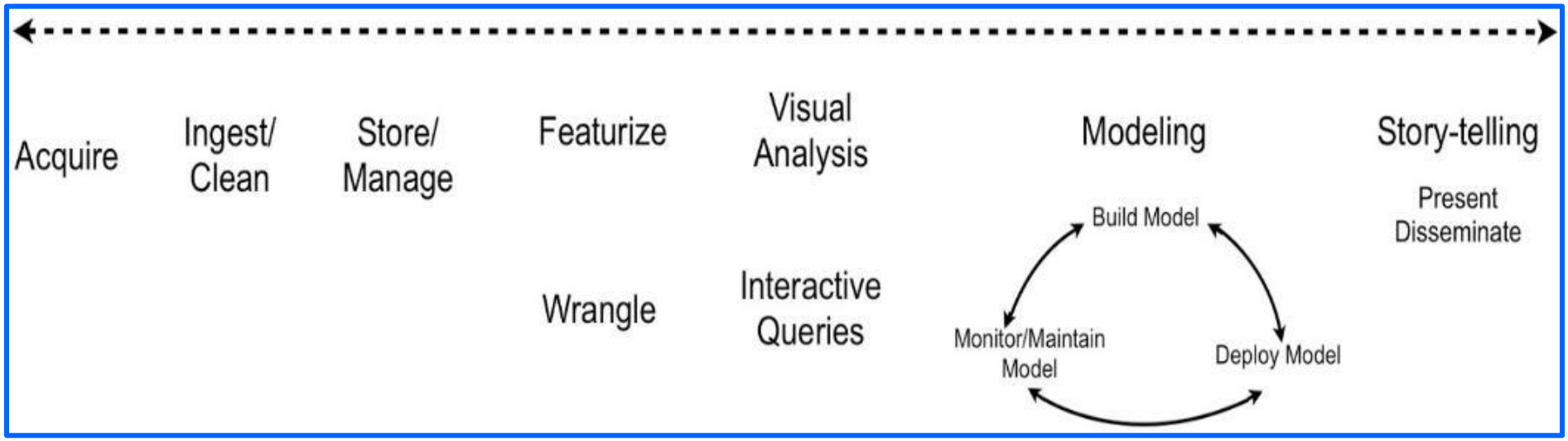
Inspire…Educate…Transform.

Introduction to Hadoop and it's Components

*"Big Data" is **high-volume**, **high-velocity** and **high-variety** information assets that demand **cost-effective**, innovative forms of information processing for **enhanced insight** and decision making.*

Gartner

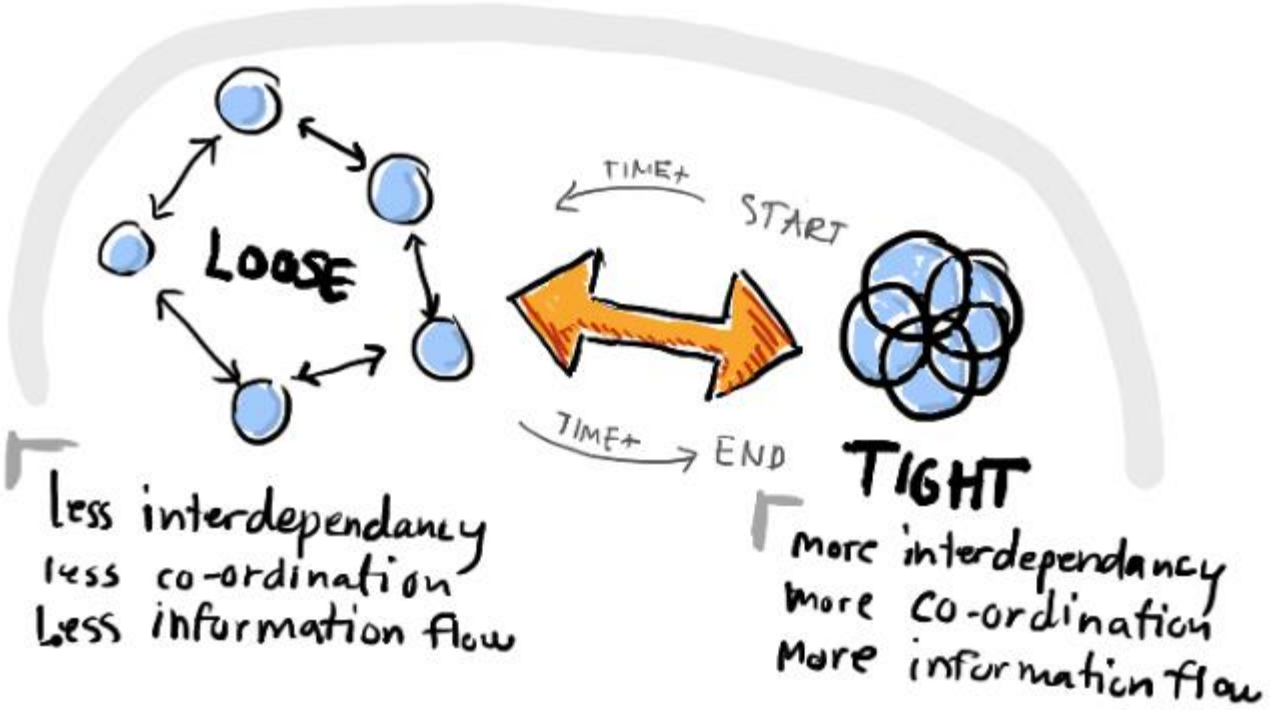
Foundations & Overview



How is each of these stages affected by Big Data?

https://www.oreilly.com/ideas/data-analysis-just-one-component-of-the-data-science-workflow

Foundations & Overview



Foundations & Overview

	Tightly coupled system	*Loosely coupled system*
1	It has shared memory concept	It has distributed memory concept
2	Contention is high in tightly coupled	Contention is low in loosely coupled
3	It has low scalability	It has high scalability
4	It has low delay	It has high delay
5	Data rate in tightly coupled system is high	Data rate in loosely coupled system is low
6	Cost of tightly coupled system is high	Cost of loosely coupled system is low
7	It has dynamic interconnection network	It has static interconnection network
8	It operates on Single Operating System	It operates on Multiple Operating System
9	In tightly coupled system cache memory assign according to the need of processing	In loosely coupled system, each process have its own cache memory
10	Throughput is high in tightly coupled	Throughput is low in loosely coupled
11	Security is high in tightly coupled	Security is low in loosely coupled
12	Low space in this architecture	High space in this architecture
13	Power consumption is lower than loosely coupled system	Power consumption is higher than tightly coupled system
14	Not reusable in the case of flexibility	Reusable in the case of flexibility
15	Ex., Zeon processor	Ex., Beowulf cluster

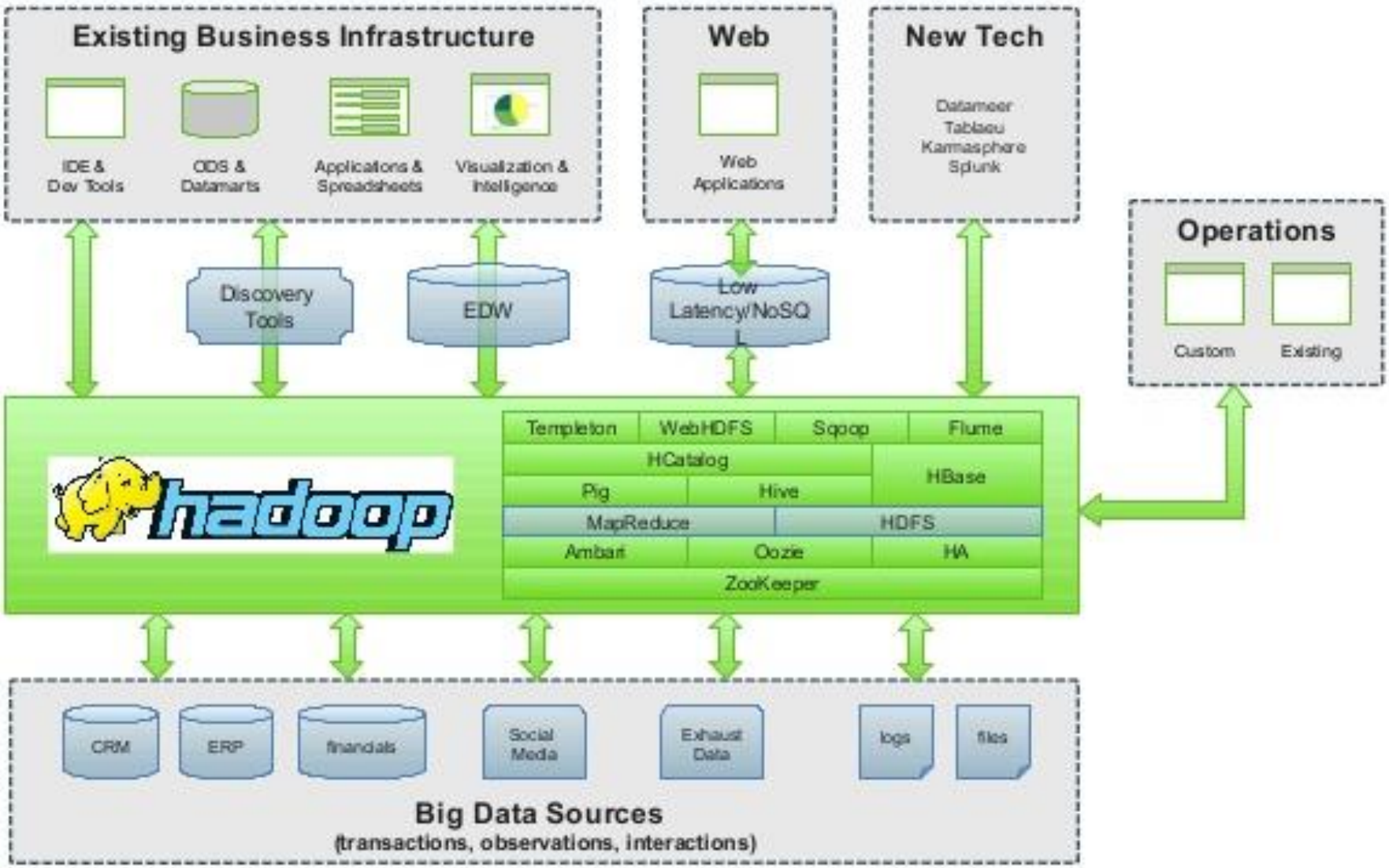
Hadoop & Spark Ecosystems

Hadoop: 30,000 feet view (contd.)

- Distribute data initially
  - Let processors / nodes work on local data
  - Minimize data transfer over network
  - Replicate data multiple times for increased availability
- Write applications at a high level
  - Programmers should not have to worry about network programming, temporal dependencies, low level infrastructure, etc
- Minimize talking between nodes (*share-nothing)*

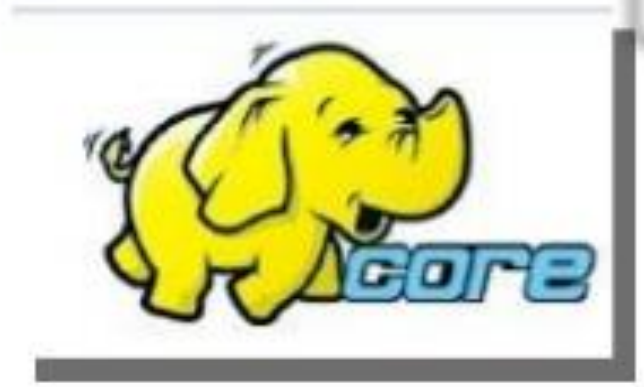
Hadoop in Enterprise Data Architectures

(The Hortonworks View)

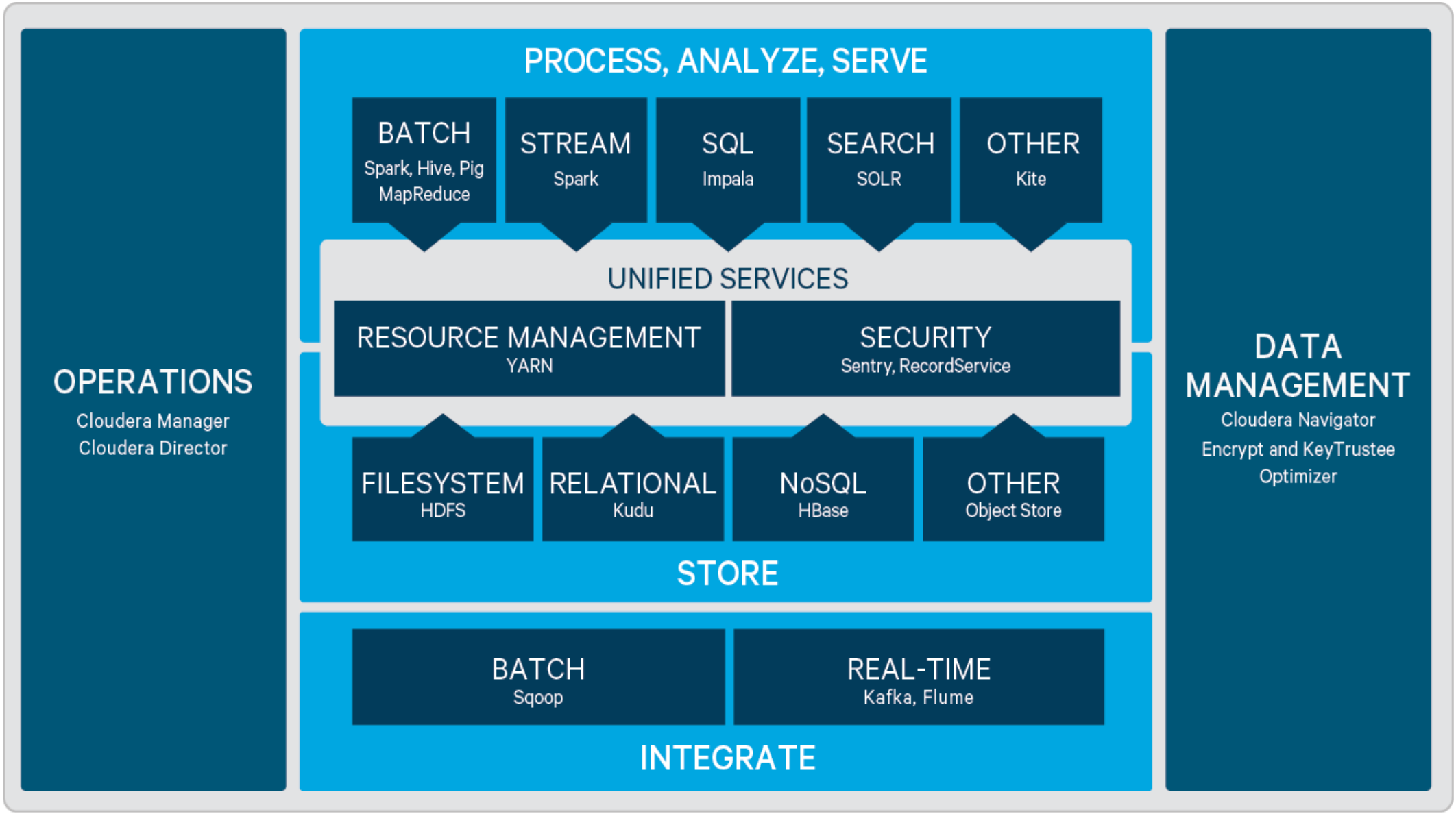


Hadoop Distributions

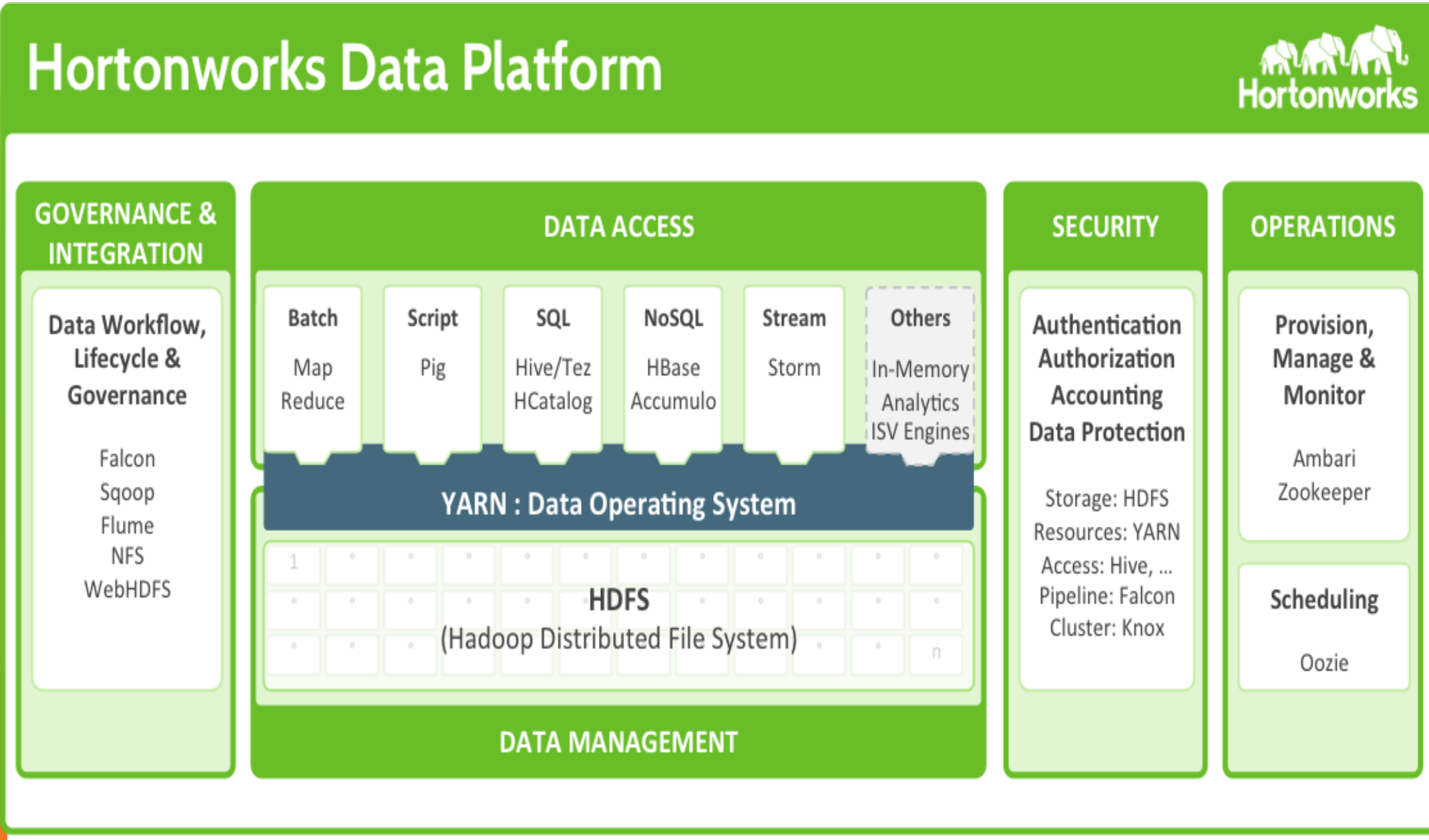
- Open Source
  - Apache
- Commercial
  - Cloudera
  - Hortonworks
  - MapR
  - AWS MapReduce
  - Microsoft HDInsight



