

# Understanding Hadoop

TDWI Chicago 2015 World  
Conference

May 4<sup>th</sup> 2015

# Big Data Concepts

## Introduction to Big Data Platforms

Big Data Introduction

The Computation Requirements

The Beginnings

## Companies Introducing Computation Platforms

Google - MapReduce, GFS & HummingBird

Nutch - The Origins of Hadoop

Yahoo - The Hadoop Platform

Facebook - Cassandra & NoSQL

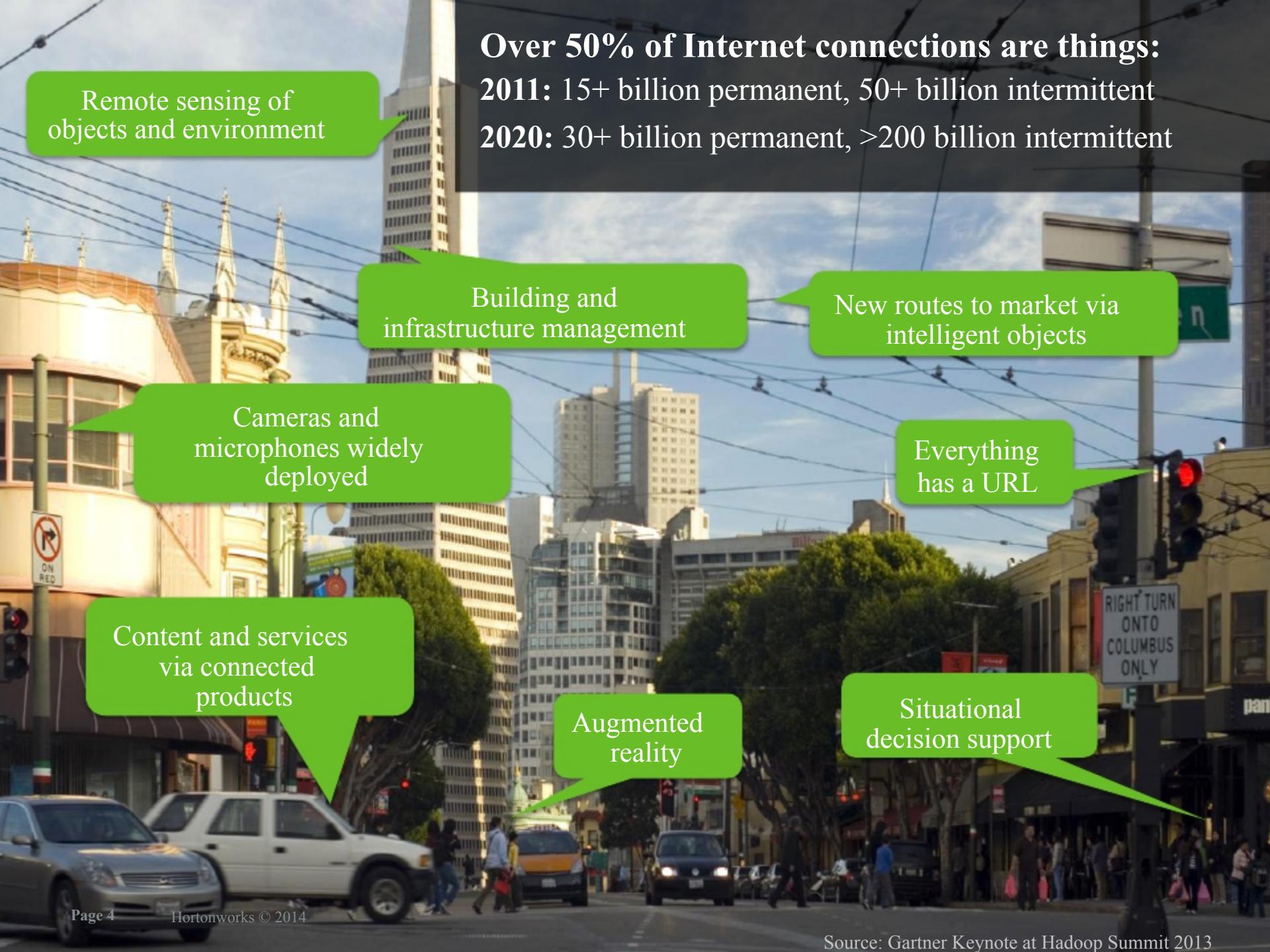
LinkedIn – Kafka Platform

## Hadoop Launch in Apache

Cloudera

MapR

Hortonworks



Remote sensing of objects and environment

## Over 50% of Internet connections are things:

2011: 15+ billion permanent, 50+ billion intermittent

2020: 30+ billion permanent, >200 billion intermittent

Building and infrastructure management

New routes to market via intelligent objects

Cameras and microphones widely deployed

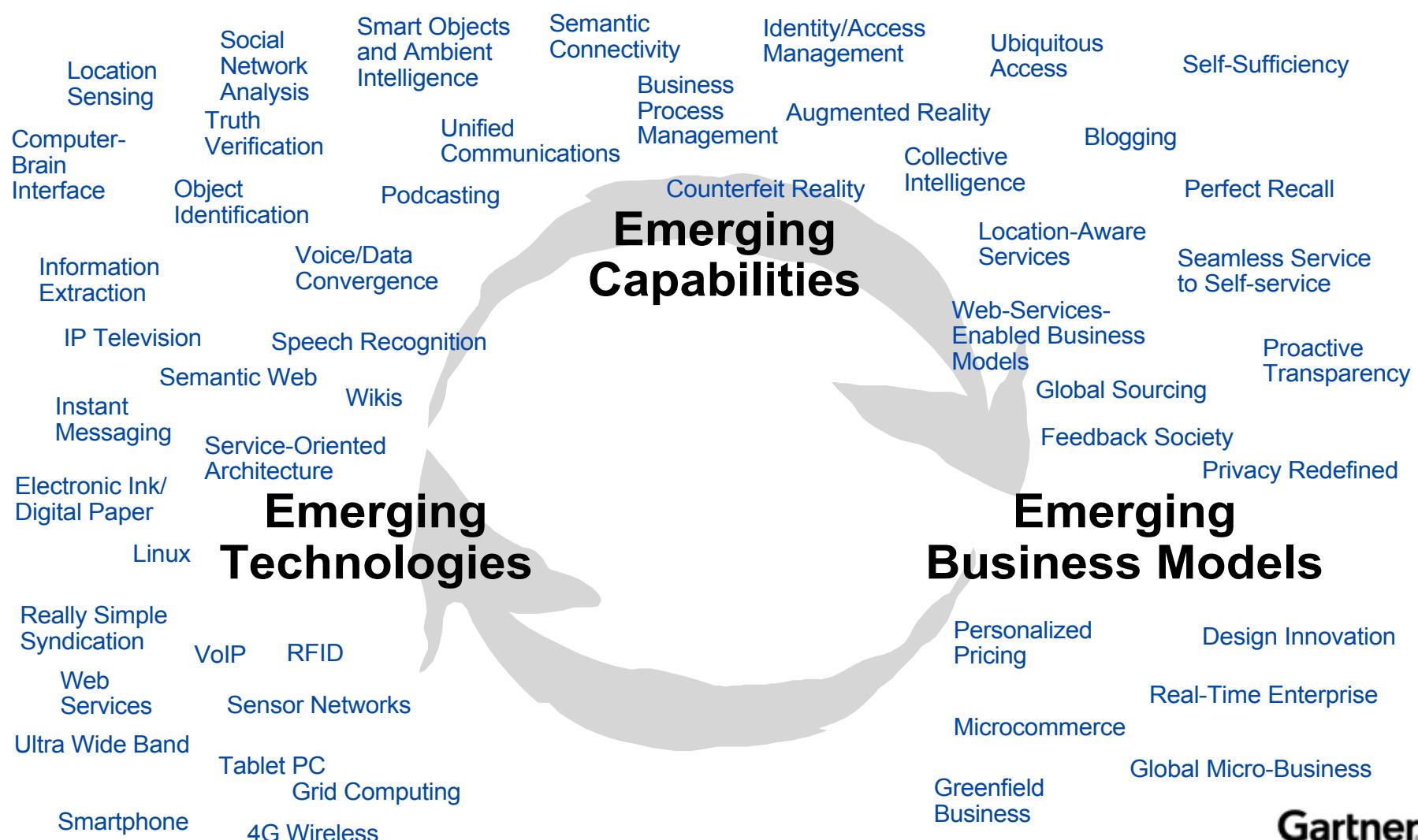
Everything has a URL

Content and services via connected products

Augmented reality

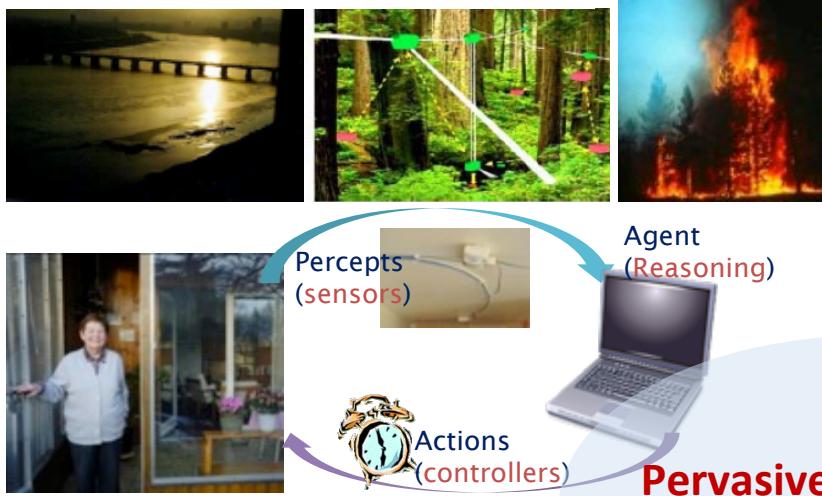
Situational decision support

# Disruption: It's Not Just About Technologies



# Smart Sensing, Reasoning and Decision-making

## Environment Sensing



## Emergency Response



Situation Awareness:  
Humans as  
sensors  
feed multi-  
modal data  
streams



Credit: Photo by US Geological Survey

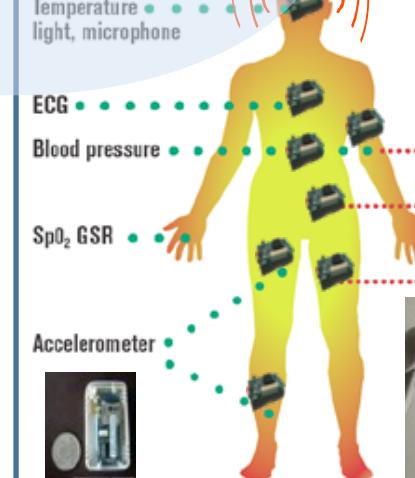
## Computing

## People-Centric Sensing



## Social

## Informatics



## Smart Health Care



# **THE BEGINNINGS**

# Netscape

The screenshot shows the Netscape Navigator Version 2.02 interface. The title bar reads "Netscape - [Version 2.02]". The menu bar includes File, Edit, View, Go, Bookmarks, Options, Directory, Window, and Help. Below the menu is a toolbar with Back, Forward, Home, Reload, Images, Open, Print, Find, and Stop buttons. The location bar shows "Location: about:". Below it is a navigation bar with links to What's New!, What's Cool, Handbook, Net Search, Net Directory, and Software. The main content area displays the Netscape Navigator logo (a stylized 'N' on a hill) and the text "Netscape Navigator™ Version 2.02 Copyright © 1994-1995 Netscape Communications Corporation, All rights reserved." It also includes links for license and feedback. The page footer contains Java and RSA security notices, and a disclaimer about U.S. Government restrictions.

**Netscape Navigator™**  
Version 2.02  
Copyright © 1994-1995 Netscape Communications Corporation, All rights reserved.  
This software is subject to the license agreement set forth in the [license](#). Please read and agree to all terms before using this software.  
Report any problems through the [feedback page](#).

Netscape Communications, Netscape, Netscape Navigator and the Netscape Communications logo are trademarks of Netscape Communications Corporation.

---

 **JAVA COMPATIBLE**  
Contains Java™ software developed by Sun Microsystems, Inc.  
Copyright © 1992-1995 Sun Microsystems, Inc. All Rights Reserved.

---

 **RSA**  
PUBLIC KEY CRYPTOSYSTEM  
Contains security software from RSA Data Security, Inc.  
Copyright © 1994 RSA Data Security, Inc. All rights reserved.  
This version supports International security with RSA Public Key Cryptography, MD2, MD5, RC4.

---

Any provision of Netscape Software to the U.S. Government is with "Restricted rights" as follows: Use, duplication or disclosure by the Government is subject to restrictions set forth in subparagraphs (a) through (d) of the Commercial Computer Restricted Rights clause at FAR 52.227-19 when applicable, or in subparagraph (c) (1) (ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013, and in similar clauses in the NASA FAR Supplement. Contractor/manufacturer is Netscape Communications Corporation, 501 East Middlefield Road, Mountain View, California, 94041.

# Search



Google Search

I'm Feeling Lucky

From the streets to the web. [Street art from around the world.](#)

# Yahoo

Make Yahoo! your home page | Get Yahoo! Toolbar with Anti-Spy

Yahoo! is testing a new home page. Send feedback.

## YAHOO!

Search:  Web Search

Y! Answers: How many calories do you have to lose to lose one pound?

**My Yahoo!** **My Mail**

**Autos**  
**Entertainment**  
**Finance**  
**Games**  
**GeoCities**  
**Groups**  
**Horoscopes**  
**HotJobs**  
**Kids**  
**Maps**  
**Personals**  
**Photos**  
**Real Estate**  
**Shopping**  
**Sports**  
**Travel**  
**Yellow Pages**  
**All Yahoo! Services**

**Featured** **Entertainment** **Sports** **Money**

**Count to 10**  
Panicking about how to pay off your tax bill? Don't do this. [more](#)  
[Video: Valuable deductions](#)  
[How to avoid an audit](#)  
[Y! Finance: Tax tools and checklists](#)

**Upper Tax tips: The 10 worst ways to pay off your bill**  
[Who is making the millions?](#)

**Training kids to be wise with their money**  
[C'mon, get happy and get rich](#)

**In the News** **World** **Video**  
10:35am, Sun Mar 26  

- Bombing in southeast Iraq kills student
- Nigeria OKs handover of Liberian warlord
- Immigration march draws 500,000 in L.A.
- Gunman kills six, self in Seattle home
- Indians fear losing native-friendly clinics
- Cell phone use up in Iraq, but network shaky
- PBS documentary features Dana Reeve intro

[» More News](#)

**Markets:** Dow: +0.1% Nasdaq: +0.6%

**Marketplace**  
 Earn your degree online Associate, Bachelor's, Master's and Certificate. Top colleges and universities Start today.  
The average US credit score: 678. The cost to see yours: \$0.  
Today's home equity offer - \$30,000 loan for under \$150/month - Start now.

Check your mail: [Sign In](#)  
Free mail: [Sign Up](#) Set location

**Mail** **Messenger** **Music**  
**Weather** **Local** **Movies**

**Yahoo! Music Unlimited**  
OVER A MILLION SONGS.  
5 BUCKS A MONTH.  
[» Try it free](#)  
Download their latest: Madonna, Chris Brown, and More

**DELL**  
**\$499 DESKTOP**  
AFTER INSTANT SAVINGS FROM DELL HOME  
**\$299 NOTEBOOK**  
AFTER MAIL-IN REBATE FROM DELL HOME  
[Dell Home Deals](#) • [Ad Feedback](#)

**Yahoo! Pulse: Top Cities in Europe**

  
1. Paris  
2. London  
3. Rome  
4. Amsterdam  
5. Barcelona  
6. Florence  
7. Dublin  
8. Venice  
9. Edinburgh  
10. Istanbul  
[» More Cities](#)

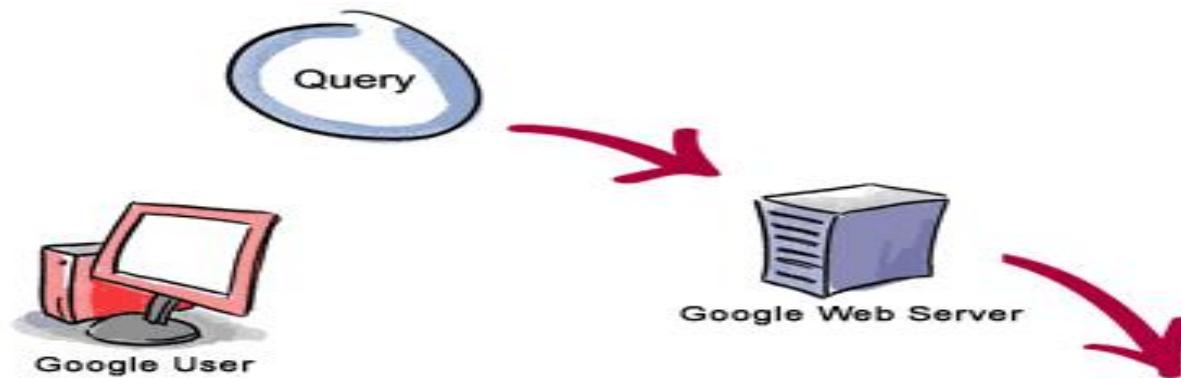
# Nutch

**Highly extensible, highly scalable Web crawler**

Nutch is a well matured, production ready Web crawler. Nutch 1.x enables fine grained configuration, relying on Apache Hadoop™ data structures, which are great for batch processing.

[Download](#)

# Search Process



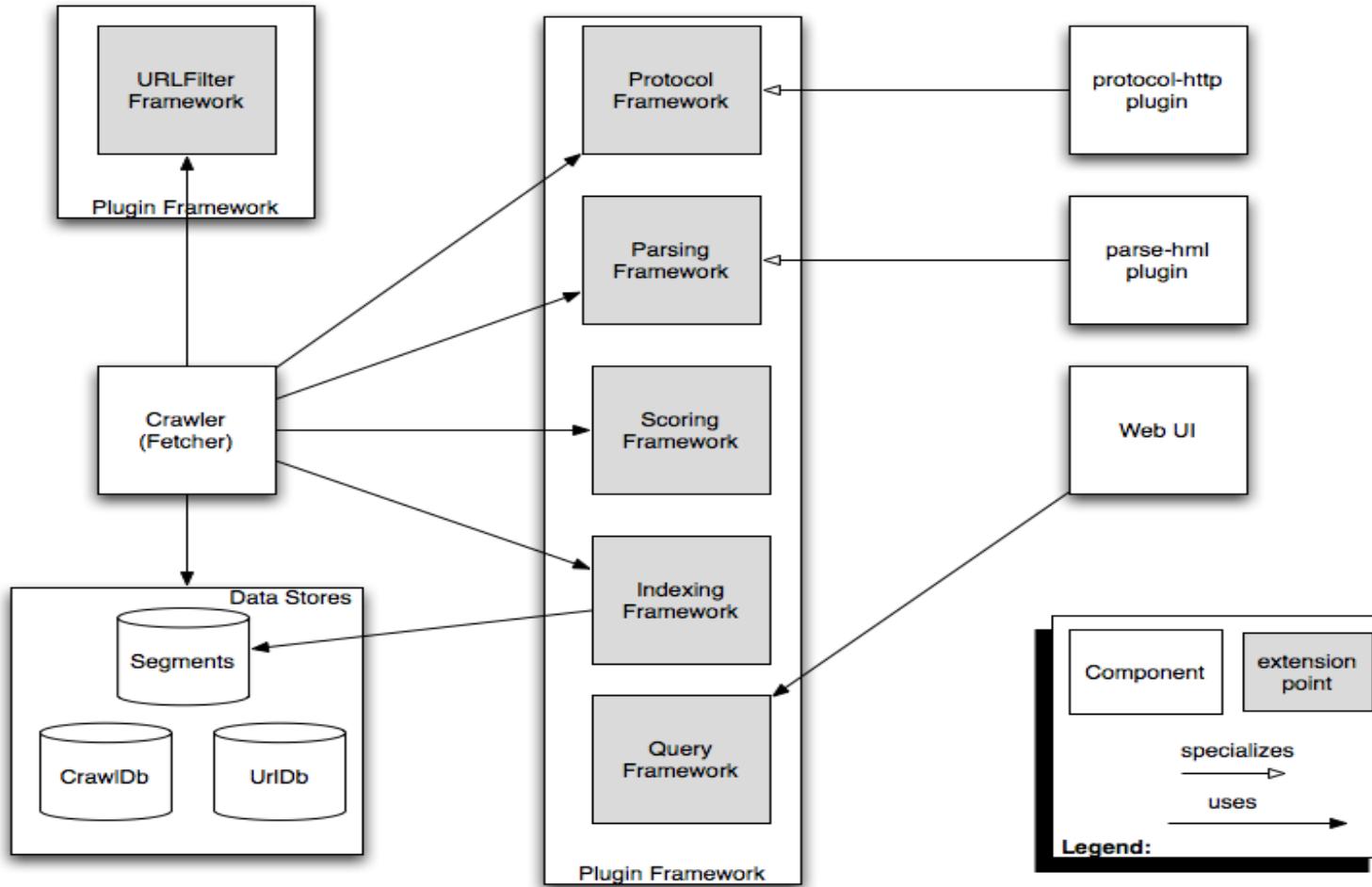
3. The search results are returned to the user in a fraction of a second.

1. The web server sends the query to the index servers. The content inside the index servers is similar to the index in the back of a book - it tells which pages contain the words that match the query.

2. The query travels to the doc servers, which actually retrieve the stored documents. Snippets are generated to describe each search result.



# Nutch Architecture



# Facebook

A photograph of a diverse group of nine people gathered around a long wooden table for a meal. They are eating, talking, and laughing. The setting is an outdoor patio with a yellow wall in the background decorated with colorful balloons and a red sunburst mural. A white tablecloth covers the table, which is set with various dishes, glasses, and a large pitcher of juice. In the bottom left corner, there is a large, semi-transparent blue square containing the white Facebook 'f' logo. To the right of the logo, the word "Facebook" is written in white, followed by a blue checkmark icon and the text "Product/Service". Below this, there is a navigation bar with five items: "Timeline", "About", "Photos", "Facebook Stories" (which is underlined in blue), and "More ▾".

Facebook   
Product/Service

Timeline    About    Photos    Facebook Stories    More ▾

# LinkedIn

**LinkedIn**

Email address  Password [Forgot your password?](#)

**Join the world's largest professional network.**



**Get started – it's free.**

Registration takes less than 2 minutes.

First name

Last name

Email address

Password (6 or more characters)

By clicking Join Now, you agree to LinkedIn's User Agreement, Privacy Policy and Cookie Policy.

Find a colleague:

First name

Last name

LinkedIn member directory: [a](#) [b](#) [c](#) [d](#) [e](#) [f](#) [g](#) [h](#) [i](#) [j](#) [k](#) [l](#) [m](#) [n](#) [o](#) [p](#) [q](#) [r](#) [s](#) [t](#) [u](#) [v](#) [w](#) [x](#) [y](#) [z](#) [more](#) | [Browse members by country](#)

# Cloudera

Downloads   Developers   Community

Sign In   Register   Contact Us   Support

Contact Sales: 866-843-7207

Search



cloudera

PRODUCTS & SERVICES

TRAINING

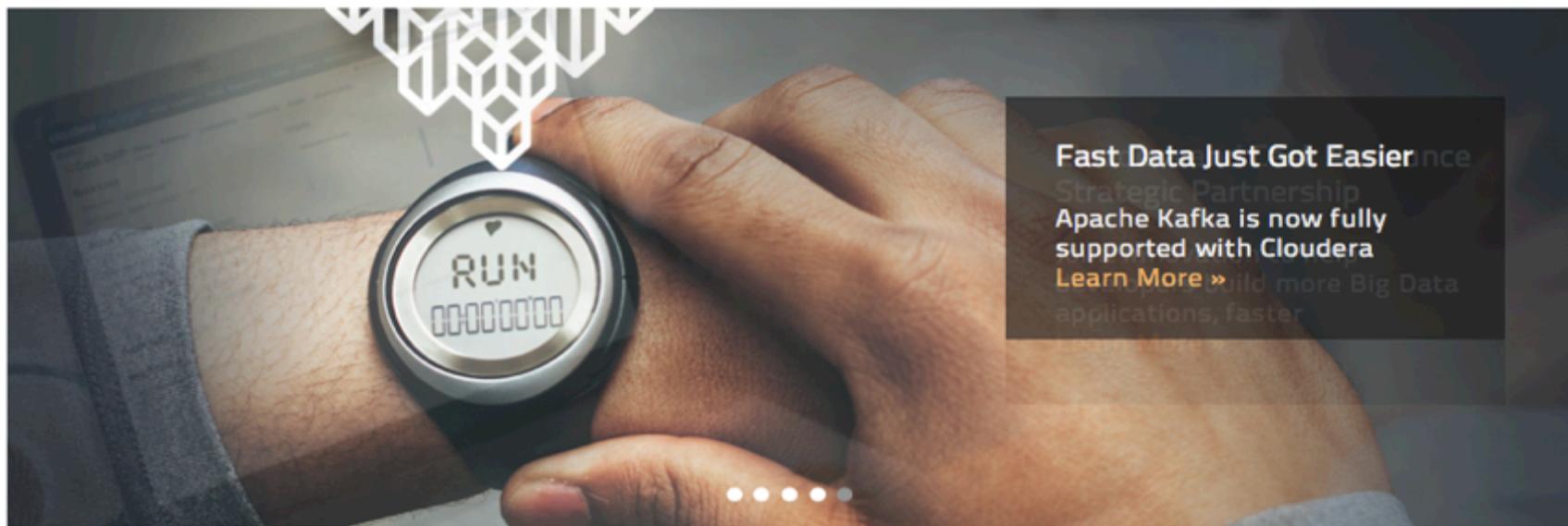
SOLUTIONS

CUSTOMERS

PARTNERS

RESOURCES

BLOGS



Fast Data Just Got Easier

Strategic Partnership  
Apache Kafka is now fully supported with Cloudera  
[Learn More »](#)

**Sandbox Free Hadoop**  
Cloudera Live, the fastest way to learn Hadoop  
[Try it »](#)

**Open Standards Commitment**  
Cloudera upgrades ASF sponsorship to Platinum  
[Learn More »](#)

**Webinar: Project Rhino**  
Enhancing Data Protection for Hadoop  
[Watch Now »](#)

# MapR

[Blog](#) | [App Gallery](#) | [Developer Central](#) | [Support](#) |  | [Login](#) |  ▾



[Why Hadoop](#)   [Products](#)   [Solutions](#)   [Training](#)   [Resources](#)   [Partners](#)   [Company](#)

Get Real with Hadoop

[Learn more](#) ⓘ

10 REASoNs  
TO BE a



# Hortonworks



Solutions

Products

Services

Partners

Get Started

Resources



## We do Hadoop.

Enabling the Data-First Enterprise

### GET THE LATEST

Industries Do Hadoop:  
Reveal the value in your data.



[Sign Up Now »](#)

### OPTIMIZE DATA ARCHITECTURE

Reduce costs by moving data  
and processing to Hadoop.



[Get Started »](#)

### ADVANCED ANALYTIC APPS

Find new opportunities with new  
types of data.



