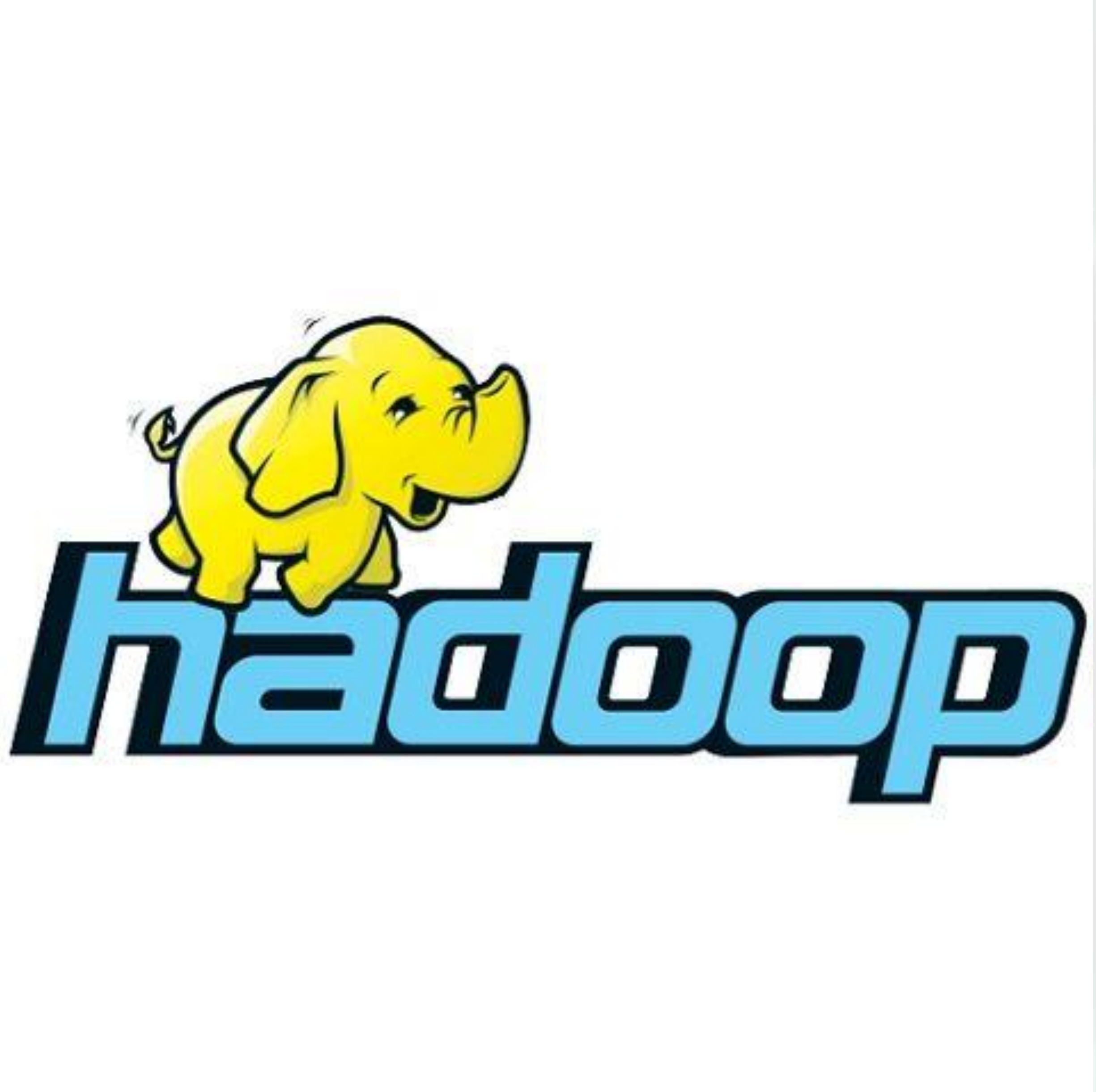
fondazione INUIT TOR VERGATA

What is Hadoop ?



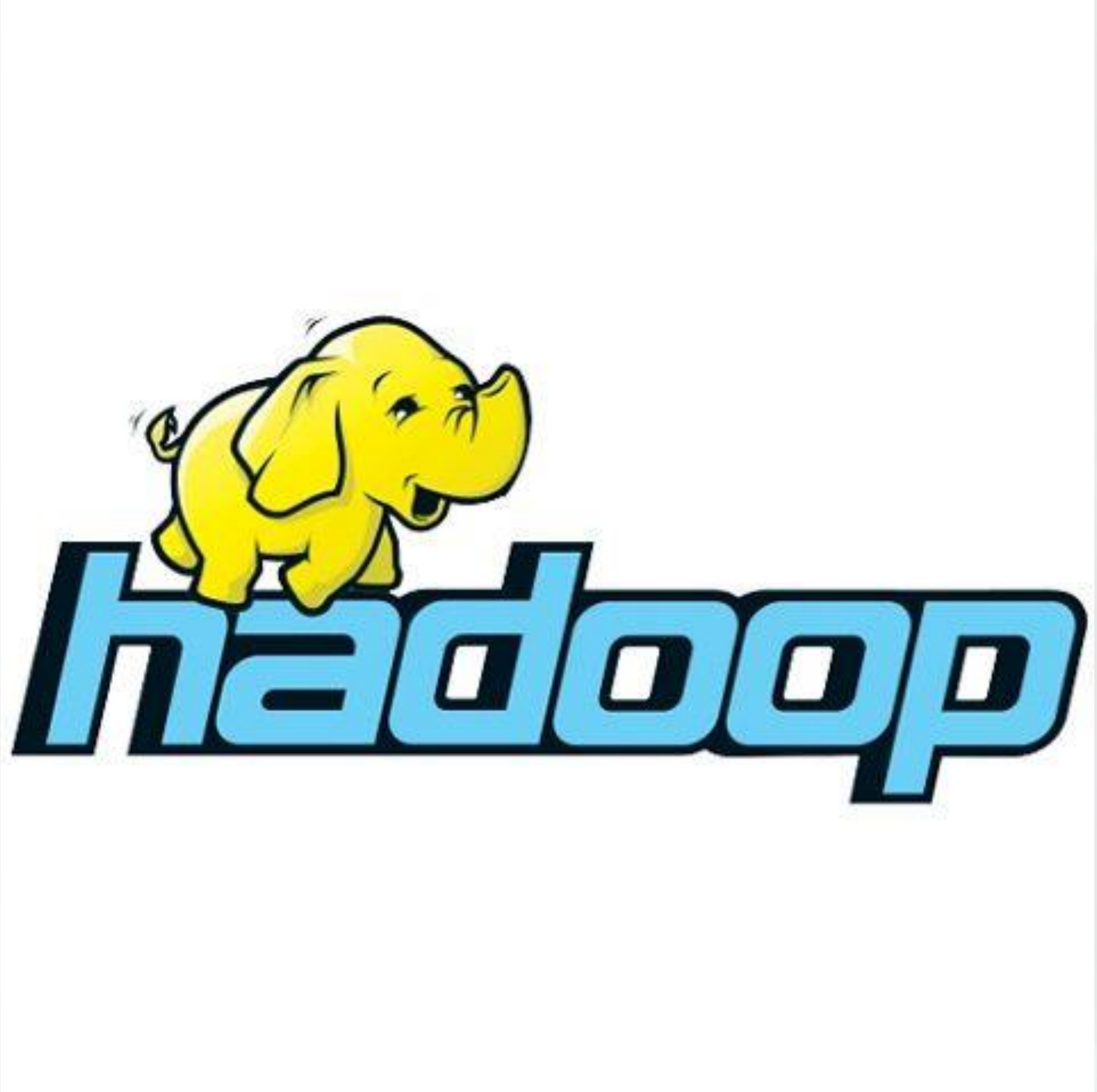
- Hadoop is an Open source platform written in Java for distributed processing of large data on a cluster of servers built from commodity hardware.
- Many people assume that Hadoop is big data. It’s not. There was big data before Hadoop and there continues to be big data without Hadoop. However, Hadoop is a huge player now with big data.
- 90 percent of the world’s data was created in the past two years. Organizations need to store and analyse massive amounts of structured and unstructured data from disparate data sources—data too massive to manage effectively with traditional relational databases. Hadoop is a great tool to help with this task.
- Hadoop can reach massive scalability by exploiting a simple distribution architecture and coordination model. Huge clusters can be made up using (cheap) commodity hardware