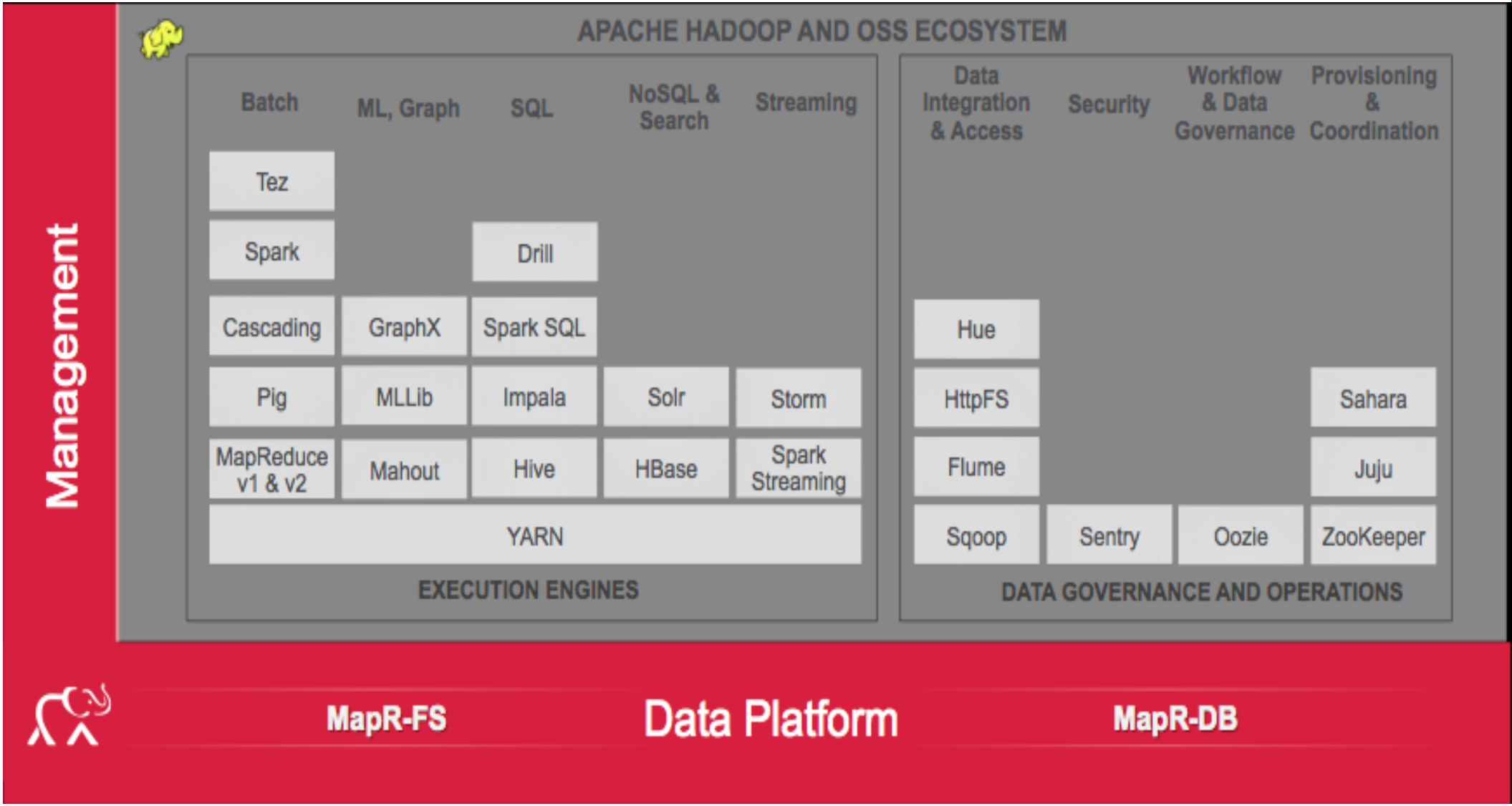
Cloudera Distribution of Hadoop



Hortonworks Data Platform

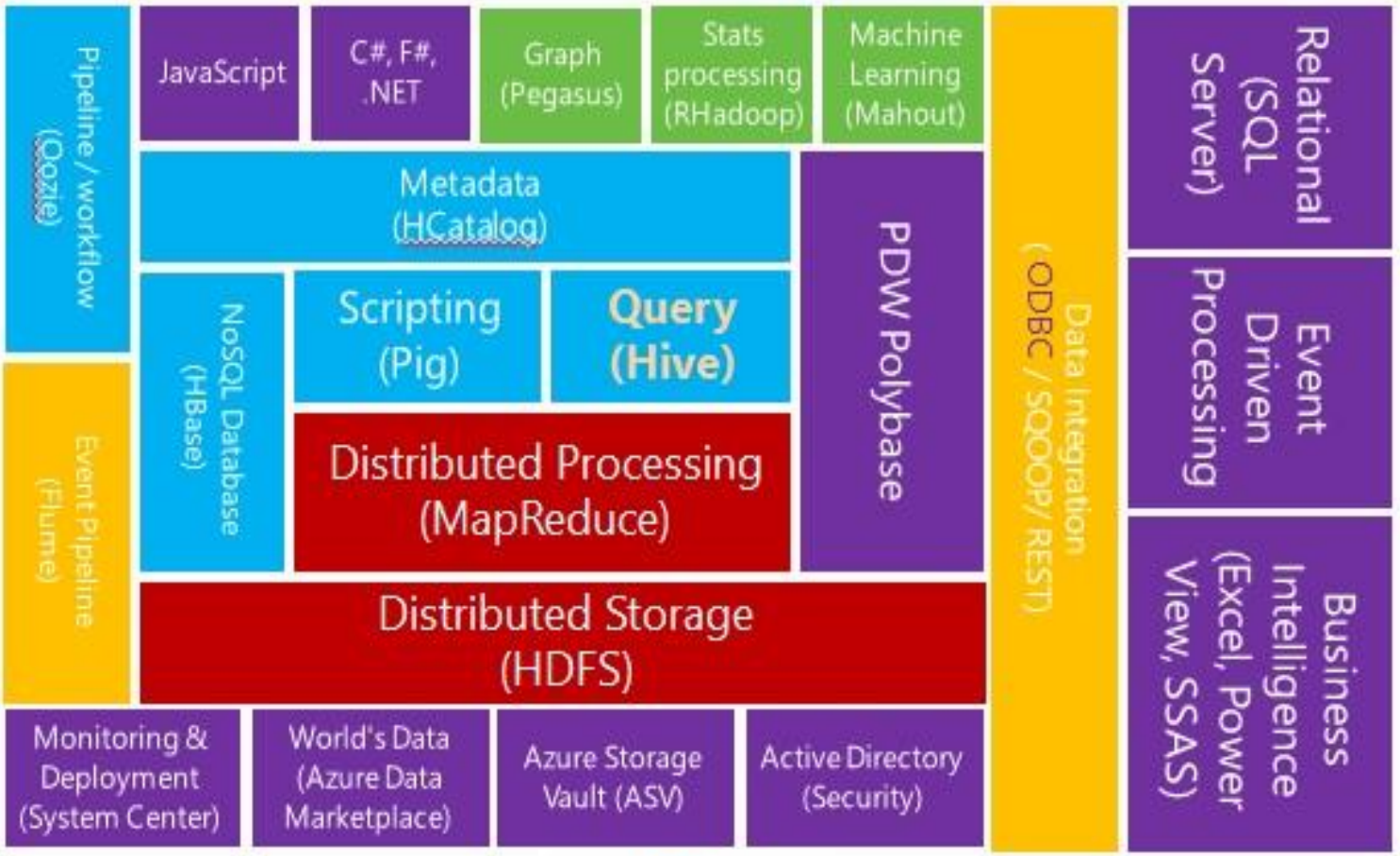


MapR Distribution



AWS big data portfolio

HDINSIGHT / HADOOP Eco-System



Legend
Red = Core Hadoop
Blue = Data processing
Purple = Microsoft integration points and value adds
Orange = Data Movement
Green = Packages

Windows Azure

GFS & HDFS

A New Need & Breed of DFS

NFS, etc.

Independence
Small Scale
Variety of workloads

GFS

Cooperation
Large scale
Very specific, well-understood workloads

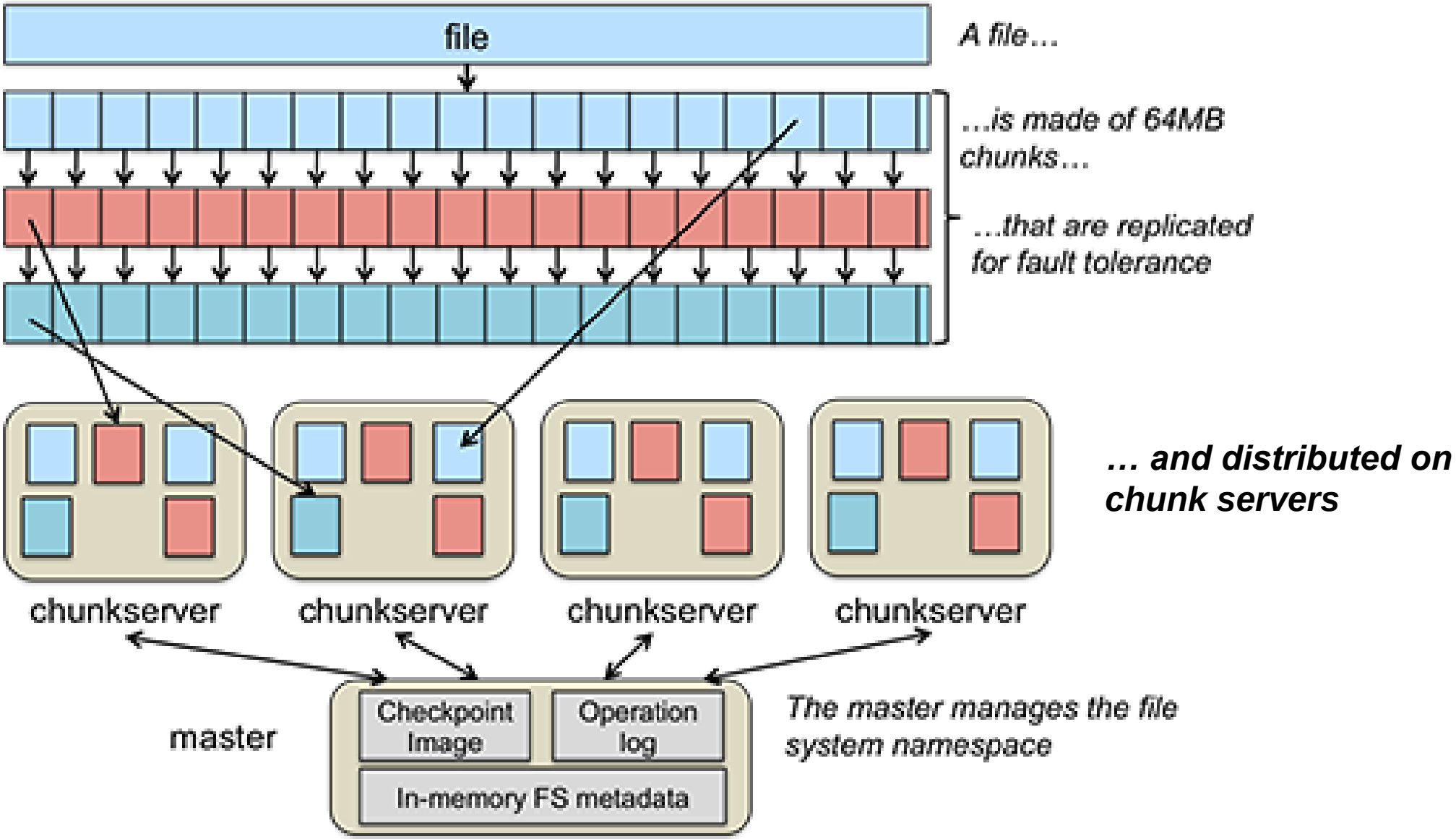
The Google File System

Sanjay Ghemawat, Howard Gobioff, and Shun-Tak Leung

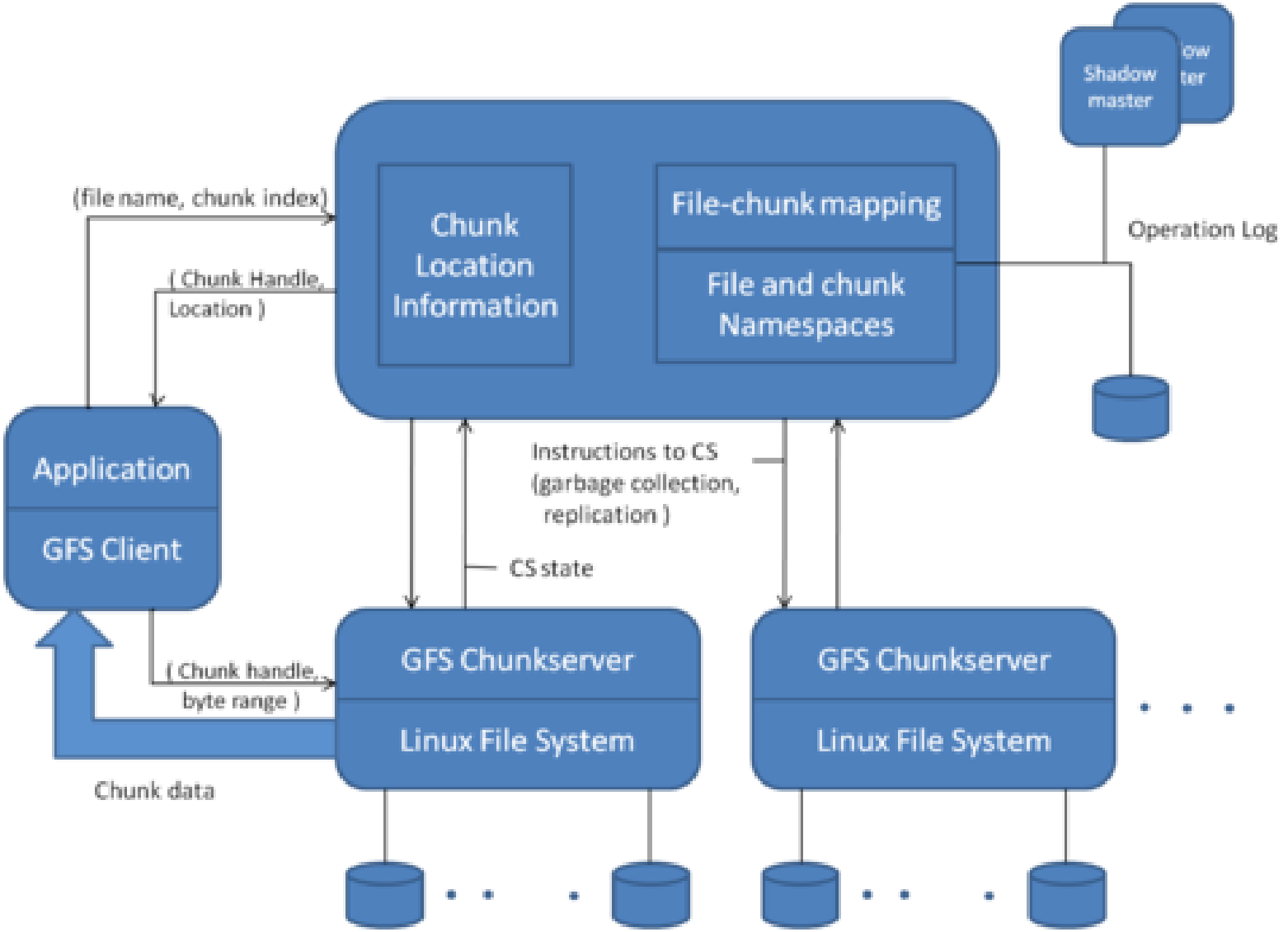
Google*

Sanjay Ghemawat 2003

GFS


file
A file…
…is made of 64MB chunks…
…that are replicated for fault tolerance
chunkserver
chunkserver
chunkserver
chunkserver
… and distributed on chunk servers
master
Checkpoint image
Operation log
In-memory FS metadata
The master manages the file system namespace

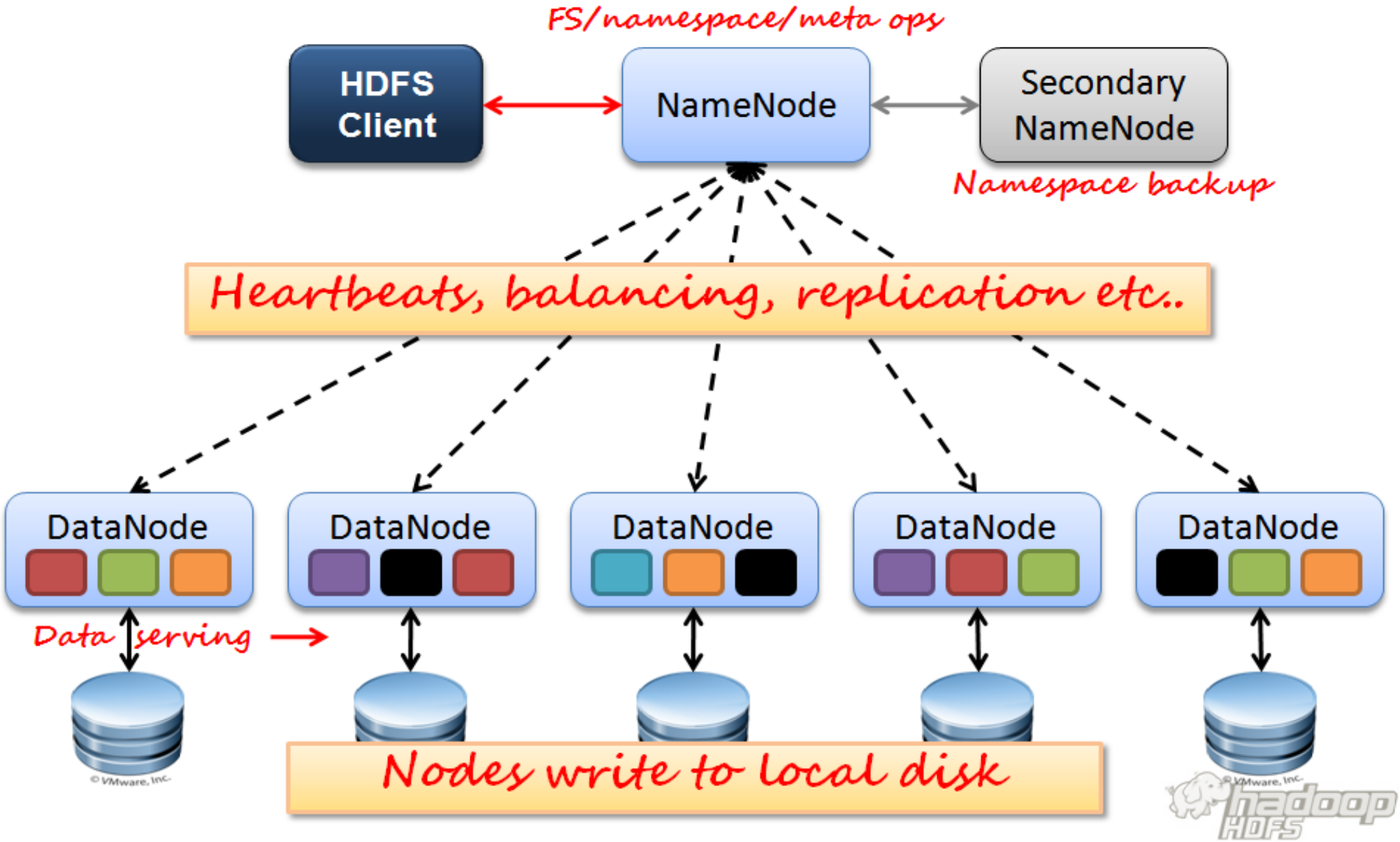
GFS: Masters, Workers and Clients



GFS Master Responsibilities

- Metadata storage
- Namespace management/locking
- Periodic communication with chunkservers
  - give instructions, collect state, track cluster health
- Chunk creation, re-replication, rebalancing
  - balance space utilization and access speed
  - spread replicas across racks to reduce correlated failures
  - re-replicate data if redundancy falls below threshold
  - rebalance data to smooth out storage and request load
- Garbage Collection
  - simpler, more reliable than traditional file delete
  - master logs the deletion, renames the file to a hidden name
  - lazily garbage collects hidden files
- Stale replica deletion
  - detect “stale” replicas using chunk version numbers