[Get Started »](#)

# Module 2

Basics of Hadoop

Ecosystem Introduction

- Why Hadoop - Volume, Variety & Veracity

- What Features - High Performance Storage, Computing & Data

- Who Uses - Security, Governance & Platform Features

Components

- Infrastructure Components - Servers, Configuration & Cluster

- Node Setup & Replication

- Software Components

What and Why

- What Components are Essential

- Why Do You Need the Components

- Why Do You Need to Manage the Components

# Hadoop Ecosystem

---



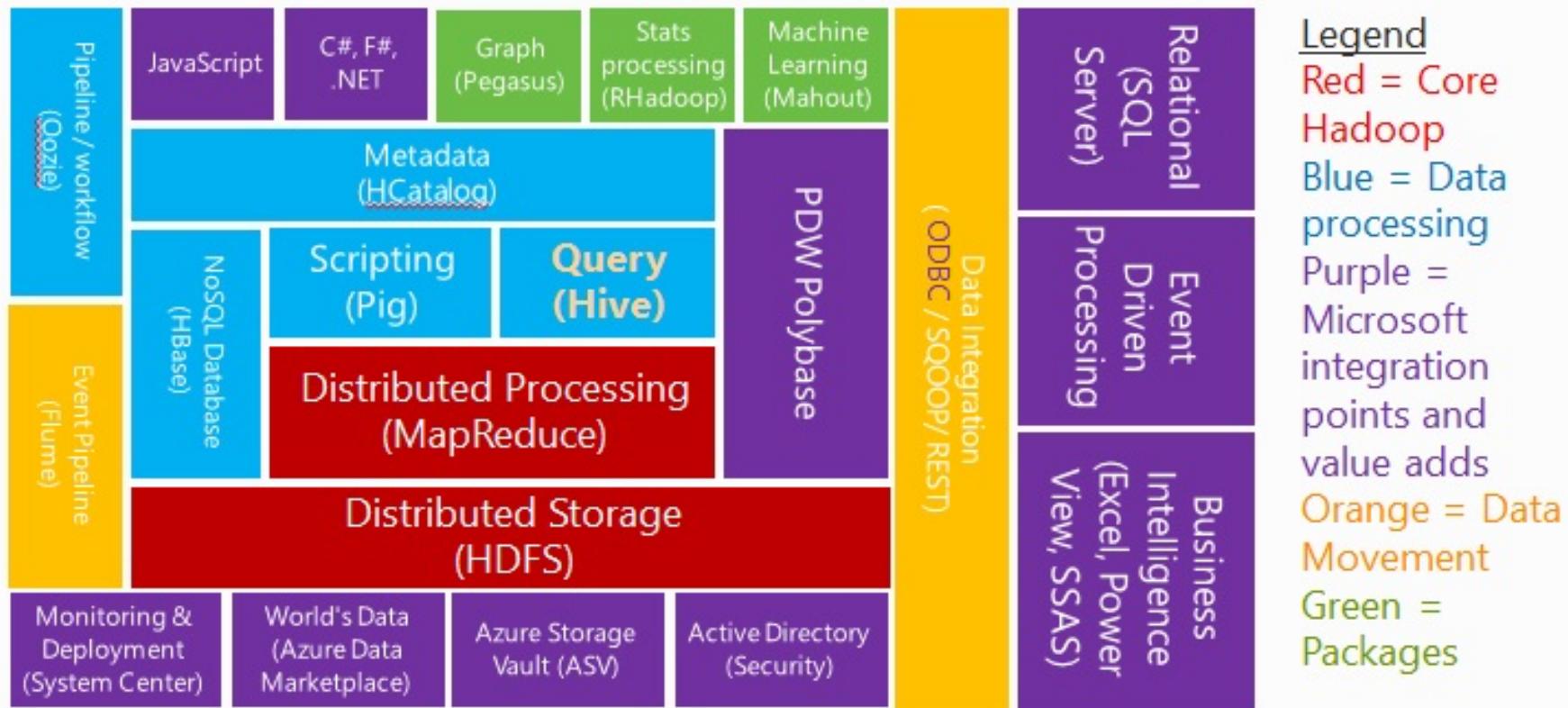
# Vendor integration

---



# Microsoft

## HDINSIGHT / HADOOP Eco-System



# **BASICS OF HADOOP**

# Data Today

Social



Mobile



User Monitoring



Online Experience



Real Time Data



# BIG Data Example

To: Bob.Collins@bankwithus.com

Dear Mr. Collins,

This email is in reference to my bank account which has been efficiently handled by your bank for more than five years. There has been no problem till date until last week the situation went out of the hand.

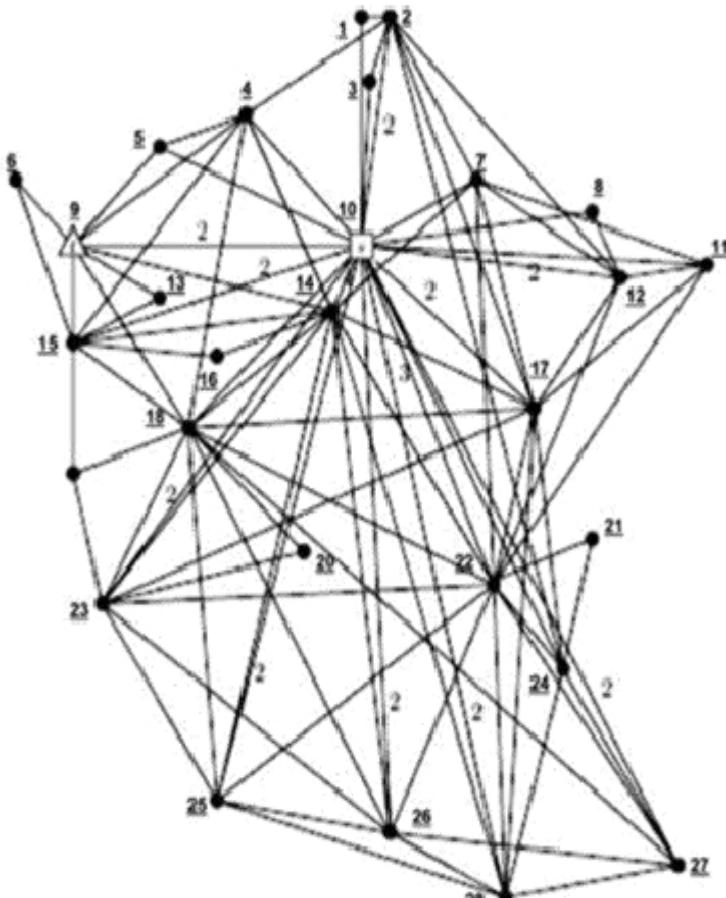
I have deposited one of my high amount cheque to my bank account no: 65656512 which was to be credited same day but due to your staff carelessness it wasn't done and because of this negligence my reputation in the market has been tarnished. Furthermore I had issued one payment cheque to the party which was showing bounced due to "Insufficient balance" just because my cheque didn't make on time.

My relationship with your bank has matured with the time and it's a shame to tell you about this kind of services are not acceptable when it is question of somebody's reputation. I hope you got my point and I am attaching a copy of the same for further rapid procedures and remit into my account in a day.

Yours sincerely

Daniel Carter

Ph: 564-009-2311



# Big Data Example

- We will often imply additional information in spoken language by the way we place stress on words.
- The sentence "I never said she stole my money" demonstrates the importance stress can play in a sentence, and thus the inherent difficulty a natural language processor can have in parsing it.
  - "I never said she stole my money" - Someone else said it, but *I* didn't.
  - "I **never** said she stole my money" - I simply didn't ever say it.
  - "I never **said** she stole my money" - I might have implied it in some way, but I never explicitly said it.
  - "I never said **she** stole my money" - I said someone took it; I didn't say it was she.
  - "I never said she **stole** my money" - I just said she probably borrowed it.
  - "I never said she stole **my** money" - I said she stole someone else's money.
  - "I never said she stole my **money**" - I said she stole something, but not my money
- Depending on which word the speaker places the stress, this sentence could have several distinct meanings.

# Features

- Infrastructure Requirements
  - Scalable platform
  - Database independence
  - Fault tolerant architectures
  - Low cost of acquisition and store
  - Supported by standard toolsets
- Workload Demands
  - Process dynamic data content
  - Process unstructured data
  - Systems that can scale up and scale out with high volume data
  - Perform complex operations within reasonable response time

# Features

- Recover from failure
- Sustained execution
- Self Healing Capabilities
- Machine Learning
- Consistency Maintenance
- Security
- Governance
- High Performance Storage – Processing Speeds
- High Speed Computing

# Partial List of Users



The New York Times



YAHOO!

eHarmony



Autodesk

contextWebs

twitter

amazon.com

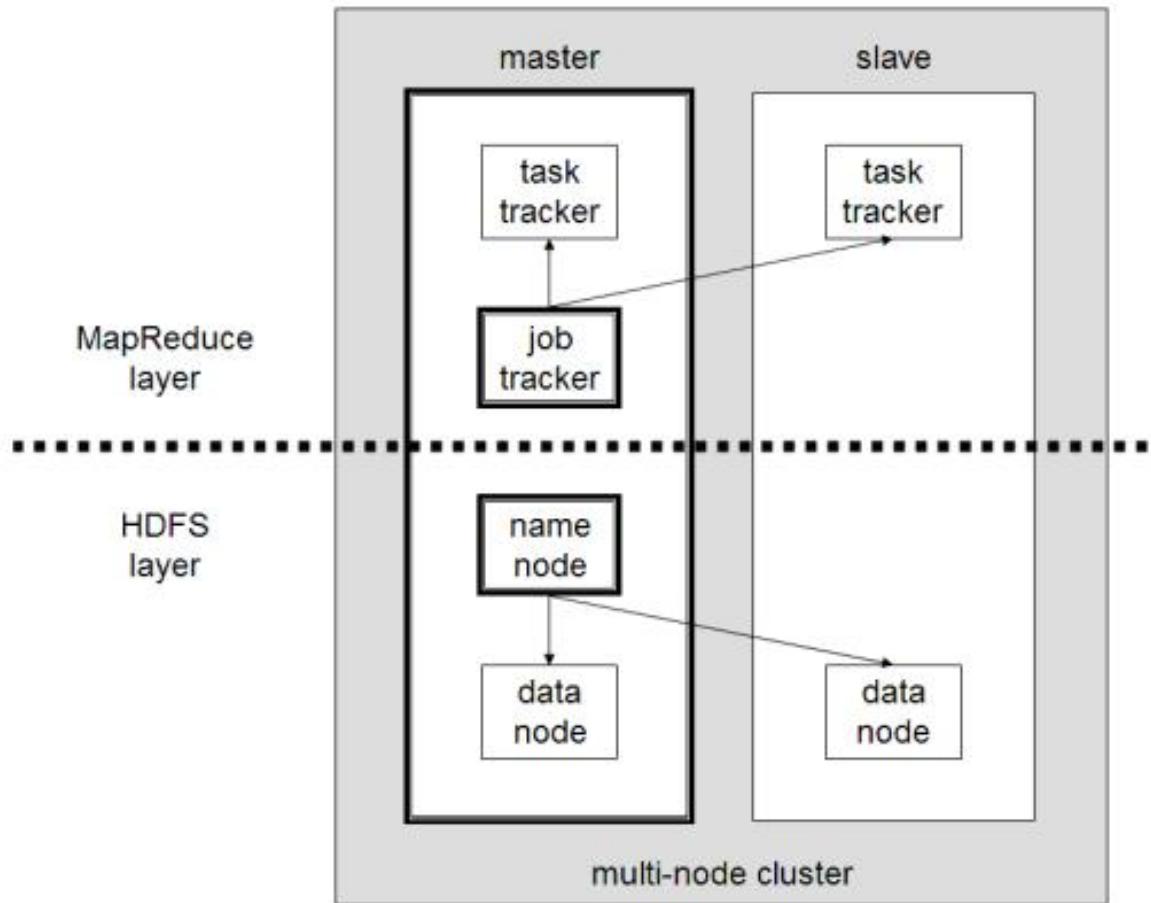
GROUPON

JPMORGAN CHASE & CO.