Hadoop timeline



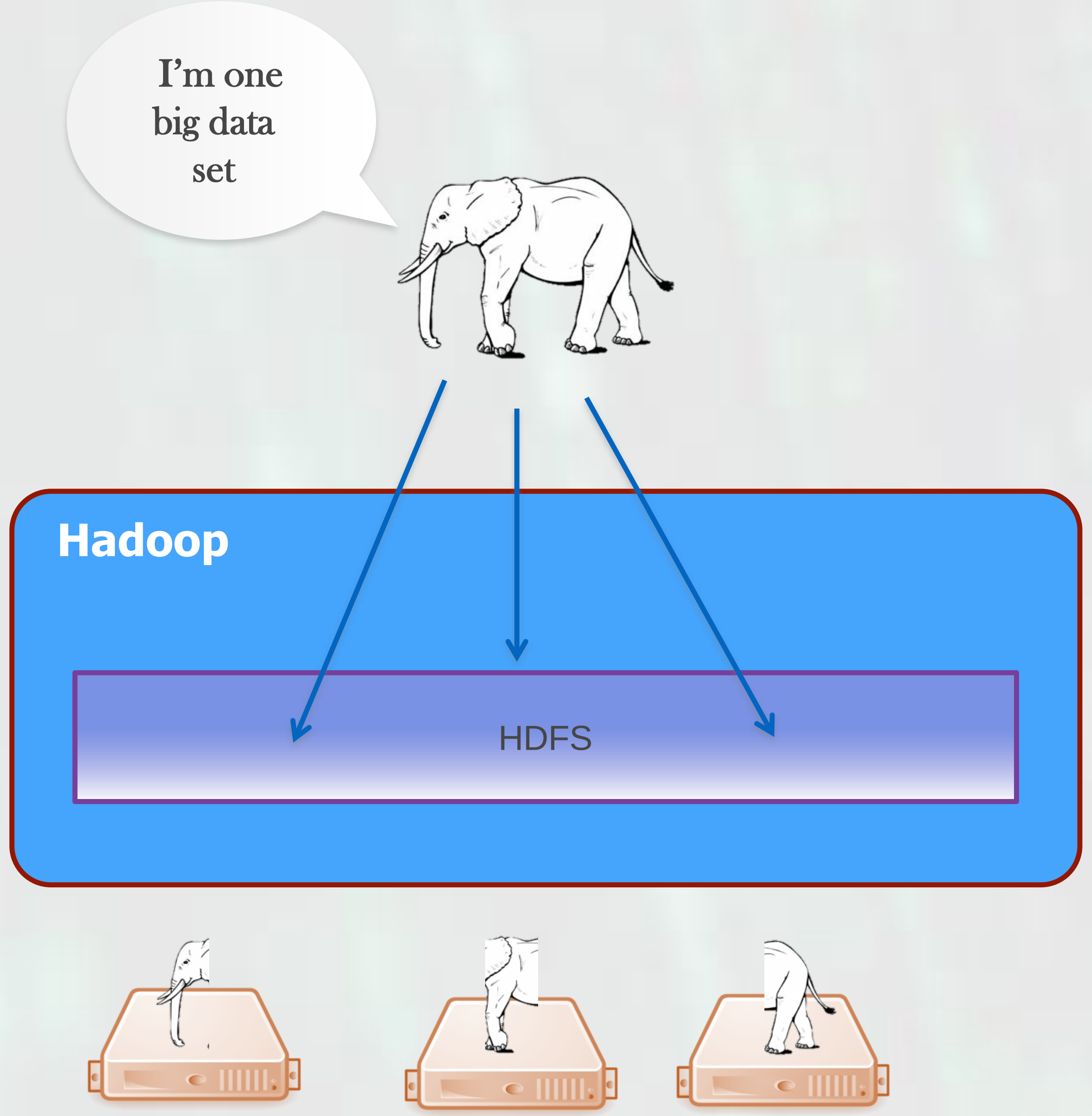
- **Dec 2004**: Dean/Ghemawat (Google) publishes MapReduce paper
- **2005**: Doug Cutting and Mike Cafarella (Yahoo) create Hadoop, at first only to extend Nutch (the name is derived from Doug’s son’s toy elephant)
- **2006**: Yahoo runs Hadoop on 5-20 nodes
- **March 2008**: Cloudera founded
- **July 2008**: Hadoop wins TeraByte sort benchmark (1st time a Java program won this competition)
- **April 2009**: Amazon introduces “Elastic MapReduce” as a service on S3/EC2
- **June 2011**: Hortonworks founded

Hadoop timeline (cont..)



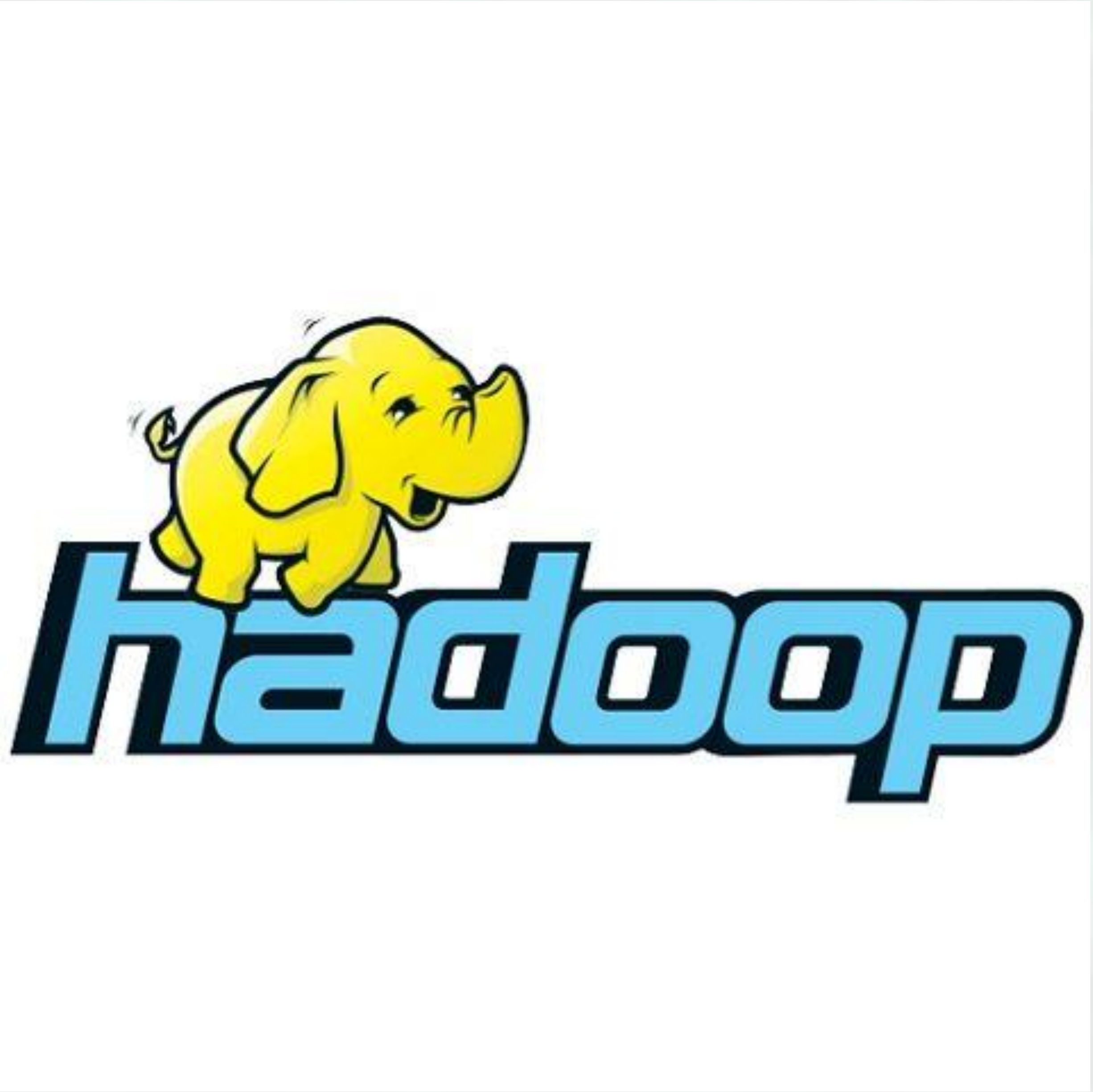
- **December 2011**: Apache Hadoop release 1.0.0
- **June 2012**: Facebook claims “biggest Hadoop cluster”, totalling more than 100 PetaBytes in HDFS
- **2013**: Yahoo runs Hadoop on 42,000 nodes, computing about 500,000 MapReduce jobs per day
- **October 2013**: Apache Hadoop release 2.2.0
- **December 2017**: Apache Hadoop version 3.0
- **October 2018**: Hortonworks will join forces with its arch-rival Cloudera to create a single company with about $730 million in annual revenue, 2,500 customers, and a $5.2 billion market.