HDFS CDH3: Open source reimplementation of GFS



Name Node Metadata

- Metadata in Memory
  - The entire metadata is in main memory
  - No demand paging of metadata
- Types of metadata
  - List of files
  - List of Blocks for each file
  - List of Data Nodes for each block
  - File attributes, e.g. creation time, replication factor
- A Transaction Log
  - Records file creations, file deletions etc

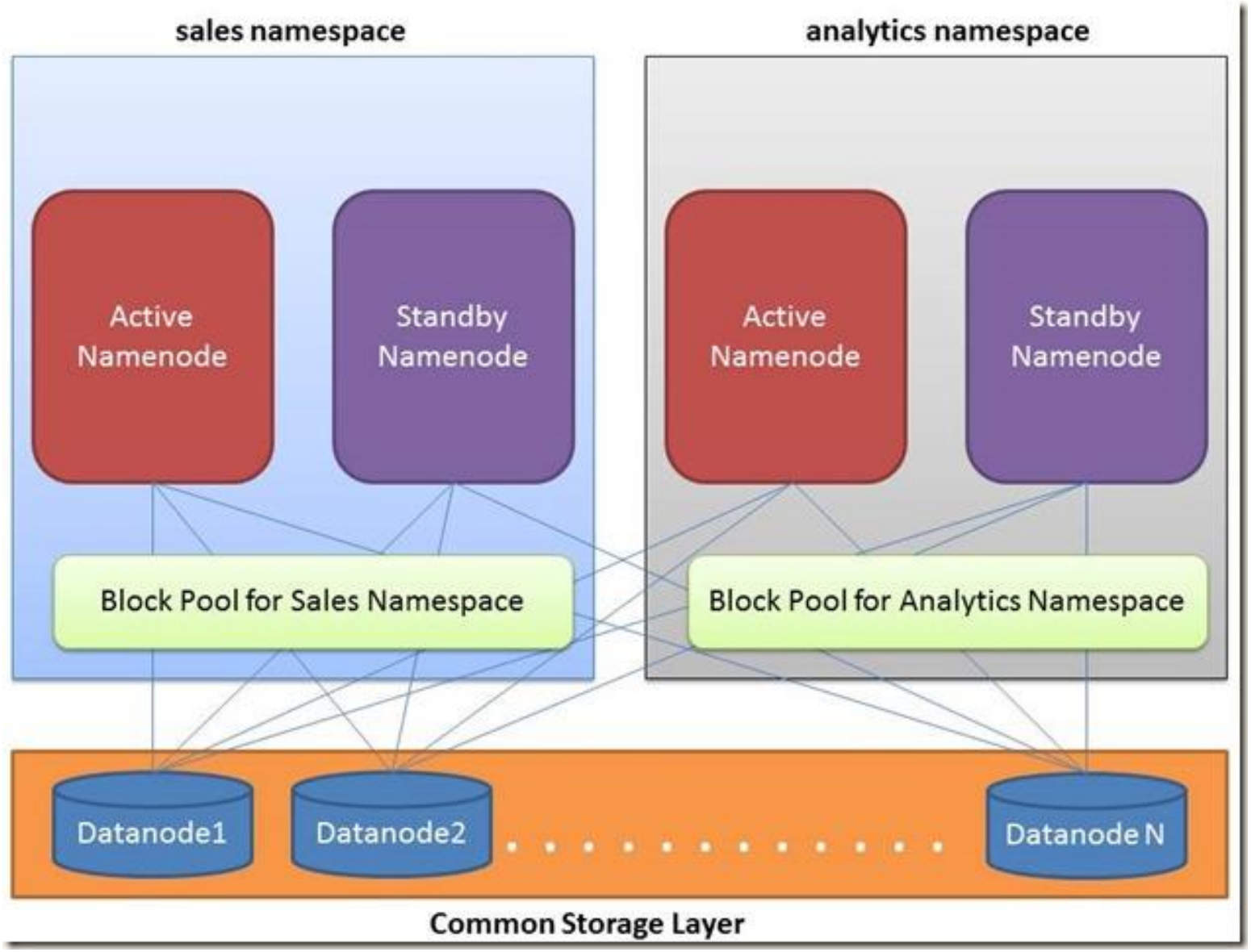
Block Placement

- Current Strategy
  - One replica on local node
  - Second replica on a remote rack
  - Third replica on same remote rack
  - Additional replicas are randomly placed
- Clients read from nearest replicas
- Would like to make this policy pluggable

HDFS: More Elaboration

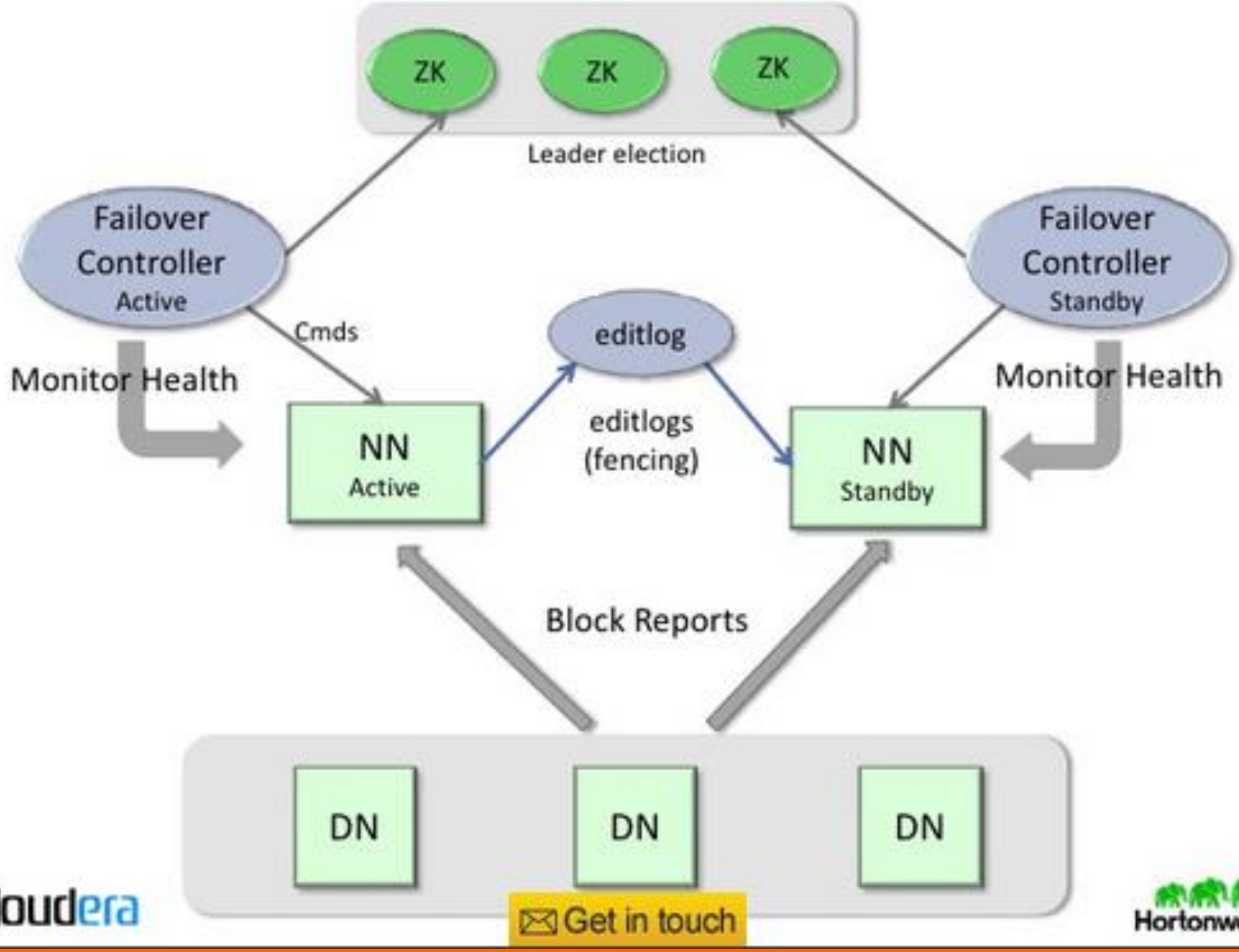
- **Heart beats**
  - Data Nodes send hear beat to the Name Node
  - Once every 3 seconds
  - Name Node uses heartbeats to detect Data Node failure
- **Rebalancing**: % disk full on Data Nodes should be similar
  - Usually run when new DataNodes are added
  - Cluster is online when Rebalancer is active
  - Rebalancer is throttled to avoid network congestion
  - Command line tool

HDFS 2.0: High Availability, Federated



http://pe-kay.blogspot.in/2016/02/configuring-federated-hdfs-cluster-with.html

HDFS 2.0: High Availability Elaborated



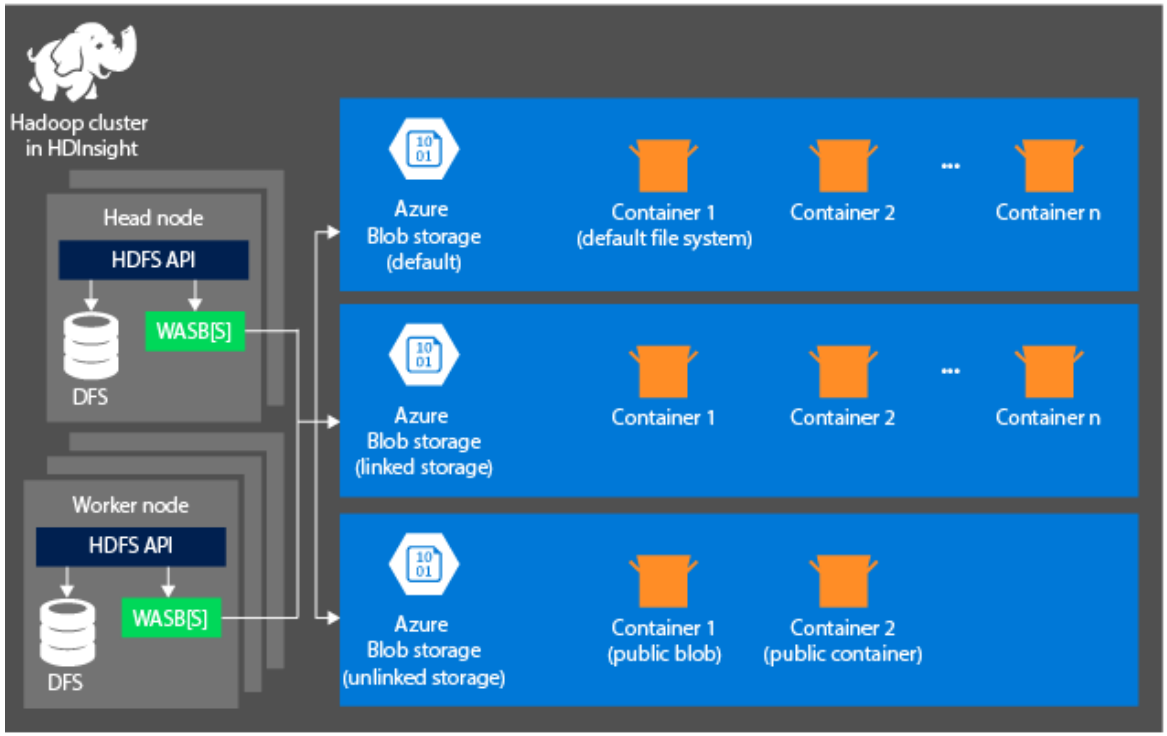
Reference: Hadoop Summit 2012 | HDFS High Availability talk

AZURE STORAGE

Microsoft Azure provides two types of storage option for HDInsight clusters.

- Blob Storage
- Data Lake Store

Azure HDInsight provides a fully featured HDFS file system over blob storage.



Source:https://docs.microsoft.com/en-us/azure/hdinsight/hdinsight-hadoop-use-blob-storage

Why Azure Storage not HDFS for HDInsight Cluster ?

Basic idea of creating the HDInsight cluster are to process data using different parallel frame works like MapReduce or Spark.

After the processing is done we can delete the clusters , and if we store data in cluster HDFS , data will be lost if we delete the cluster.

Also azure storage is highly available , scalable and low cost storage system which will solve the purpose of storing large amount of data.

Benefits of Azure Storage

Data reuse and Sharing – Data in HDFS is only accessible to resources which has have access to cluster.

Data Archiving – Safely delete the cluster with out worrying about Data.

Cost – Cluster are more expensive than storage accounts.

Elastics Scale Out – HDFS scaled out depends upon the number of nodes in the cluster while azure storage , while elastic scaling is automatic is azure storage.

Source - https://docs.microsoft.com/en-us/azure/hdinsight/hdinsight-hadoop-use-blob-storage

Azure Data Lake Store

Azure data lake store is a file system which is compatible with HDFS and works well with Hadoop eco system.

Unlimited Size : Azure data lake store do not impose any restriction on file size , file size can range from kilo bytes to peta bytes.

High Performance: Data lake store is configured for large scale analytics system, and provides a good throughput for query results over data. Read throughput is good for data lake store as it stores the files over multiple storage servers.

Compatible for All Data: Can store any type of data with out any transformations. No schema is required before to store the data.

Source - https://docs.microsoft.com/en-us/azure/hdinsight/hdinsight-hadoop-use-blob-storage

Replication for Azure Storage

Data is replicated for high availability and recovery purpose.

There are 4 kinds of replication facility available for azure storage.

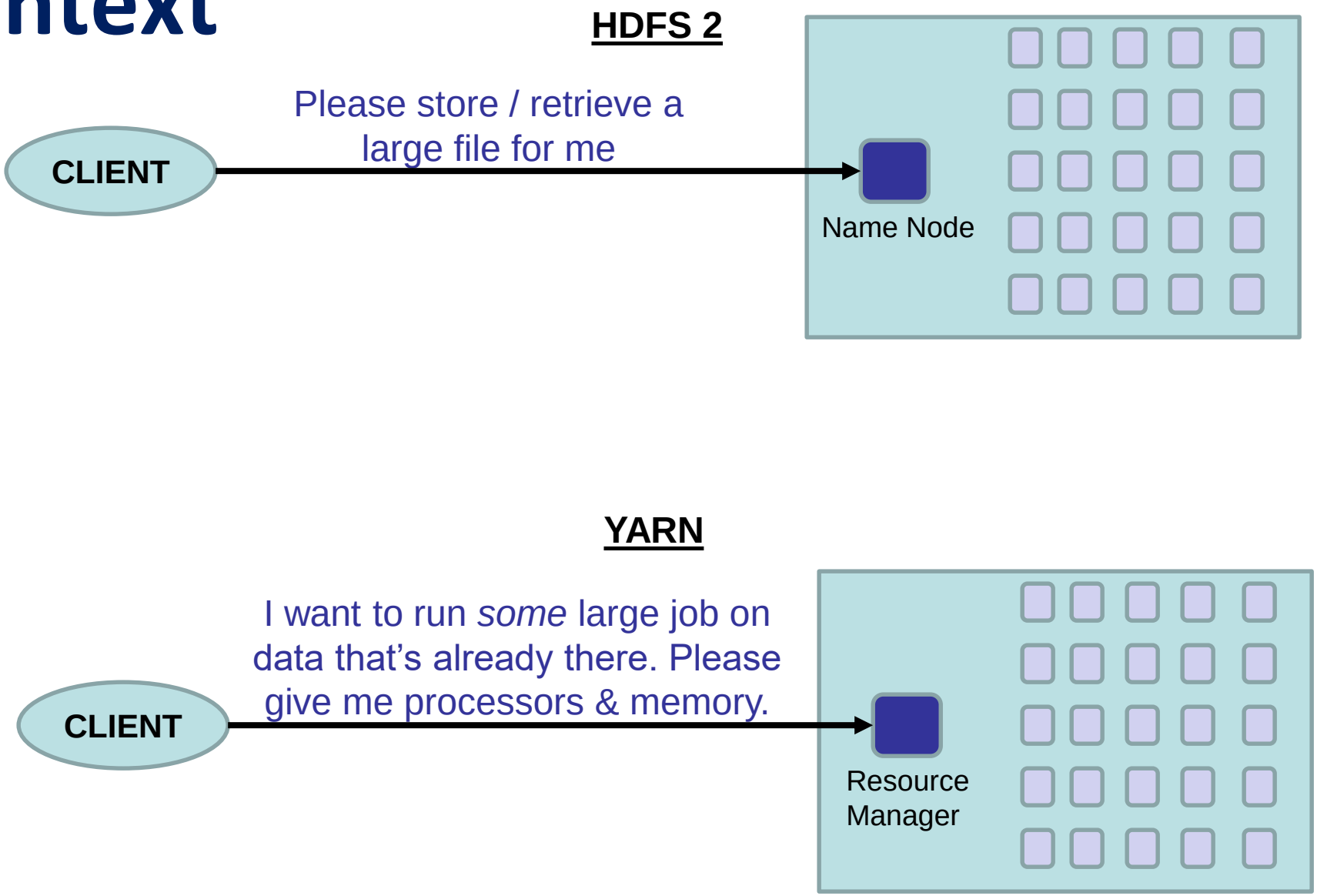
- Locally redundant (LRS)
- Zone Redundant (ZRS)
- Geo-redundant (GRS)
- Read – access Geo – Redundant (RA GRS)