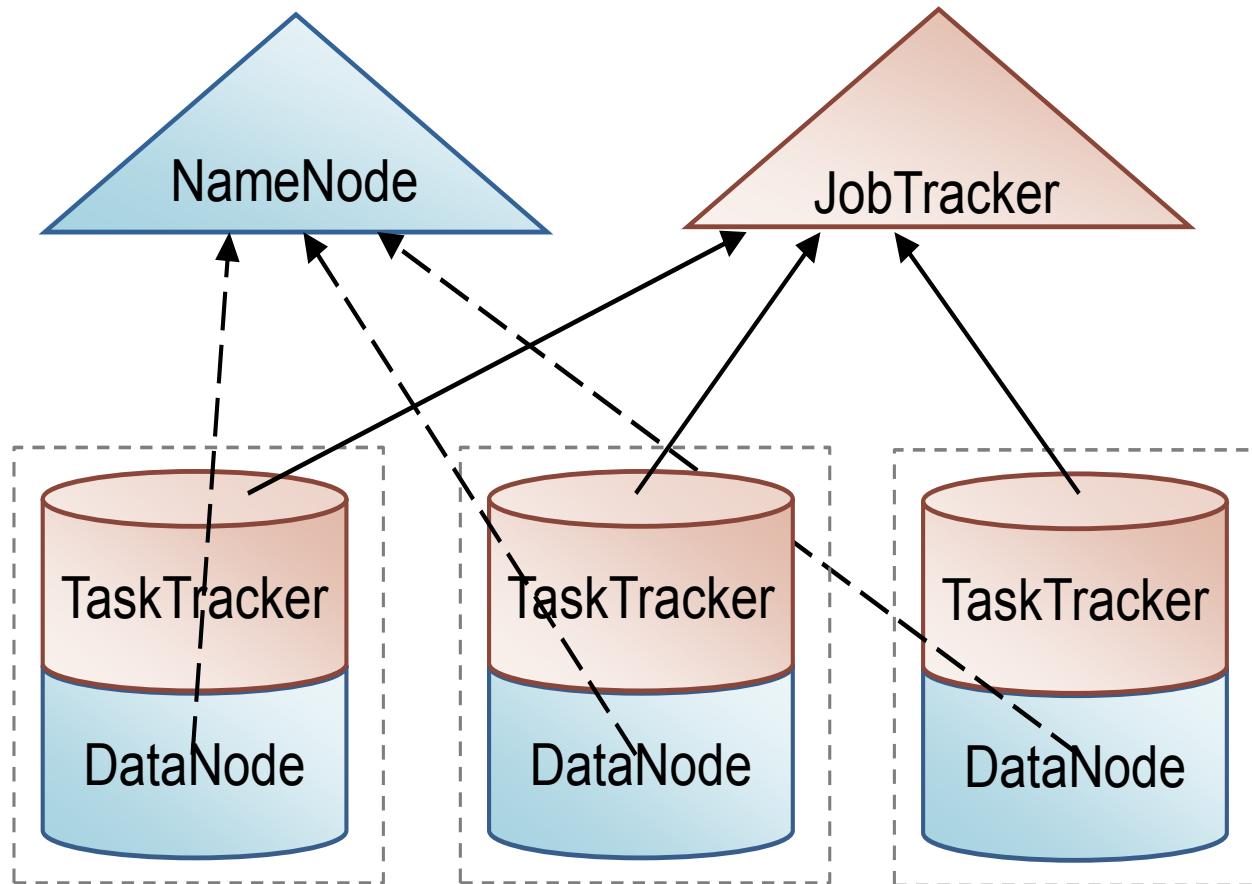
NAVTEQ



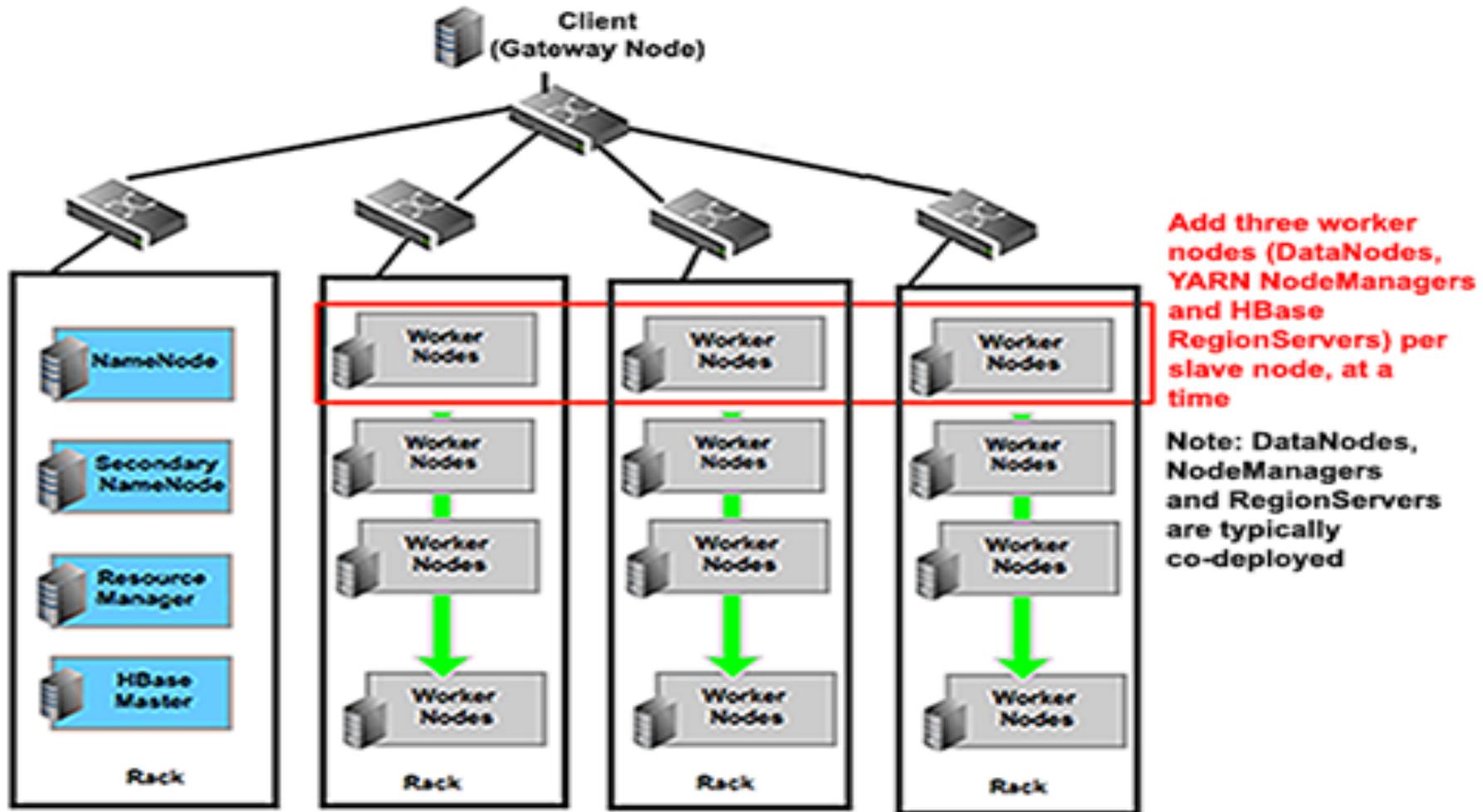
# Hadoop Architecture



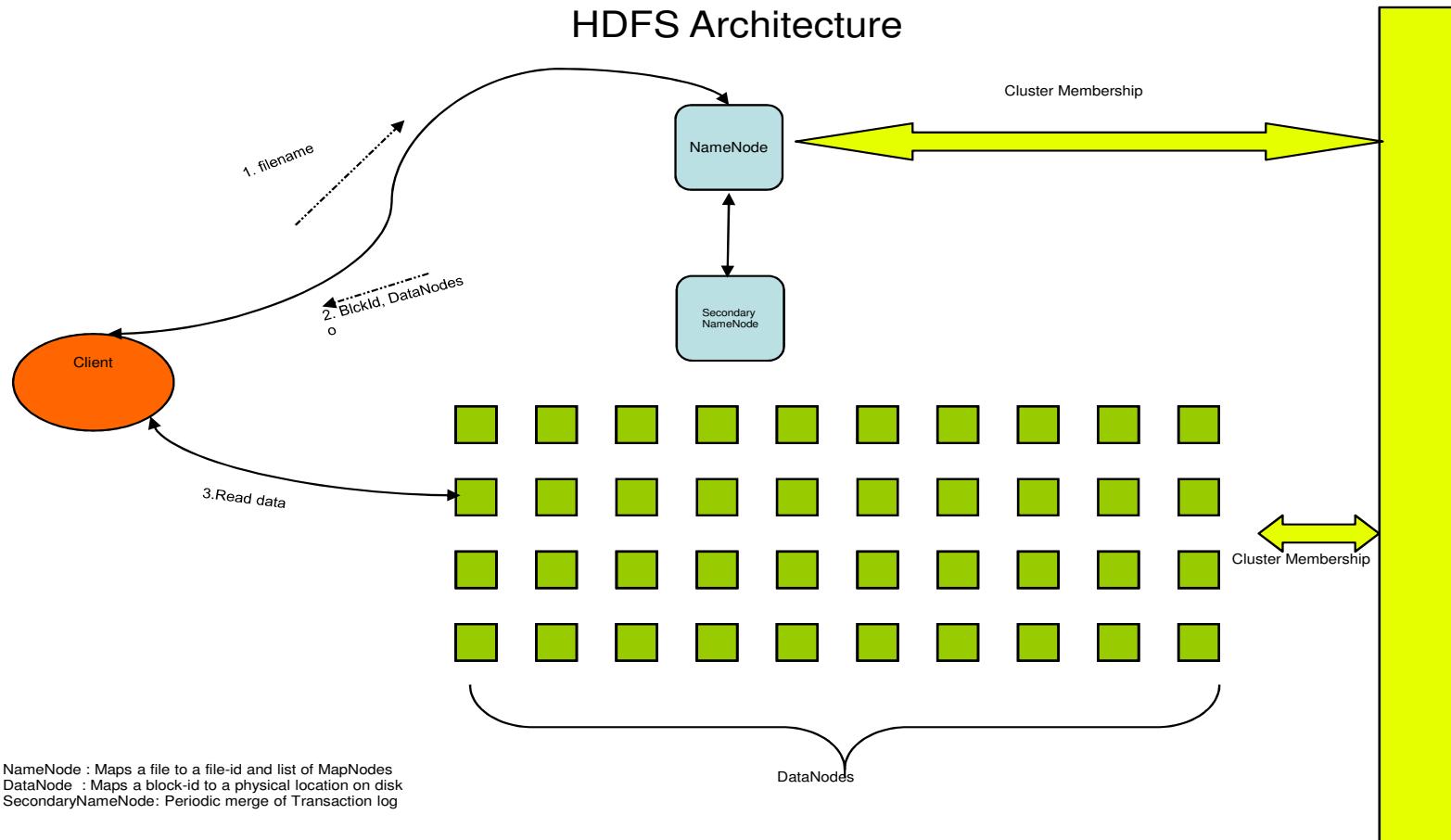
# Node Setup



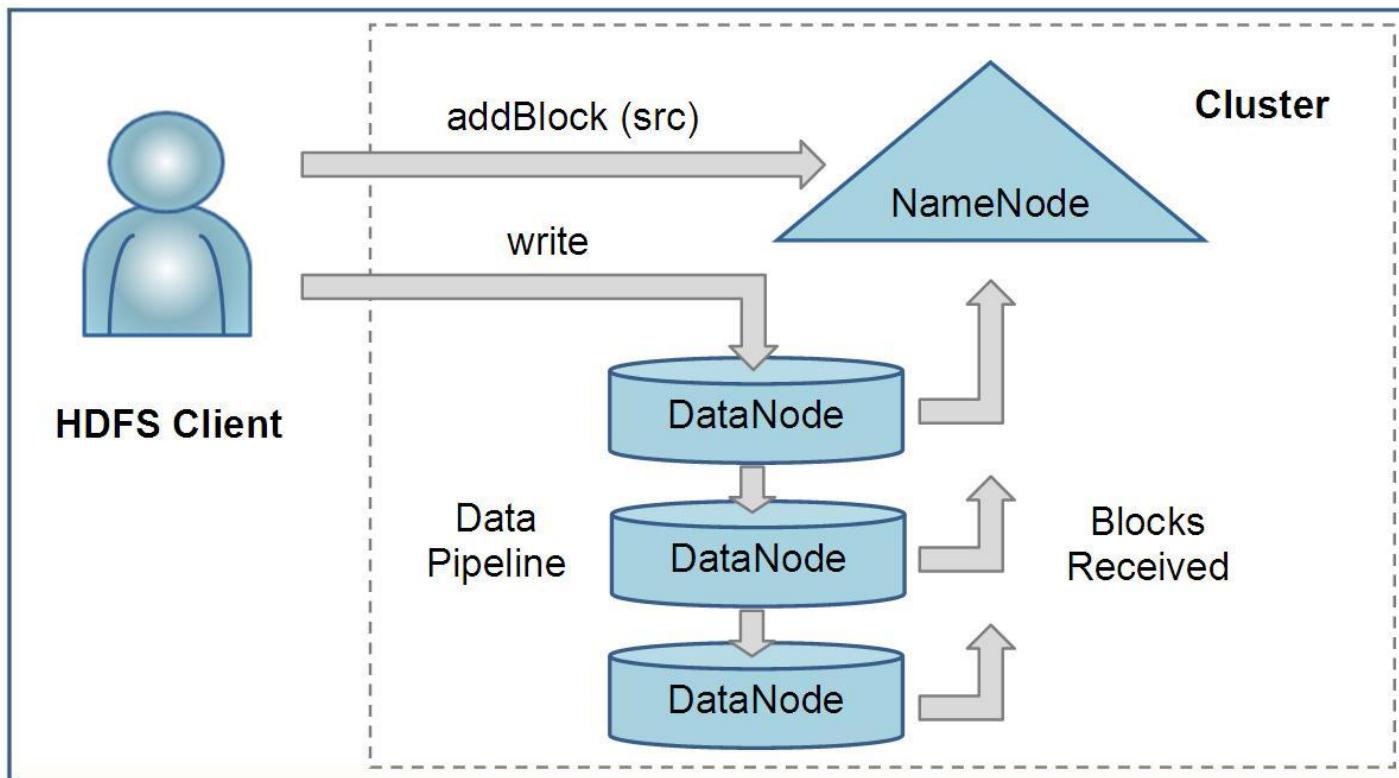
# Cluster Setup



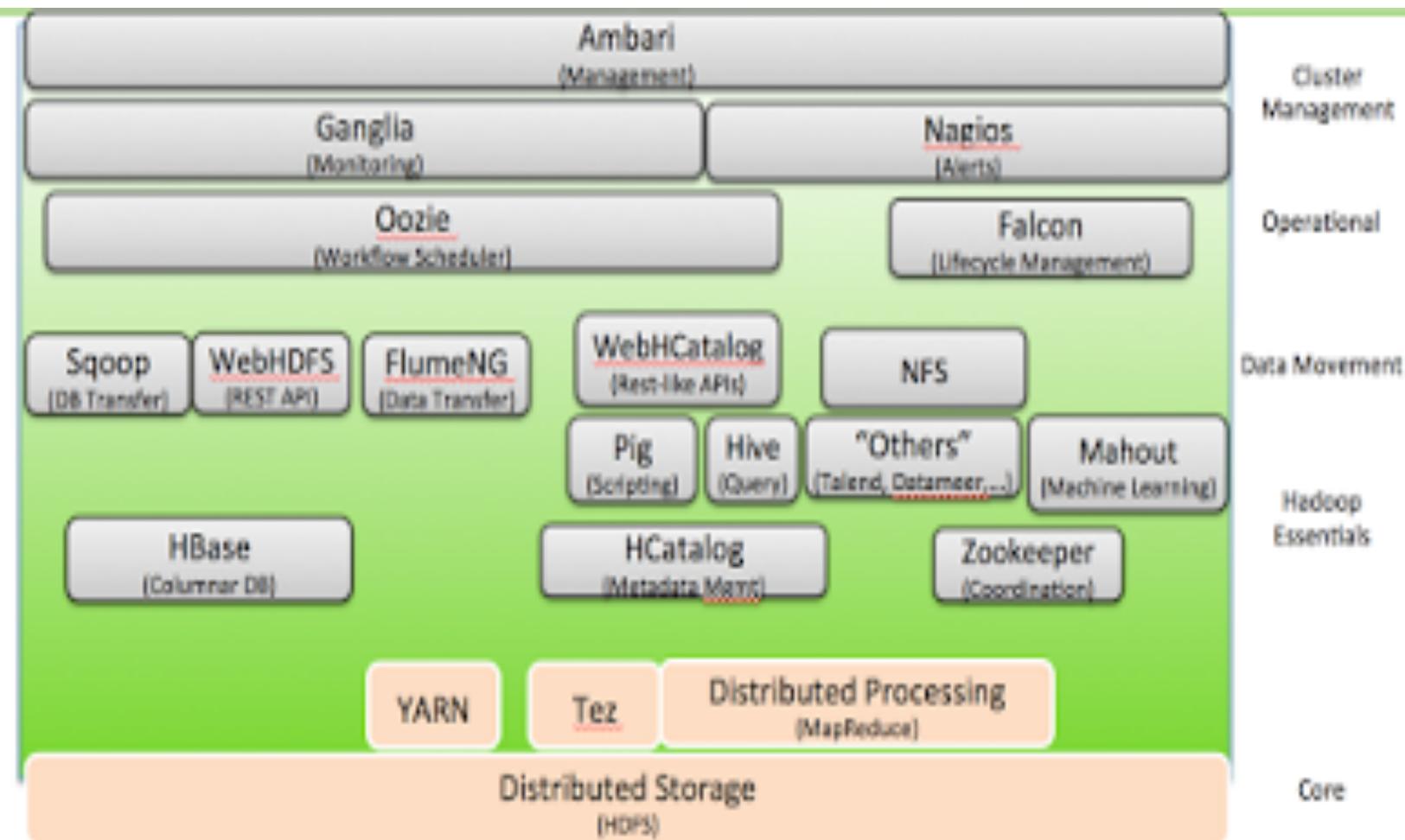
# HDFS Architecture / Replication



# Replication



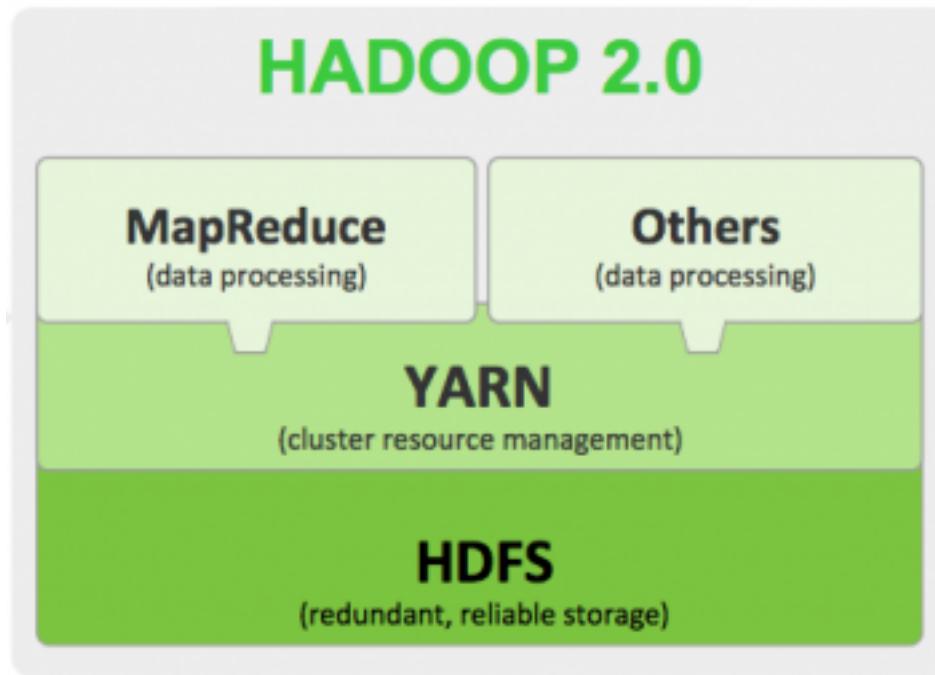
# Software Components



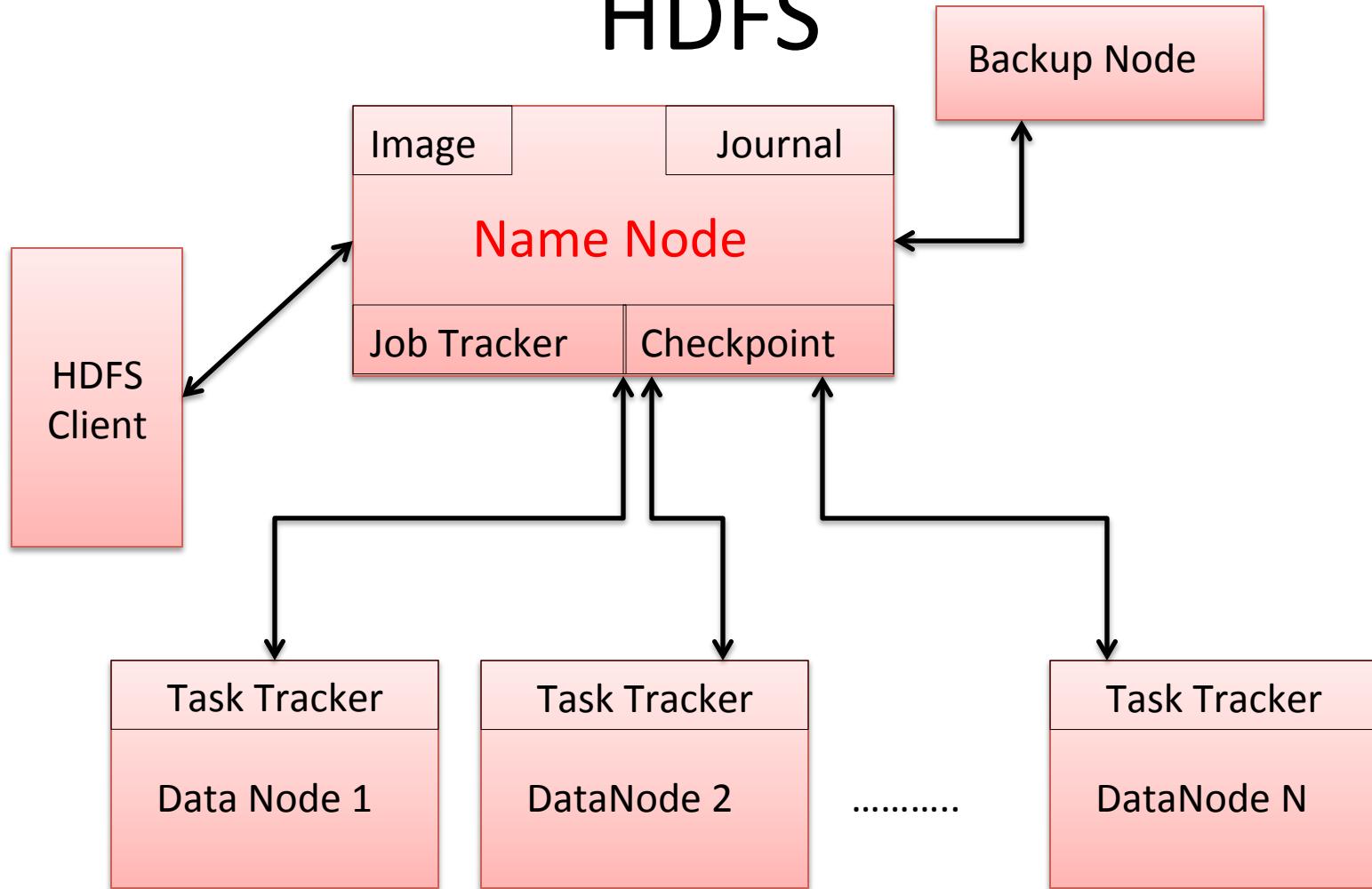
# Module 3

- Hadoop in BI and Analytics
- Core Components
  - Hadoop Core Components
    - HDFS - Setup, Management & Maintenance
    - YARN - Setup, Resource Management & Maintenance
  - MapReduce
- Hadoop Tools
  - Hbase
  - Hive
  - Pig
  - Impala
  - Drill
  - Spark
  - Shark
  - Zookeeper
  - Avro / SerDe
  - Oozie
- Flume
  - Falcon
  - Knox
  - Tez
  - Ambala
  - HAWQ
  - Mahout
- SQL on Hadoop
- Best Practices
- Tools and Distributions
- Platform Architecture
- Apache Platform Recommendation
- Distribution Architectures
- Business Intelligence Integration
- Hive and Impala
- Platfora
- Datameer
- Analytics Integration
- R Integration
- RHadoop
- SAS Integration

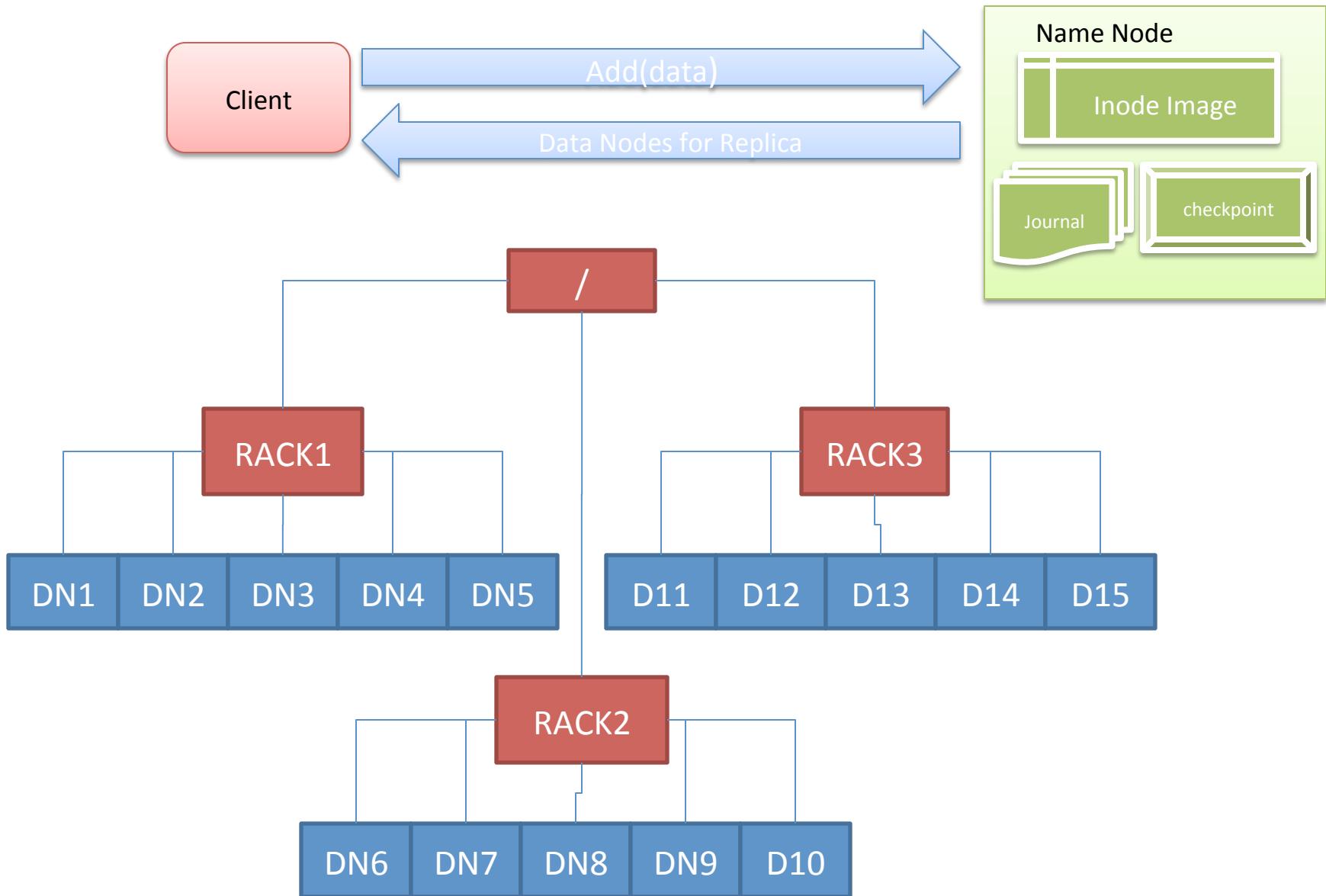
# Core Components



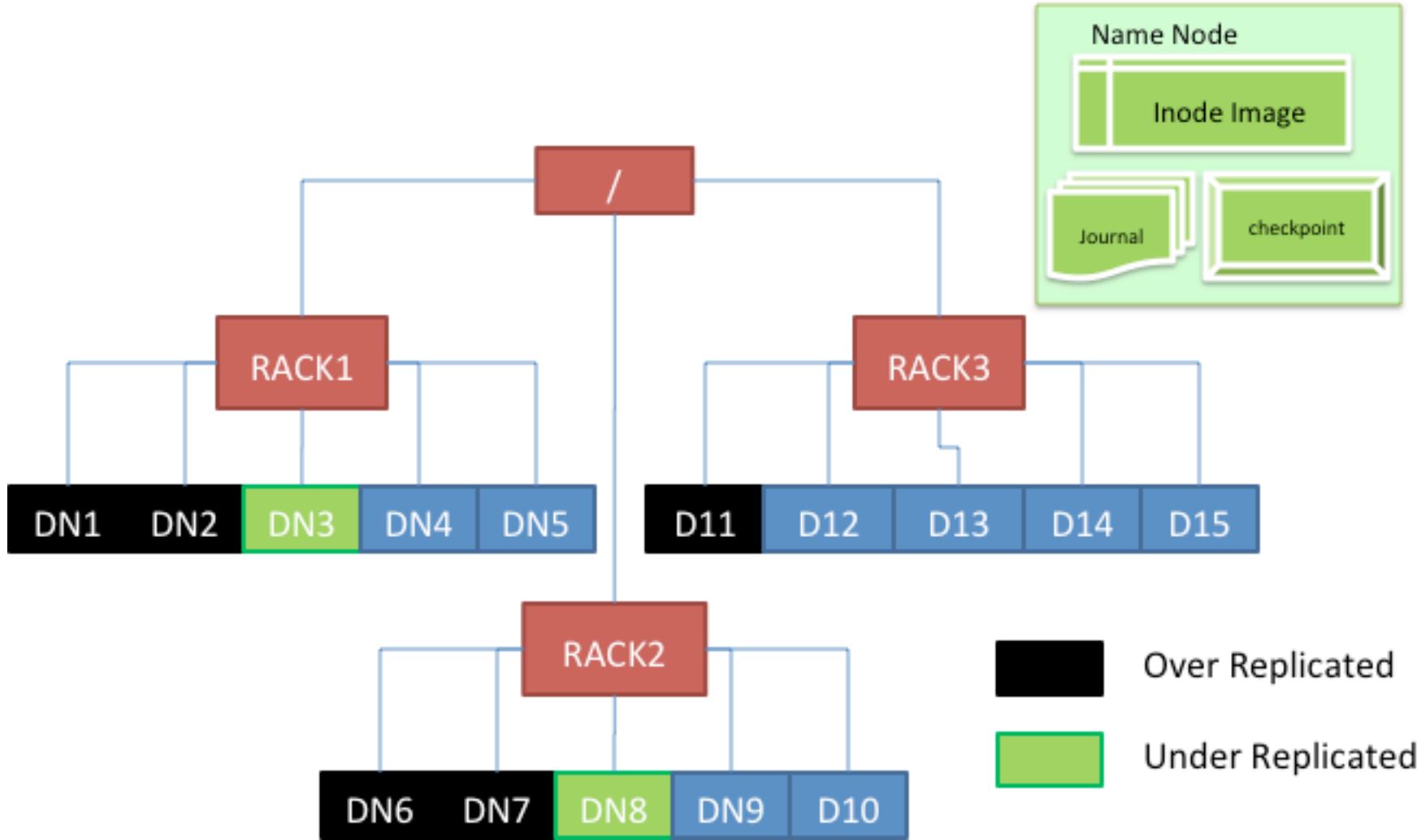
# HDFS



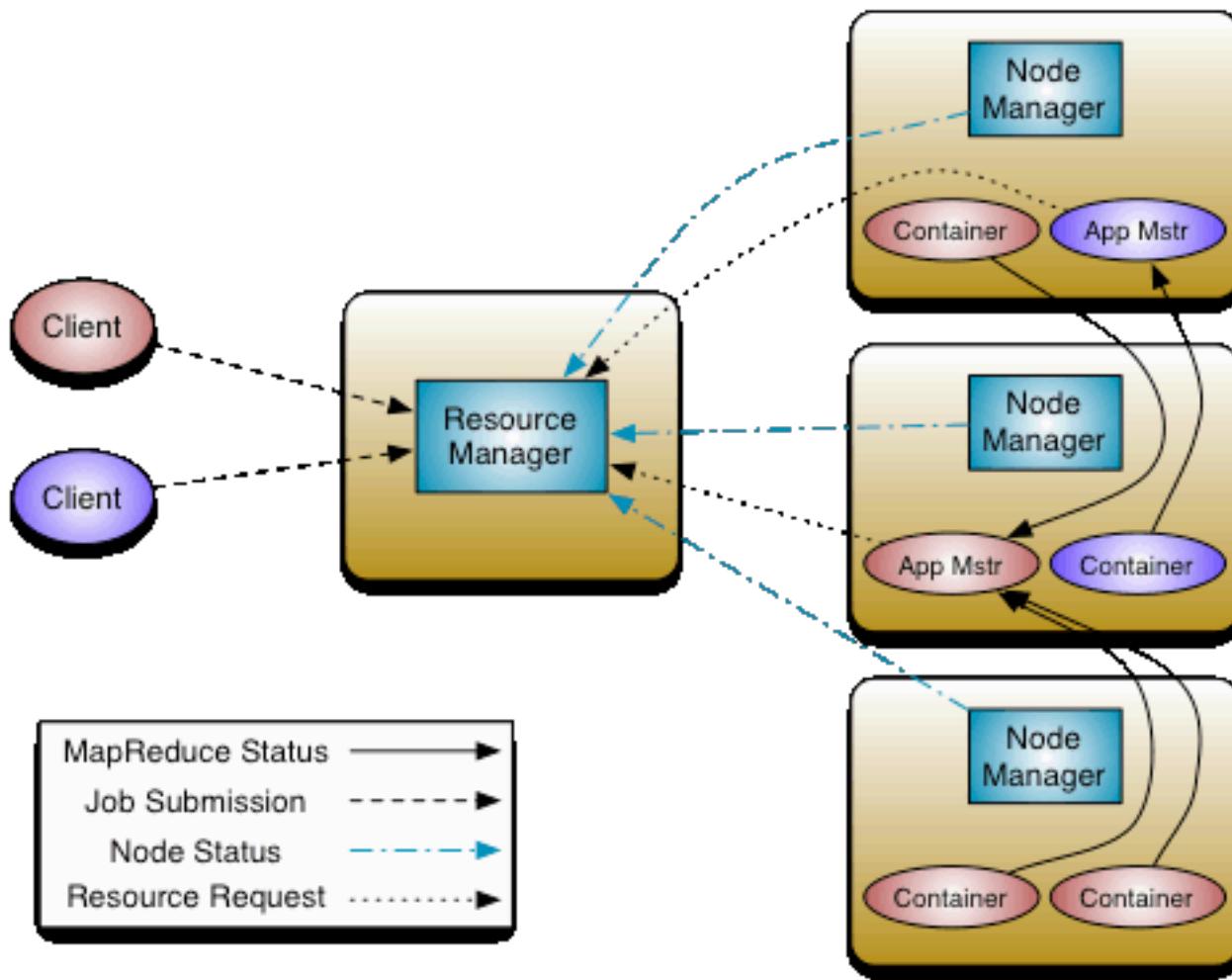
# Block Placement



# REPLICATION MANAGEMENT



# YARN



# YARN Architecture

## ResourceManager (RM)

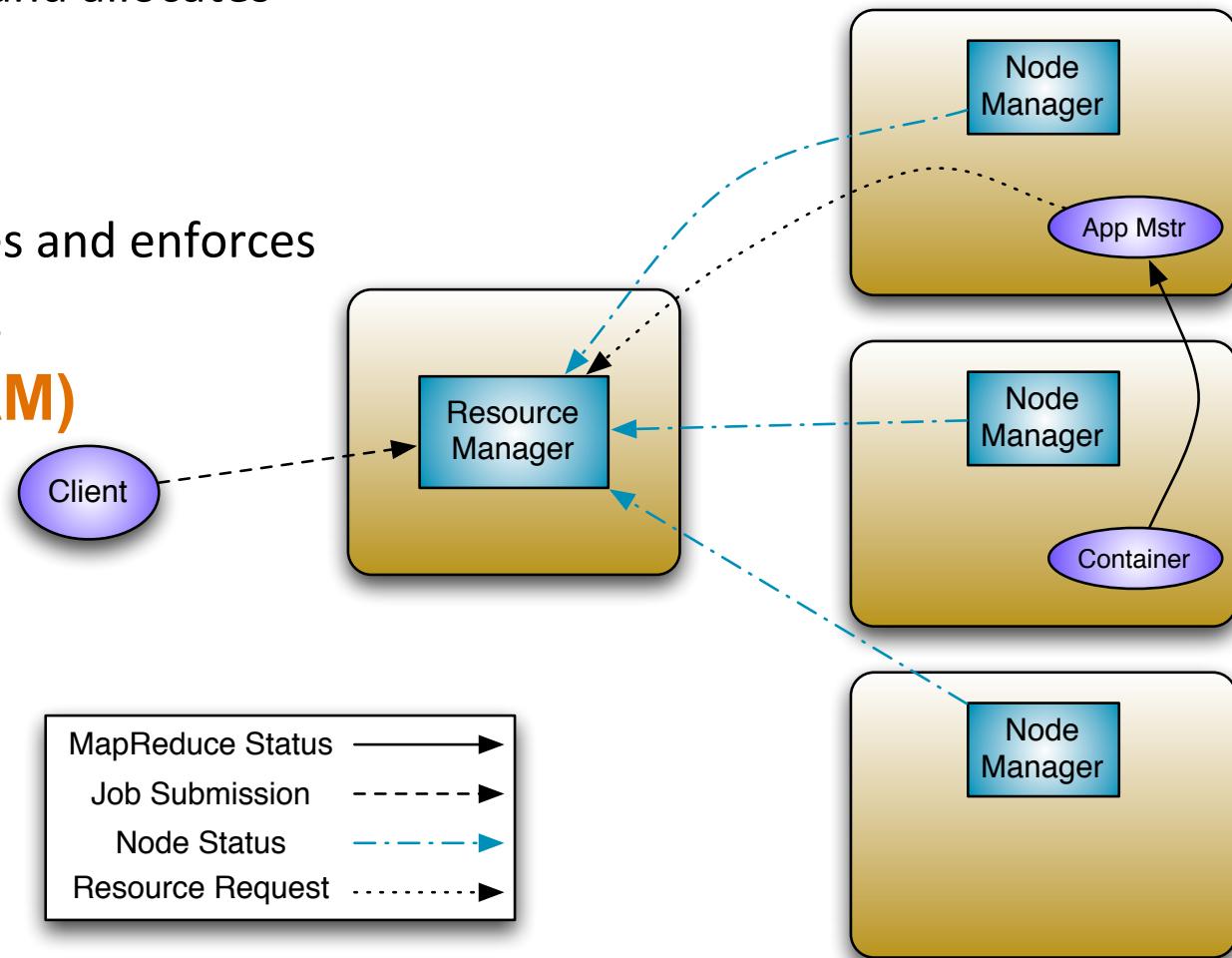
Central agent - Manages and allocates cluster resources

## NodeManager (NM)

Per-Node agent - Manages and enforces node resource allocations

## ApplicationMaster (AM)

Per-Application –  
Manages application lifecycle and task scheduling

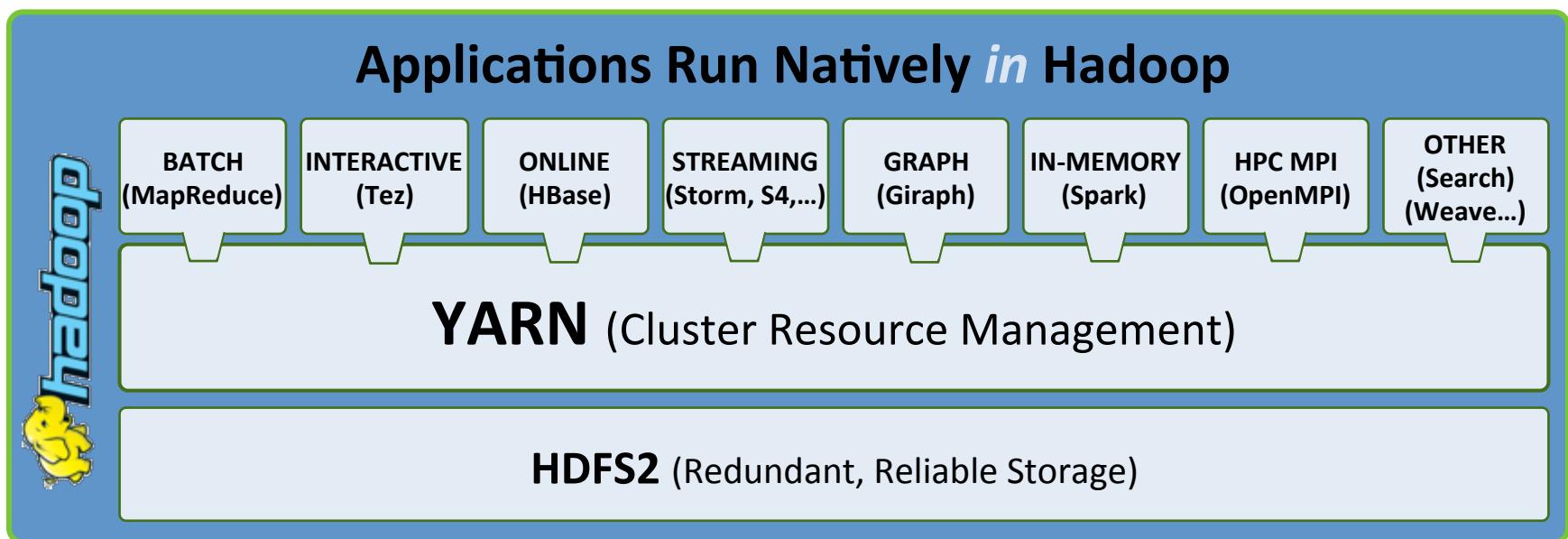


# YARN – Hadoop Beyond Batch

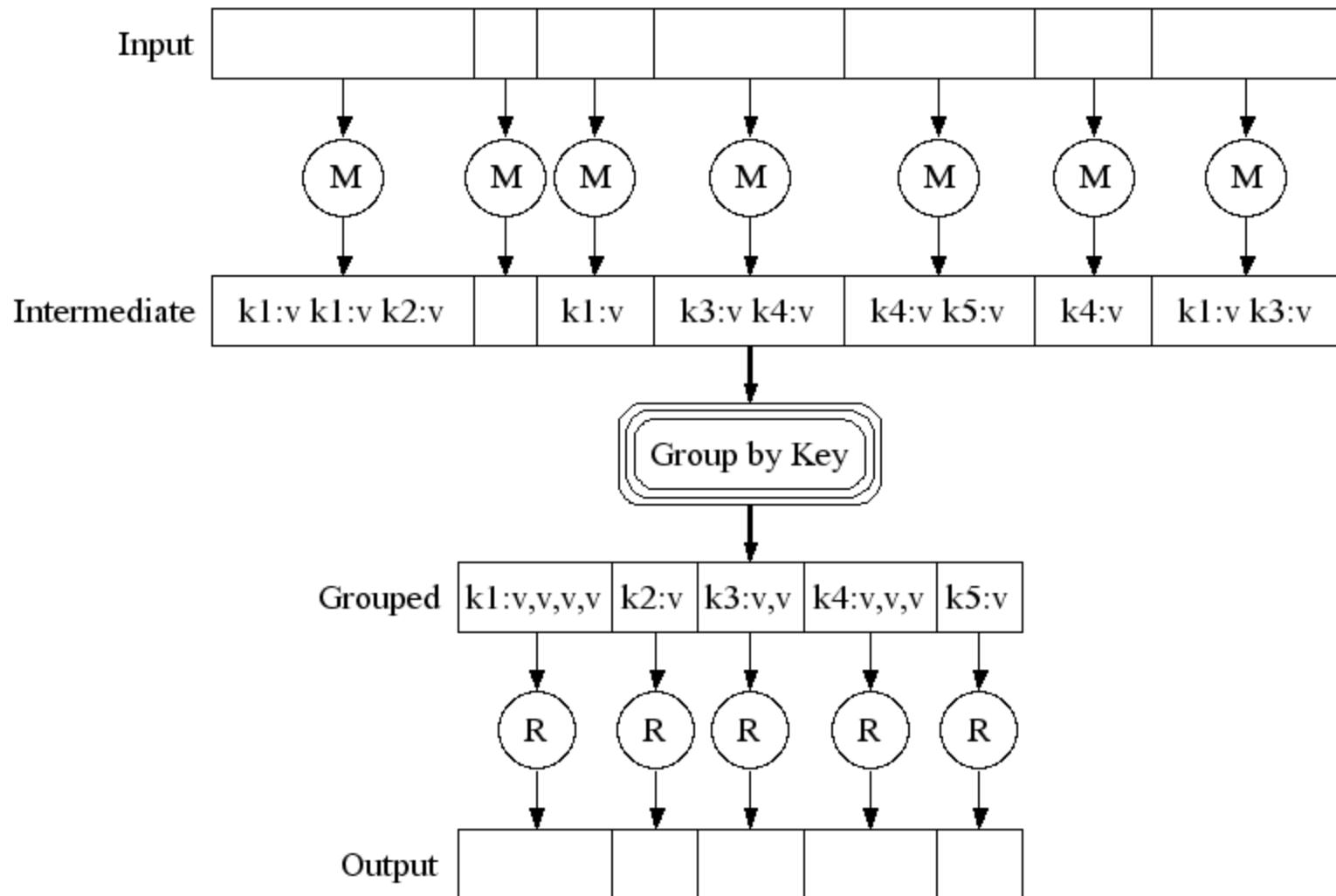
**Store ALL DATA in one place...**

**Interact with that data in MULTIPLE WAYS**

**with Predictable Performance and Quality of Service**



# The MapReduce Framework (pioneered by Google)



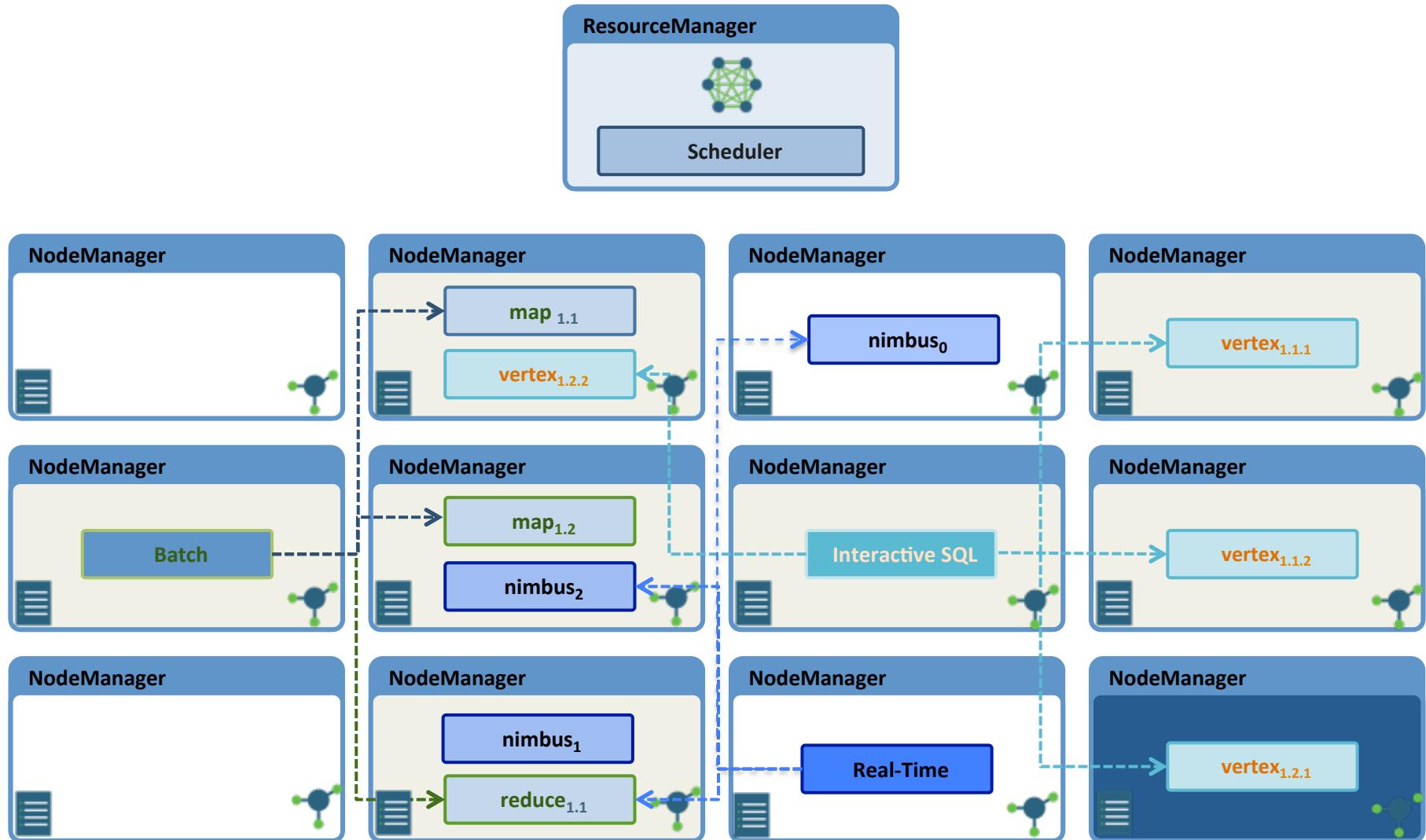
# YARN: Efficiency with Shared Services

## Yahoo! leverages YARN

- 40,000+ nodes running YARN across over 365PB of data
- ~400,000 jobs per day for about 10 million hours of compute time
- *Estimated a 60% – 150% improvement on node usage per day using YARN*
- *Eliminated Colo (~10K nodes) due to increased utilization*