Principle of Hadoop



- Hadoop is basically a middleware platforms that manages a cluster of machines
- The core components is a distributed file system (HDFS)
- Files in HDFS are split into blocks that are scattered over the cluster
- The cluster can grow indefinitely simply by adding new nodes

Hadoop Distributions



- Hadoop is an open source project promoted by the Apache Foundation. As such, it can be downloaded and used for free.
- However, all the configuration and maintenance of all the components must be done by the user, mainly with command-line tools
- Software vendors provide Hadoop distributions that facilitate in various ways the use of the platform. Distributions are normally free but there is a paid-for support.
- Additional features such as User interface, Management console and Installation tools vary from distribution to distribution

Common Hadoop Distributions

❖ **Hortonworks**

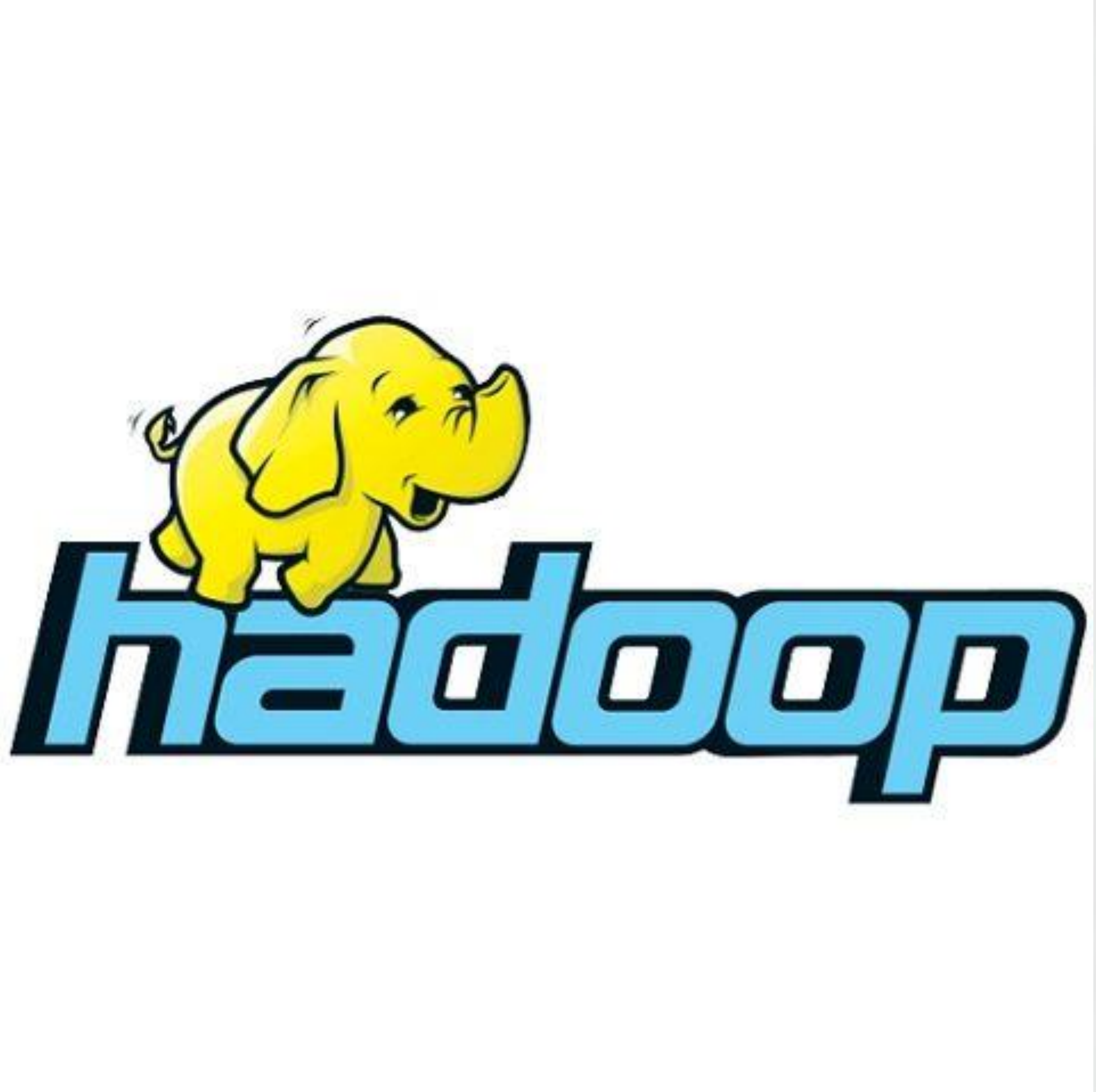
✓ Completely open-source
✓ Also have a Windows version
✓ Used in: Big Data Sandbox

❖ **Cloudera**

✓ Mostly standard Hadoop but extended with proprietary components
✓ Highlights: Cloudera Manager (console) and Impala (high-performance query)
✓ Used in: Istat Big Data Platform

- **Note**: In this course we will be using Hortonworks distribution.

Hadoop vs RDBMS



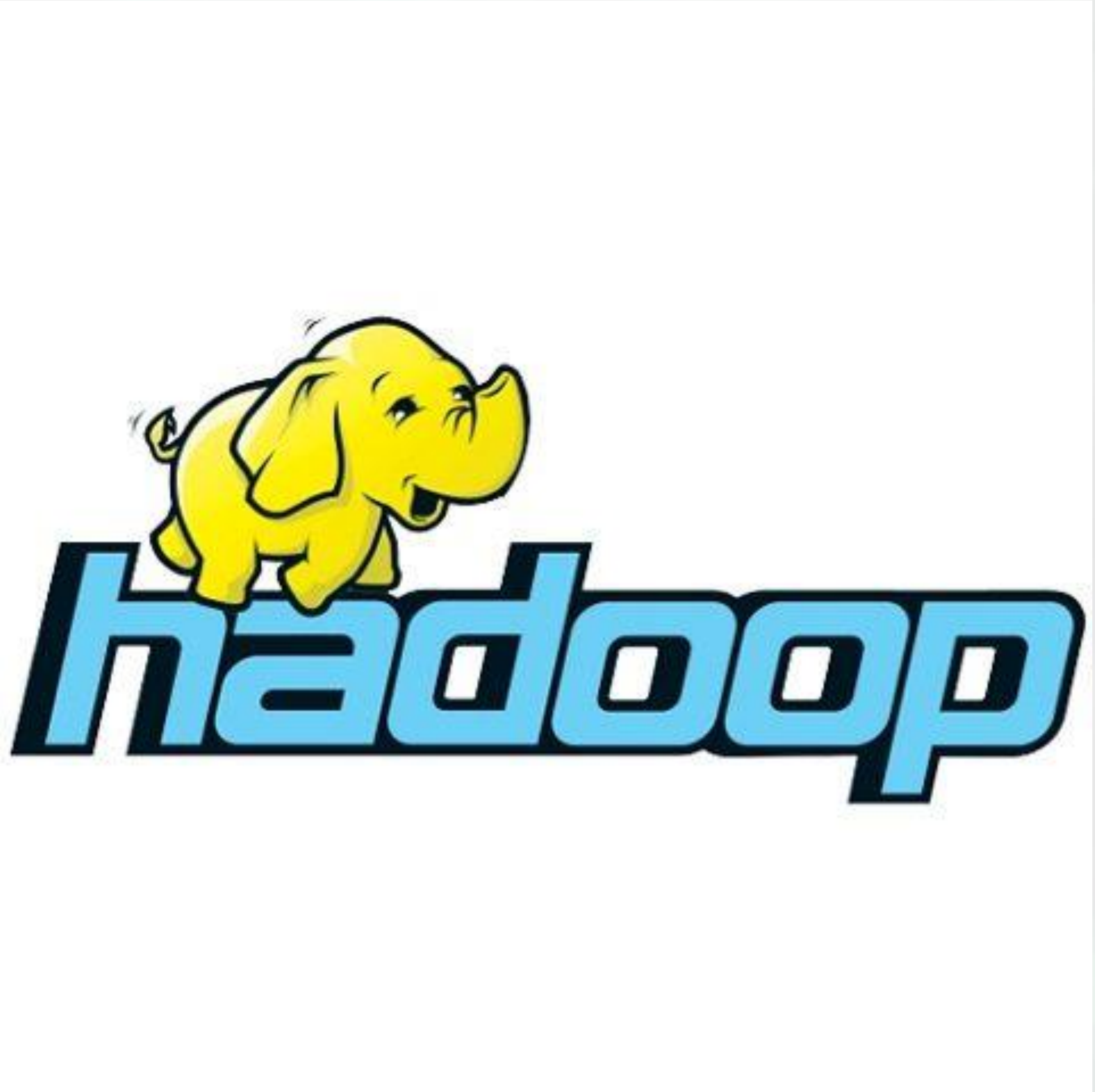
❖ Hadoop:

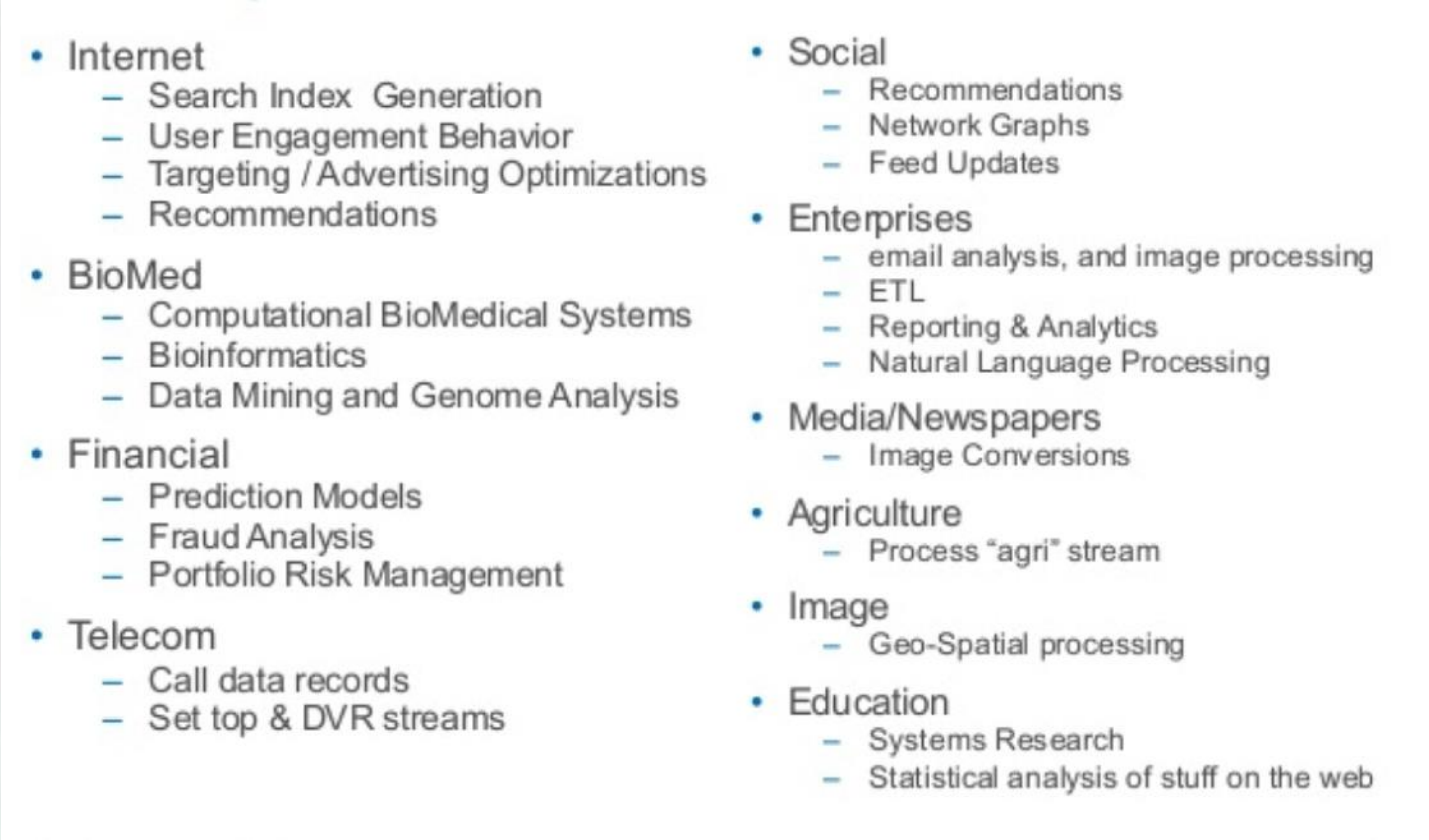
- is not transactional
- is not optimized for random access
- does not natively support data updates
- privileges long-running, batch work

❖ RDBMS:

- disk space is more expensive
- cannot scale indefinitely

Hadoop use cases:





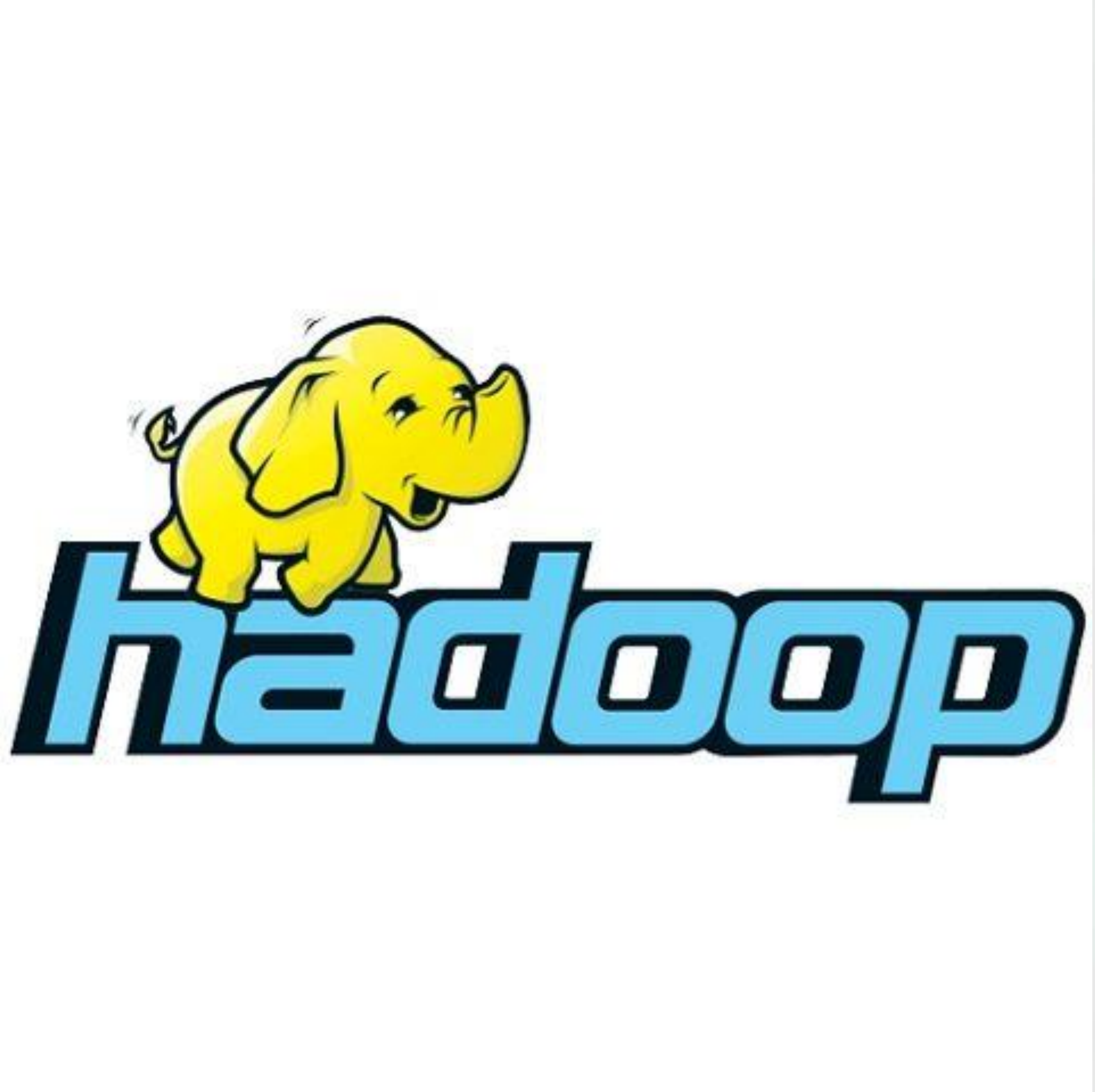
- Internet
  - Search Index Generation
  - User Engagement Behavior
  - Targeting / Advertising Optimizations
  - Recommendations
- BioMed
  - Computational BioMedical Systems
  - Bioinformatics
  - Data Mining and Genome Analysis
- Financial
  - Prediction Models
  - Fraud Analysis
  - Portfolio Risk Management
- Telecom
  - Call data records
  - Set top & DVR streams
- Social
  - Recommendations
  - Network Graphs
  - Feed Updates
- Enterprises
  - email analysis, and image processing
  - ETL
  - Reporting & Analytics
  - Natural Language Processing
- Media/Newspapers
  - Image Conversions
- Agriculture
  - Process "agri" stream
- Image
  - Geo-Spatial processing
- Education
  - Systems Research
  - Statistical analysis of stuff on the web

Source: https://www.slideshare.net/emcacademics/hadoop-101

Who Uses Hadoop?

- Amazon/A9
- AOL
- Facebook
- Fox interactive media
- Google
- IBM
- New York Times
- Microsoft
- Quantcast
- Rack space / Mailtrust
- Veoh
- Yahoo!