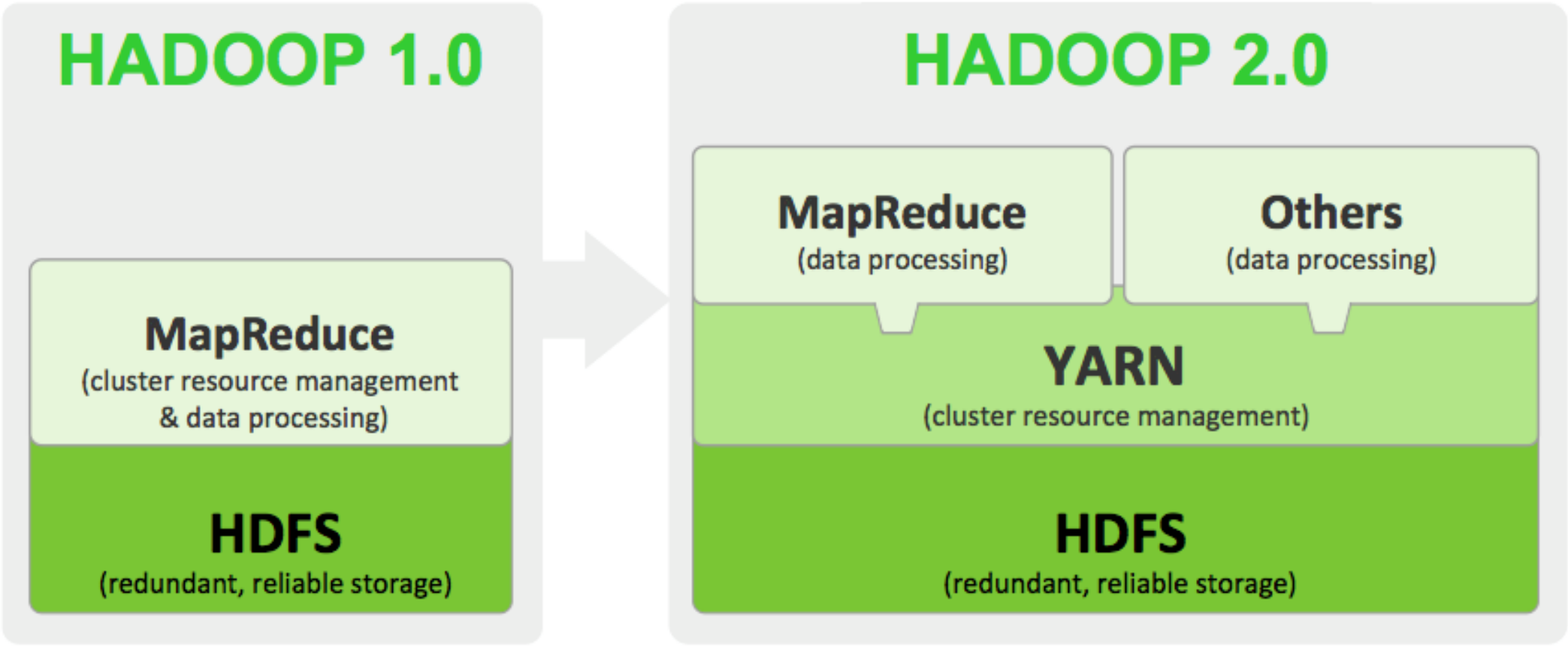
Replication strategy	LRS	ZRS	GRS	RA-GRS
Data is replicated across multiple datacenters.	No	Yes	Yes	Yes
Data can be read from a secondary location as well as the primary location.	No	No	No	Yes
Number of copies of data maintained on separate nodes.	3	3	6	6

Source - https://docs.microsoft.com/en-us/azure/storage/storage-redundancy

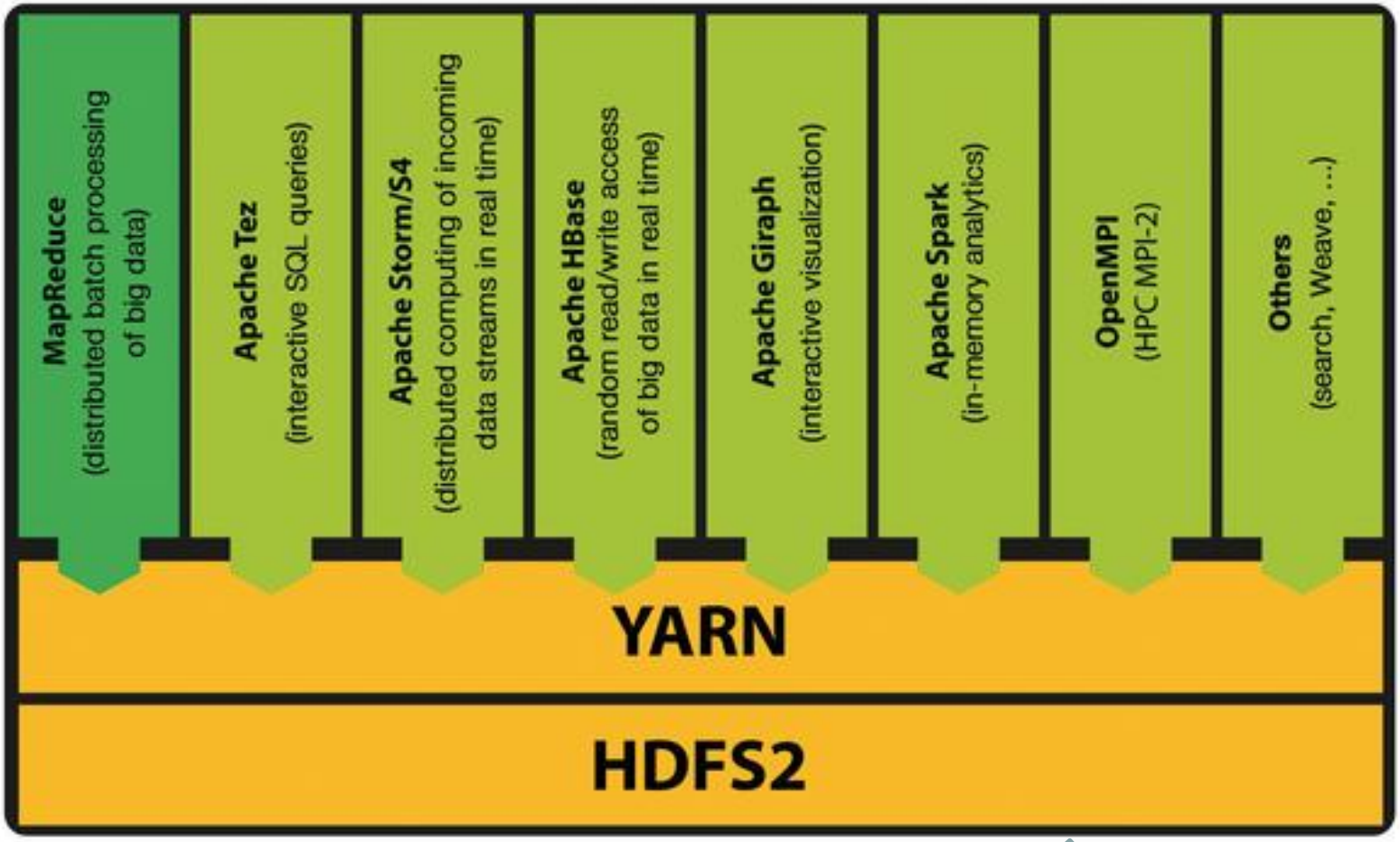
RESOURCE MANAGEMENT

Context


HDFS 2
Please store / retrieve a large file for me
CLIENT
Name Node
YARN
I want to run *some* large job on data that's already there. Please give me processors & memory.
CLIENT
Resource Manager

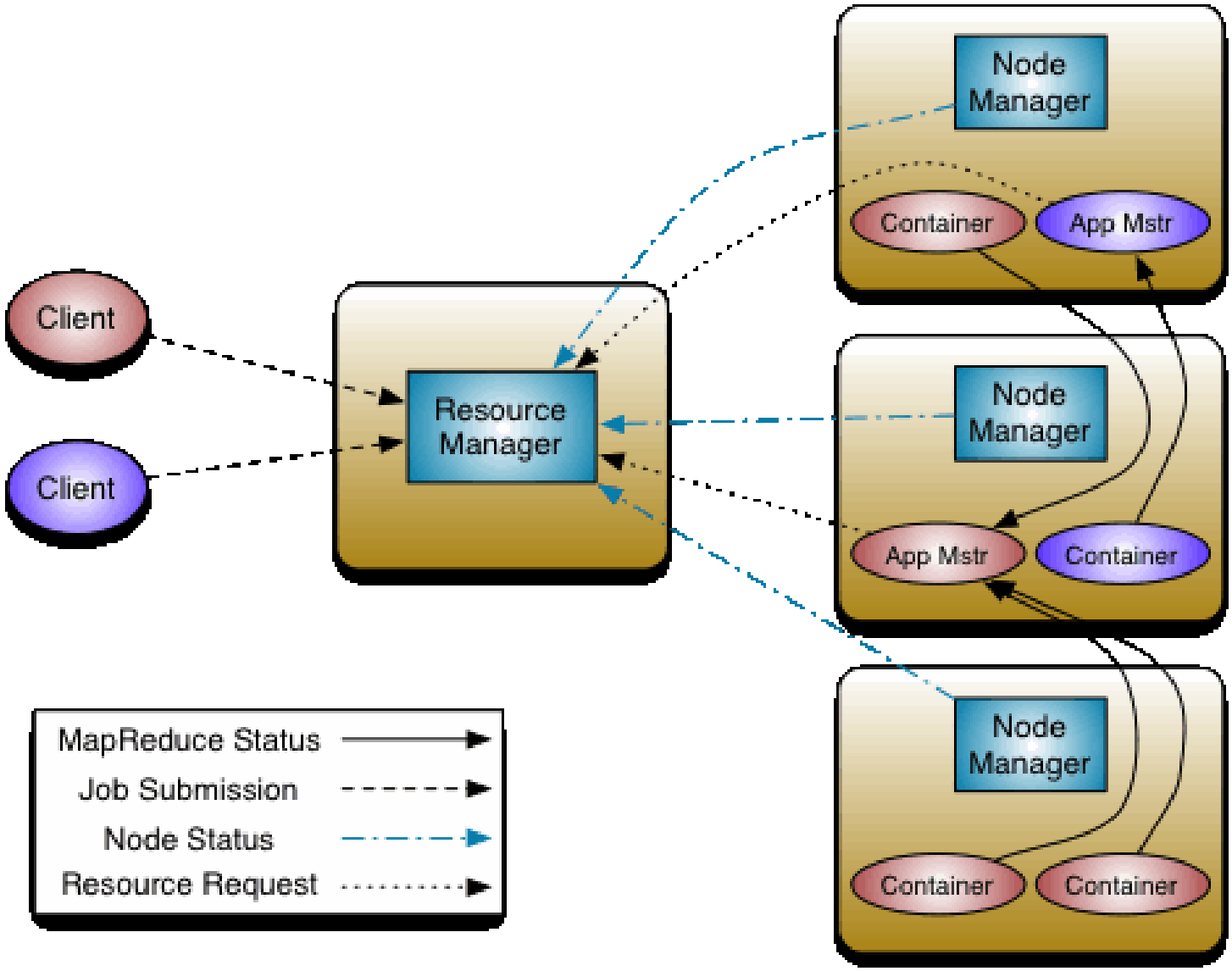

HADOOP 1.0
MapReduce
(cluster resource management
& data processing)
HDFS
(redundant, reliable storage)
HADOOP 2.0
MapReduce
(data processing)
Others
(data processing)
YARN
(cluster resource management)
HDFS
(redundant, reliable storage)

Today's Context of HDFS2, YARN, Ingestion



INGESTION

YARN Architecture



Parallelization platforms

What We Want

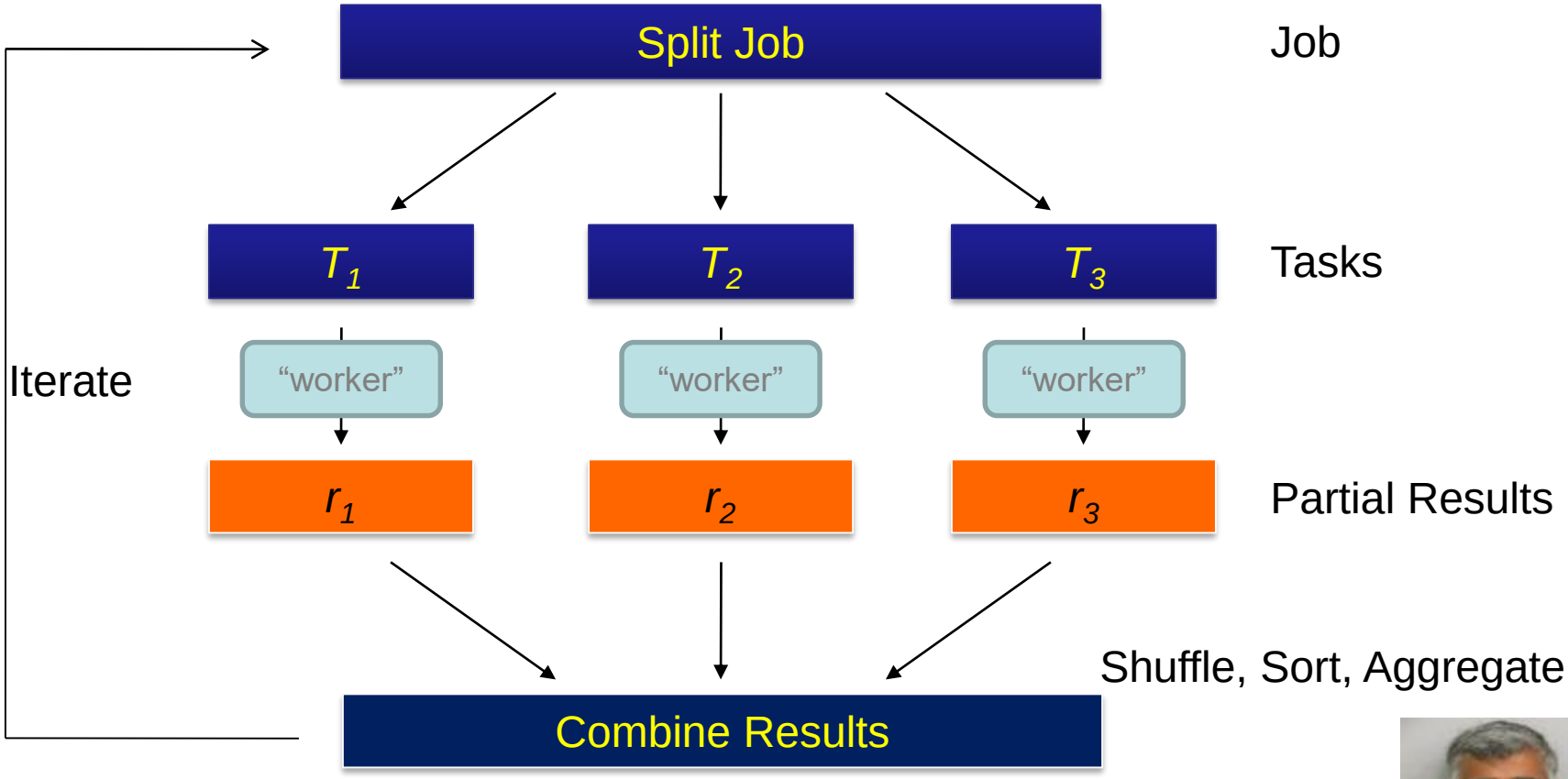
User **easily writes** intuitive instructions...

Parallelization Platform

Which get **auto-converted** into very **efficient** parallel programs...

And gets **reliably executed** on a Hadoop cluster...

Option 1: Map-Reduce

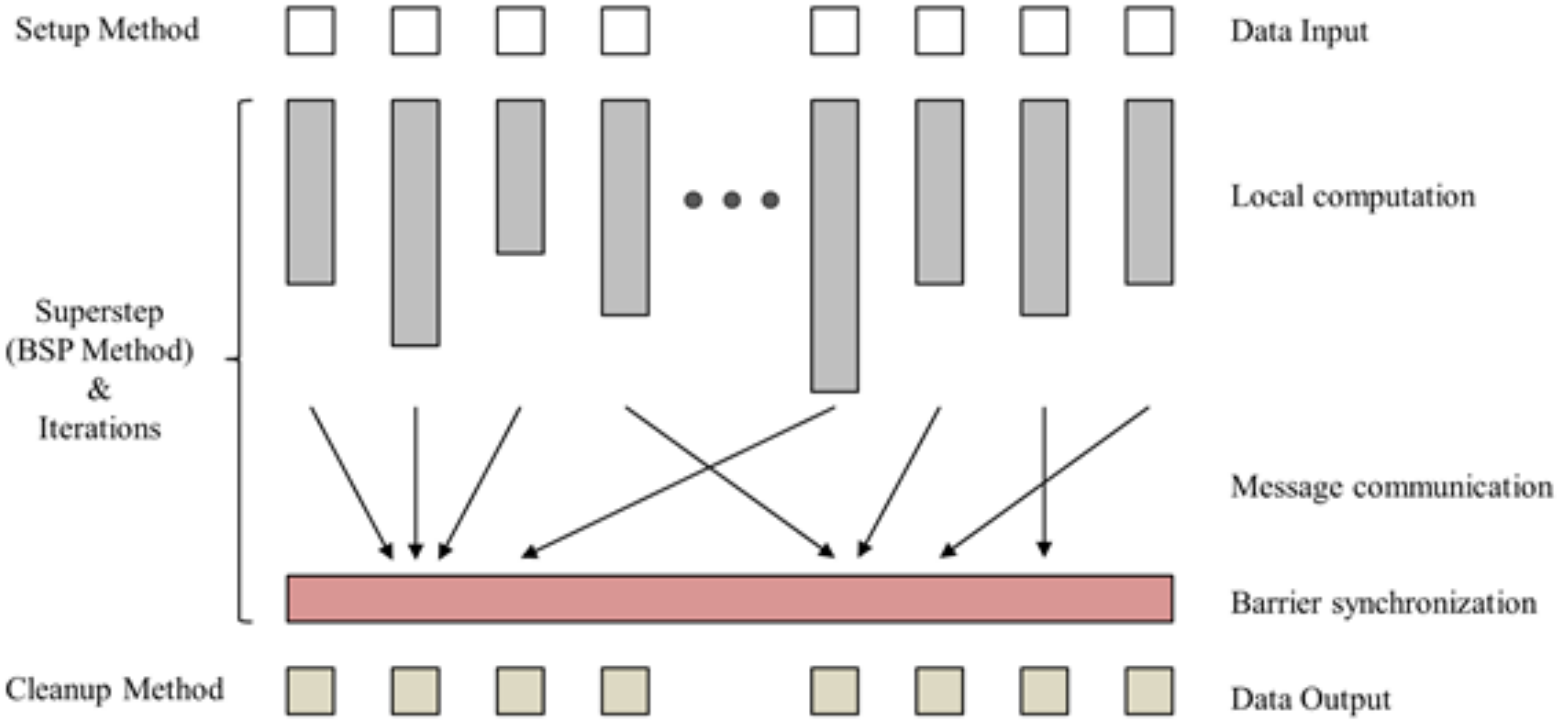


Sanjay Ghemawat

Jeffrey Dean

Tensor Flow, Spanner,
Google News, Ad Sense,
Big Table, GFS, …

Option 2: Bulk Synchronous Parallel (BSP)