For more details check out the YARN SOCC 2013 paper

# YARN as Cluster Operating System



# Multi-Tenancy is Built-in

- **Queues**
- **Economics as *queue-capacity***

- Hierarchical Queues

- **SLAs**

- Cooperative Preemption

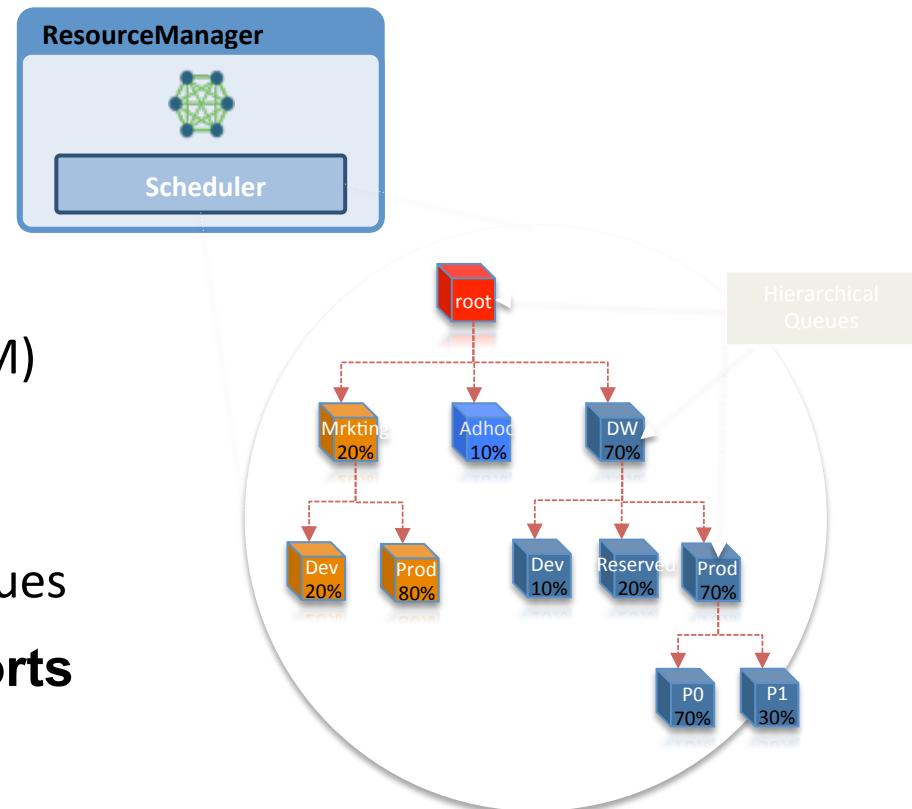
- **Resource Isolation**

- Linux: cgroups
  - Roadmap: Virtualization (Xen, KVM)

- **Administration**

- Queue ACLs
  - Run-time re-configuration for queues

**Default Capacity Scheduler supports all features**



# YARN Eco-system

## *Applications Powered by YARN*

Apache Giraph – Graph Processing

Apache Hama - BSP

Apache Hadoop MapReduce – Batch

Apache Tez – Batch/Interactive

Apache S4 – Stream Processing

Apache Samza – Stream Processing

Apache Storm – Stream Processing

Apache Spark – Iterative applications

Elastic Search – Scalable Search

Cloudera Llama – Impala on YARN

DataTorrent – Data Analysis

HOYA – HBase on YARN



There's an app for that...  
YARN App Marketplace!

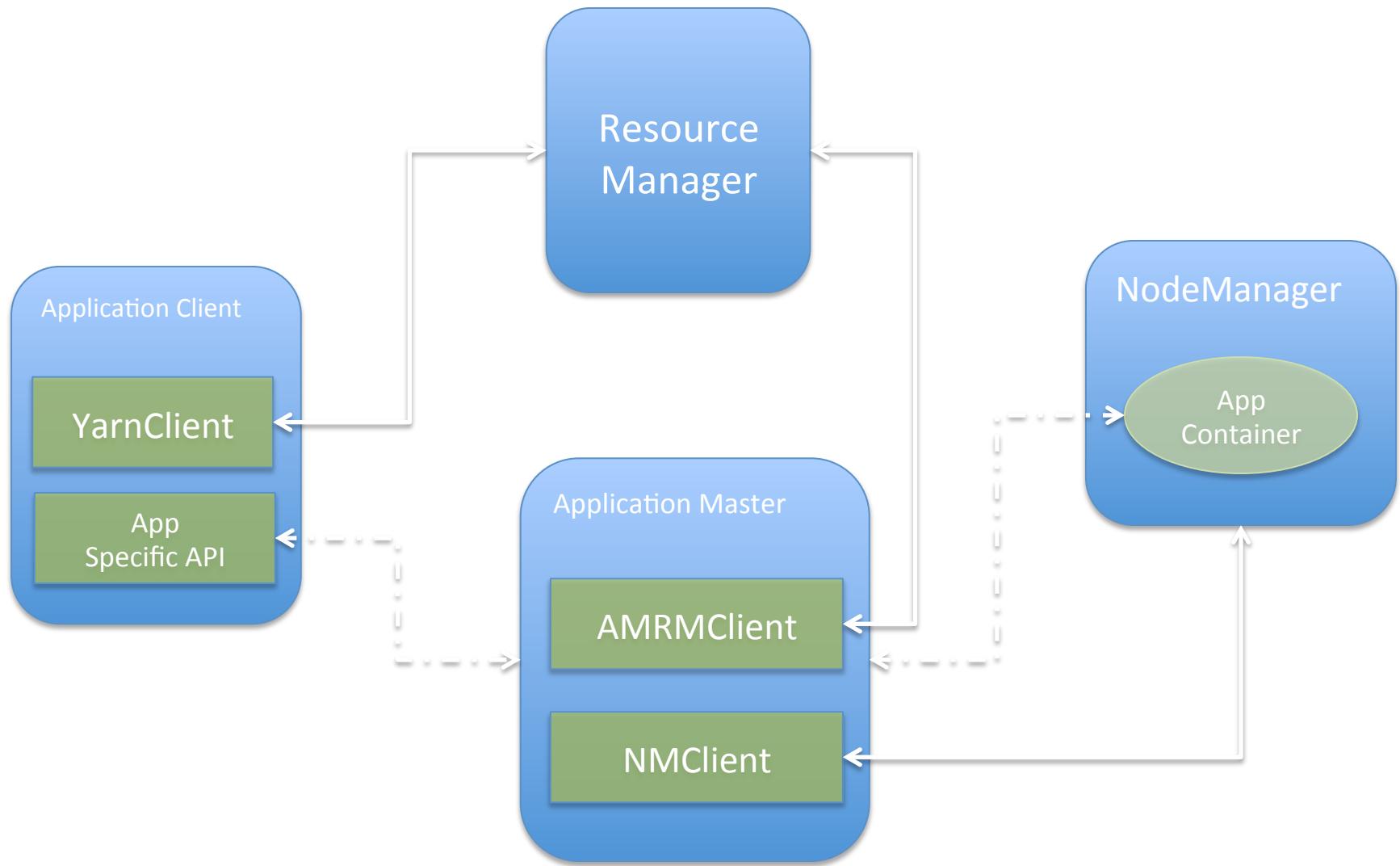
## *Frameworks Powered By YARN*

Apache Twill

REEF by Microsoft

Spring support for Hadoop 2

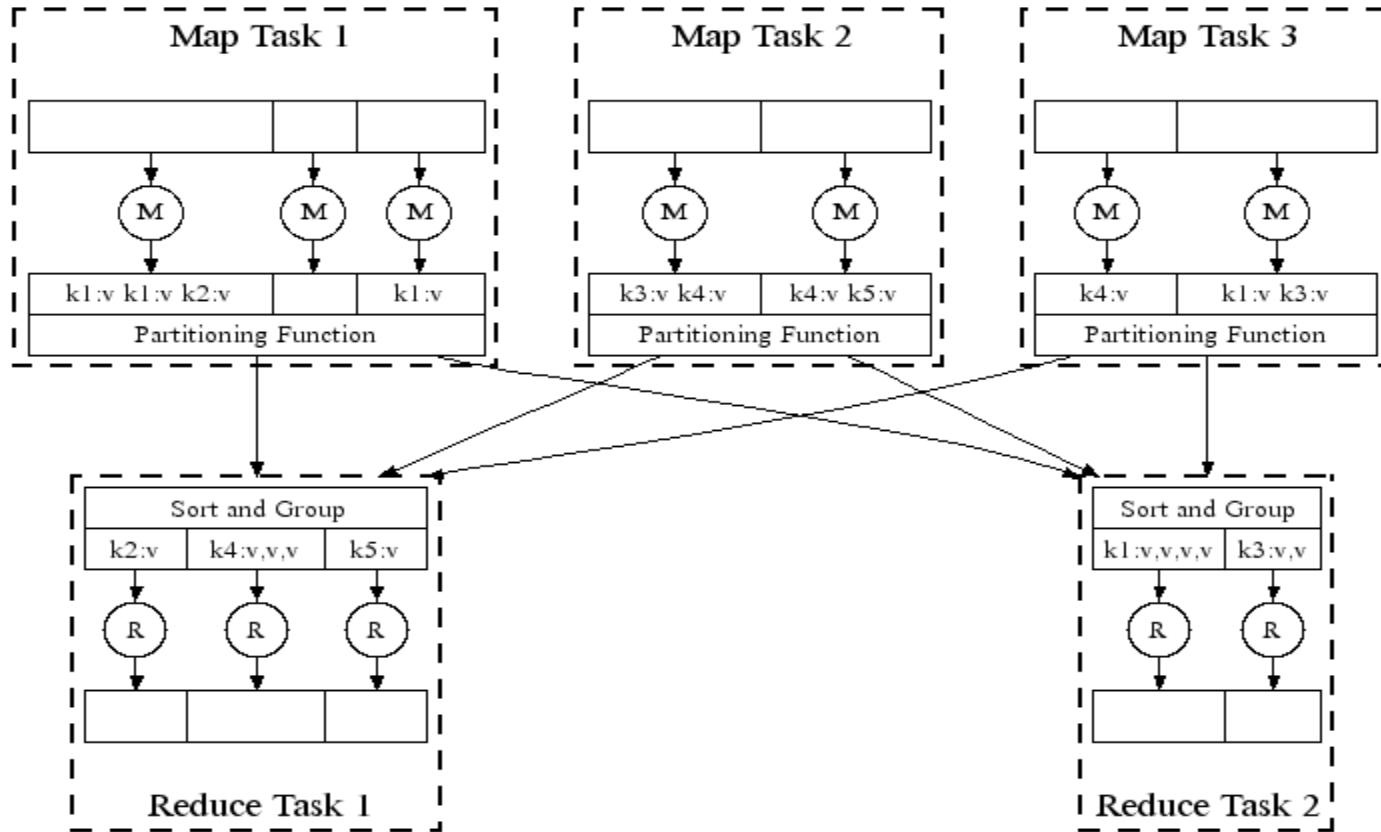
# YARN Application Lifecycle



# Key Take-Aways

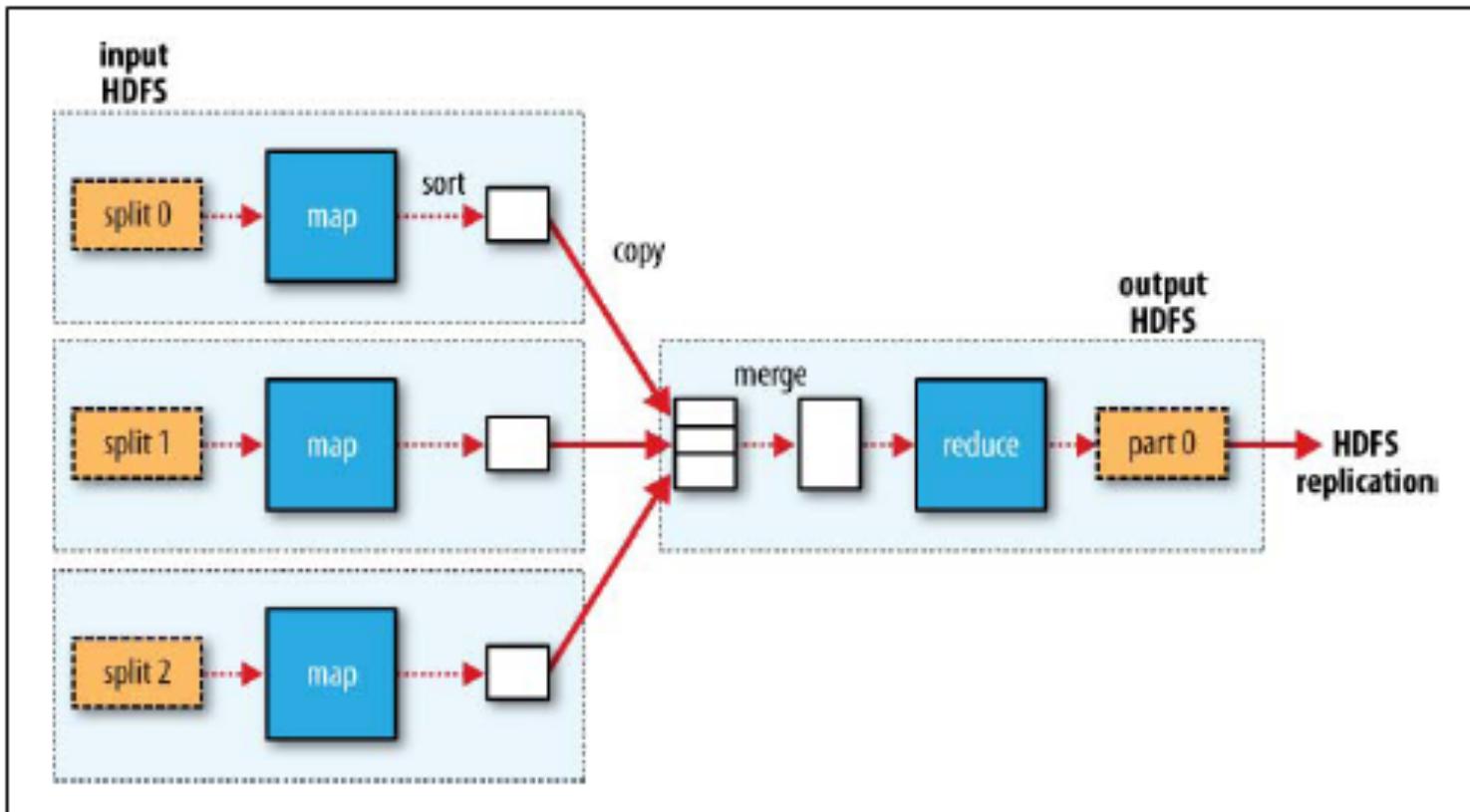
- YARN is a platform to build/run Multiple Distributed Applications in Hadoop
- YARN is completely Backwards Compatible for existing MapReduce apps
- YARN enables Fine Grained Resource Management via Generic Resource Containers.
- YARN has built-in support for multi-tenancy to share cluster resources and increase cost efficiency
- YARN provides a cluster operating system like abstraction for a modern data architecture

# Automatic Parallel Execution in MapReduce (Google)



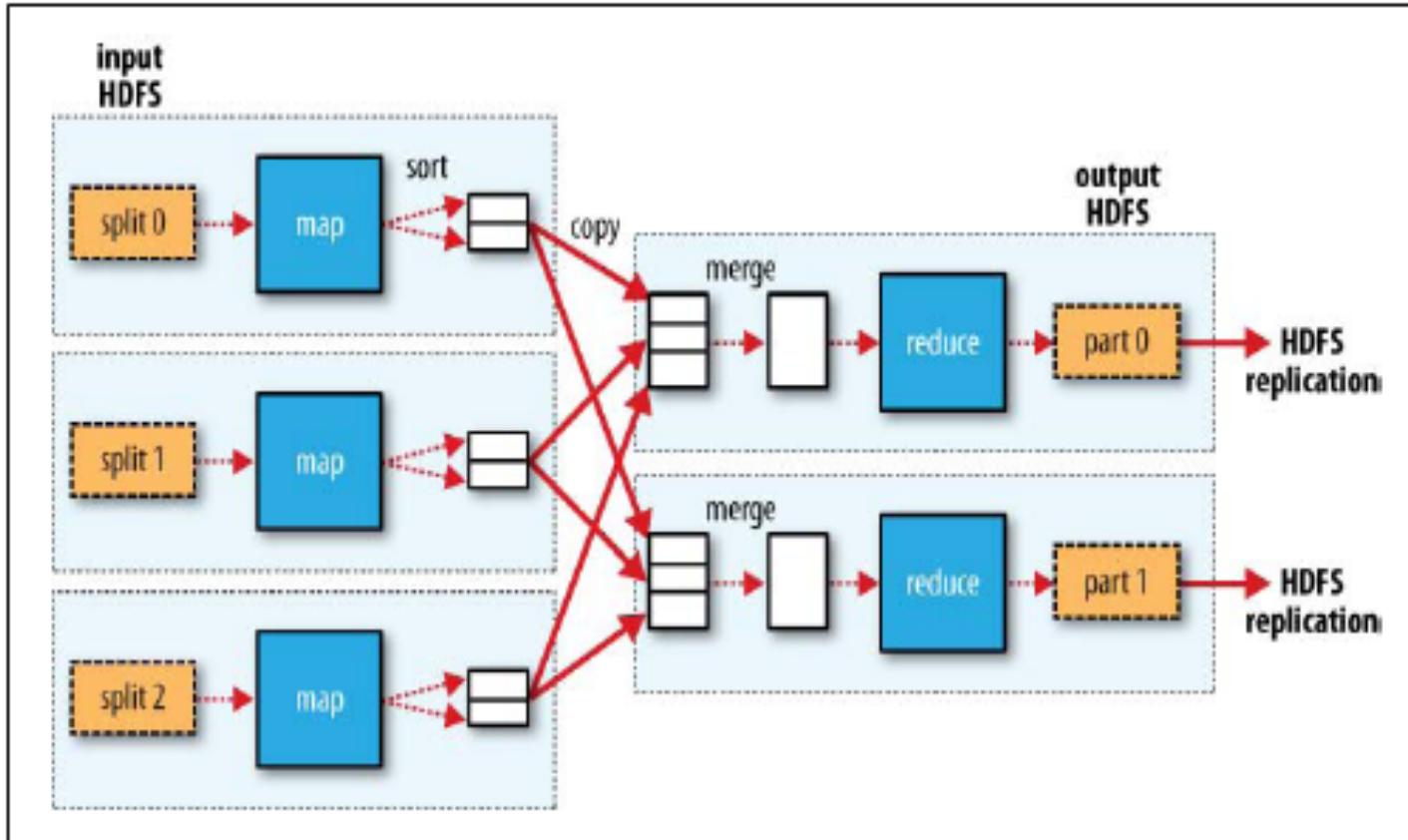
Handles failures automatically, e.g., restarts tasks if a node fails; runs multiple copies of the same task to avoid a slow task slowing down the whole job

# MapReduce in Hadoop (1)



Many Maps and a single Reduce

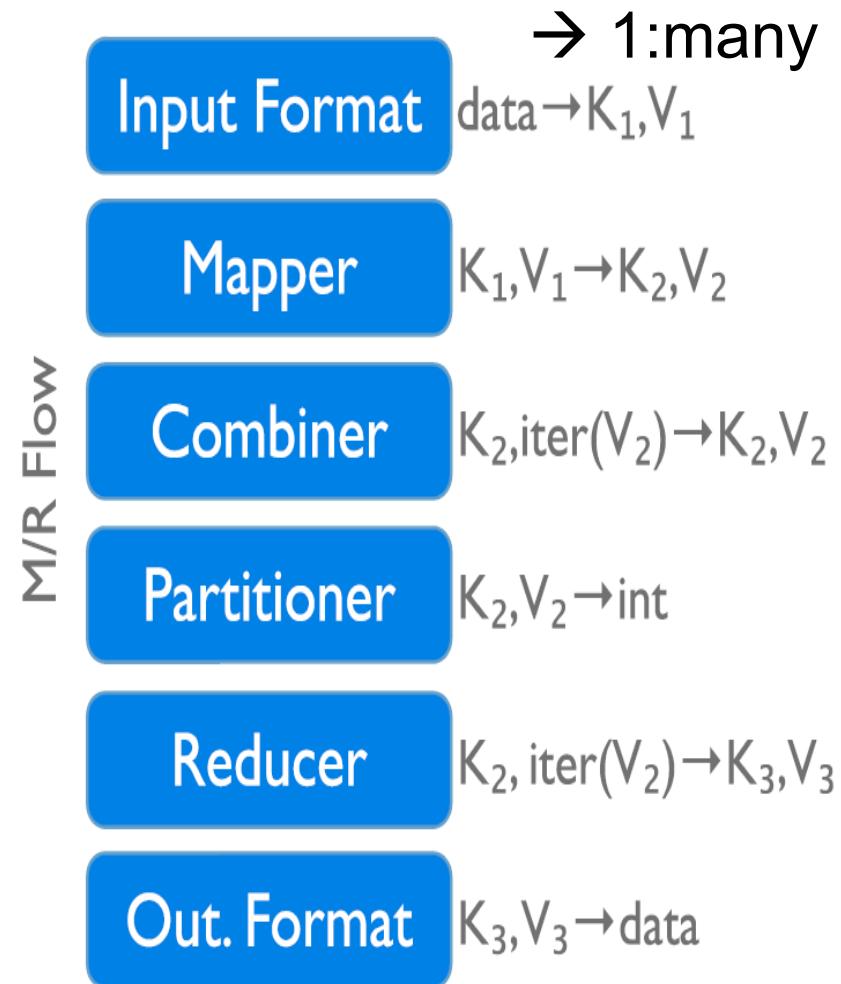
# MapReduce in Hadoop (2)



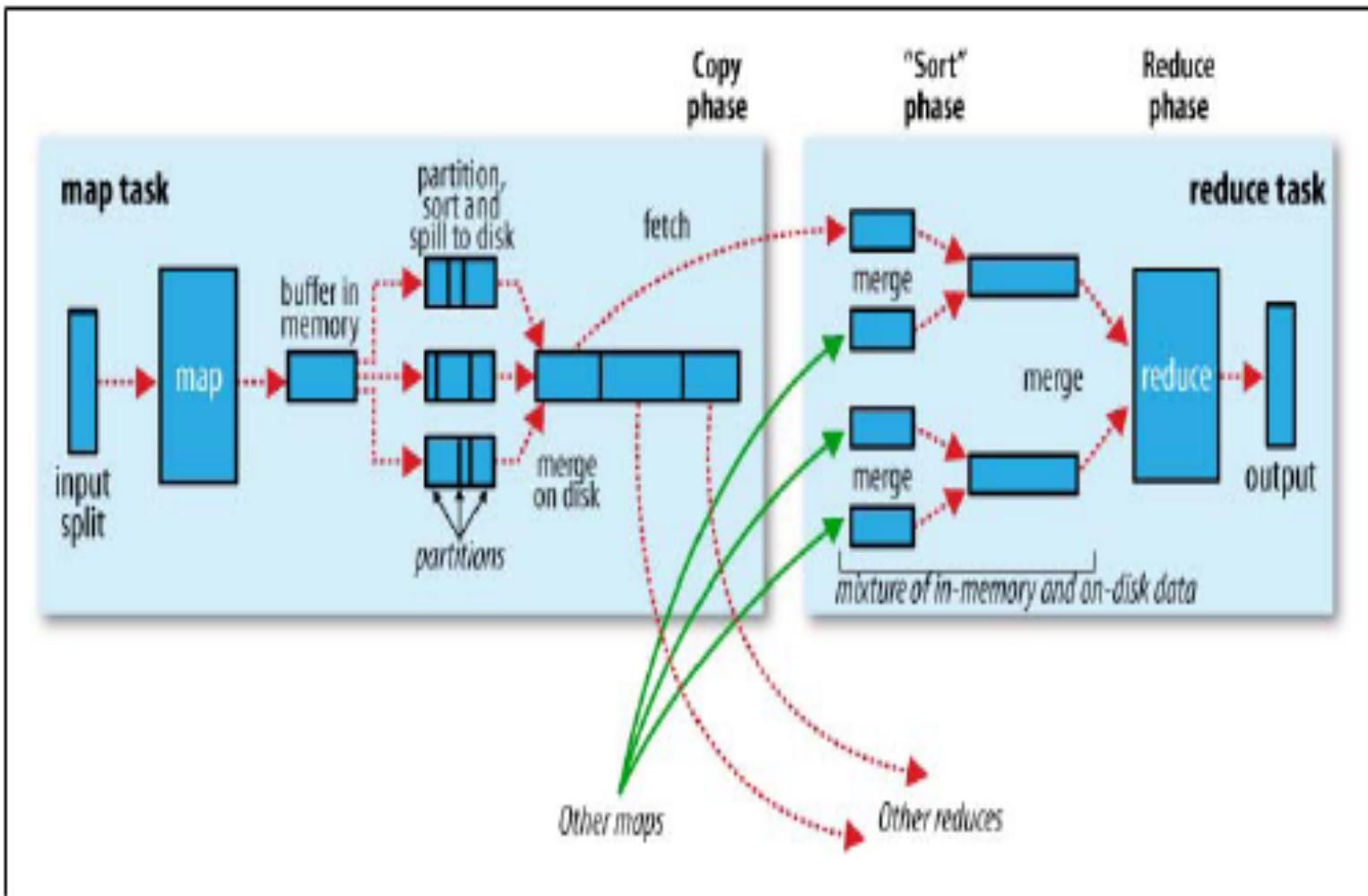
Multiple Maps and Reducers

# Data Flow in a MapReduce Program in Hadoop

- InputFormat
- Map function
- Partitioner
- Sorting & Merging
- Combiner
- Shuffling
- Merging
- Reduce function
- OutputFormat



# Shuffle & Sort - MapReduce



# Lifecycle of a MapReduce Job

```
File Edit Options Buffers Tools Java Help  
public class WordCount {  
    public static class Map extends MapReduceBase implements 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, OutputCollector<Text, IntWritable> output, Reporter reporter) throws IOException {  
            String line = value.toString();  
            StringTokenizer tokenizer = new StringTokenizer(line);  
            while (tokenizer.hasMoreTokens()) {  
                word.set(tokenizer.nextToken());  
                output.collect(word, one);  
            }  
        }  
        public static class Reduce extends MapReduceBase implements Reducer<Text, IntWritable, Text, IntWritable> {  
            public void reduce(Text key, Iterator<IntWritable> values, OutputCollector<Text, IntWritable> output, Reporter reporter) throws IOException {  
                int sum = 0;  
                while (values.hasNext()) { sum += values.next().get(); }  
                output.collect(key, new IntWritable(sum));  
            }  
        }  
        public static void main(String[] args) throws Exception {  
            JobConf conf = new JobConf(WordCount.class);  
            conf.setJobName("wordcount");  
            conf.setOutputKeyClass(Text.class);  
            conf.setOutputValueClass(IntWritable.class);  
            conf.setMapperClass(Map.class);  
            conf.setCombinerClass(Reduce.class);  
            conf.setReducerClass(Reduce.class);  
            conf.setInputFormat(TextInputFormat.class);  
            conf.setOutputFormat(TextOutputFormat.class);  
            FileInputFormat.setInputPaths(conf, new Path(args[0]));  
            FileOutputFormat.setOutputPath(conf, new Path(args[1]));  
            JobClient.runJob(conf);  
        }  
    }  
}
```

Map function

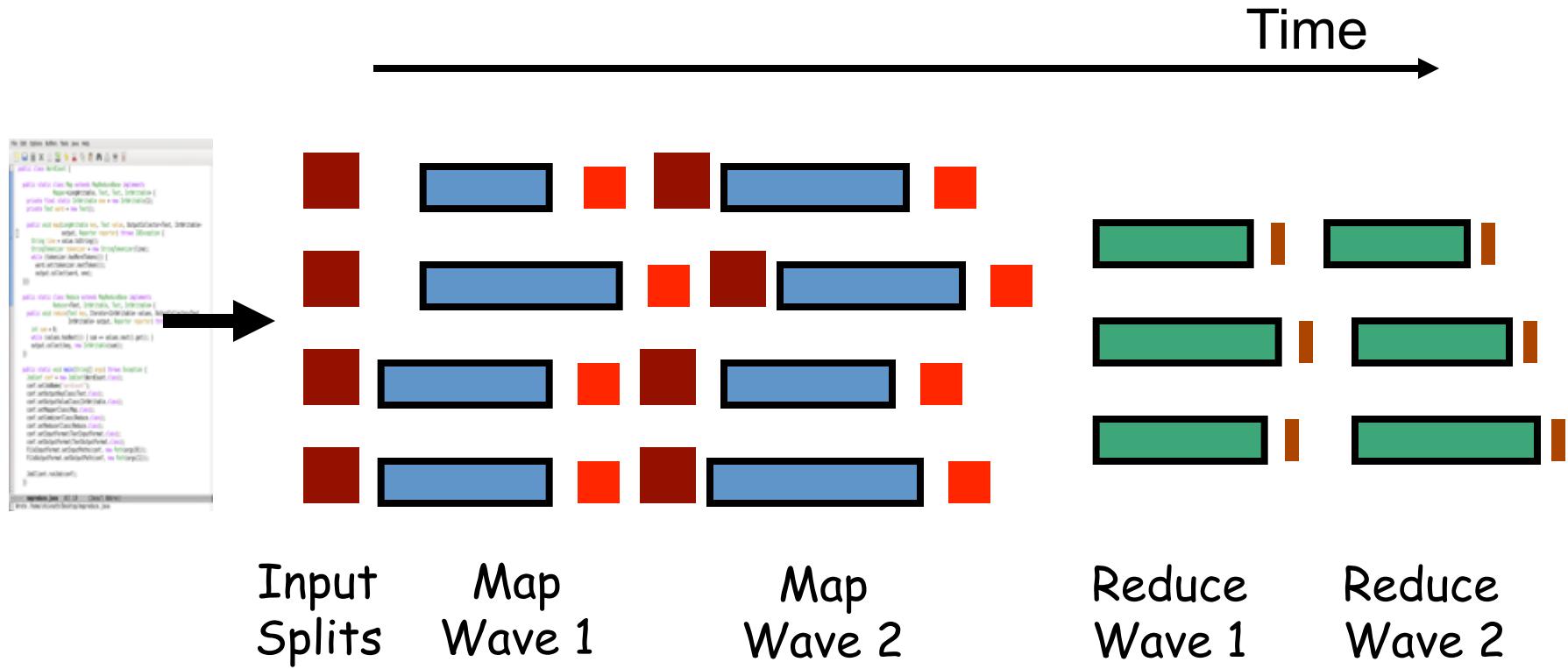
Reduce function

Run this program as a MapReduce job

Copyright © Sixth Sense Advisors Inc.