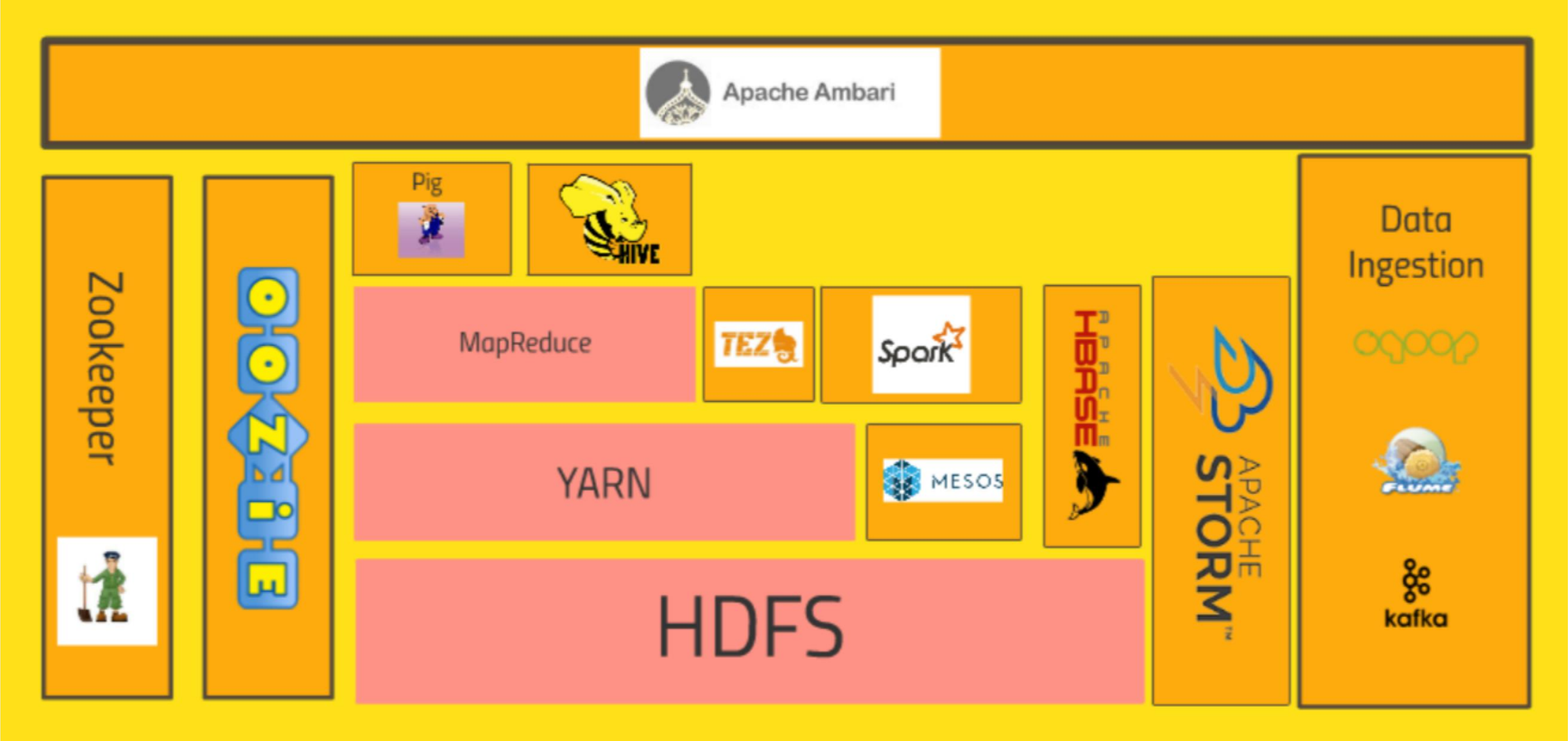
Hadoop Components



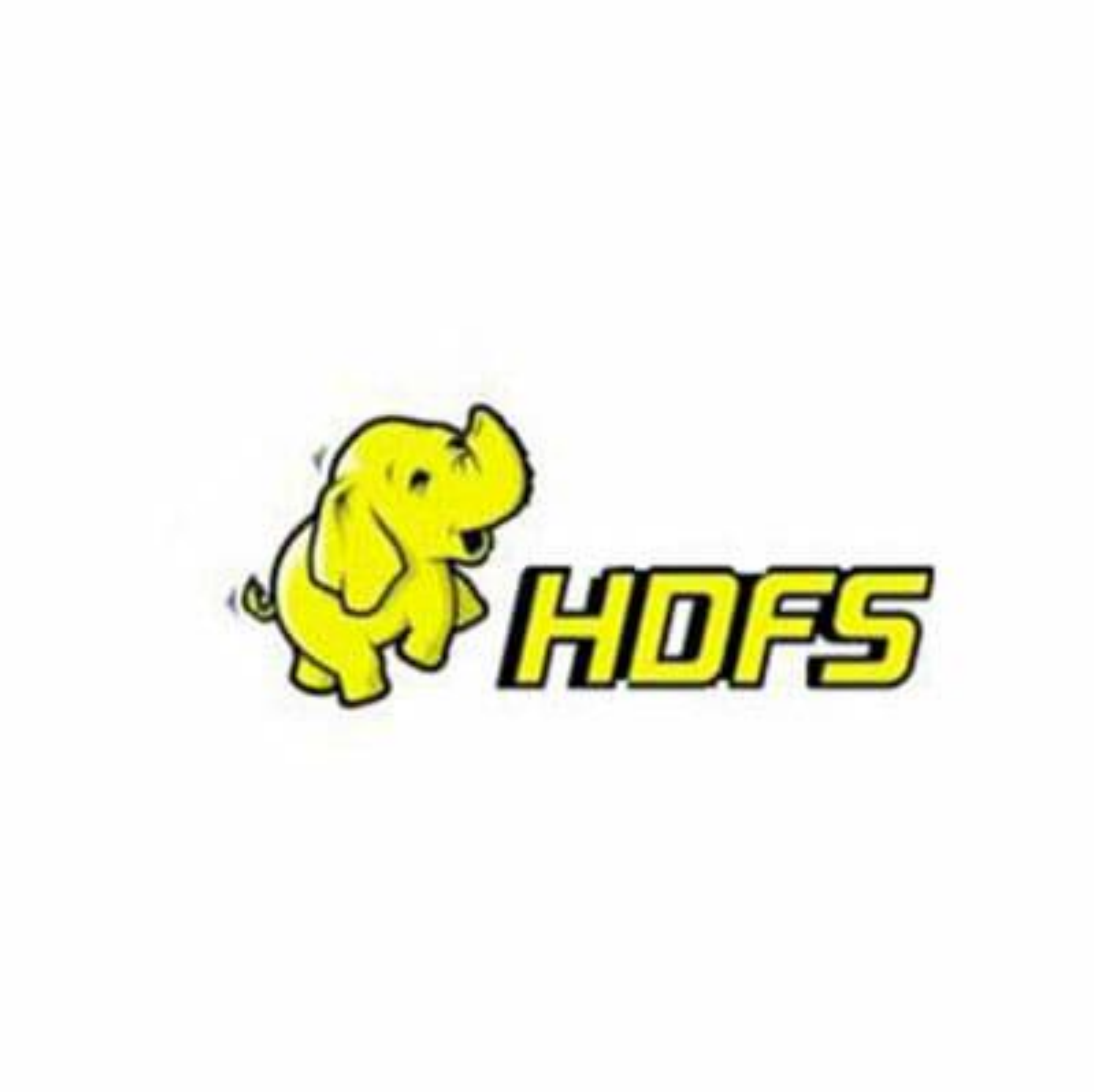
There are 3 primary components in Hadoop:

- Core Hadoop ecosystem
- Query Engines
- External data storage

Core Hadoop Ecosystem



HDFS



- Hadoop distributed file system.
- Abstraction of a file system over a cluster. Stores large amount of data by transparently spreading it on different machines.
- So, it makes all of the hard drives on our cluster look like one giant file system and not only that it actually maintains redundant copies of that data.