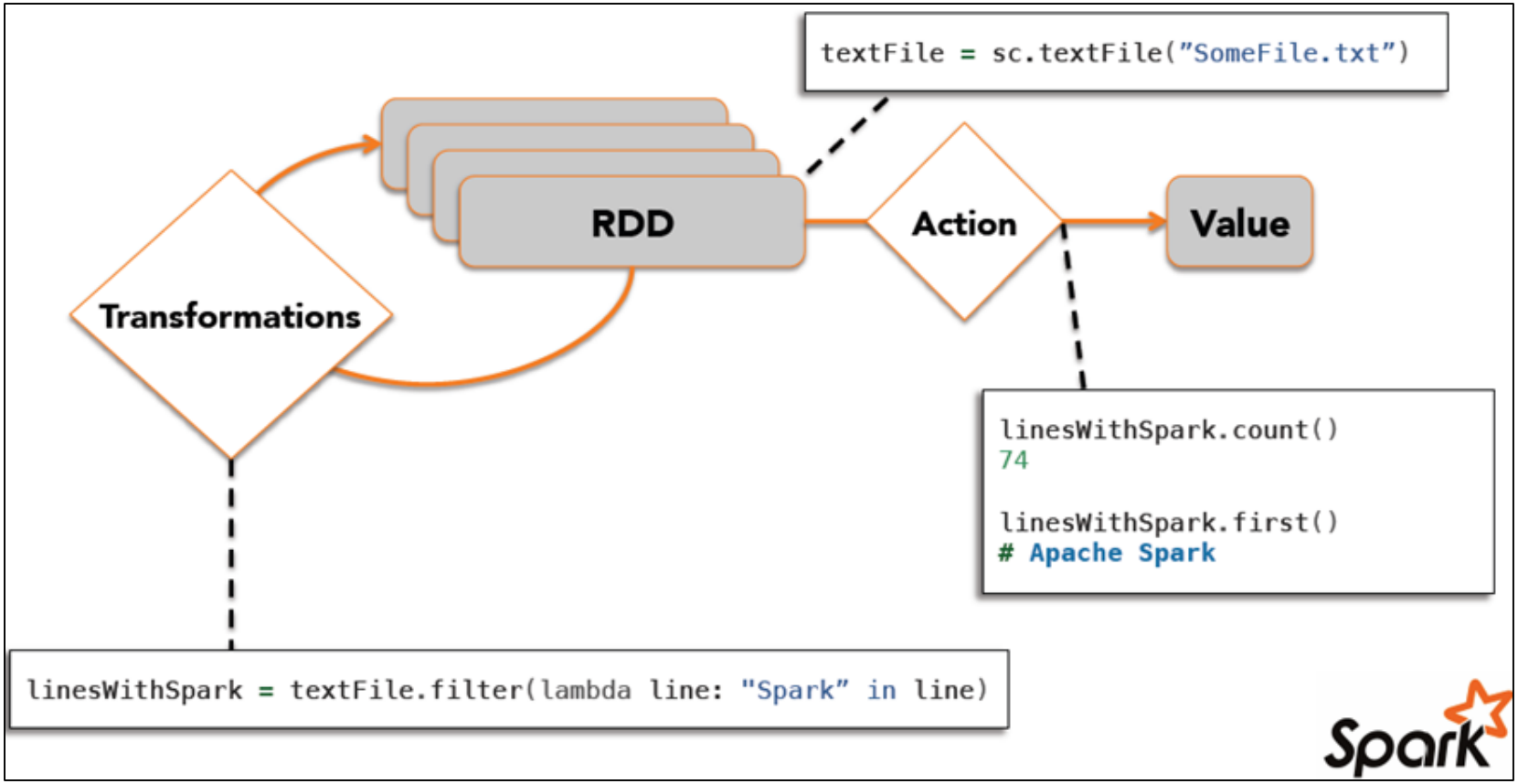


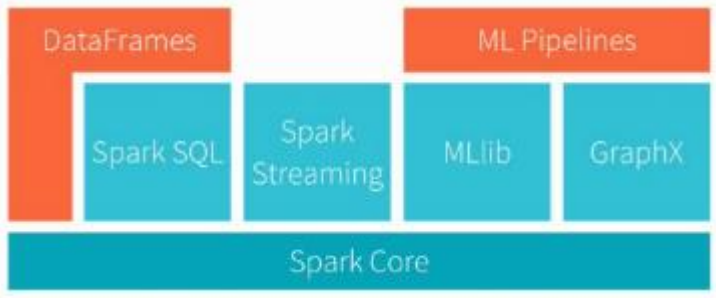
Pregel

GraphLab

Leslie Valiant (1990)

Option 3: Spark



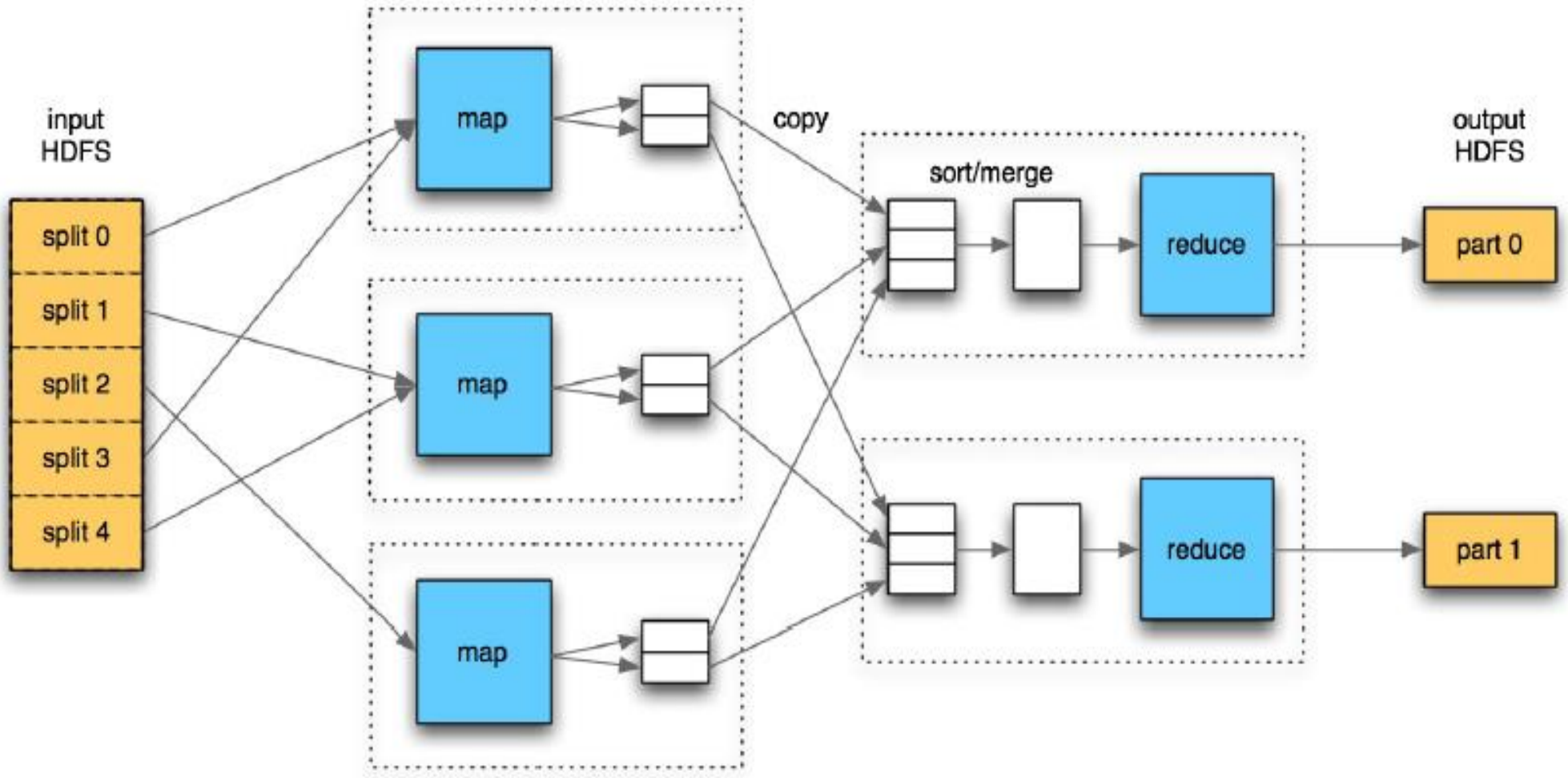


Matei Zaharia

HOW DOES MAP REDUCE WORK

Map Reduce

Map-Reduce: Solving large data problems. 2004



The Hello World of MapReduce

- **Count the number of occurrences of each word in a large amount of input data**
  - This is the 'hello world' of MapReduce programming

```
map(String input_key, String input_value)
    foreach word w in input_value:
        emit(w, 1)
```

```
reduce(String output_key,
                    Iterator<int> intermediate_vals)
    set count = 0
    foreach v in intermediate_vals:
        count += v
    emit(output_key, count)
```

Hello World - continued

- Input to the Mapper:

```
(3414, 'the cat sat on the mat')
(3437, 'the aardvark sat on the sofa')
```

- Output from the Mapper:

```
('the', 1), ('cat', 1), ('sat', 1), ('on', 1),
('the', 1), ('mat', 1), ('the', 1), ('aardvark', 1),
('sat', 1), ('on', 1), ('the', 1), ('sofa', 1)
```

- Intermediate data sent to the Reducer:

```
('aardvark', [1])
('cat', [1])
('mat', [1])
('on', [1, 1])
('sat', [1, 1])
('sofa', [1])
('the', [1, 1, 1, 1])
```

- Final Reducer output:

```
('aardvark', 1)
('cat', 1)
('mat', 1)
('on', 2)
('sat', 2)
('sofa', 1)
('the', 4)
```

Keys and Values

- Programmers specify two functions:

  map (k, v) → <k', v'>*

  reduce (k', v') → <k', v'>*

  – All values with the same key are reduced together

- **Keys and values in Hadoop are `Objects`**
- **Values are objects which implement `Writable`**
- **Keys are objects which implement `WritableComparable`**

Partition & Combine

- Both Optional:

  partition (k', number of reducers) → reducer for k'

  – Often a simple hash of the key, e.g., hash(k') mod n

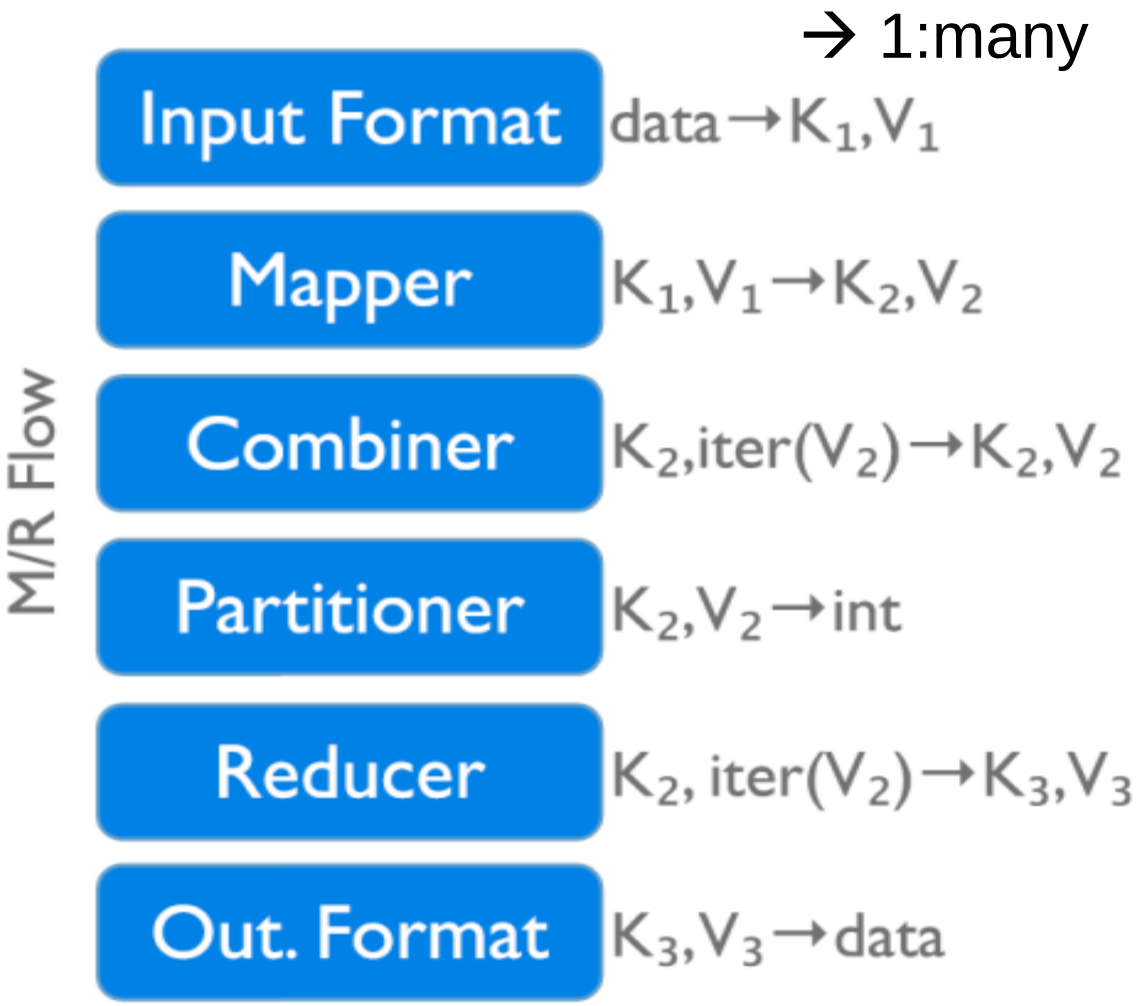
  – Divides up key space for parallel reduce operations

  combine (k', v') → <k', v'>*

  – **Mini-reducers** that run in memory after the map phase

  – Used as an optimization to reduce network traffic

Data Flow in a MapReduce Program

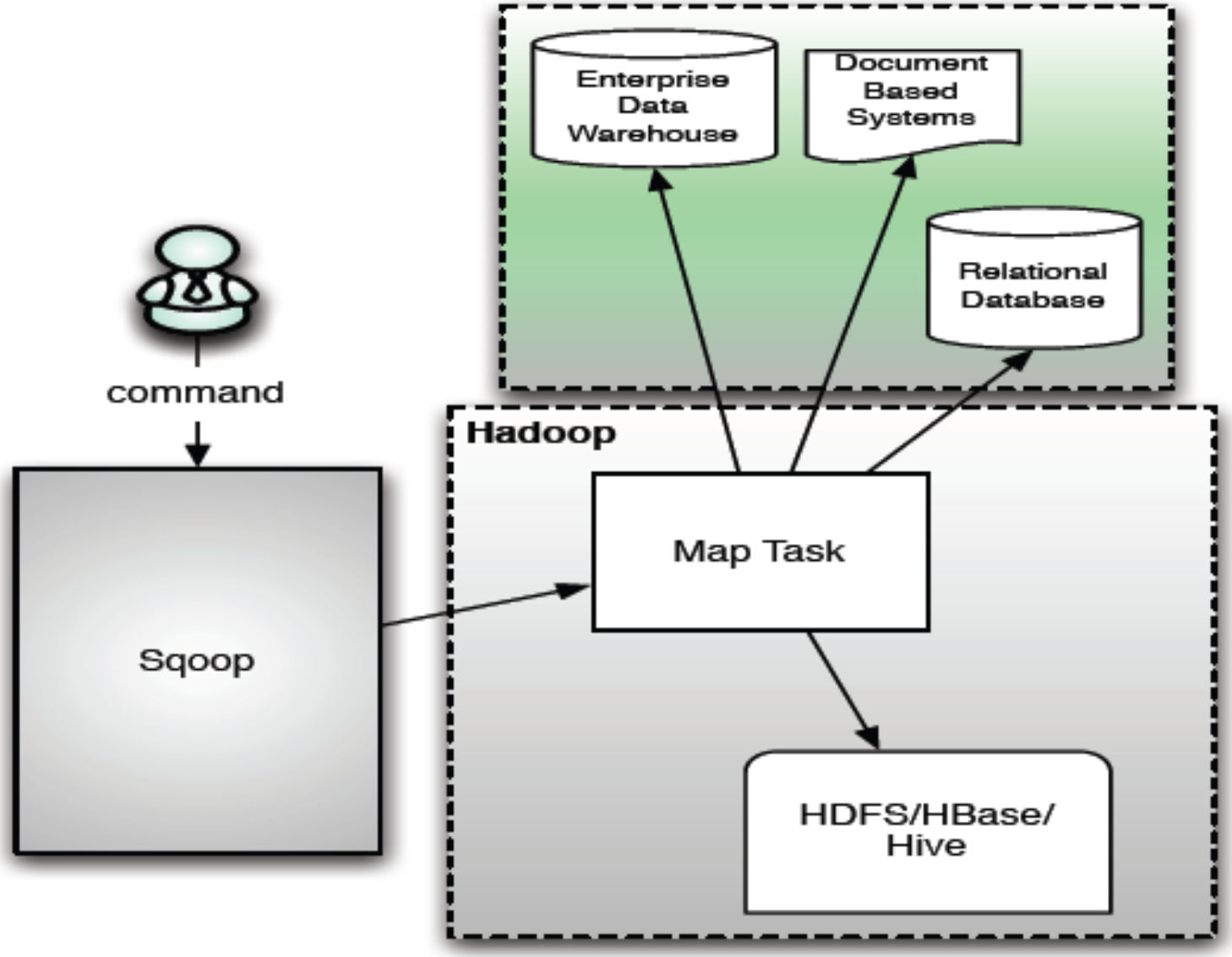


Data Ingestion

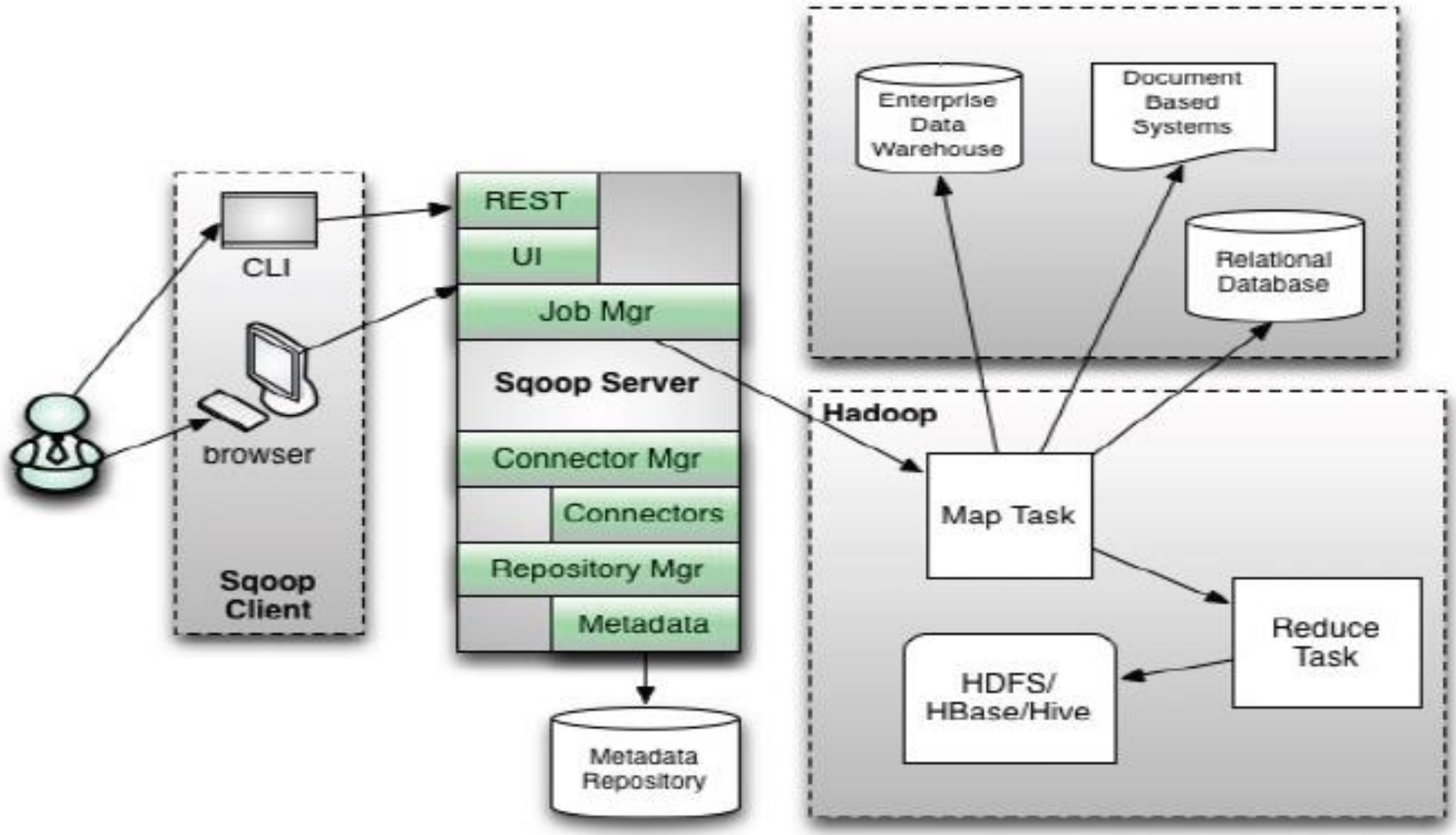
SQOOP

- RDBMS-Hadoop interoperability is key to Enterprise Hadoop adoption
- SQOOP provides a good general purpose tool for transferring data between any JDBC database and Hadoop
- SQOOP extensions can provide optimizations for specific targets