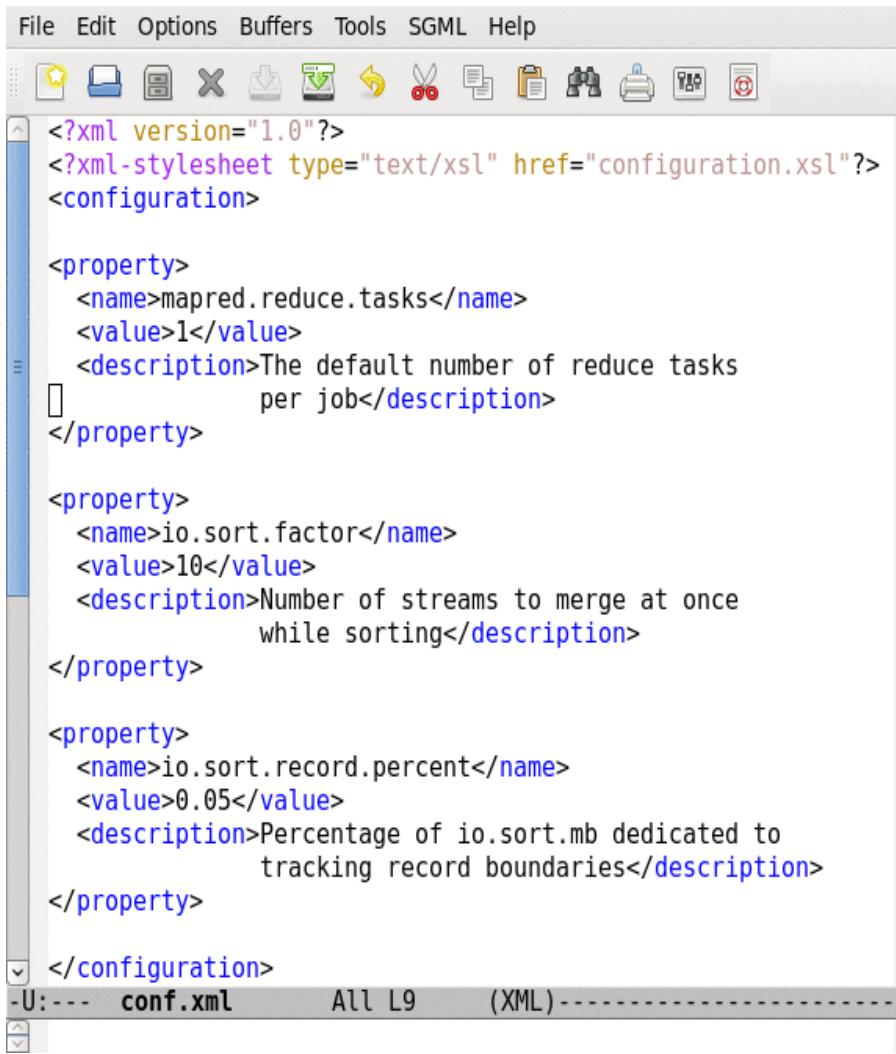
55

# Lifecycle of a MapReduce Job



How are the number of splits, number of map and reduce tasks, memory allocation to tasks, etc., determined?

# Job Configuration Parameters



The screenshot shows a text editor window with a menu bar (File, Edit, Options, Buffers, Tools, SGML, Help) and a toolbar with various icons. The main area displays an XML configuration file named 'conf.xml'. The XML code defines several properties for Hadoop, including 'mapred.reduce.tasks' (set to 1), 'io.sort.factor' (set to 10), and 'io.sort.record.percent' (set to 0.05). Each property includes a name, value, and a detailed description. The status bar at the bottom shows the file path as '-U:--- conf.xml' and the line count as 'All L9'.

```
<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<configuration>

<property>
  <name>mapred.reduce.tasks</name>
  <value>1</value>
  <description>The default number of reduce tasks
  per job</description>
</property>

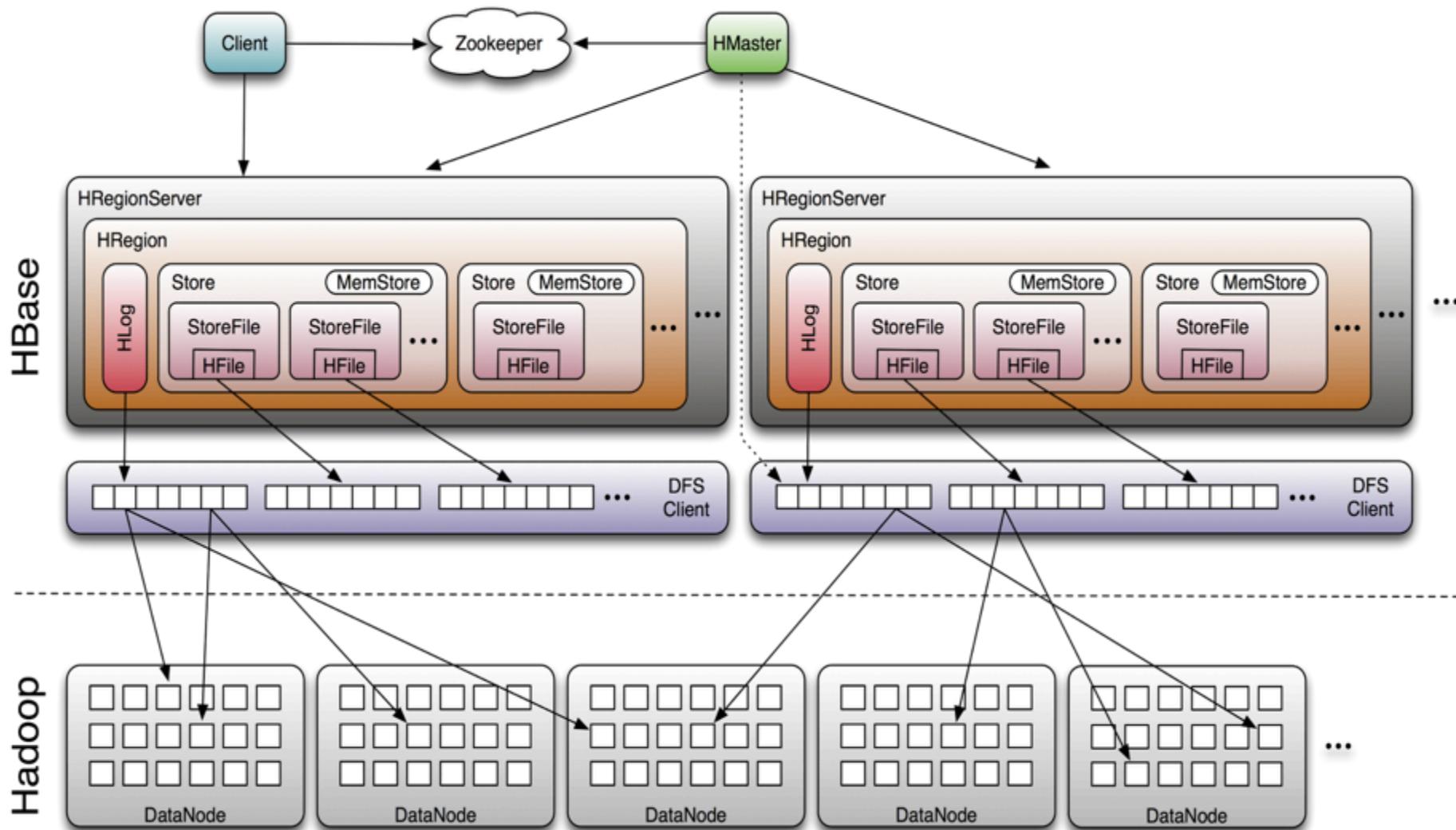
<property>
  <name>io.sort.factor</name>
  <value>10</value>
  <description>Number of streams to merge at once
  while sorting</description>
</property>

<property>
  <name>io.sort.record.percent</name>
  <value>0.05</value>
  <description>Percentage of io.sort.mb dedicated to
  tracking record boundaries</description>
</property>

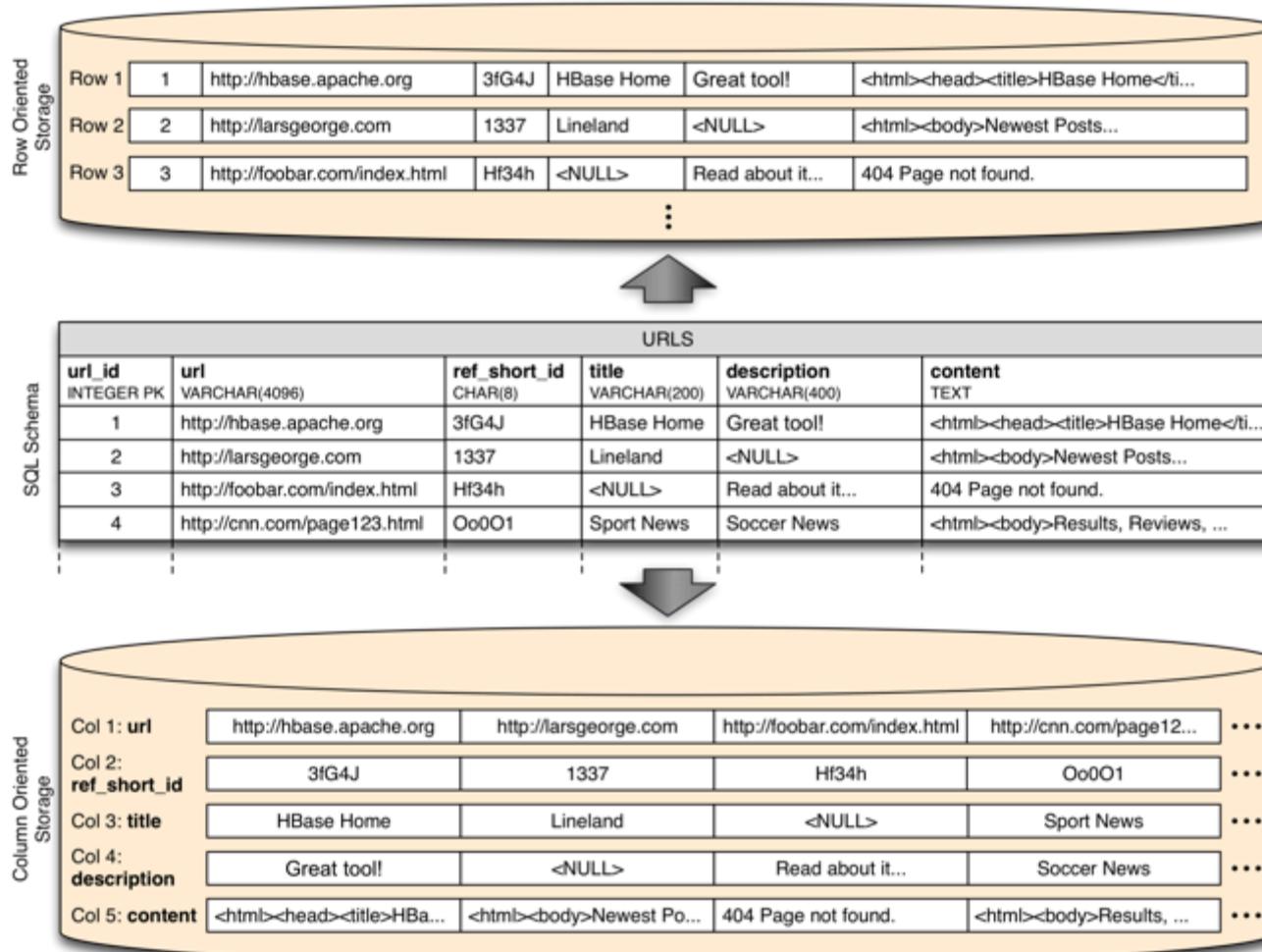
</configuration>
```

- 190+ parameters in Hadoop
- Set manually or defaults are used

# HBASE



# Column Oriented Storage



# Why HBase?

- Datasets are reaching Petabytes
- Traditional databases are expensive to scale and difficult to distribute
- Commodity hardware is cheap and powerful
- Need for random access and batch processing (which Hadoop does not offer)

# Tables

- Table is split into roughly equal sized “regions”
- Each region is a contiguous range of keys
- Regions split as they grow, thus dynamically adjusting to your data set

# Table (cont.)

- Tables are sorted by Row
- Table schema defines column families
  - Families consist of any number of columns
  - Columns consist of any number of versions
  - Everything except table name is byte[]

(Table, Row, Family:Column, Timestamp) -> Value

# Table (cont.)

- As a data structure

```
SortedMap(  
    RowKey, List(  
        SortedMap(  
            Column, List(  
                Value, Timestamp  
            )  
        )  
    )  
)
```

# HBase Open Source Stack

- ZooKeeper : Small Data Coordination Service
- HBase : Database Storage Engine
- HDFS : Distributed File system
- Hadoop : Asynchronous Map-Reduce Jobs

# Server Architecture

- Similar to HDFS
  - Master == Namenode
  - Regionserver == Datanode
- Often run these alongside each other!
- Difference: HBase stores state in HDFS
- HDFS provides robust data storage across machines, insulating against failure
- Master and Regionserver fairly stateless and machine independent

# Region Assignment

- Each region from every table is assigned to a Regionserver
- Master Duties:
  - Responsible for assignment and handling regionserver problems (if any!)
  - When machines fail, move regions
  - When regions split, move regions to balance
  - Could move regions to respond to load
  - Can run multiple backup masters

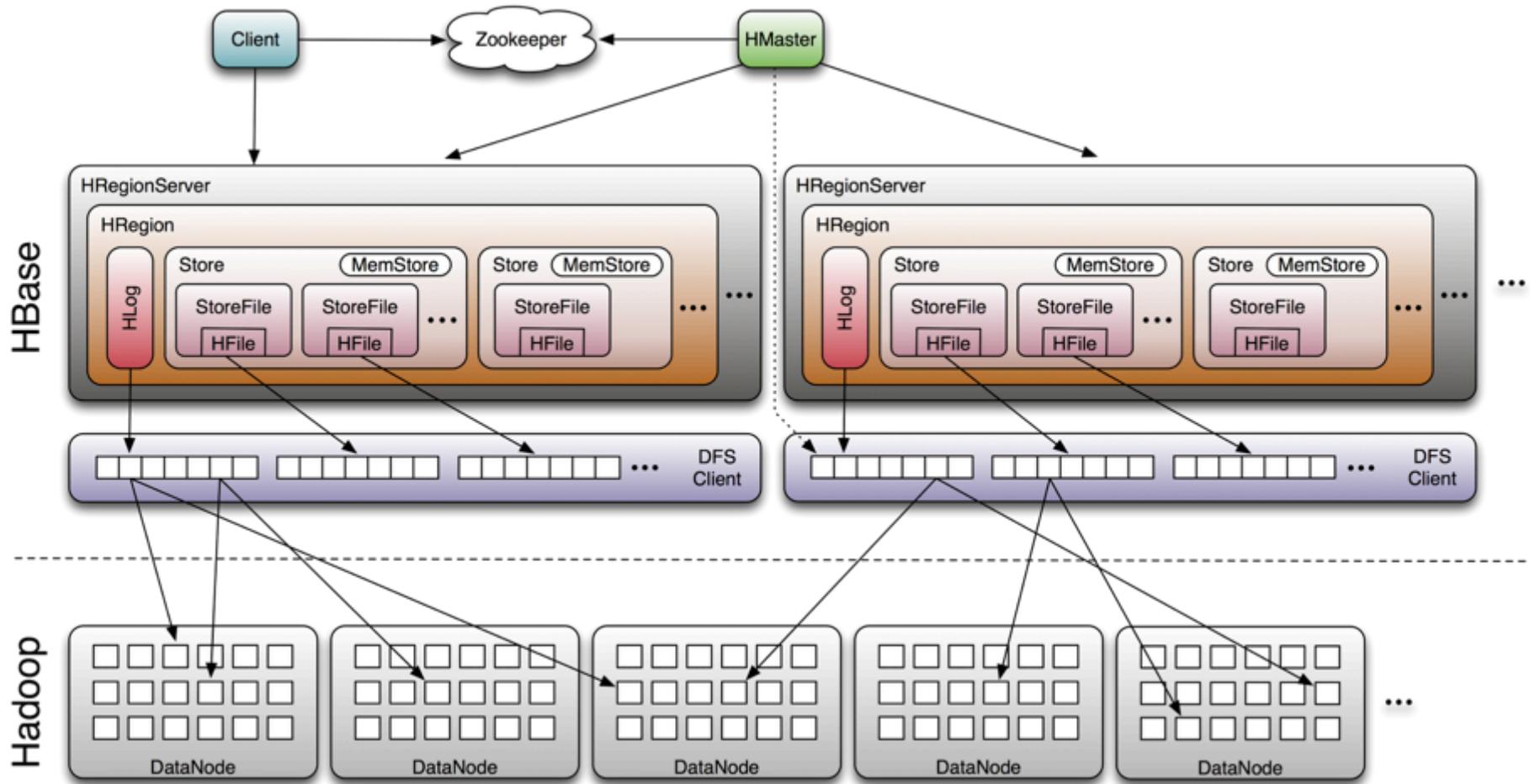
# Master

- The master does NOT
  - Handle any write request (not a DB master!)
  - Handle location finding requests
  - Not involved in the read/write path
  - Generally does very little most of the time

# Distributed Coordination

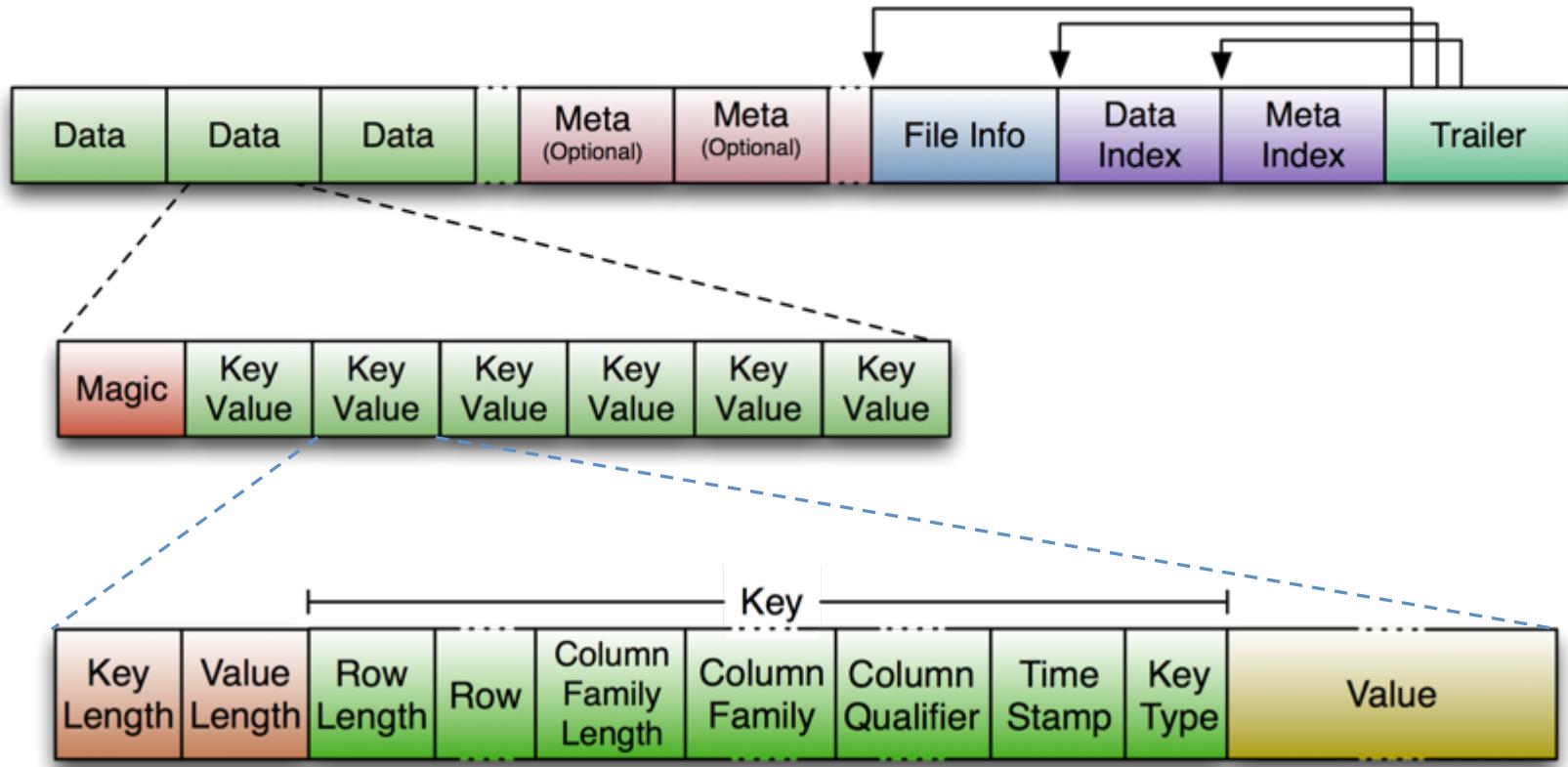
- Zookeeper is used to manage master election and server availability
- Set up as a cluster, provides distributed coordination primitives
- An excellent tool for building cluster management systems