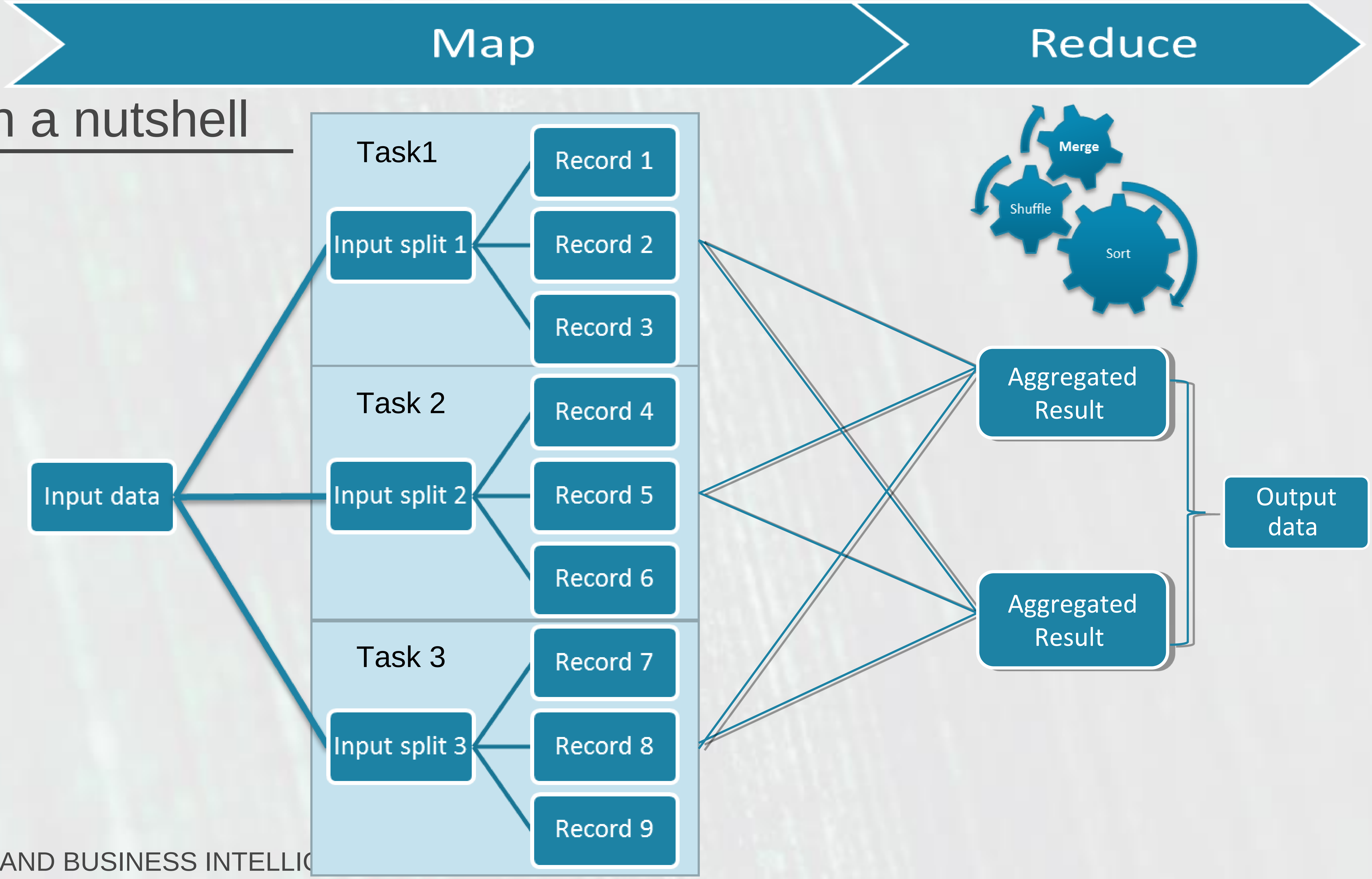
YARN

- Yet another resource negotiator
- The fundamental idea of YARN is to split up the functionalities of resource management and job scheduling/monitoring into separate daemons.
- YARN is where the data processing starts to come into play. So yarn is basically the system that manages the resources on your computing cluster.
- It's what decides what gets to run tasks when what nodes are available for extra work which nodes are not which ones are available which ones are not available so it's kind of the the heartbeat that keeps your cluster going

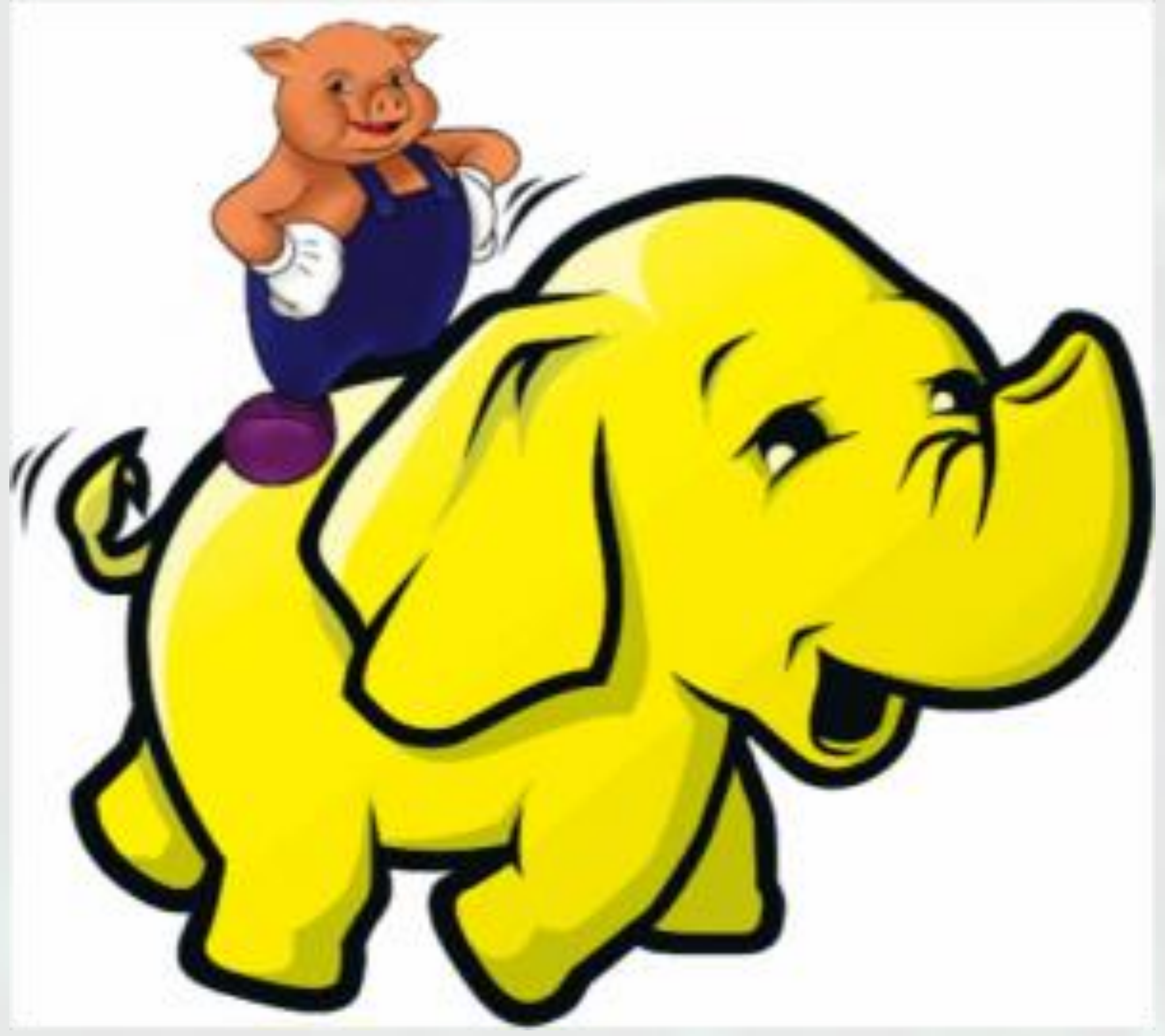
Map Reduce

- Programming metaphor that allows to process your data across an entire cluster.
- Enables parallel execution of data processing programs
- Contains mappers and reducers.
- Mappers have the ability to transform your data in parallel across your entire computing cluster in a very efficient manner.
- Reducers are what aggregate that data together.
- In a nutshell: HDFS places the data on the cluster and MapReduce does the processing work

Map Reduce in a nutshell



Pig



- Pig is a very high level programming API that allows you to write simple scripts that look a lot like SQL.
- So if you don't want to write Java or python map reduce code and you're more familiar with a scripting language that has sort of a SQL style syntax Pig is for you.
- Pig will actually transform that script into something that will run on map reduce which in turn goes through yarn and HDFS to actually process and get the data that it needs to get the answer you want.
- Just a high level scripting language that sits on top of map reduce. A high-level platform for handling any kind of data and runs on Hadoop.
- It uses PigLatin language to write programs and enables us to spend less time in writing map-reduce programs for analysing large data sets.

Hive

- Solves a similar problem to pig
- Hive works on flat files and does not support indexes and transactions Hive does not support updates and deletes. Rows can only be added incrementally
- Hive works more as a data warehouse than as a DBMS
- Execute SQL queries on the data that's stored on your Hadoop cluster even though it's not really a relational database under the hood.
- All common SQL constructs can be used Joins, subqueries, functions etc..,

Ambari

- Sits on top of everything and lets you have a view into the actual state of your cluster
- Gives you a view of your cluster and lets you visualize what's running on your cluster
- Has an interface to do host of operations such as execute hive queries or import databases into hive or execute Pig queries etc..,
- Ambari is available to easily navigate and manage different systems on Hadoop

Spark

- Most exiting technologies in Hadoop ecosystem.
- Sitting at the same level as Map reduce.
- Requires some programming knowledge.
- Write your SPARK scripts using either Python or Java or the Scala programming language, Scala being preferred.
- Extremely fast, 100x faster than traditional Hadoop approach as it relies on in-memory computing So if you need to very quickly and efficiently and reliably process data on your cluster SPARK is a really good choice for that.
- Very versatile it can do things like handle SQL queries that can do machine learning across an entire cluster of information Can also handle streaming data in real time.