Sqoop 1 Architecture



Sqoop 2 Architecture



Sqoop 2 Themes

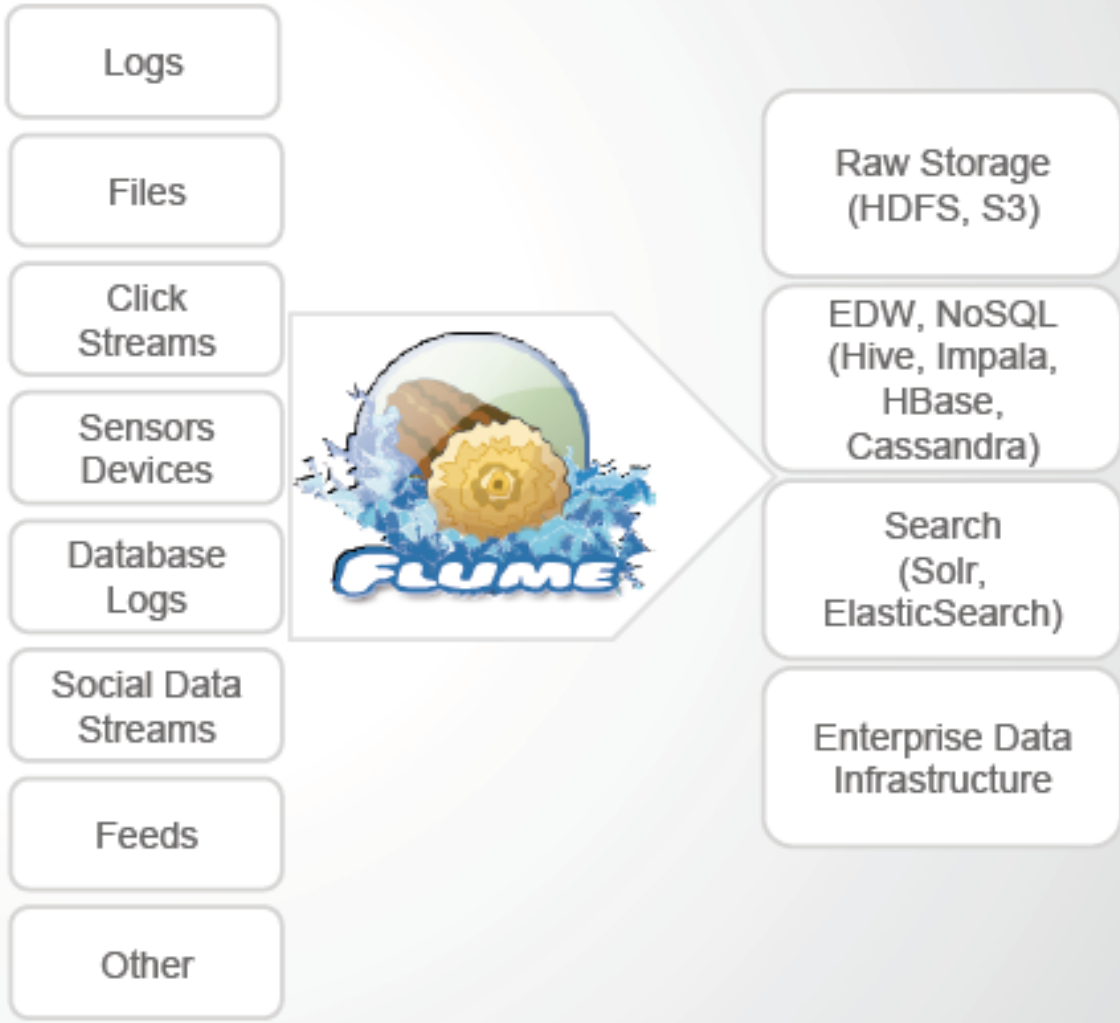
- Same goal: transfer data around
- Ease of Use
  - Sqoop as a Service
  - Domain Specific Interactions without too many args
- Ease of Extension
  - No low-level Hadoop knowledge needed
  - Uniform functionality of connectors, no functional overlap between connectors
- Security and Separation of Concerns
  - Role based access and use

Some slides from Apache Flume – Data aggregation at scale. Arvind Prabhakar, Stream Set's 2015 talk.

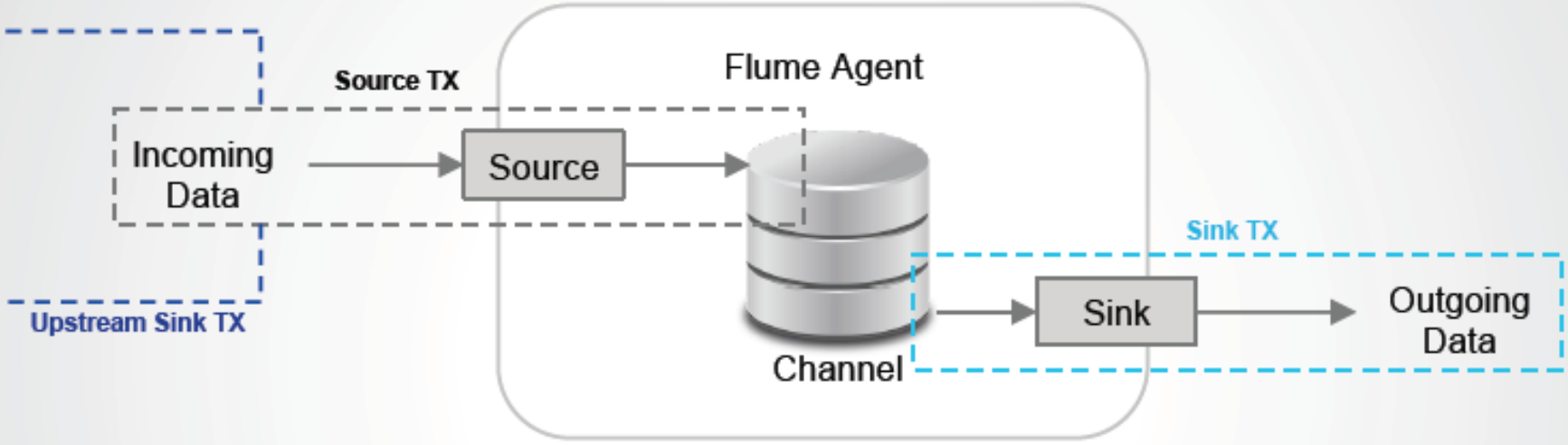
Apache Flume is a **continuous data ingestion system** that is...

- *open-source,*
- *reliable,*
- *scalable,*
- *manageable,*
- *customizable,*

...and designed for ***Big Data* ecosystem**.



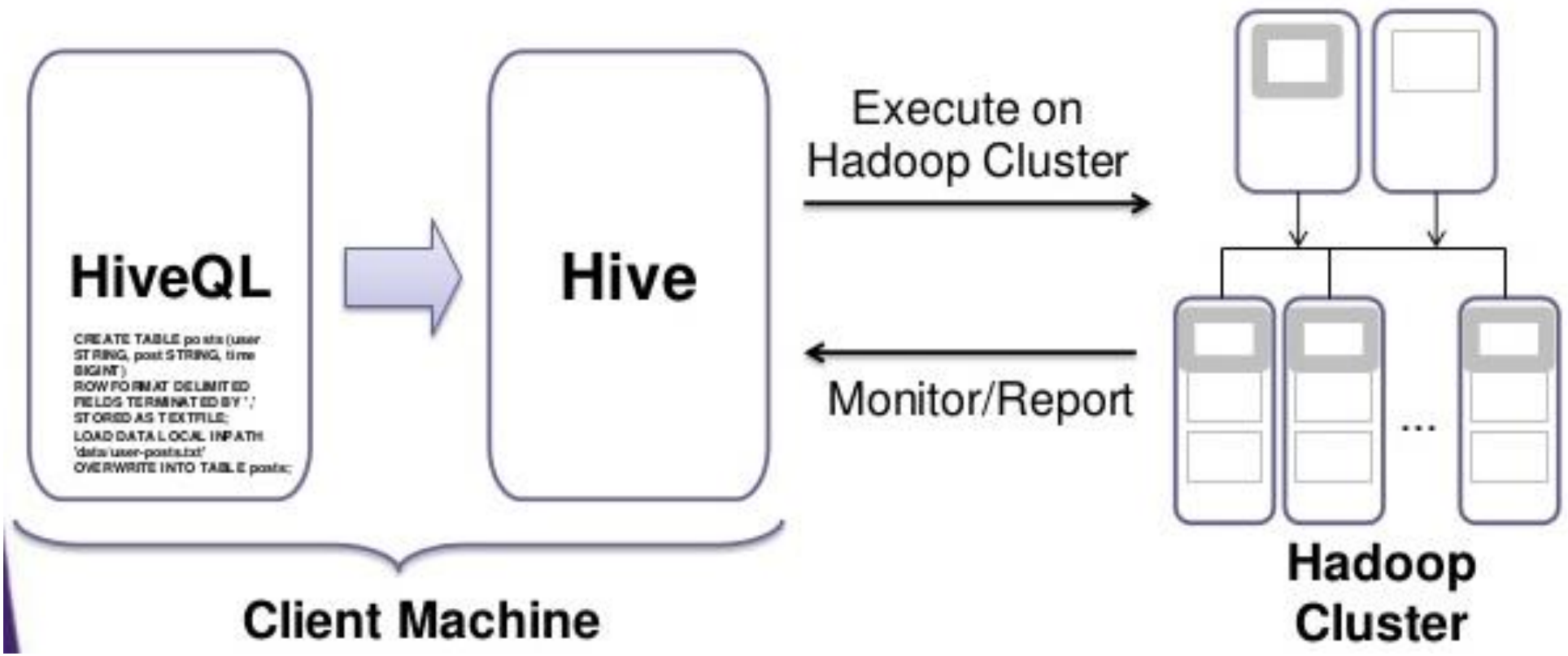
Transactional Data Exchange



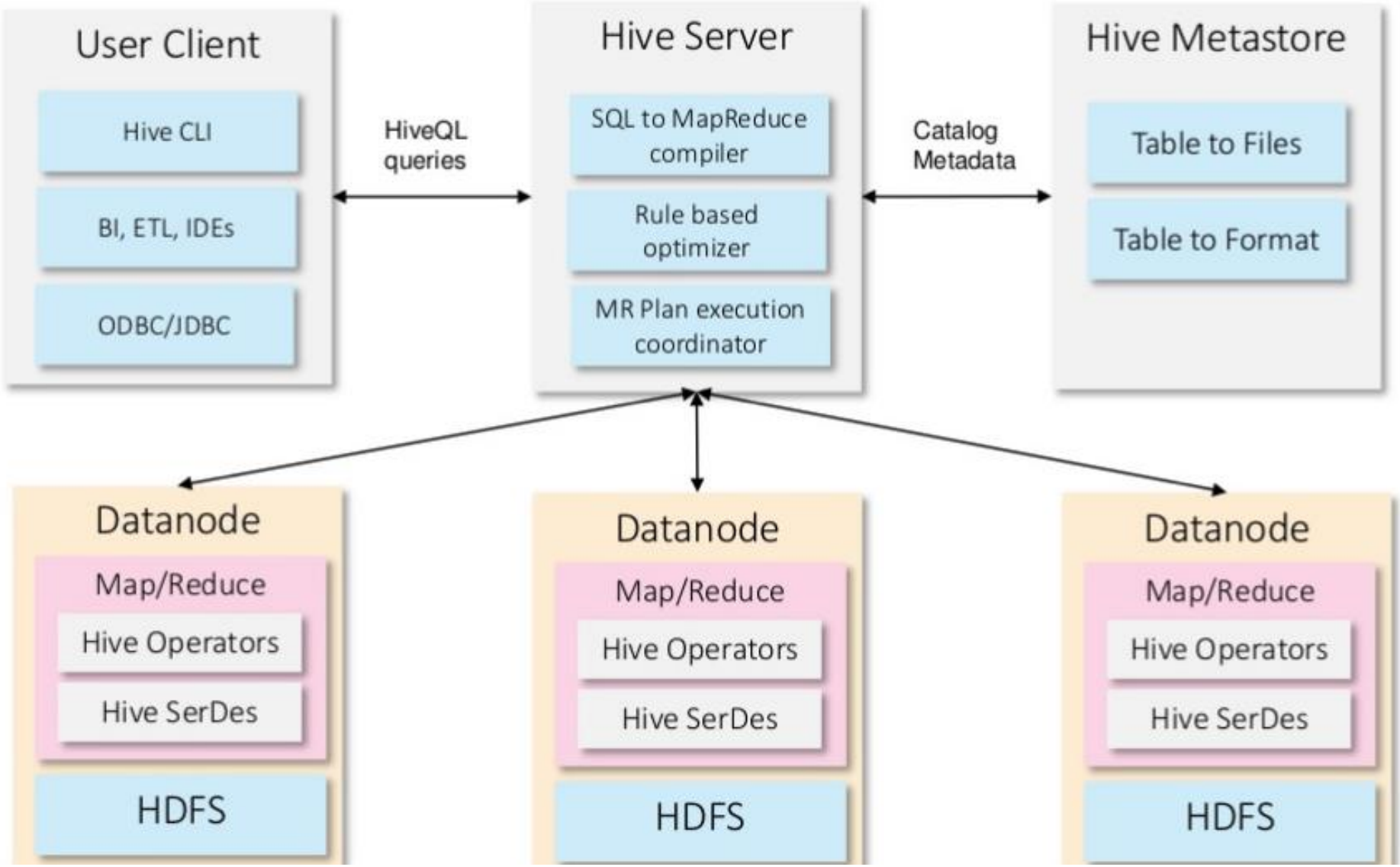
- Source uses transactions to write to the channel
- Sink uses transactions to remove data from the channel
- Sink transaction commits only after successful transfer of data
- This ensures no data loss in Flume pipeline


HIVE

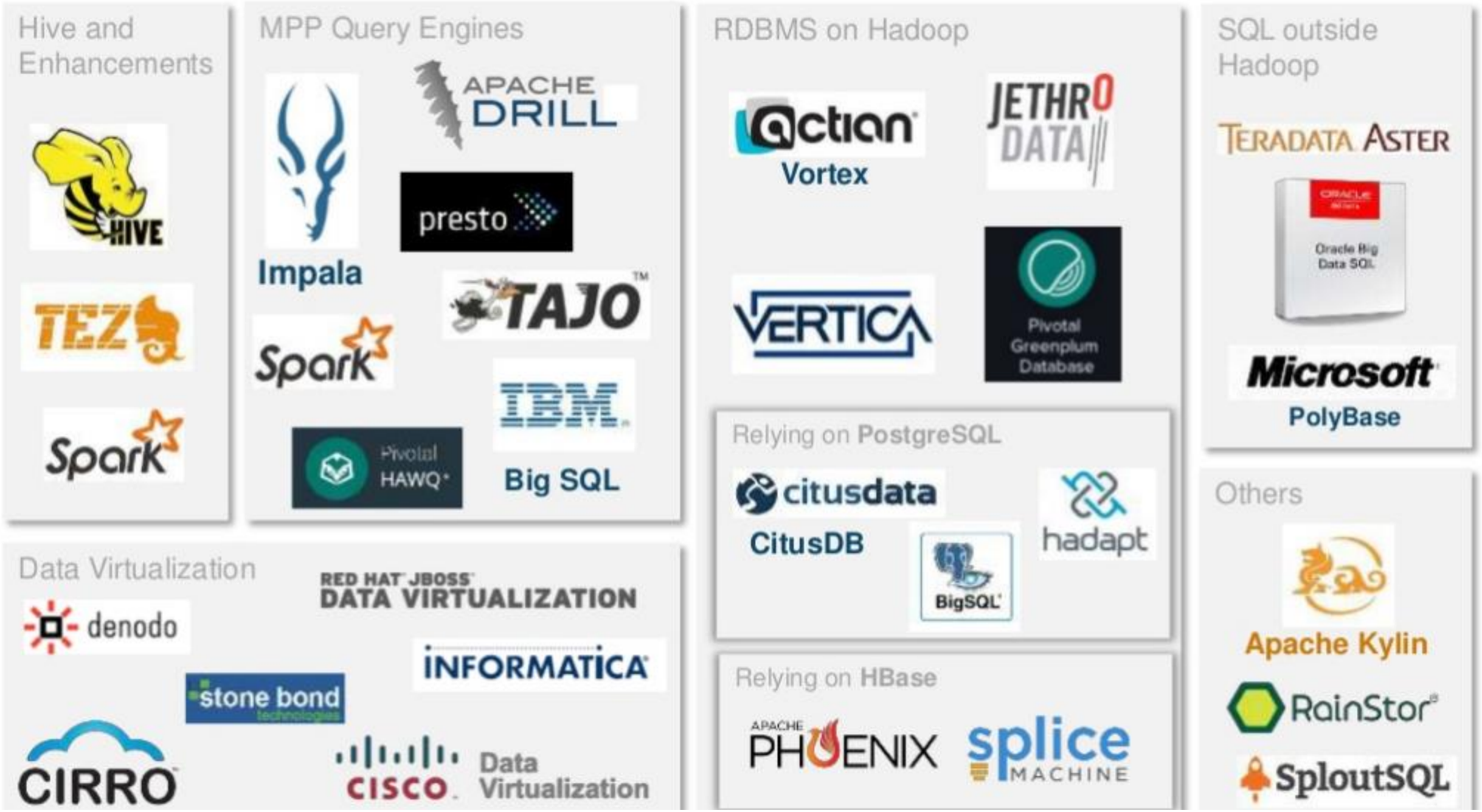
Hive Execution Model: Translate to Map-Reduce