# HBase Architecture

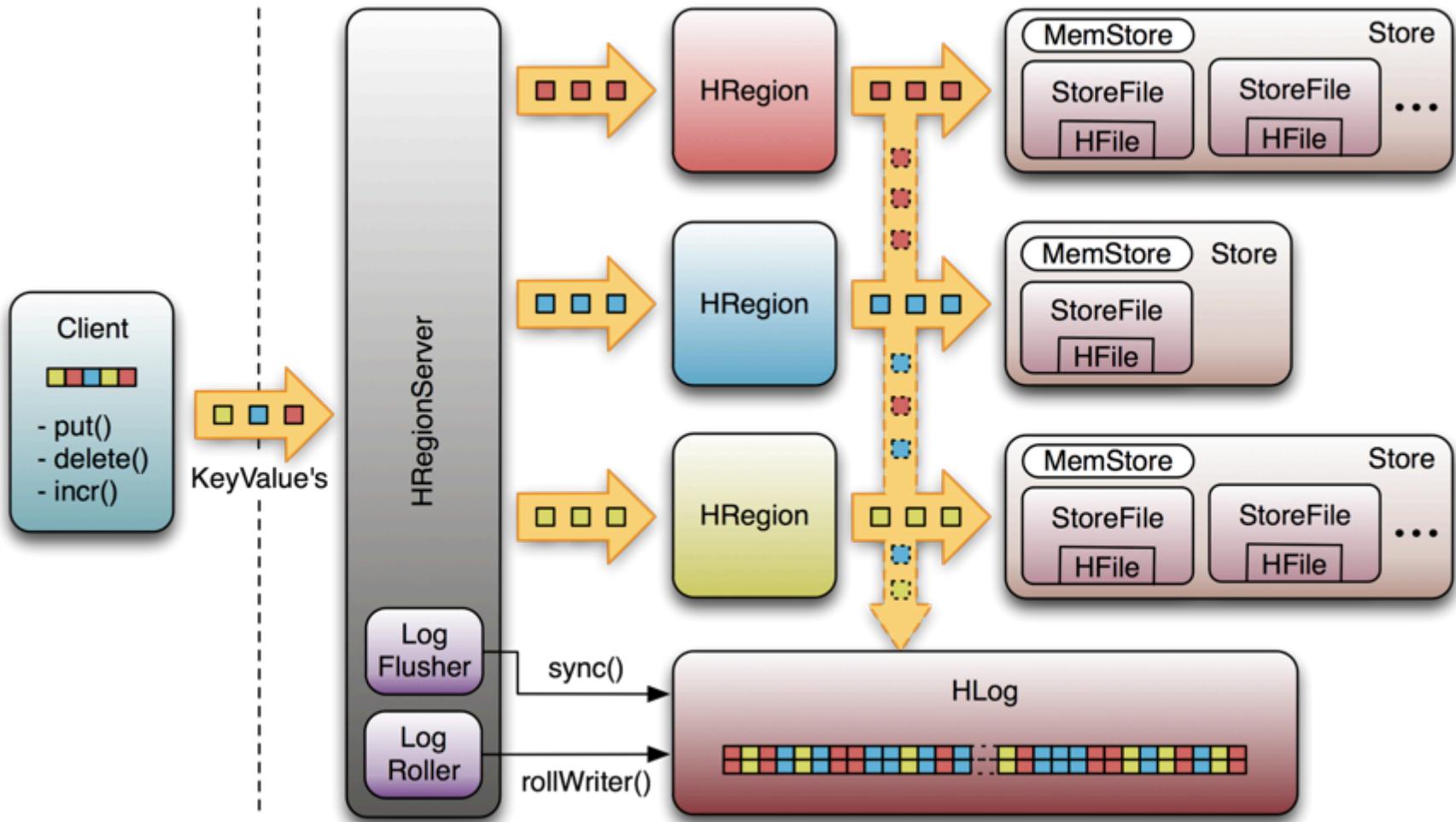


<http://www.larsgeorge.com/2009/10/hbase-architecture-101-storage.html>

# How data actually stored

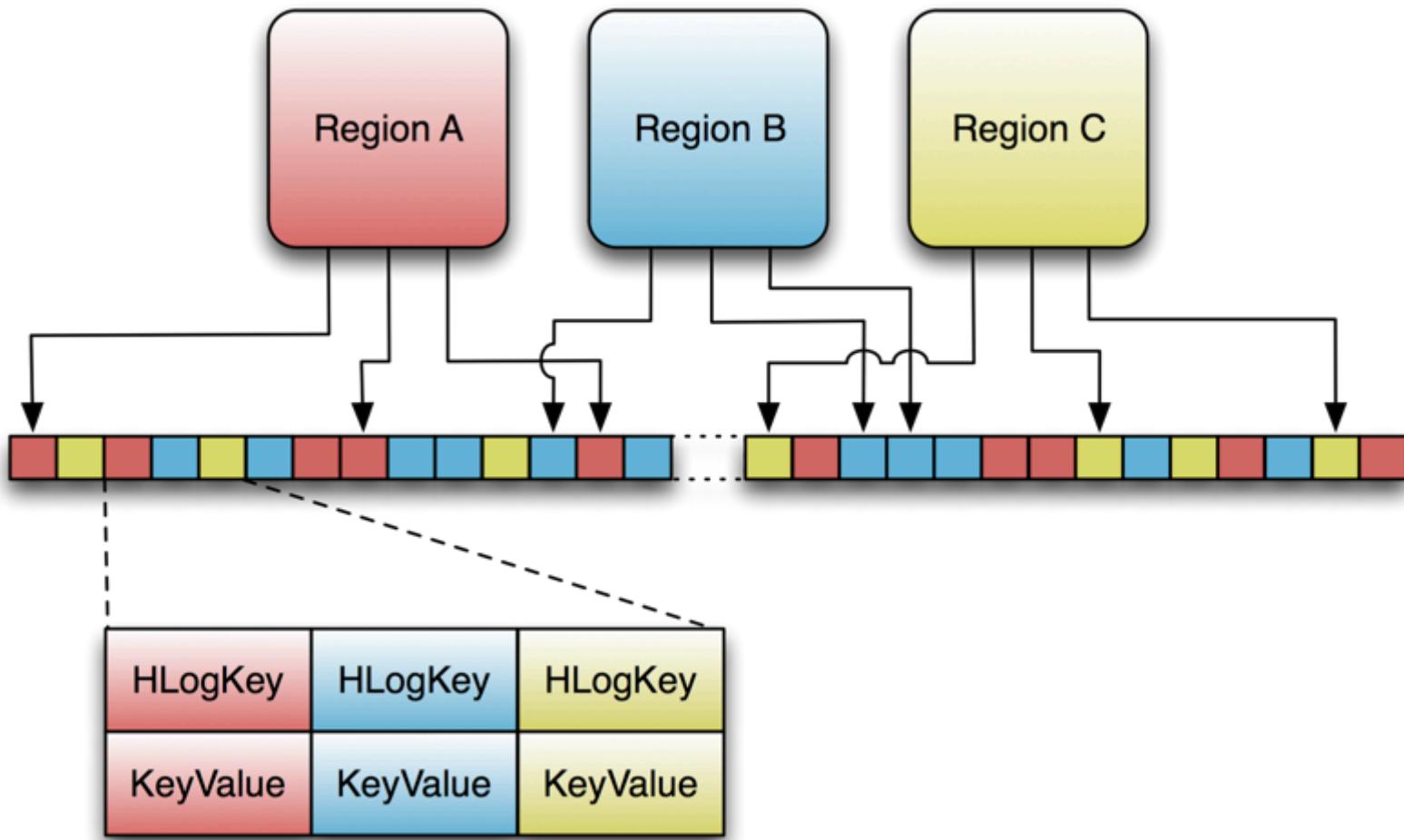


# Write-ahead-Log

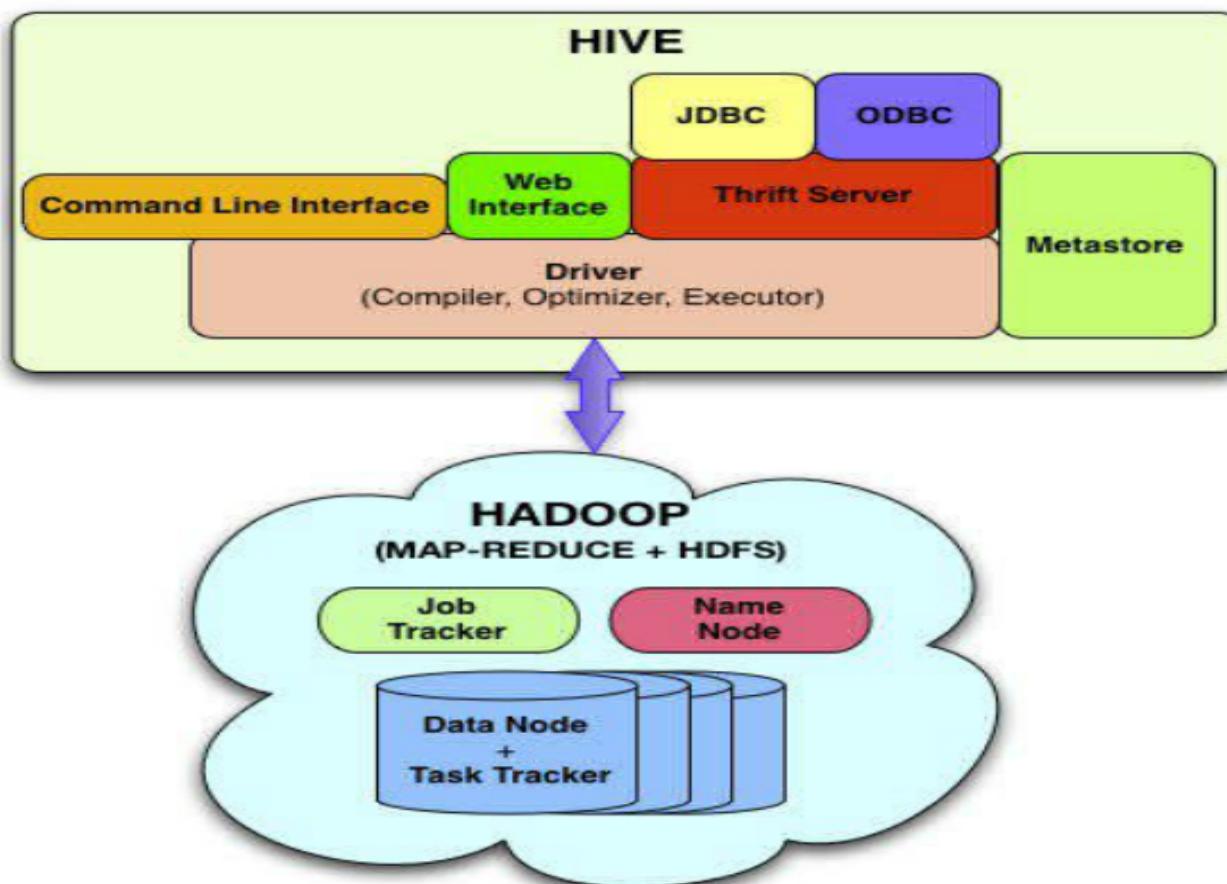


<http://www.larsgeorge.com/2010/01/hbase-architecture-101-write-ahead-log.html>

# HLog

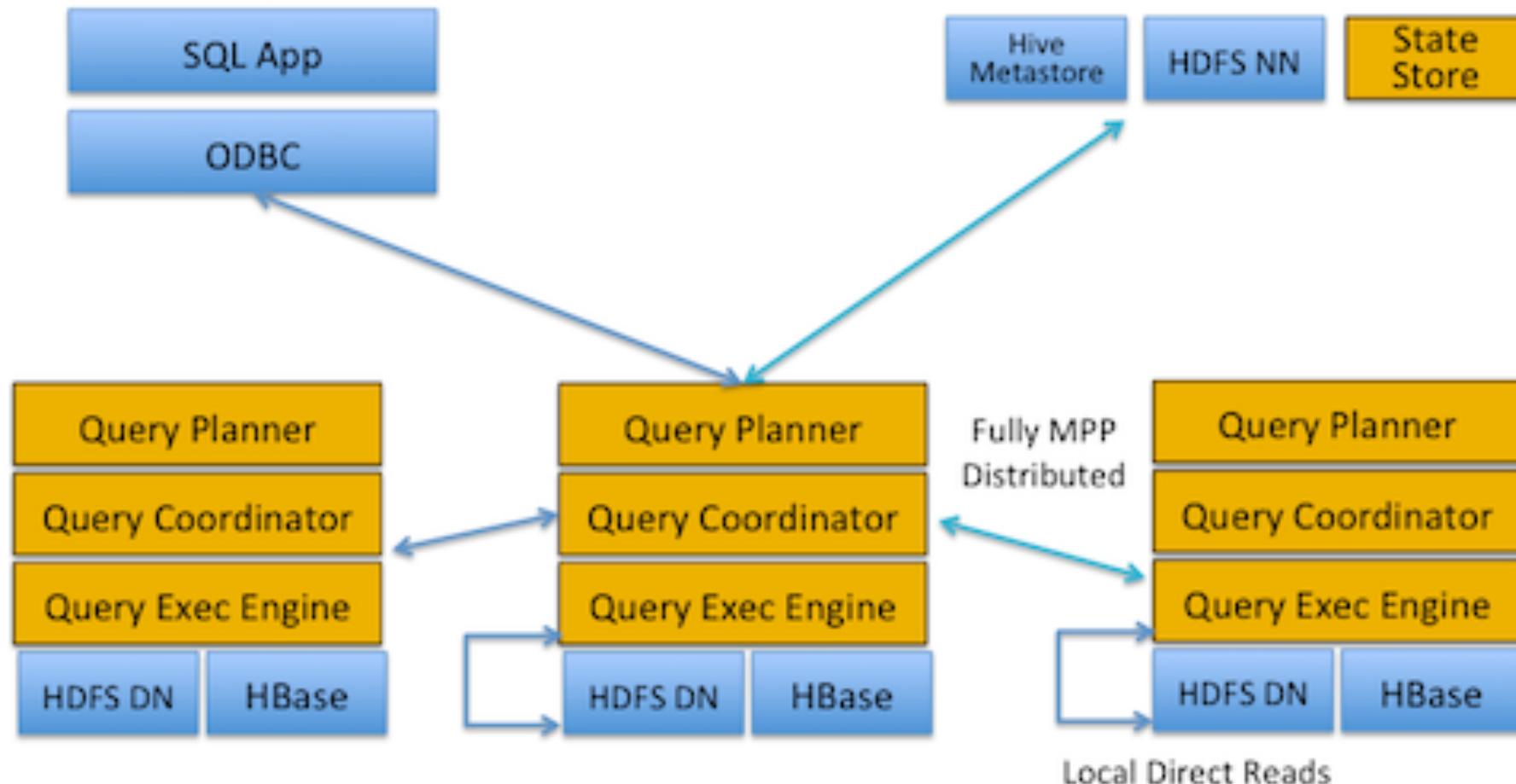


# Hive



# Impala

Common Hive SQL and interface



# What is PIG?

- Pig is a platform for analyzing large data sets that consists of a high-level language for expressing data analysis programs
- Pig generates and compiles a Map/Reduce program(s) on the fly.

# Why PIG?

- **Ease of programming** - It is trivial to achieve parallel execution of simple, "embarrassingly parallel" data analysis tasks. Complex tasks comprised of multiple interrelated data transformations are explicitly encoded as data flow sequences, making them easy to write, understand, and maintain.

# Architecture

## Pig Latin

```
A = LOAD 'myfile'  
      AS (x, y, z);  
B = FILTER A by x > 0;  
C = GROUP B BY x;  
D = FOREACH A GENERATE  
      x, COUNT(B);  
STORE D INTO 'output';
```



## pig.jar:

- parses
- checks
- optimizes
- plans execution
- submits jar to Hadoop
- monitors job progress

Execution Plan  
Map:  
Filter

Reduce:  
Count



# File Formats

- PigStorage
- Custom Load / Store Functions

# Installing PIG

- Download / Unpack tarball ([pig.apache.org](http://pig.apache.org))
- Install RPM / DEB package ([cloudera.com](http://cloudera.com))

# Running Pig

- **Grunt Shell:** Enter Pig commands manually using Pig's interactive shell, Grunt.
- **Script File:** Place Pig commands in a script file and run the script.
- **Embedded Program:** Embed Pig commands in a host language and run the program.

# Run Modes

- Local Mode: To run Pig in local mode, you need access to a single machine.
- Hadoop (mapreduce) Mode: To run Pig in hadoop (mapreduce) mode, you need access to a Hadoop cluster and HDFS installation.

# Sample PIG script

```
A = load 'passwd' using PigStorage(':');  
B = foreach A generate $0 as id;  
store B into 'id.out';
```

# Sample Script With Schema

- A = LOAD 'student\_data' AS (name: chararray,  
age: int, gpa: float);
- B = FOREACH A GENERATE  
myudfs.UPPER(name);

# Eval Functions

- AVG
- CONCAT
- Example
- COUNT
- COUNT\_STAR
- DIFF
- IsEmpty
- MAX
- MIN
- SIZE
- SUM
- TOKENIZE

# Math Functions

- # Math Functions
- ABS
- ACOS
- ASIN
- ATAN
- CBRT
- CEIL
- COSH
- COS
- EXP
- FLOOR
- LOG
- LOG10
- RANDOM
- ROUND
- SIN
- SINH
- SQRT
- TAN
- TANH

# Pig Types

Pig Type	Java Class
bytearray	DataByteArray
chararray	String
int	Integer
long	Long
float	Float
double	Double
tuple	Tuple
bag	DataBag
map	Map<Object, Object>

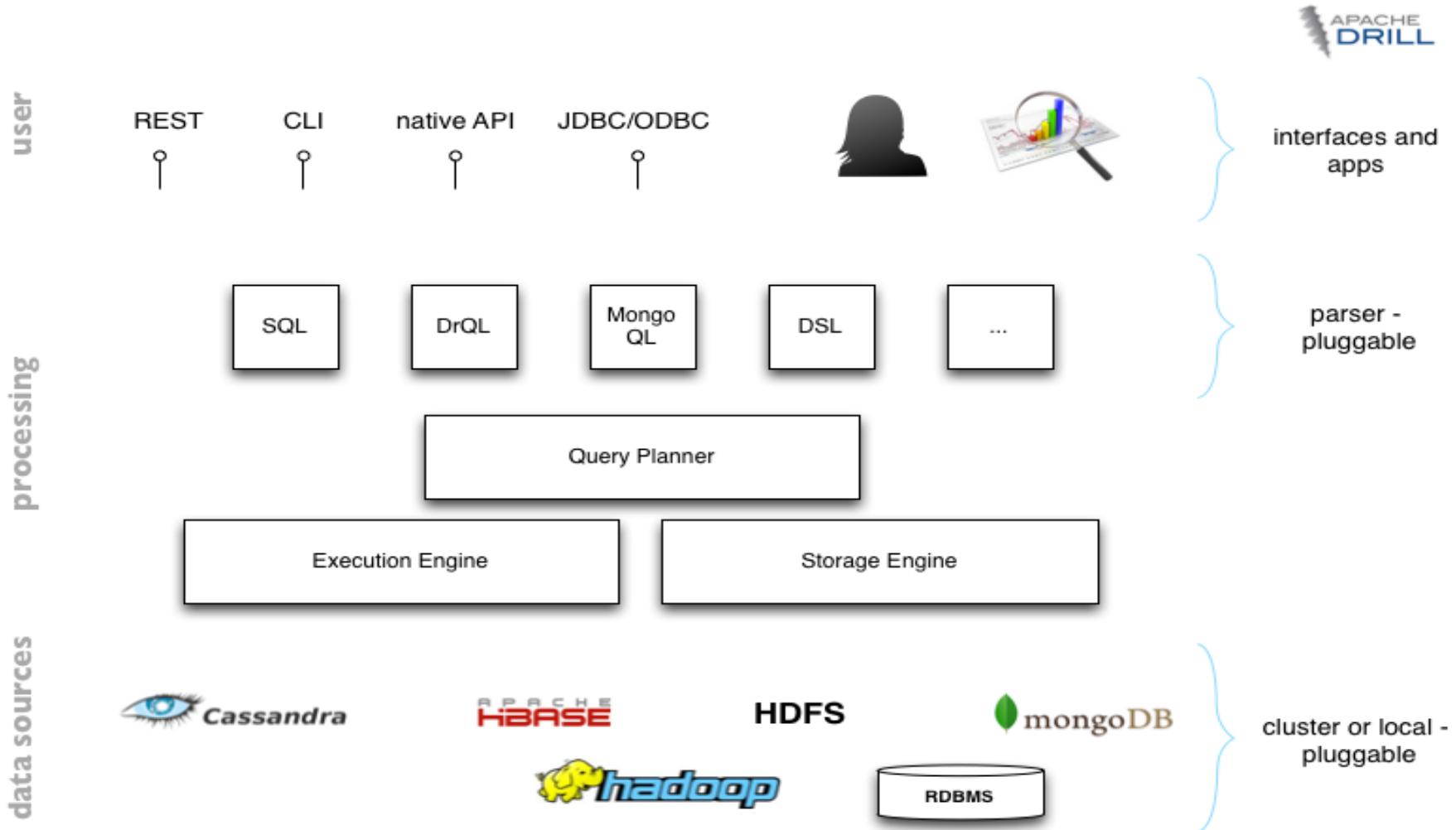
# What is PIG UDF?

- UDF - User Defined Function
- Types of UDF's:
  - Eval Functions (extends `EvalFunc<String>`)
  - Aggregate Functions (extends `EvalFunc<Long>` implements `Algebraic`)
  - Filter Functions (extends `FilterFunc`)
- `UDFContext`
  - Allows UDFs to get access to the `JobConf` object
  - Allows UDFs to pass configuration information between instantiations of the UDF on the front and backends.

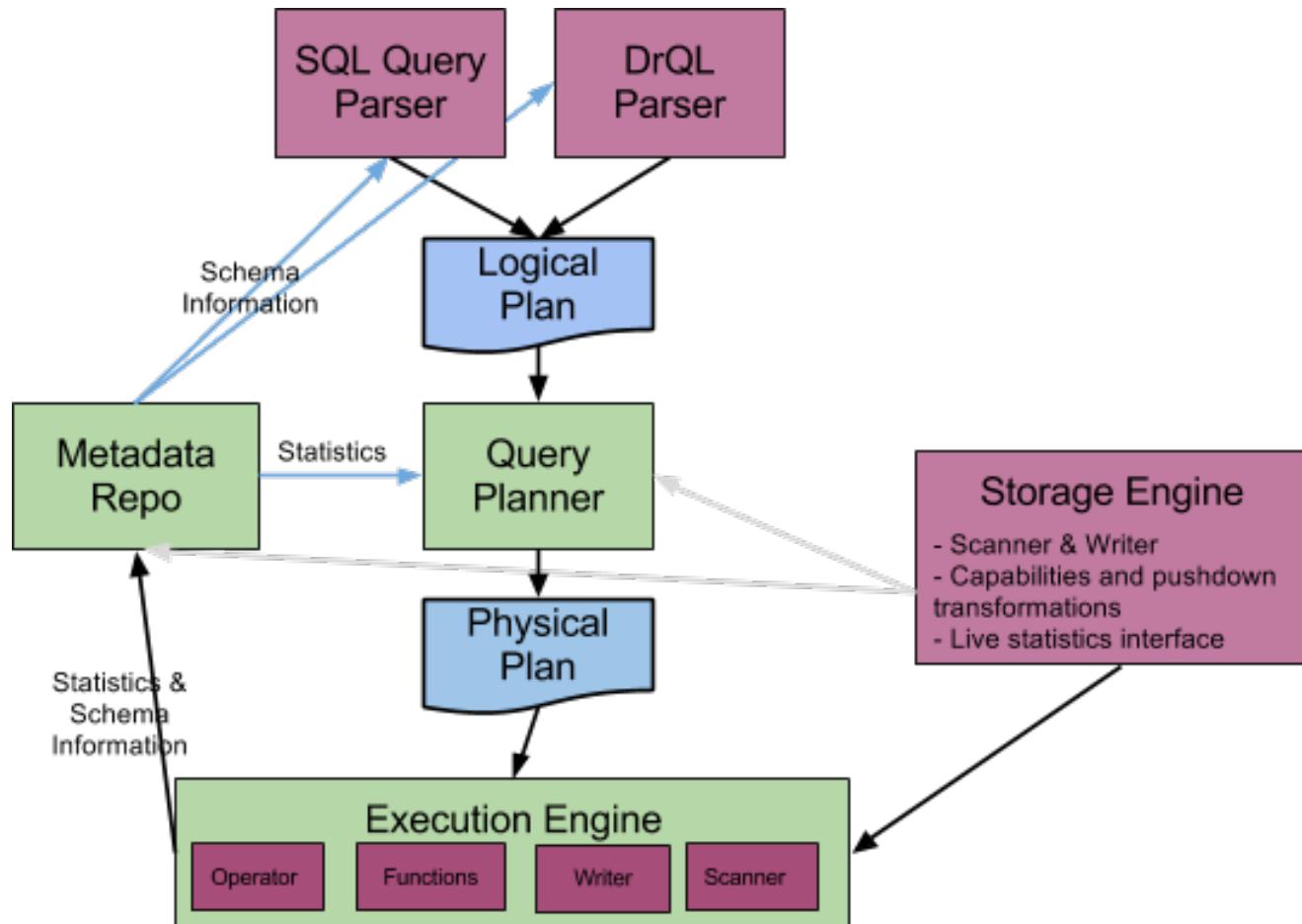
# DRILL

- Self Service Interactive Analysis
- Designed for Big Data
- Fast low latency queries
- Complements existing tools like Pig, Hive, Map Reduce
- Distributed
  - Query planning
  - Scheduling
  - Execution
- User Interfaces
  - REST
  - CLI ( command line )
  - Native API
  - JDBC / ODBC
- Processing Interfaces
  - SQL
  - DrQL ( Drill query language )
  - MongoQL
  - DSL
- Data Interfaces
  - Cassandra
  - Hbase
  - Hadoop HDFS
  - MongoDB
  - RDBMS

# Architecture



# Processing



# What is Spark?

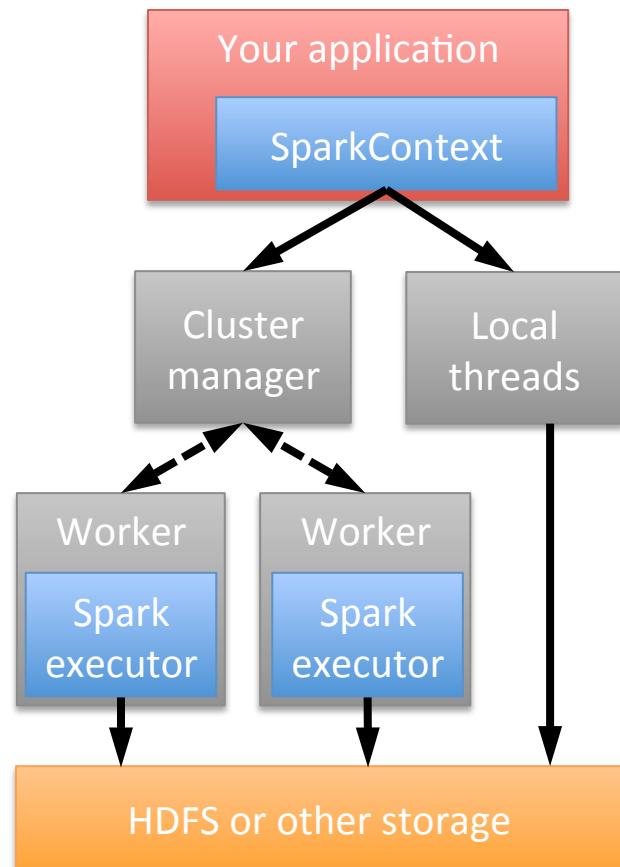
- Not a modified version of Hadoop
- Separate, fast, MapReduce-like engine
  - In-memory data storage for very fast iterative queries
  - General execution graphs and powerful optimizations
  - Up to 40x faster than Hadoop
- Compatible with Hadoop's storage APIs
  - Can read/write to any Hadoop-supported system, including HDFS, HBase, SequenceFiles, etc

# What is Shark?

- Port of Apache Hive to run on Spark
- Compatible with existing Hive data, metastores, and queries (HiveQL, UDFs, etc)
- Similar speedups of up to 40x

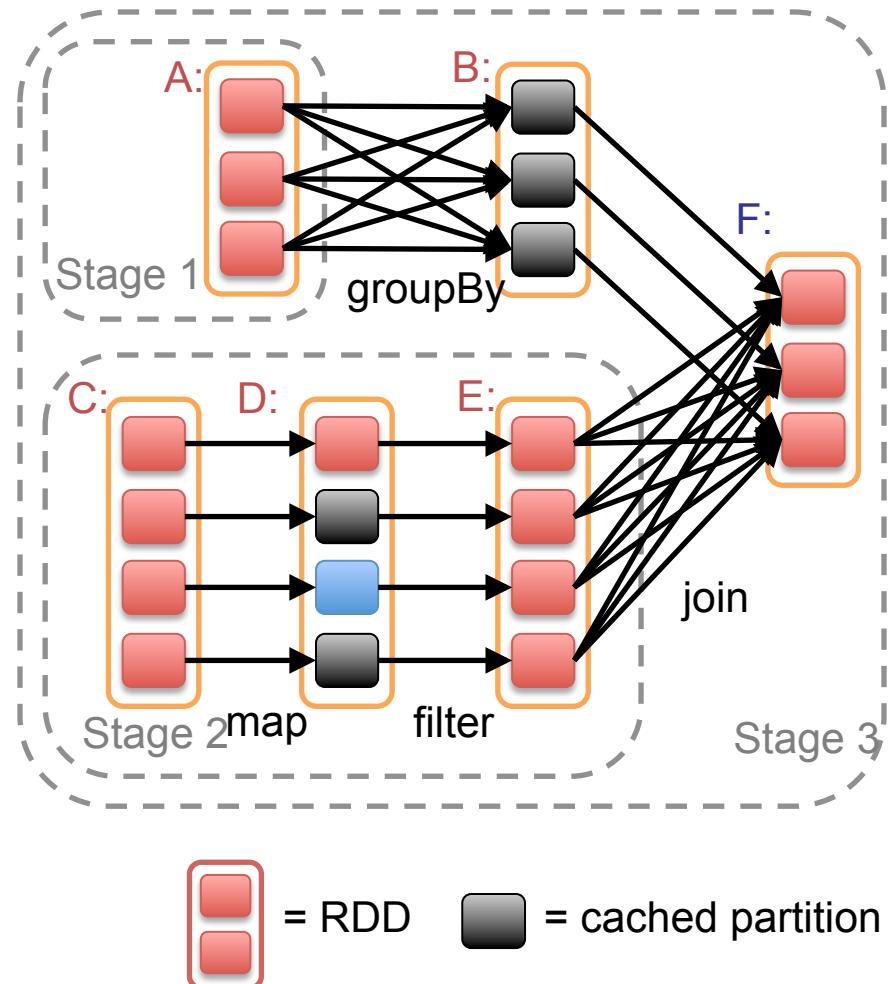
# Software Components

- Spark runs as a library in your program (one instance per app)
- Runs tasks locally or on a cluster
  - Standalone deploy cluster, Mesos or YARN
- Accesses storage via Hadoop InputFormat API
  - Can use HBase, HDFS, S3, ...



# Task Scheduler

- Supports general task graphs
- Pipelines functions where possible
- Cache-aware data reuse & locality
- Partitioning-aware to avoid shuffles



# Hadoop Compatibility

- Spark can read/write to any storage system / format that has a plugin for Hadoop!
  - Examples: HDFS, S3, HBase, Cassandra, Avro, SequenceFile
  - Reuses Hadoop's InputFormat and OutputFormat APIs
- APIs like `SparkContext.textFile` support filesystems, while `SparkContext.hadoopRDD` allows passing any Hadoop JobConf to configure an input source

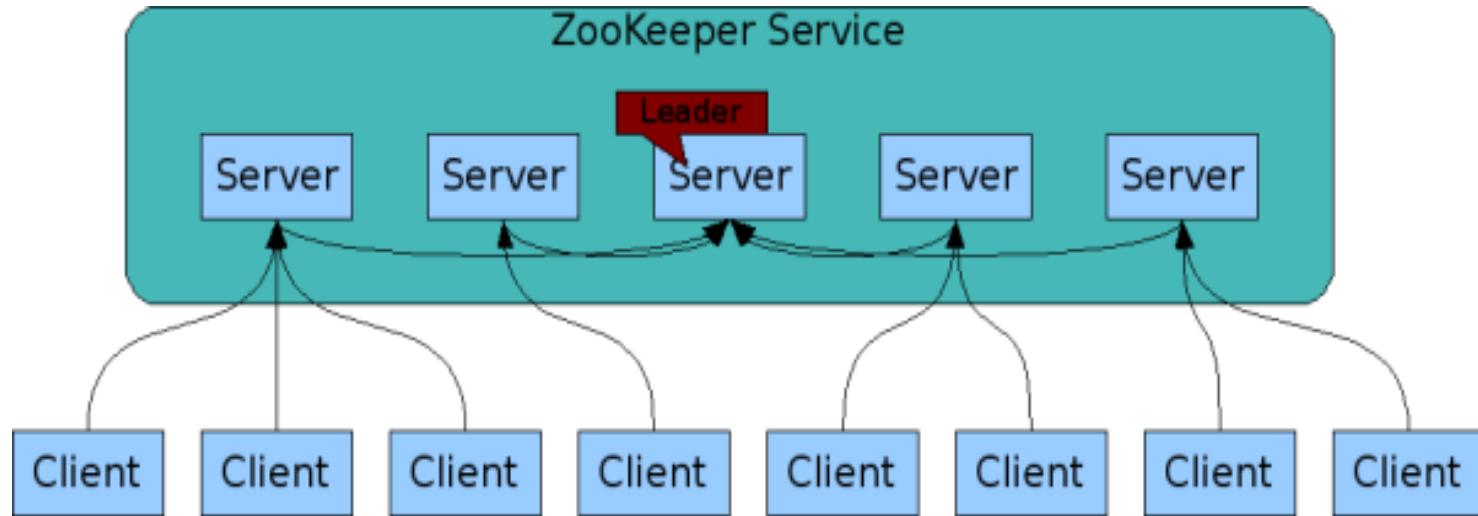
# ZooKeeper?