Tez

- Similar to Spark
- Uses directed acyclic graph due to which it can produce more optimal plans for actually executing queries
- Used in conjunction with Hive to accelerate it so Hive through Tez can be faster than Hive through map reduce.

HBase

- Sits off to the side and it's a way of exposing the data on your cluster to transactional platforms.
- NoSQL database appropriate for hitting from a web application doing all types of transactions.

Apache Storm

- Processing streaming data say if you have streaming data such as sensors or web logs , using storm we can process that in real time.
- Using storm we can update our machine learning models or transform data in to a database all in real time as it comes in.

Oozie

- For scheduling jobs on your cluster
- So when you have more complicated operations that require loading data into hive and then integrating that with Pig and maybe querying it with SPARK and then transforming the results into HBase , Oozie can manage that all for you and make sure that it runs reliably on a consistent basis.

Zookeeper

- For coordinating everything on your cluster.
- For keeping track of which nodes are up which nodes are down.
- It's a very reliable way of just kind of keeping track of shared states across your cluster that different applications can use To maintain reliable and consistent performance across the cluster.

Tools for data ingestion:

- Get data into your cluster and onto HDFS from external sources

❖ **Sqoop**:

- To tie your Hadoop database in to a RDBMS so it acts as a connector between Hadoop and your legacy databases

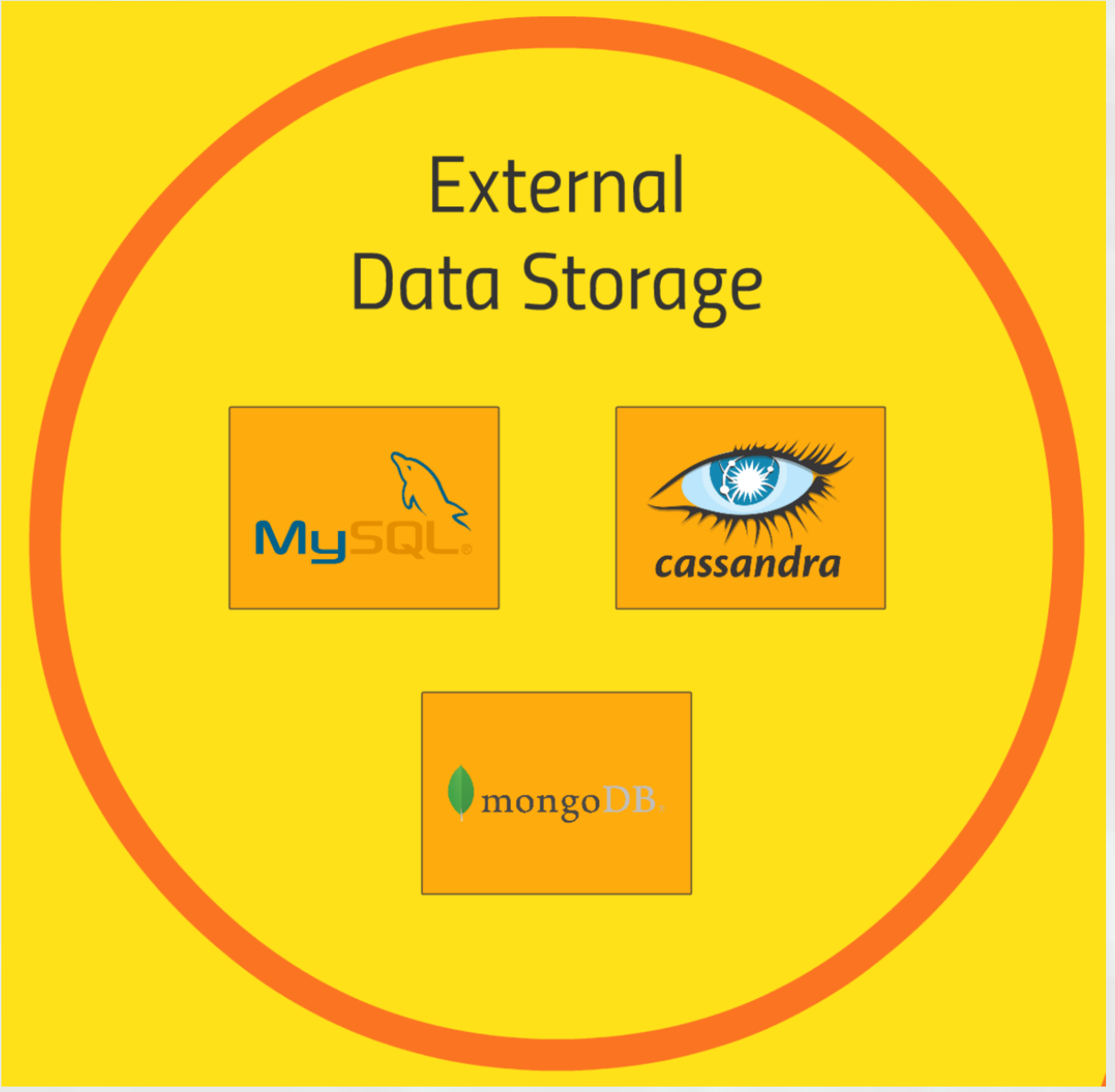
❖ **Flume**:

- It's a way of actually transporting Web logs at a very large scale and very reliably to your cluster. So let's say you have a fleet of web servers Flume can actually listen to the web logs coming in from those web servers in real time and publish them into your cluster in real time for processing by something like storm or spark streaming.

❖ **Kafka**:

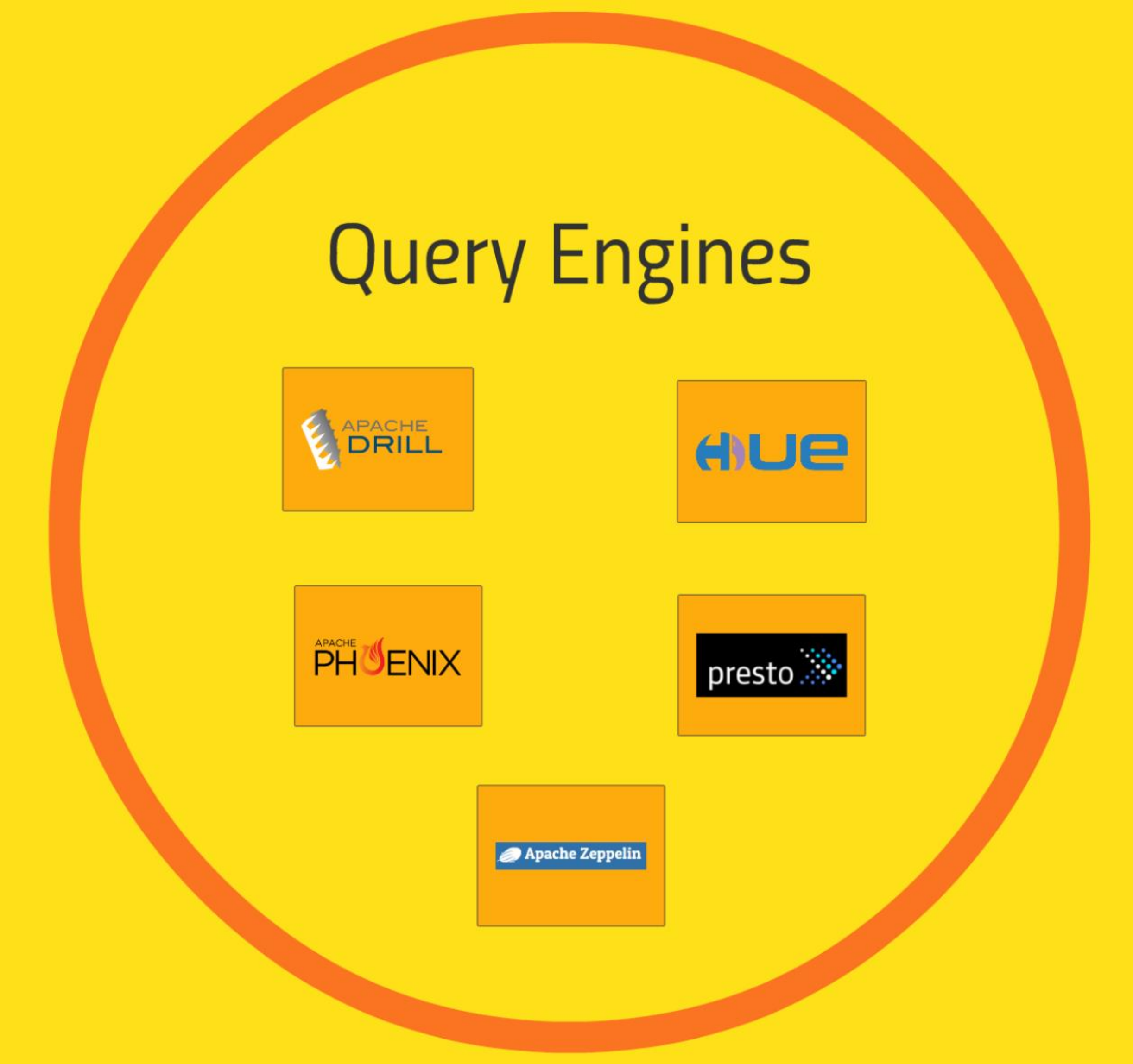
- A distributed publish-subscribe messaging system designed for processing of real-time activities stream data (logs, social media streams).