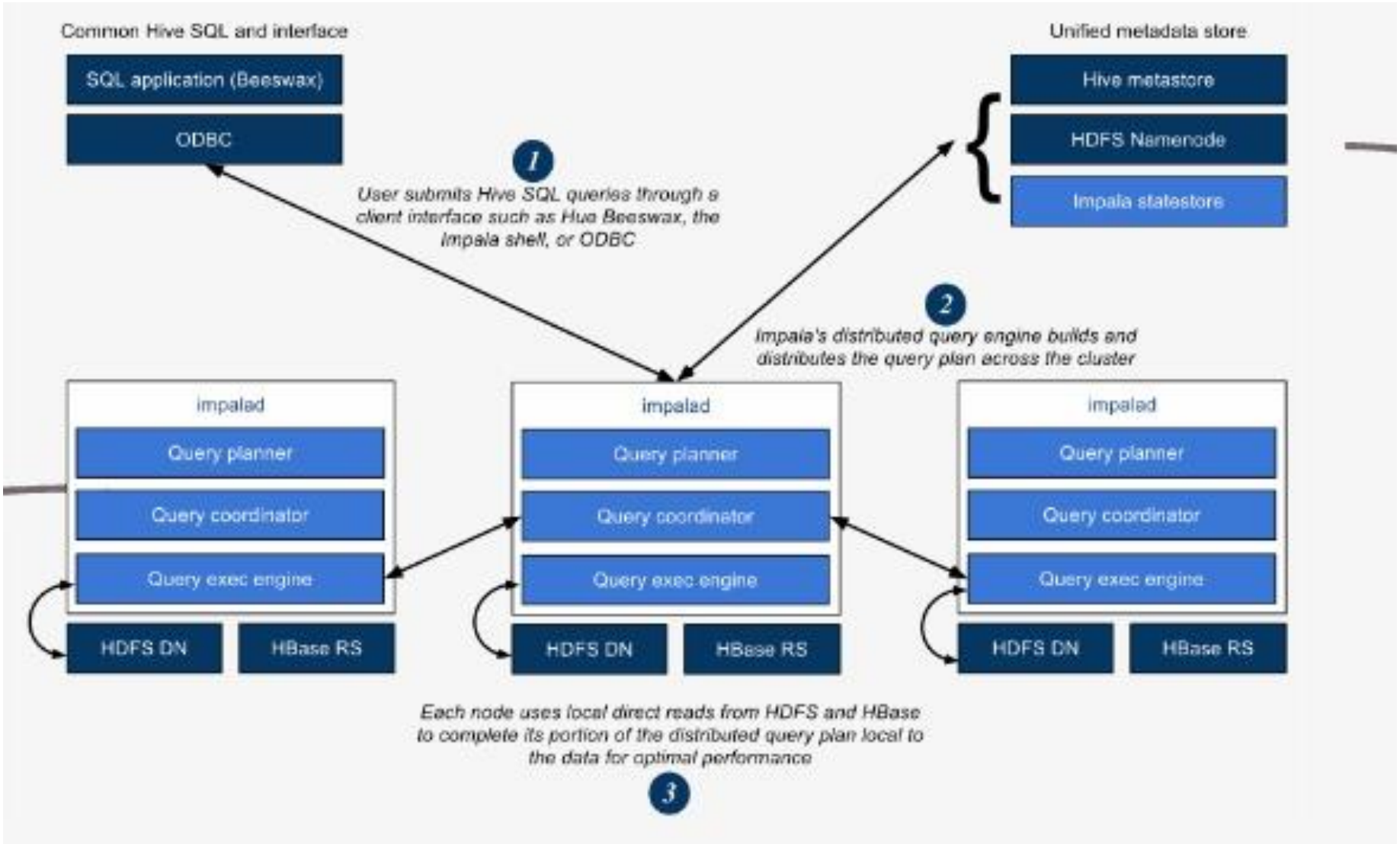
Hive Architecture



SQL on Hadoop Landscape



Cloudera Impala



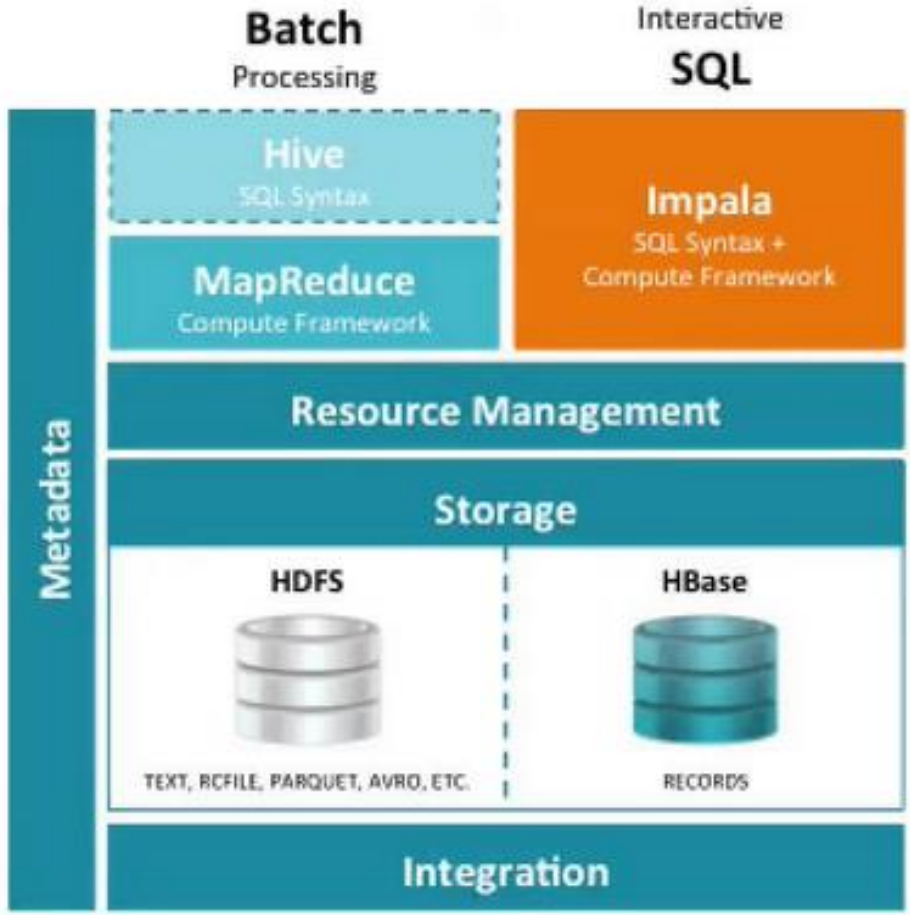
Impala and Hive

Shares Everything Client-Facing

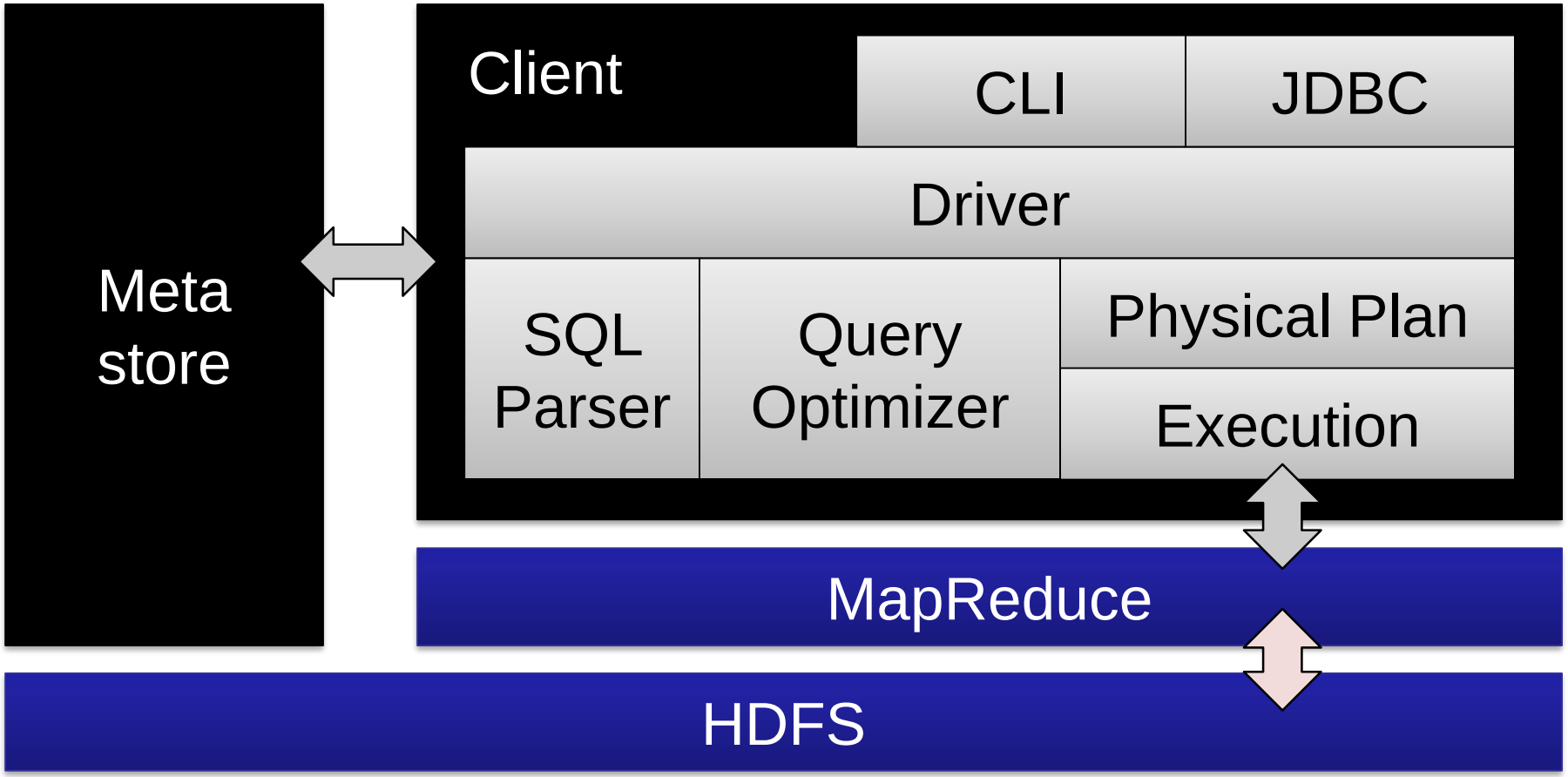
- Metadata (table definitions)
- ODBC/JDBC drivers
- SQL syntax (Hive SQL)
- Flexible file formats
- Machine pool
- Hue GUI

But Built for Different Purposes

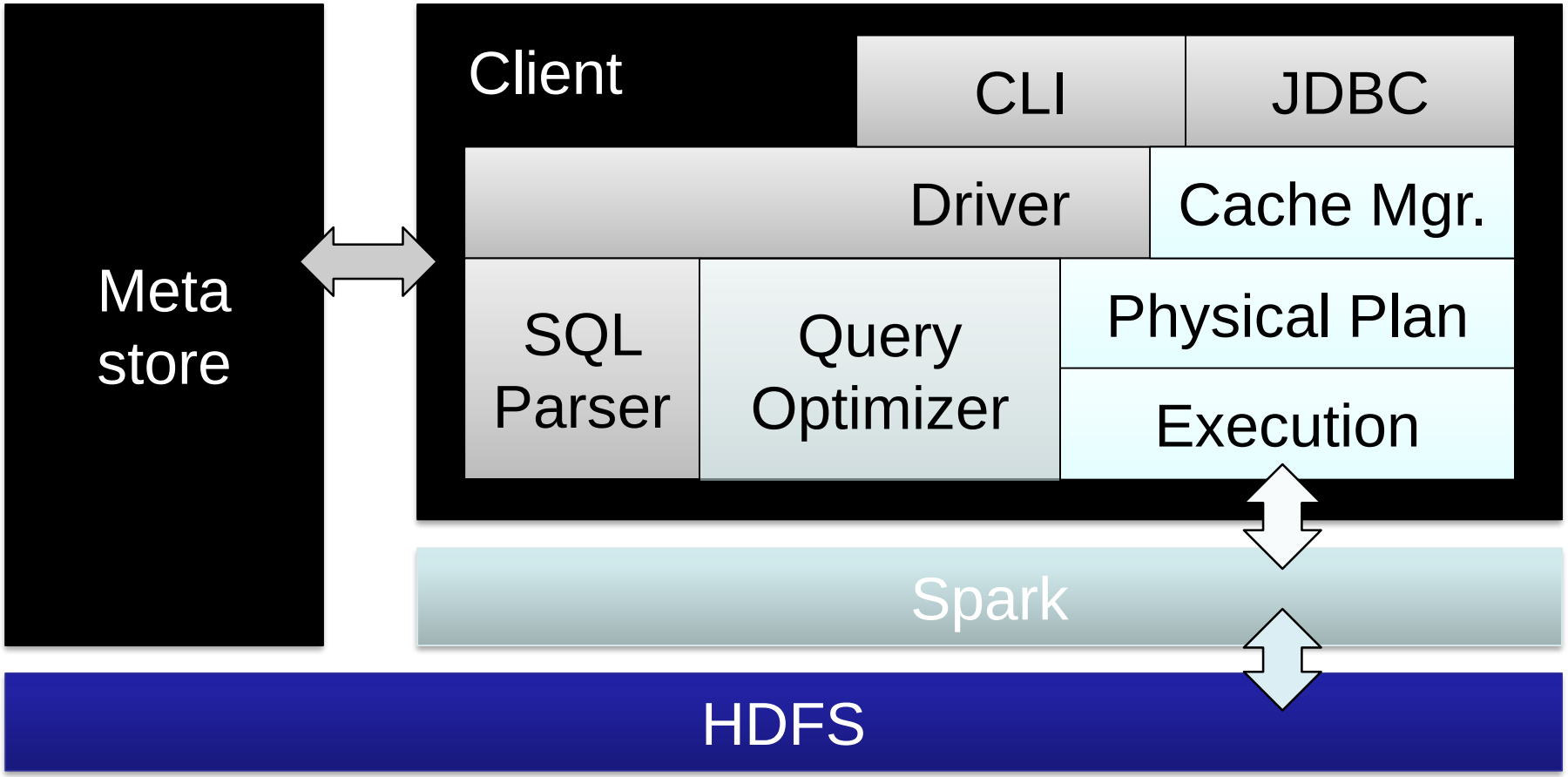
- **Hive:** runs on MapReduce and ideal for batch processing
- **Impala:** native MPP query engine ideal for interactive SQL



Hive Architecture



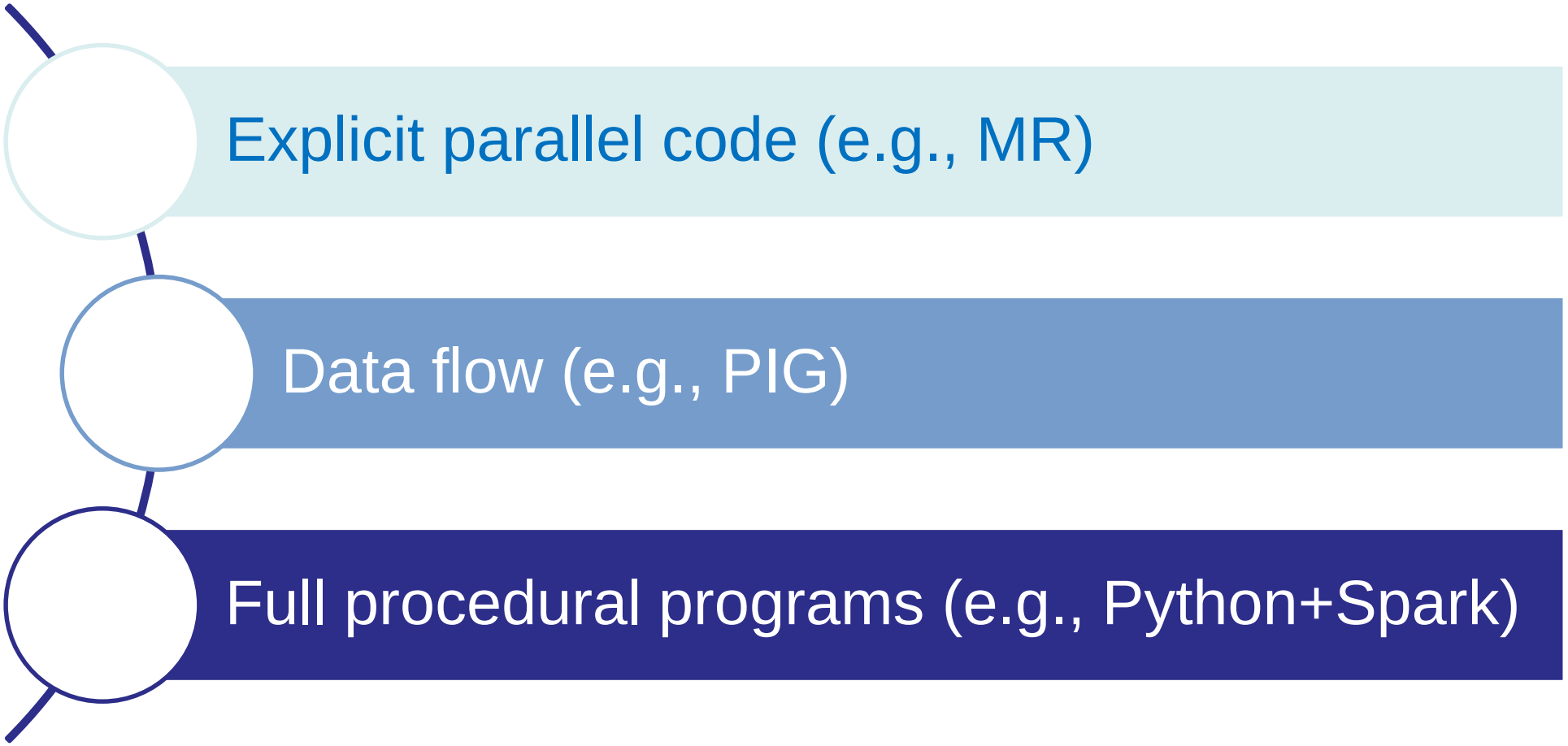
SQL on Spark Architecture



[Engle et al, SIGMOD 2012]

PROGRAMMING / SCRIPTING ON HADOOP

Evolution of Non-SQL programming on Hadoop




Y!