- An open source, high-performance coordination service for distributed applications.
- Exposes common services in simple interface:
  - naming
  - configuration management
  - locks & synchronization
  - group services
    - ... *developers don't have to write them from scratch*
- Build your own on it for specific needs.

# ZooKeeper Use Cases

- Configuration Management
  - Cluster member nodes bootstrapping configuration from a centralized source in unattended way
  - Easier, simpler deployment/provisioning
- Distributed Cluster Management
  - Node join / leave
  - Node statuses in real time
- Naming service – e.g. DNS
- Distributed synchronization - locks, barriers, queues
- Leader election in a distributed system.
- Centralized and highly reliable (simple) data registry

# The ZooKeeper Service



- ZooKeeper Service is replicated over a set of machines
- All machines store a copy of the data (in memory)
- A leader is elected on service startup
- Clients only connect to a single ZooKeeper server & maintains a TCP connection.
- Client can read from any Zookeeper server, writes go through the leader & needs majority consensus.

Image: <https://cwiki.apache.org/confluence/display/ZOOKEEPER/ProjectDescription>

# The ZooKeeper Data Model

- ZooKeeper has a hierachal name space.
- Each node in the namespace is called as a *ZNode*.
- Every ZNode has data (given as byte[]) and can optionally have children.

```
parent : "foo"
  |-- child1 : "bar"
  |-- child2 : "spam"
  `-- child3 : "eggs"
      `-- grandchild1 : "42"
```

- ZNode paths:
  - canonical, absolute, slash-separated
  - no relative references.
  - names can have Unicode characters

# ZNodes

- Maintain a stat structure with version numbers for data changes, ACL changes and timestamps.
- Version numbers increases with changes
- Data is read and written in its entirety

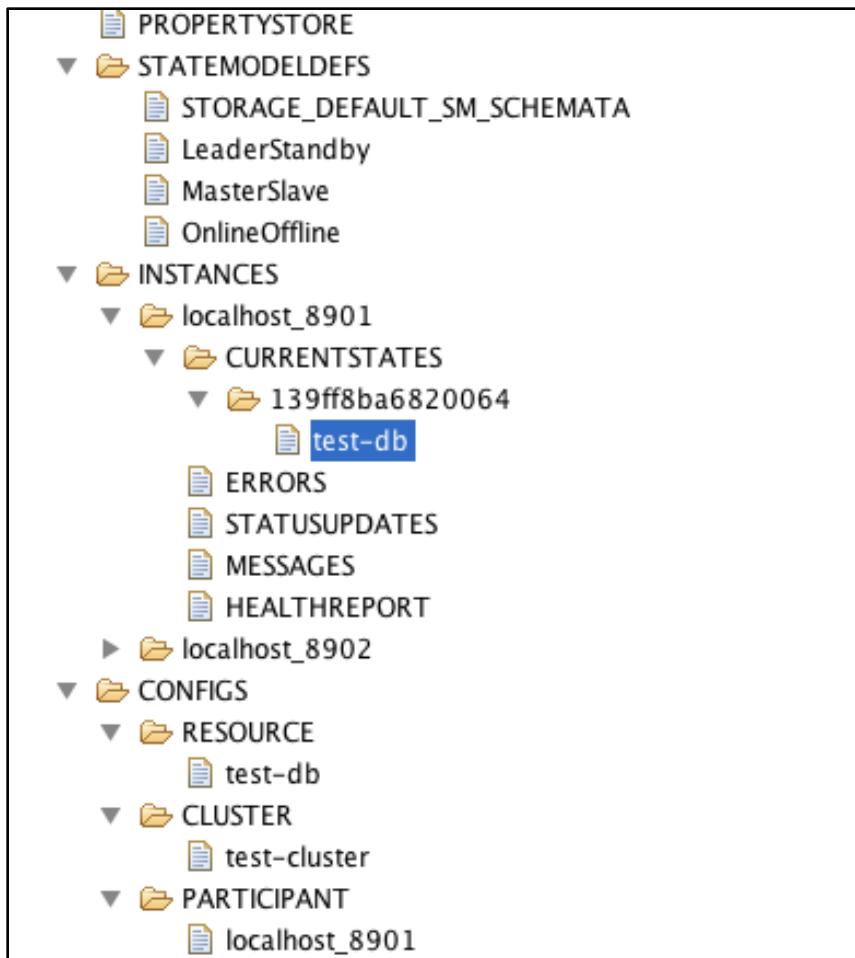
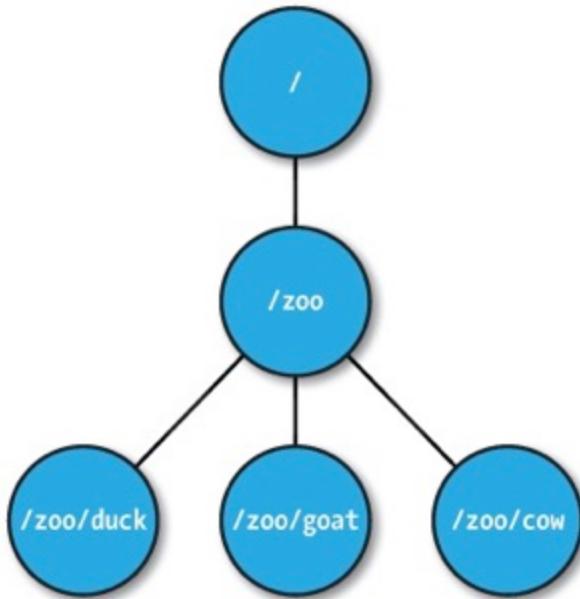


Image: <http://helix.incubator.apache.org/Architecture.html>

# ZNode Operations

Operation	Type
create	Write
delete	Write
exists	Read
getChildren	Read
getData	Read
setData	Write
getACL	Read
setACL	Write
sync	Read

***ZNodes are the main entity that a programmer access.***

# ZNode Watches

- Clients can set watches on znodes:
  - NodeChildrenChanged
  - NodeCreated
  - NodeDataChanged
  - NodeDeleted
- Changes to a znode trigger the watch and ZooKeeper sends the client a notification.
- Watches are one time triggers.
- Watches are always ordered.
- Client sees watched event before new znode data.
- Client should handle cases of latency between getting the event and sending a new request to get a watch.

# API Synchronicity

- API methods are sync as well as async

- Sync:

```
exists("/test-cluster/CONFIGS", null);
```

- Async:

```
exists("/test-cluster/CONFIGS", null, new StatCallback() {  
    @Override  
    public processResult(int rc, String path, Object ctx, Stat stat)  
    {  
        //process result when called back later  
    }  
}, null);
```

# AVRO

- Avro is a serialization framework developed within Apache's Hadoop project. It uses JSON for defining data types and protocols, and serializes data in a compact binary format. Its primary use is in Apache Hadoop, where it can provide both a serialization format for persistent data.
- Avro provides good way to convert unstructured and semi-structured data into a structured way using schemas

# Schema

- Schema Features
  - Primitive types (null, boolean, int, long, float, double, bytes, string)
  - Records
    - { "type": "record",
      - "name": "LongList",
      - [ {"name": "value", "type": "long"},
      - {"name": "description", "type": "string"}]
    - }
    - Others (Enums, Arrays, Maps, Unions, Fixed)
- Example
- Schema description:
  - {  
  "name": "User",  
  "type": "record",  
  "fields": [  
    {"name": "FirstName", "type": "string", "doc": "First Name"},  
    {"name": "LastName", "type": "string"},  
    {"name": "isActive", "type": "boolean", "default": true},  
    {"name": "Account", "type": "int", "default": 0} ]  
  – }

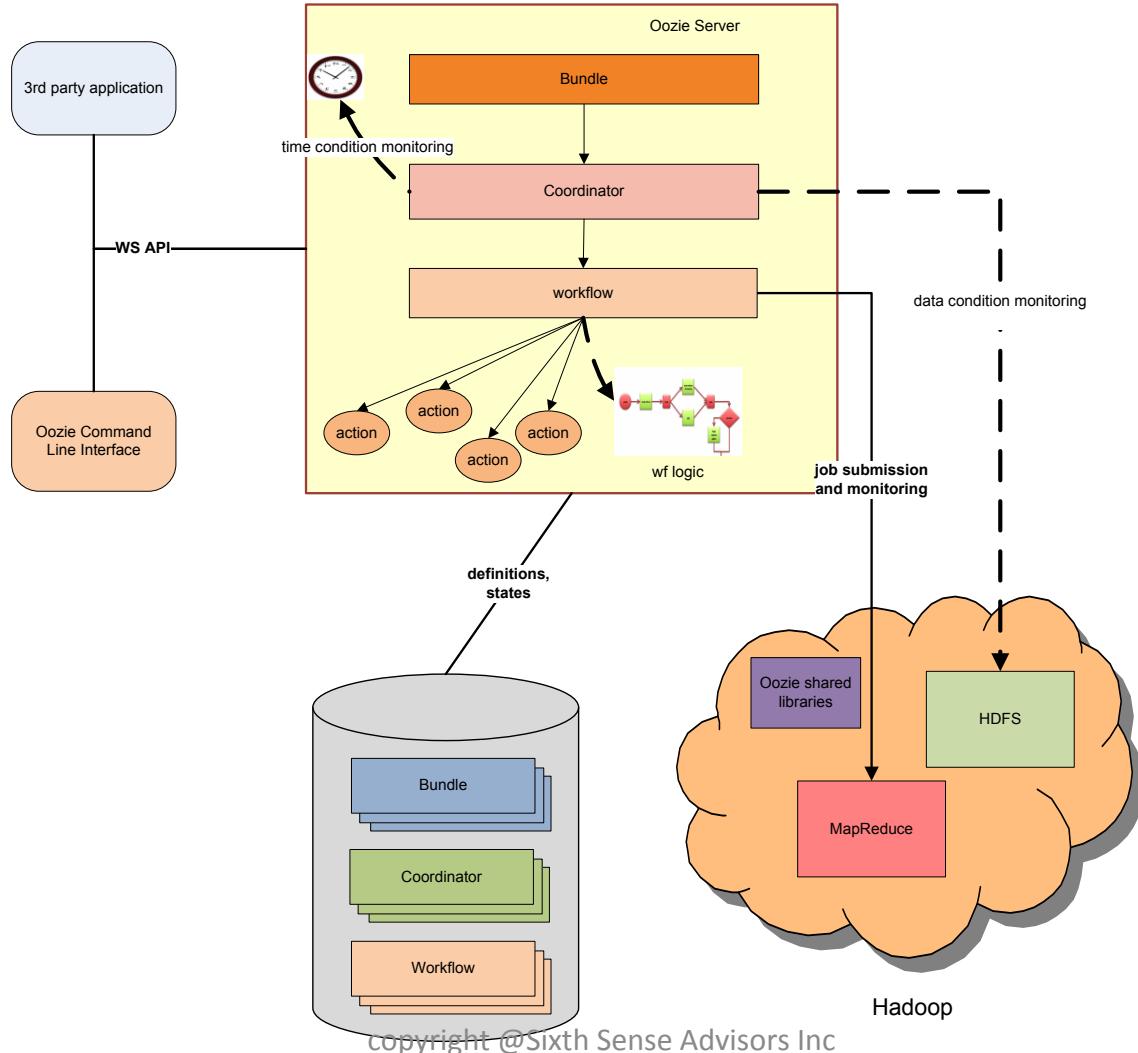
# AVRO

- Dynamic
- Expressive
- Efficient
- Specification based design
- Language implementations are fairly solid
- Serialization or RPC or both

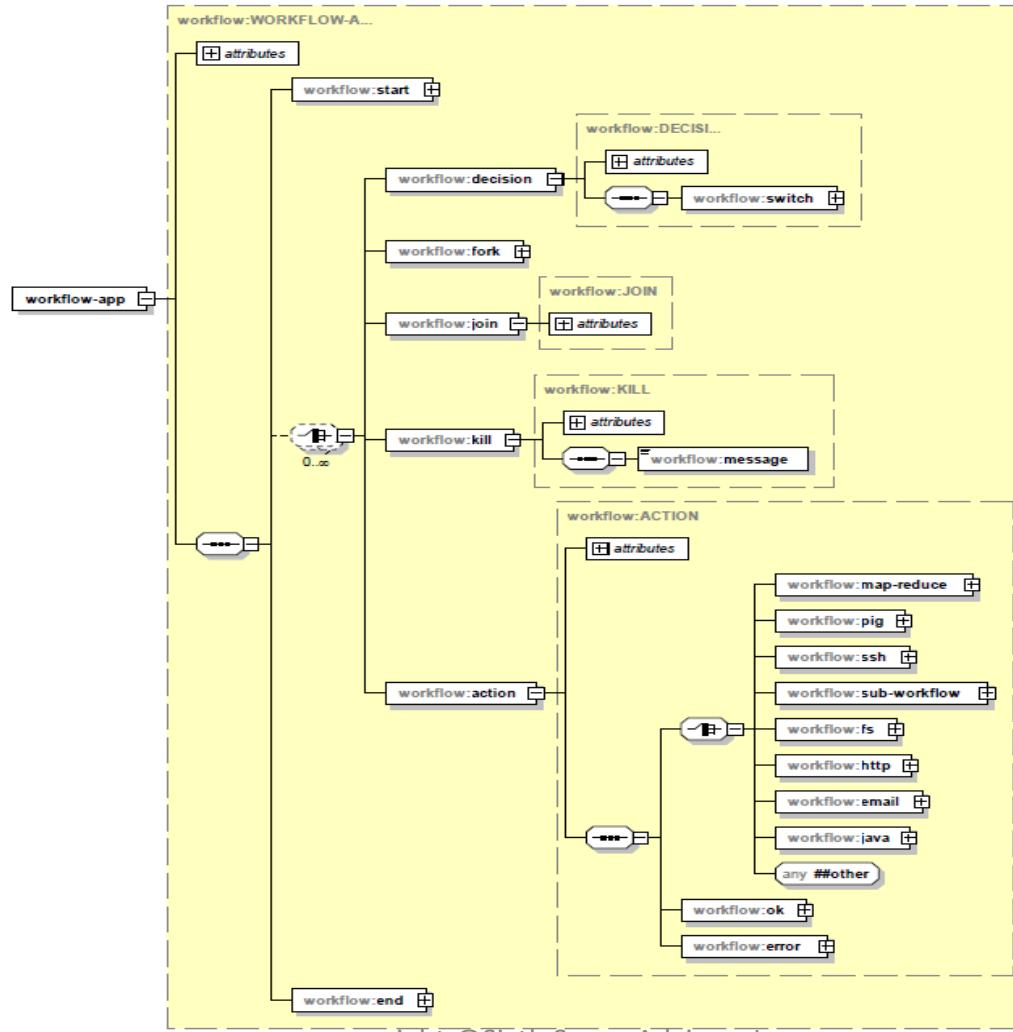
# Oozie

- Oozie is a workflow/coordination system to manage Apache Hadoop jobs.
- A single Oozie server implements all four functional Oozie components:
  - Oozie workflow
  - Oozie coordinator
  - Oozie bundle
  - Oozie SLA.

# Oozie Architecture



# Oozie workflow



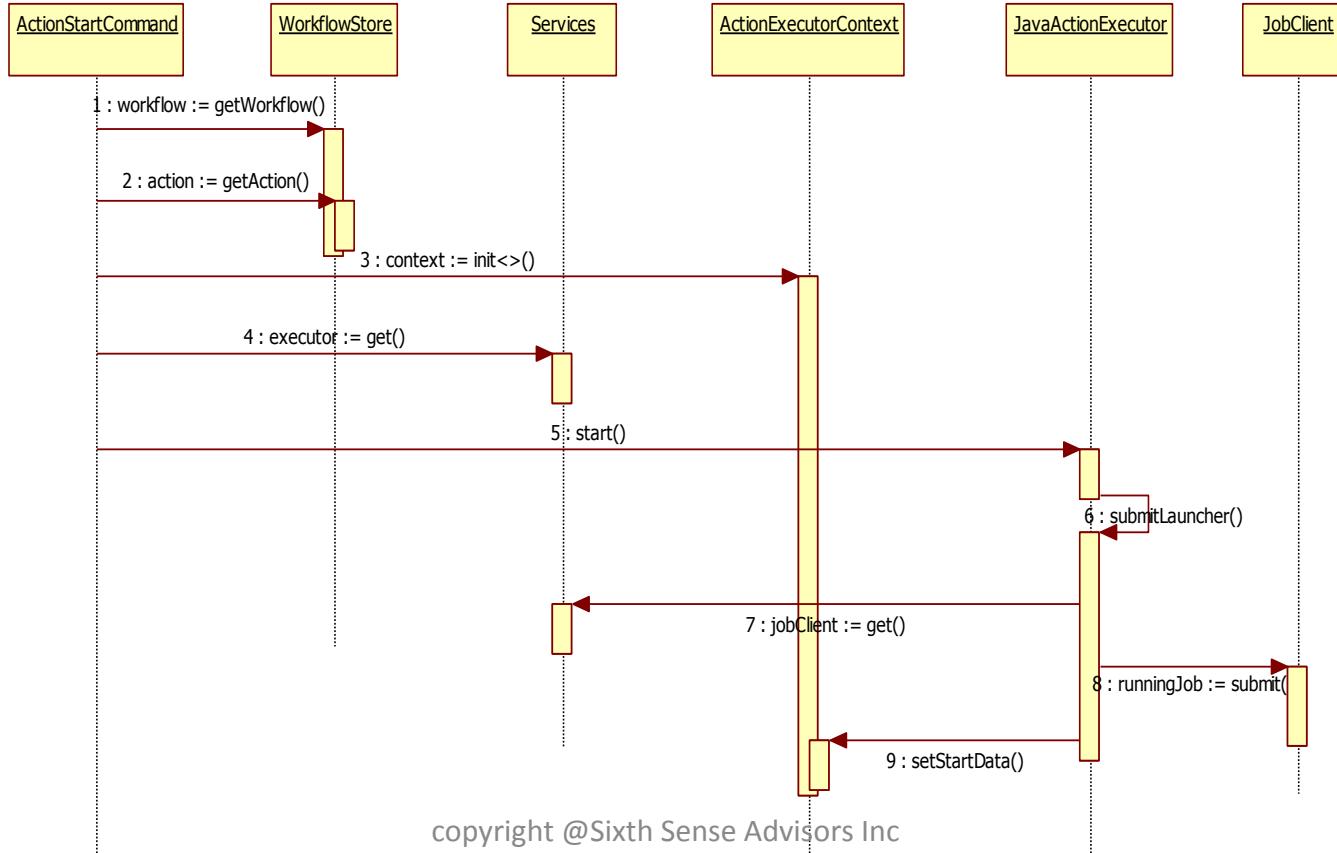
# Workflow Language

Flow-control node	XML element type	Description
Decision	workflow:DECISION	expressing “switch-case” logic
Fork	workflow:FORK	splits one path of execution into multiple concurrent paths
Join	workflow:JOIN	waits until every concurrent execution path of a previous fork node arrives to it
Kill	workflow:kill	forces a workflow job to kill (abort) itself

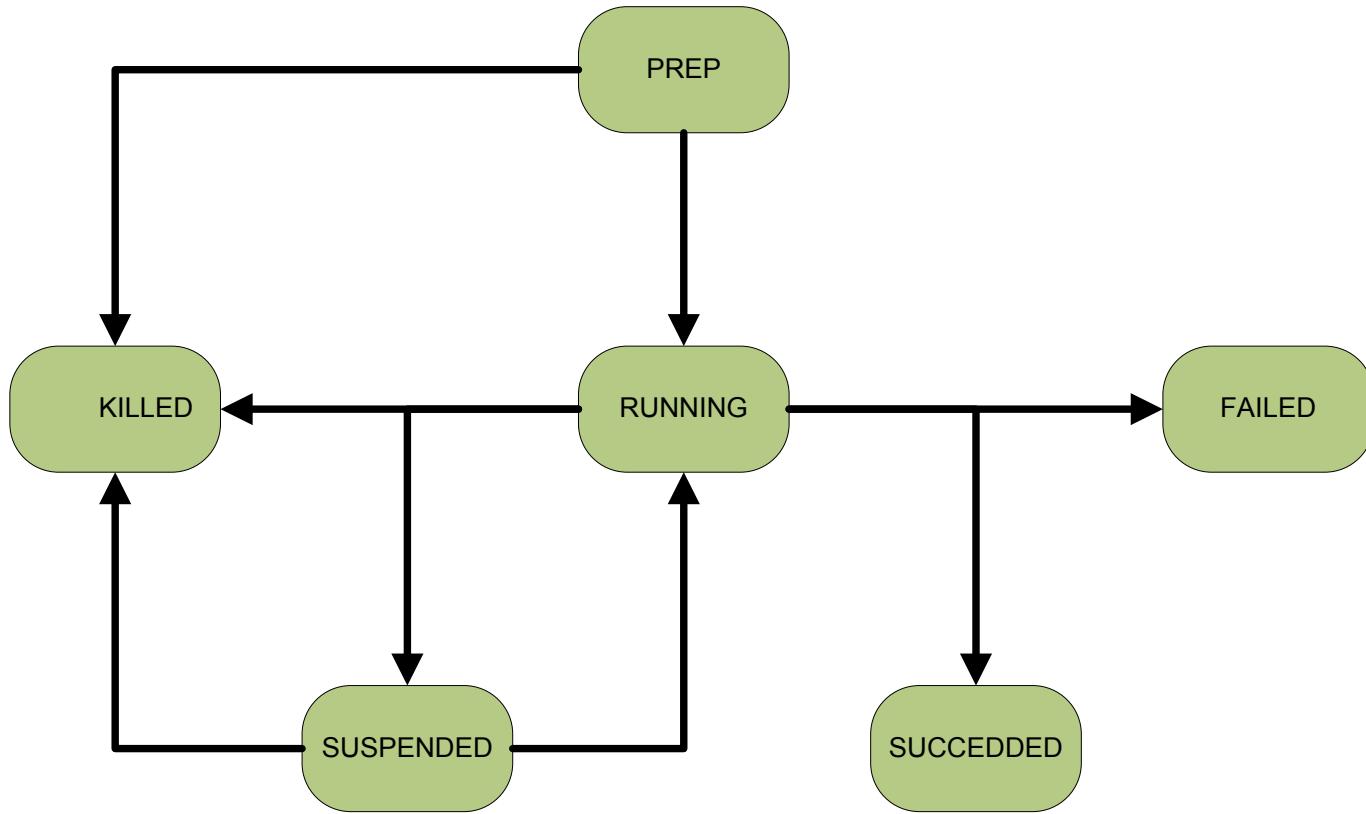
Action node	XML element type	Description
java	workflow:JAVA	invokes the main() method from the specified java class
fs	workflow:FS	manipulate files and directories in HDFS; supports commands: move, delete, mkdir
MapReduce	workflow:MAP-REDUCE	starts a Hadoop map/reduce job; that could be java MR job, streaming job or pipe job
Pig	workflow:pig	runs a Pig job
Sub workflow	workflow:SUB-WORKFLOW	runs a child workflow job
Hive *	workflow:HIVE	runs a Hive job
Shell *	workflow:SHELL	runs a Shell command
ssh *	workflow:SSH	starts a shell command on a remote machine as a remote secure shell
Sqoop *	workflow:SQOOP	runs a Sqoop job
Email *	workflow:EMAIL	sending emails from Oozie workflow application
Distcp ?		Under development (Yahoo)

# Workflow actions

- Oozie workflow supports two types of actions:
  - Synchronous, executed inside Oozie runtime
  - Asynchronous, executed as a Map Reduce job.



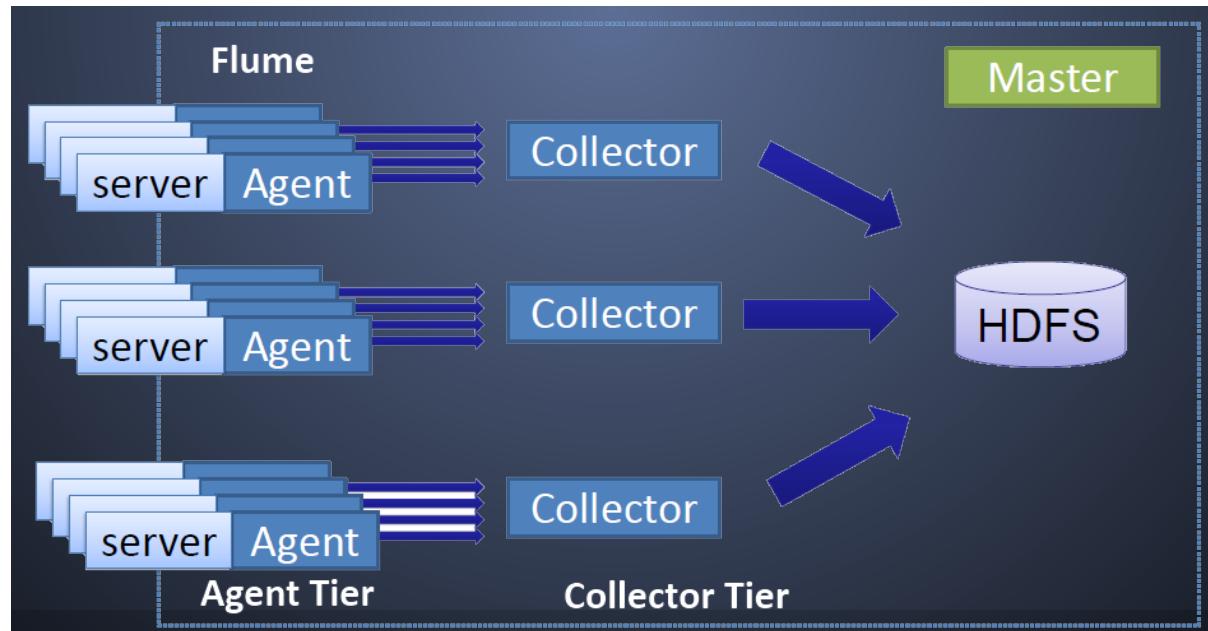
# Workflow lifecycle



# Flume

Flume is a distributed data collection service that gets flows of data (like logs) from their source and aggregates them to where they have to be processed.

Goals: reliability, scalability, extensibility and manageability.