External data storage



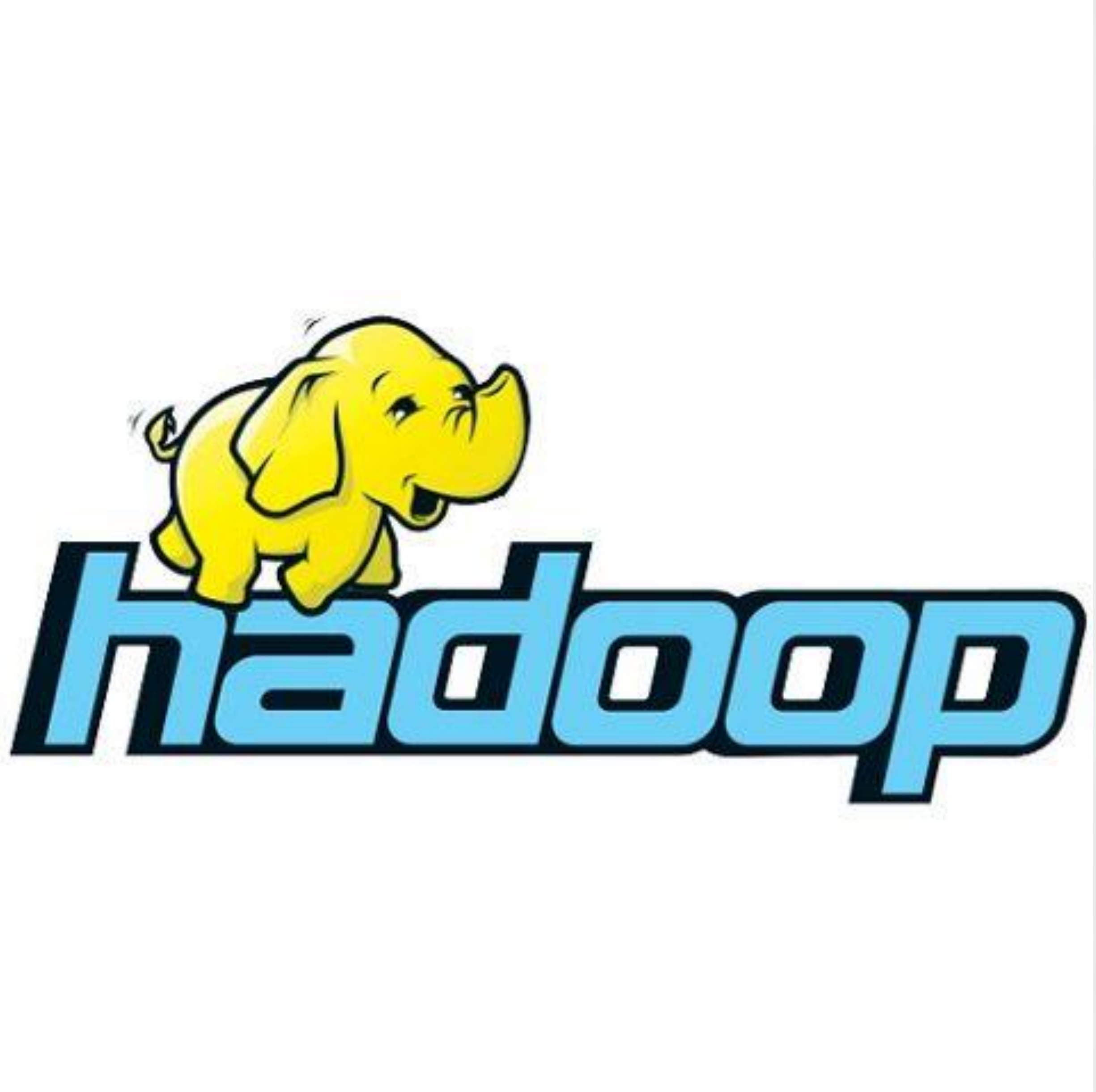
- MySQL
- Cassandra
- MongoDB

Query Engines



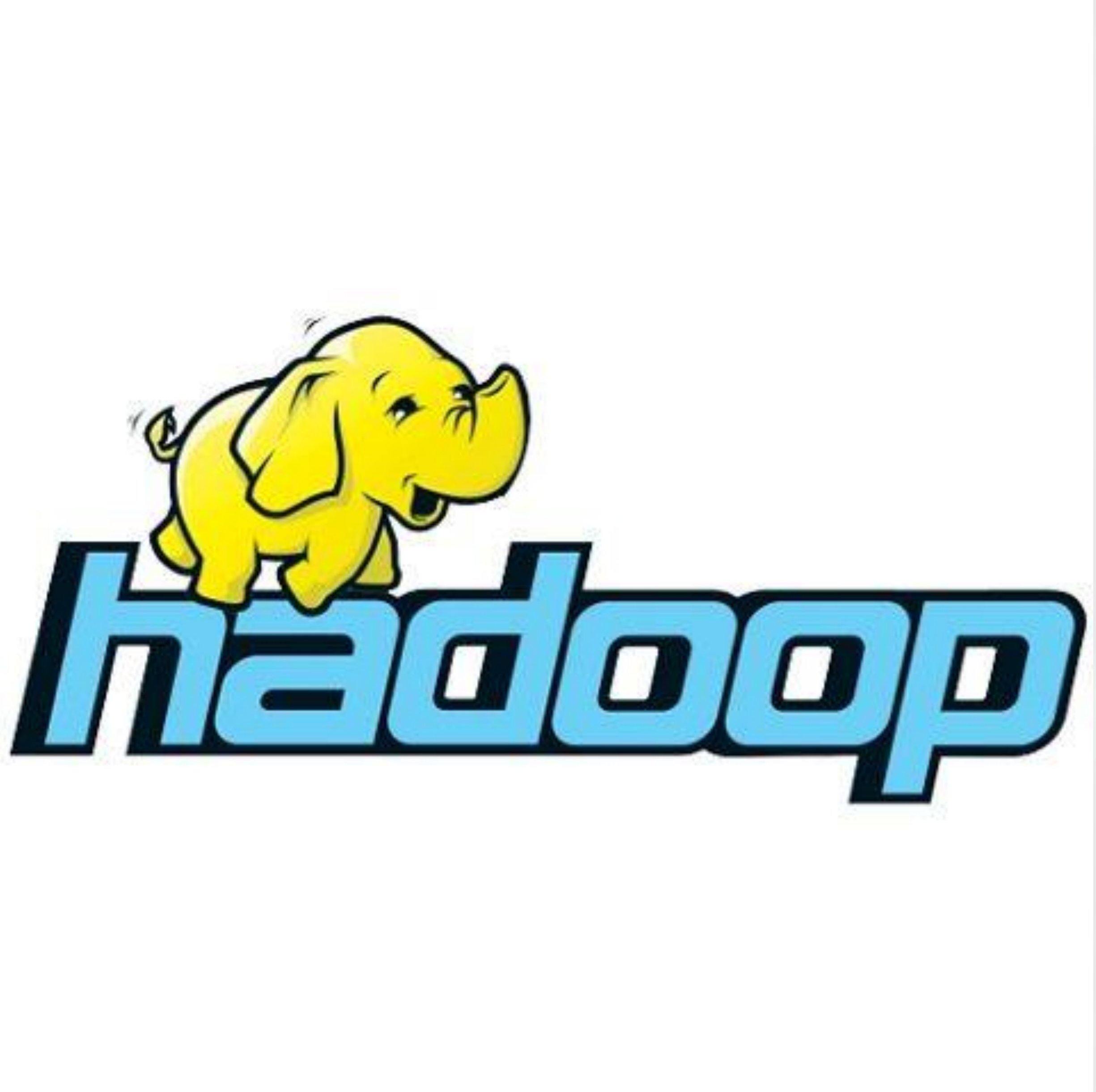
- Apache Drill
- Hue
- Apache Phoenix
- Presto
- Apache Zepplin

Hadoop Installation modes



- Local mode
- Pseudo distributed mode
- Fully distributed mode

Hadoop Pros and Cons



❖ Good for:

- Repetitive tasks on big size data

❖ Not good for:

- Replacing a RDMBS
- Complex processing requiring various phases and/or iterations
- Processing small to medium size data

Master Executive di II Livello

BIG DATA ANALYSIS AND BUSINESS INTELLIGENCE

Grazie

Vamsi Krishna Varma Gunturi
Deep learning researcher at ISTAT
vamsivarmagunturi@gmail.com

fondazione INUIT TOR VERGATA