Example Data Analysis Task

Find the top 10 most visited pages in each category

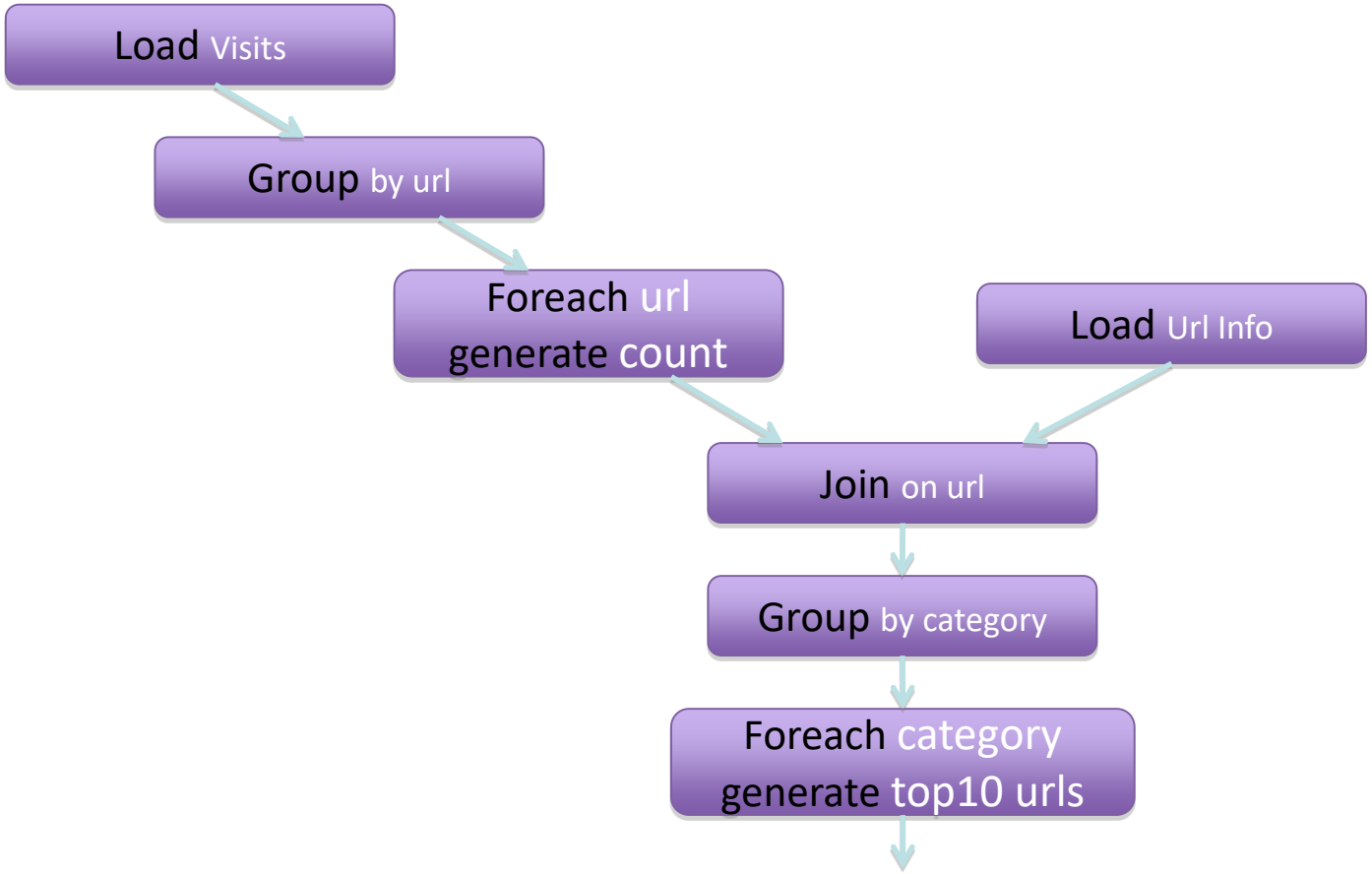
Visits

User	Url	Time
Amy	cnn.com	8:00
Amy	bbc.com	10:00
Amy	flickr.com	10:05
Fred	cnn.com	12:00

Url Info

Url	Category	PageRank
cnn.com	News	0.9
bbc.com	News	0.8
flickr.com	Photos	0.7
espn.com	Sports	0.9

Data Flow


Load Visits
Group by url
Foreach url generate count
Load Url Info
Join on url
Group by category
Foreach category generate top10 urls

In Pig Latin

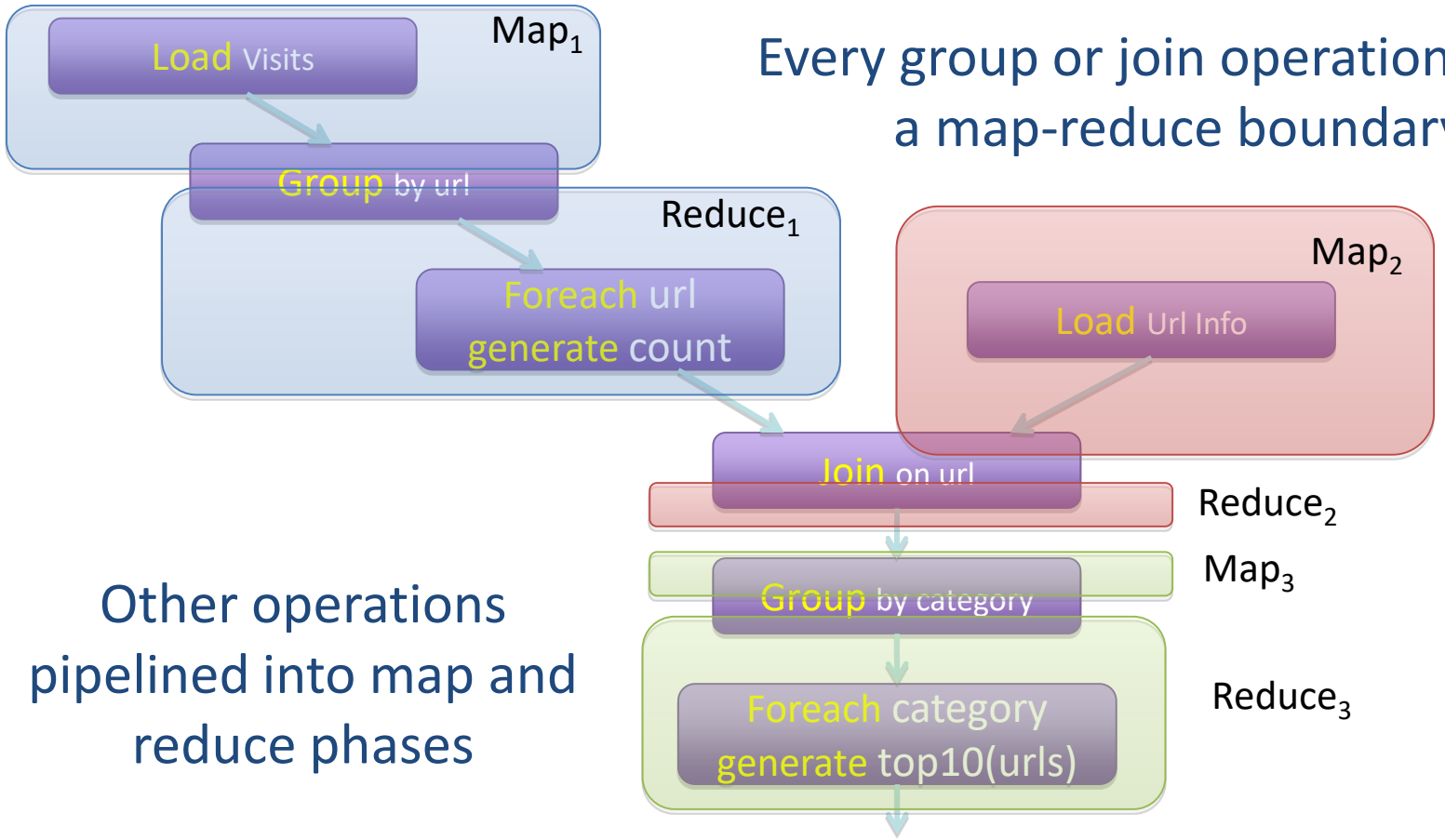
```
visits       = load '/data/visits' as (user, url, time);
gVisits      = group visits by url;
visitCounts  = foreach gVisits generate url, count(visits);

urlInfo      = load '/data/urlInfo' as (url, category, pRank);
visitCounts  = join visitCounts by url, urlInfo by url;

gCategories = group visitCounts by category;
topUrls = foreach gCategories generate top(visitCounts,10);

store topUrls into '/data/topUrls';
```

Compilation into Map-Reduce



Every group or join operation forms a map-reduce boundary

Other operations pipelined into map and reduce phases

What is Azure HD Insight

What is Azure HD Insight

- It is a Microsoft Product
- Fully managed cloud apache Hadoop service
- We can create clusters for Spark, Hive, MapReduce, Storm, Kafka etc.
- Have provisions for cluster management as well as enterprise level security, so we will get a service which is enterprise ready.

In generalized term we can say that Azure HD Insight is a service that provides us options for deploy Hadoop clusters in cloud environment , more specifically in azure cloud

High-level Overview of Hadoop Clusters in HD Insight

If we take a 30,000 feet overview of what is Hadoop , we can say “open source frame work which allows distributed storage of frame work and programming models for parallel processing of the distributed data”.

Azure HD Insight is giving us provision to leverage this powerful technology in an easy and efficient manner.

Below are few of the cluster type that we can build in azure HD Insight

Apache Hadoop	Uses HDFS, YARN and simple MapReduce programming model
Apache Spark	Leverages Spark Parallel and in memory computing facilities
Apache HBase	NoSQL database built on top of Hadoop that provides massive storage and efficient processing
Apache Kafka	Open source platform for handling streaming application
Apache Storm	Distributed system for processing large stream of data quickly.

Source https://docs.microsoft.com/en-us/azure/hdinsight/hdinsight-hadoop-introduction#overview

Different Utilities in HD Insight Clusters

Ambari	Management and Monitoring utilities
Hive and Hcatalog	Table and storage management layer, sql like functionalities
Map Reduce	Distributed processing frame work
Oozie	Work flow management
Phoneix	Relational database flavour on top of Hbase
Pig	Scripting language for easy implementation of map reduce transformations
YARN	Hadoop facility for resource management
Zookeeper	Process co-ordination in distributed system

Source https://docs.microsoft.com/en-us/azure/hdinsight/hdinsight-hadoop-introduction#overview

Storage facilities for Azure HD Insight

We are starting with this concept because to start with HD Insight Cluster we first need to place our data to perform different kind op operations in it.

We can store in our data in two different ways either we can use Azure Storage or we can use HDFS for storing our data. Point to remember that when ever we are storing our data in HDFS it will be lost as soon as we delete our cluster.

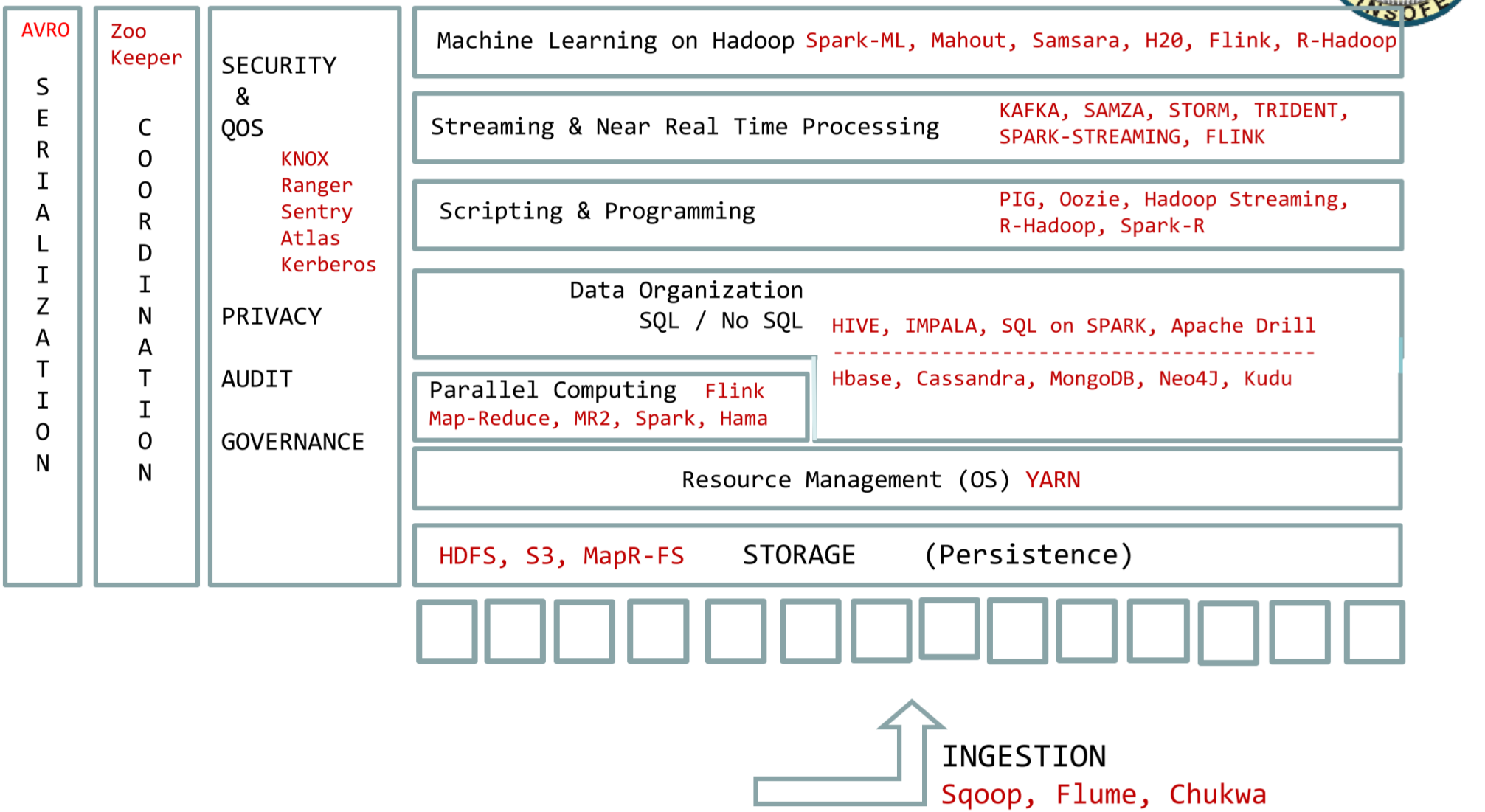
If we store our data in Azure storage it will be available for us after we delete the cluster.

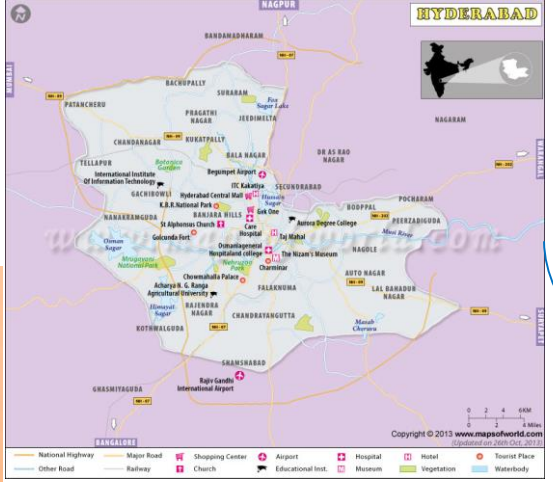
For our training purpose we will focus about azure storage.

What is Azure Storage

- It is massively scalable storage so we can store hundreds of tera bytes of data for big data analysis purpose.
- You pay only the storage you used , so if you store less data you pay less.
- It is elastic in nature so we do not have to concerned about sudden growth of data
- It is an auto-partitioning system which automatically handles the load balance based upon traffic. As demands grows more resources will be allocated.
- It is accessible from any kind of platform, whether your application is running on cloud or stand alone desktop or on premise server.
- It supports windows as well as Linux operating system
- It supports multiple programming languages

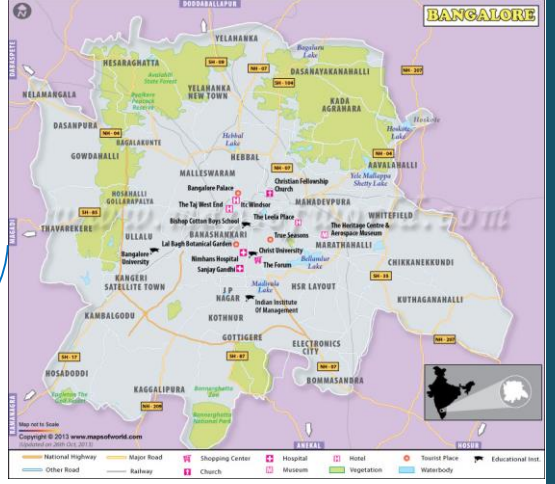
Summary


AVRO
SERIALIZATION
Zoo Keeper
COORDINATION
SECURITY & QOS
KNOX
Ranger
Sentry
Atlas
Kerberos
PRIVACY
AUDIT
GOVERNANCE
Machine Learning on Hadoop Spark-ML, Mahout, Samsara, H20, Flink, R-Hadoop
Streaming & Near Real Time Processing
KAFKA, SAMZA, STORM, TRIDENT, SPARK-STREAMING, FLINK
Scripting & Programming
PIG, Oozie, Hadoop Streaming, R-Hadoop, Spark-R
Data Organization
SQL / No SQL
HIVE, IMPALA, SQL on SPARK, Apache Drill
Hbase, Cassandra, MongoDB, Neo4J, Kudu
Parallel Computing Flink
Map-Reduce, MR2, Spark, Hama
Resource Management (OS) YARN
HDFS, S3, MapR-FS STORAGE (Persistence)
INGESTION
Sqoop, Flume, Chukwa



INSOFE

Inspire...Educate...Transform.



HYDERABAD

Office and Classrooms

Plot 63/A, Floors 1&2, Road # 13, Film Nagar,
Jubilee Hills, Hyderabad - 500 033
+91-9701685511 (Individuals)
+91-9618483483 (Corporates)

BENGALURU

Office

Incubex, #728, Grace Platina, 4th Floor, CMH Road,
Indira Nagar, 1st Stage, Bengaluru – 560038
+91-9502334561 (Individuals)
+91-9502799088 (Corporates)

Classroom

KnowledgeHut Solutions Pvt. Ltd., Reliable Plaza,
Jakkasandra Main Road, Teacher's Colony, 14th Main
Road, Sector – 5, HSR Layout, Bengaluru - 560102

Social Media

Web:	http://www.insofe.edu.in
Facebook:	https://www.facebook.com/insofe
Twitter:	https://twitter.com/Insofeedu
YouTube:	http://www.youtube.com/InsofeVideos
SlideShare:	http://www.slideshare.net/INSOFE
LinkedIn:	http://www.linkedin.com/company/international-school-of-engineering

This presentation may contain references to findings of various reports available in the public domain. INSOFE makes no representation as to their accuracy or that the organization subscribes to those findings.