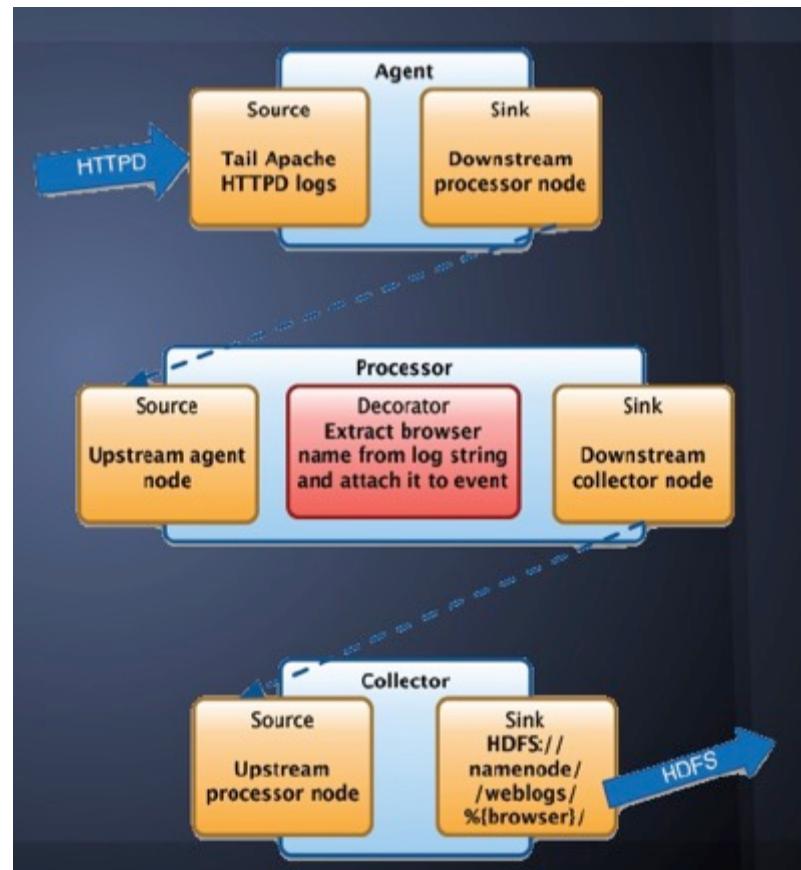
# Flume Architecture

Source: Examples: Console, Exec, Syslog, IRC, Twitter, other nodes

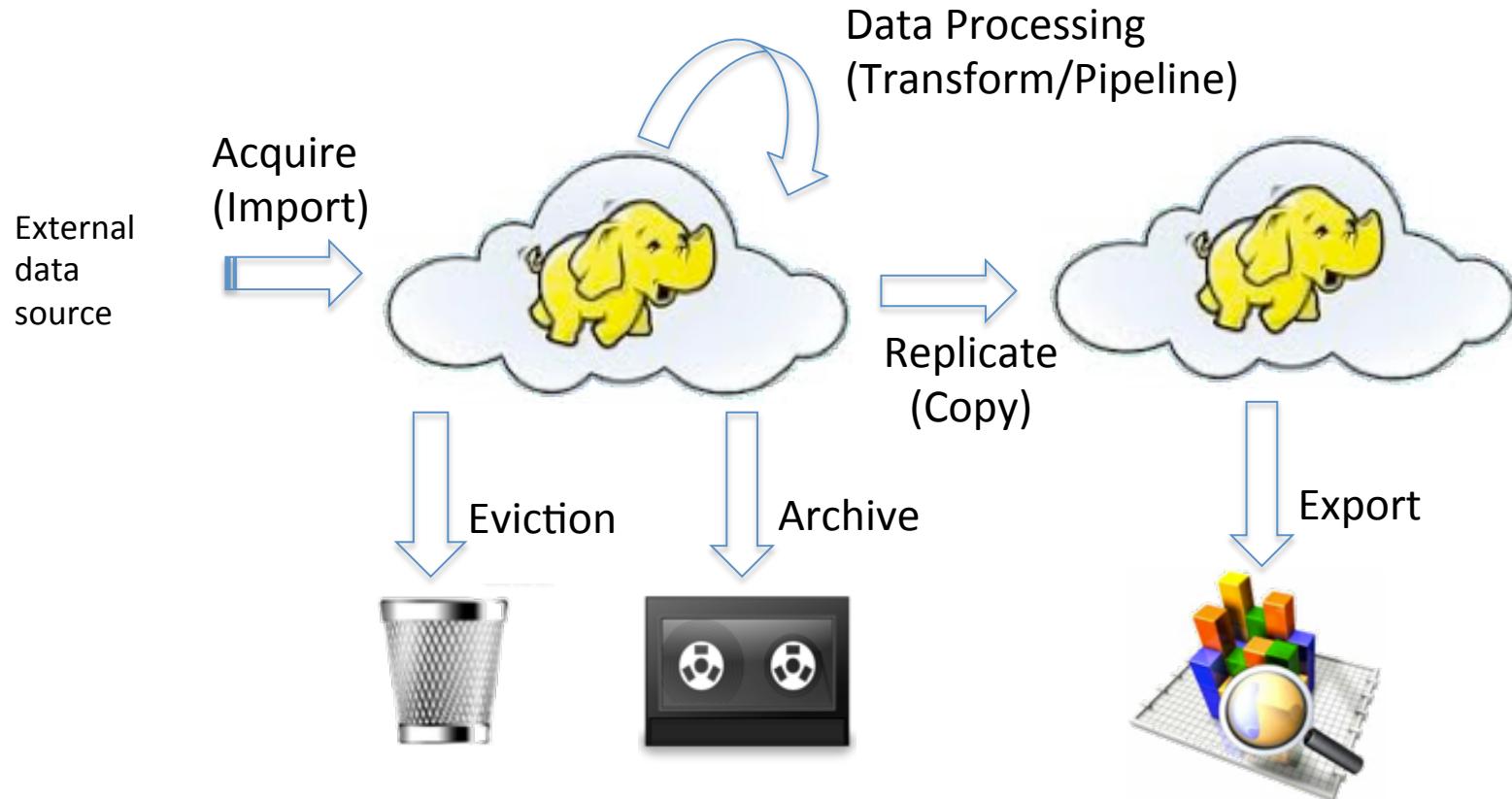
Sink: Examples: Console, local files, HDFS, S3, other nodes

Deco: Examples: wire batching, compression, sampling, projection, extraction

- Agent: receives data from an application.
- Processor (optional): intermediate processing.
- Collector: write data to permanent storage.



# Data Processing Landscape



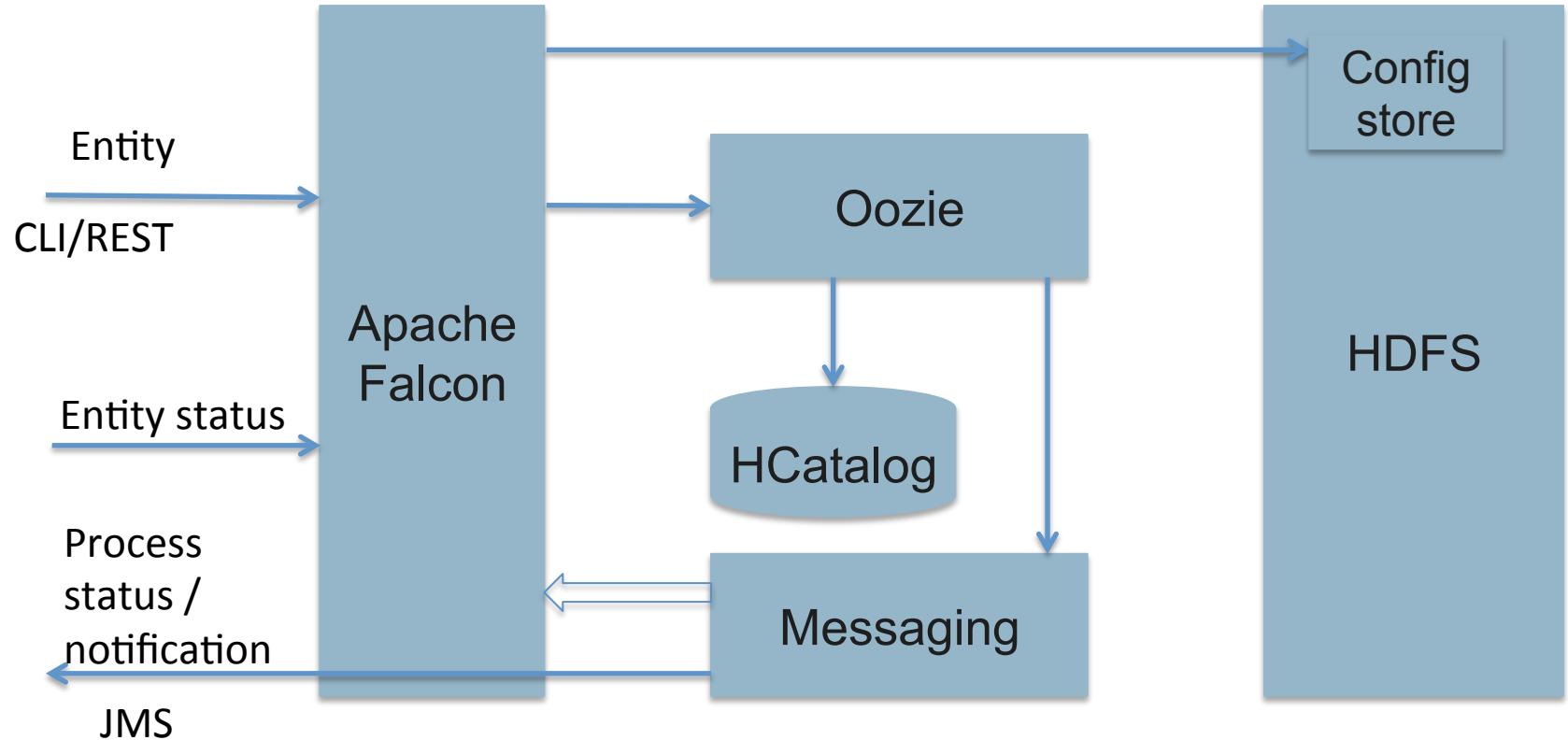
# Falcon: One-stop Shop for Data Management

Apache Falcon	
Provides	Orchestrates
Data Management Needs	Tools
Multi Cluster Management	Oozie
Replication	Sqoop
Scheduling	Distcp
Data Reprocessing	Flume
Dependency Management	Map / Reduce
Eviction	Hive and Pig Jobs
Governance	

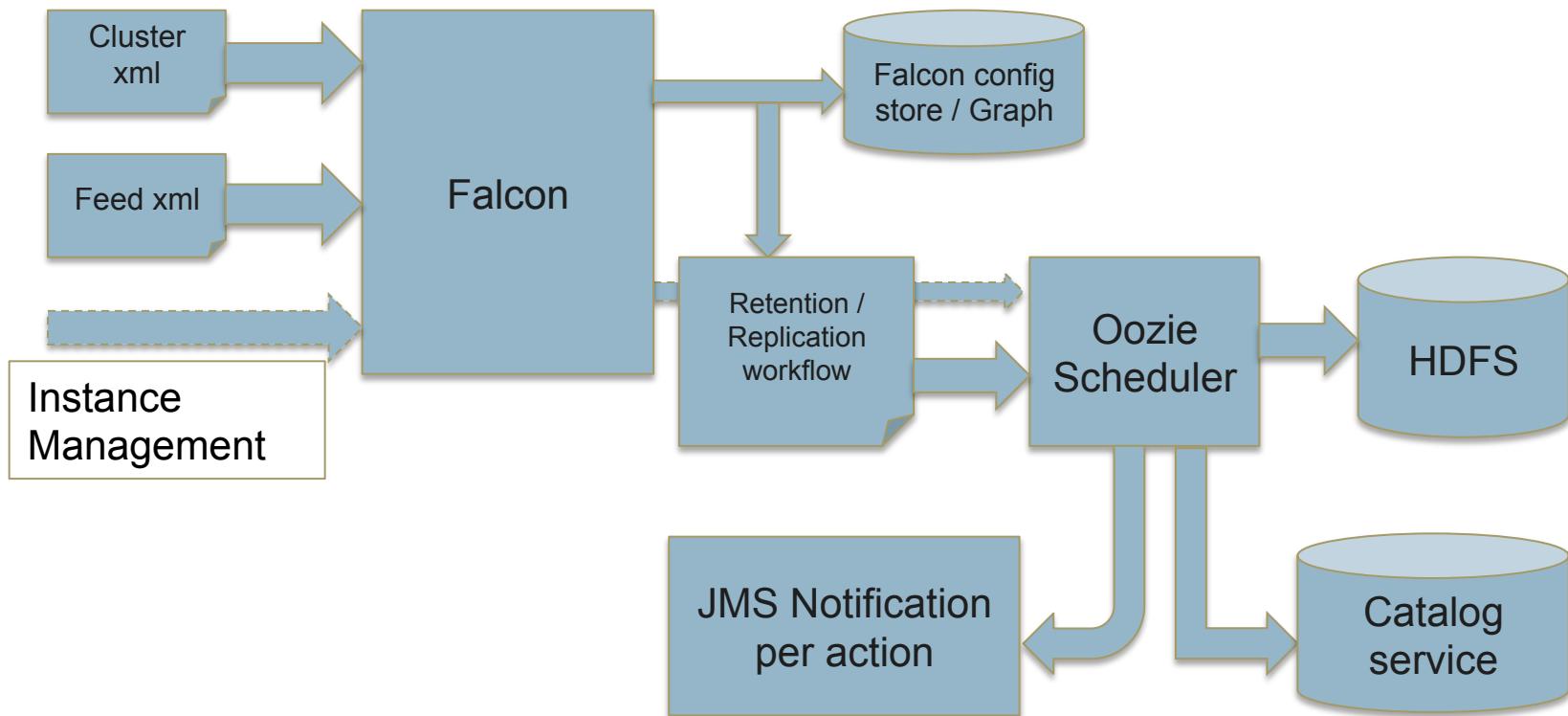
Falcon provides a single interface to orchestrate data lifecycle.  
Sophisticated DLM easily added to Hadoop applications.

Slide Courtesy: Hortonworks

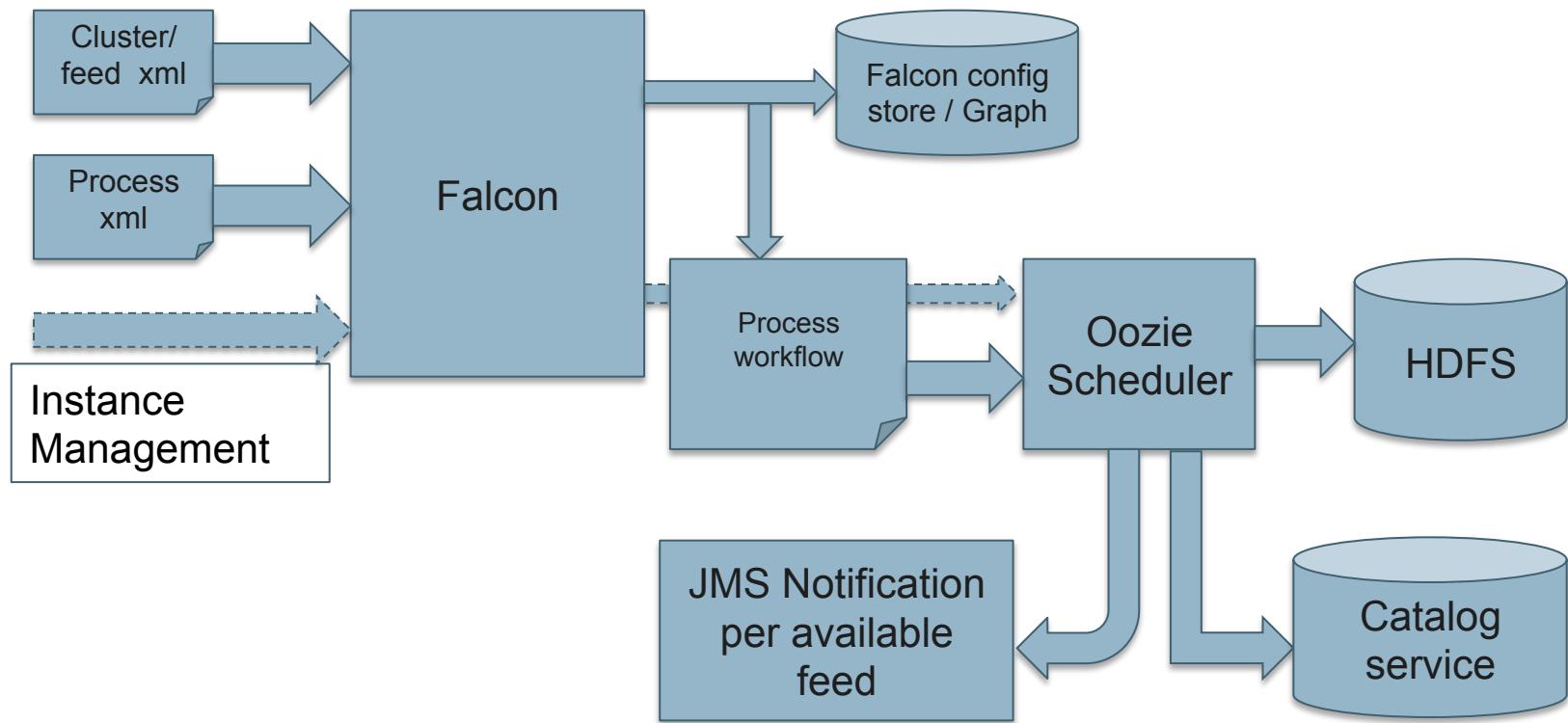
# High Level Architecture



# Feed Schedule



# Process Schedule



# Physical Architecture

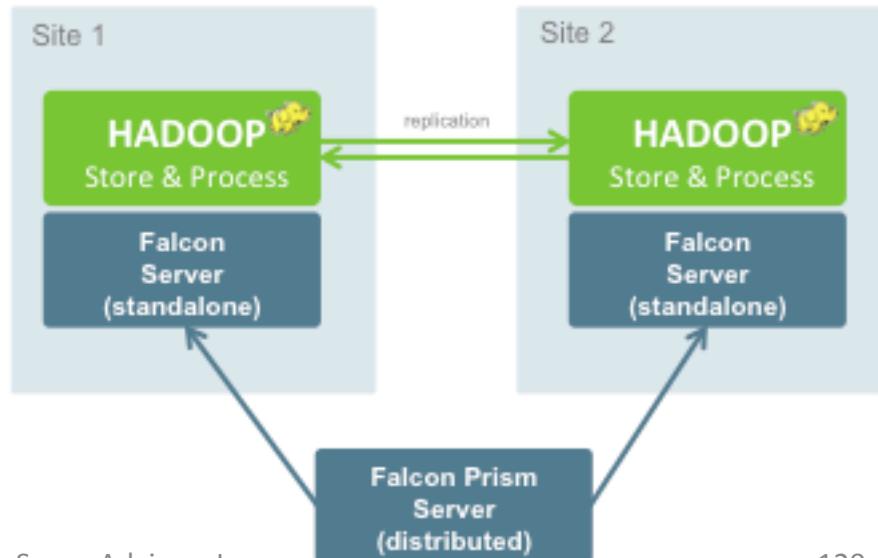
- **STANDALONE**

- Single Data Center
- Single Falcon Server
- Hadoop jobs and relevant processing involves only one cluster



- **DISTRIBUTED**

- Multiple Data Centers
- Falcon Server per DC
- Multiple instances of hadoop clusters and workflow schedulers



# Knox

- The Apache Knox Gateway is an extensible reverse proxy framework for securely exposing REST APIs and HTTP based services at a perimeter
- It is also referenced as Perimeter Based Security, the perimeter is outside of the core Hadoop ecosystem
- Out of the box features:
  - Support for several of the most common Hadoop services
  - Integration with enterprise authentication systems
  - Extensible

# Why Knox?

## Enhanced Security

- Protect network details
- Partial SSL for non-SSL services
- WebApp vulnerability filter

## Centralized Control

- Central REST API auditing
- Service-level authorization
- Alternative to SSH “edge node”

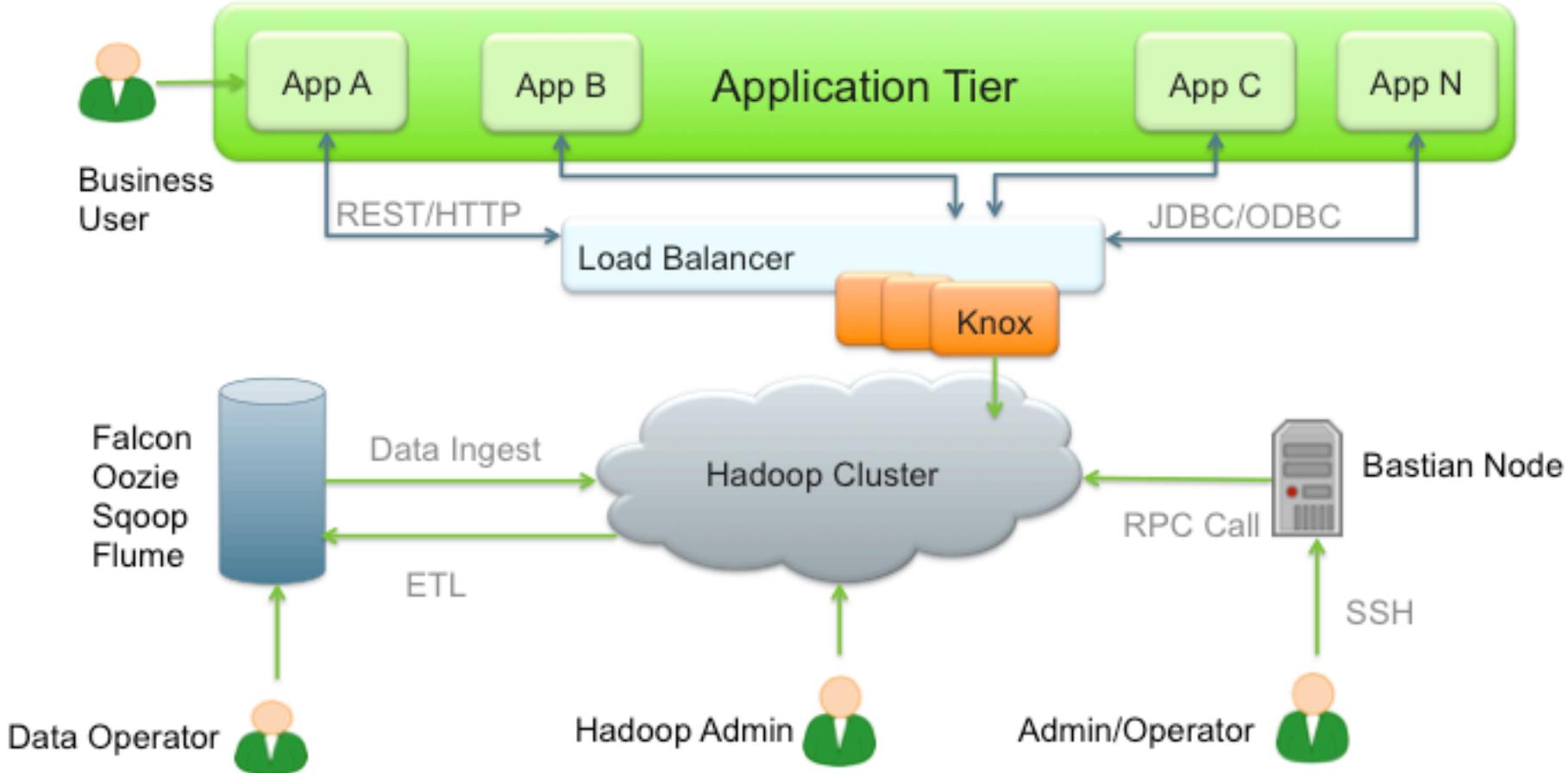
## Simplified Access

- Kerberos encapsulation
- Extends API reach
- Single access point
- Multi-cluster support
- Single SSL certificate

## Enterprise Integration

- LDAP integration
- Active Directory integration
- SSO integration
- Apache Shiro extensibility
- Custom extensibility

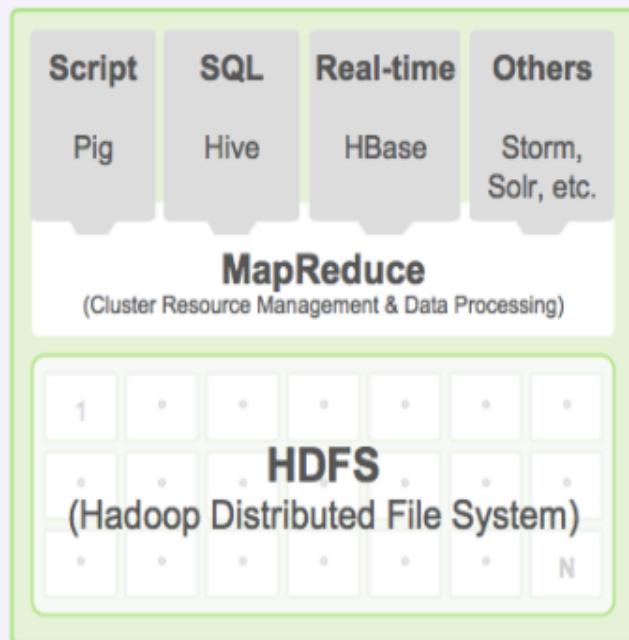
# Knox API Architecture



# Tez

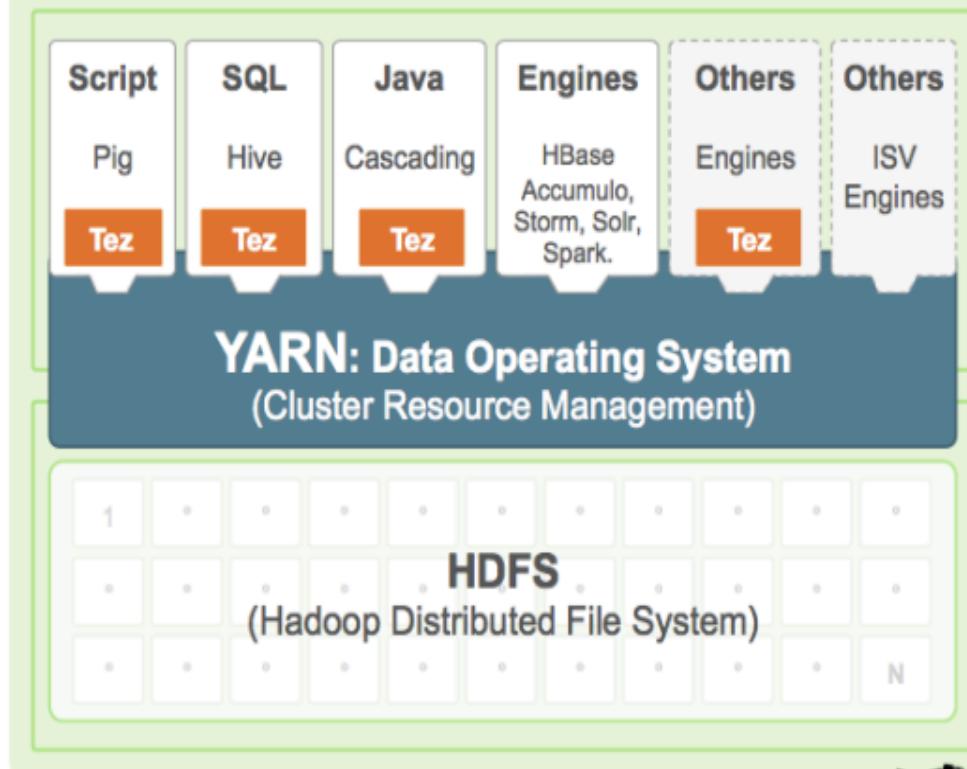
## Hadoop 1

- Silos & Largely batch
- Single Processing engine



## Hadoop 2 w/ Tez

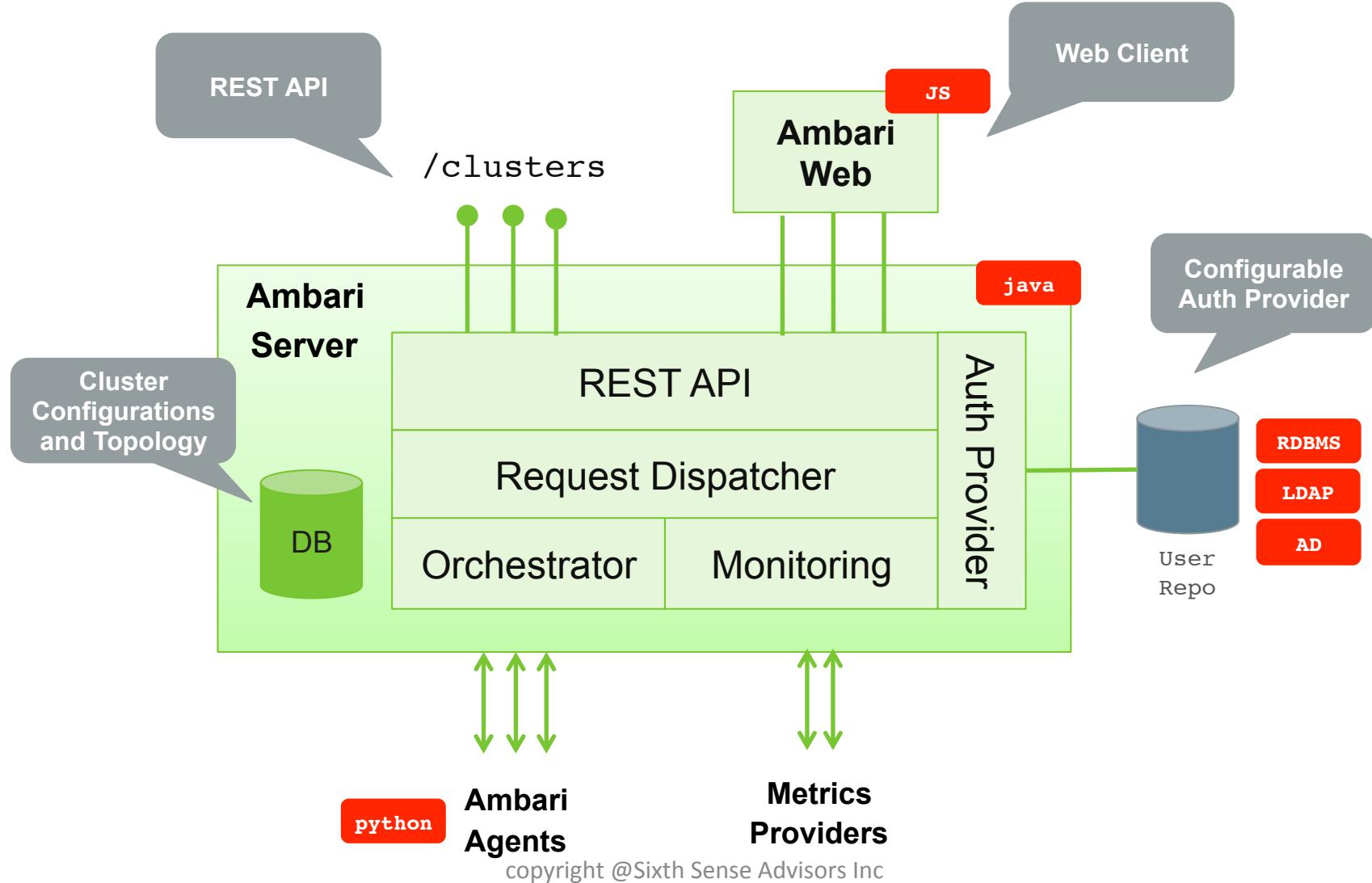
- Multiple Engines, Single Data Set
- Batch, Interactive & Real-Time



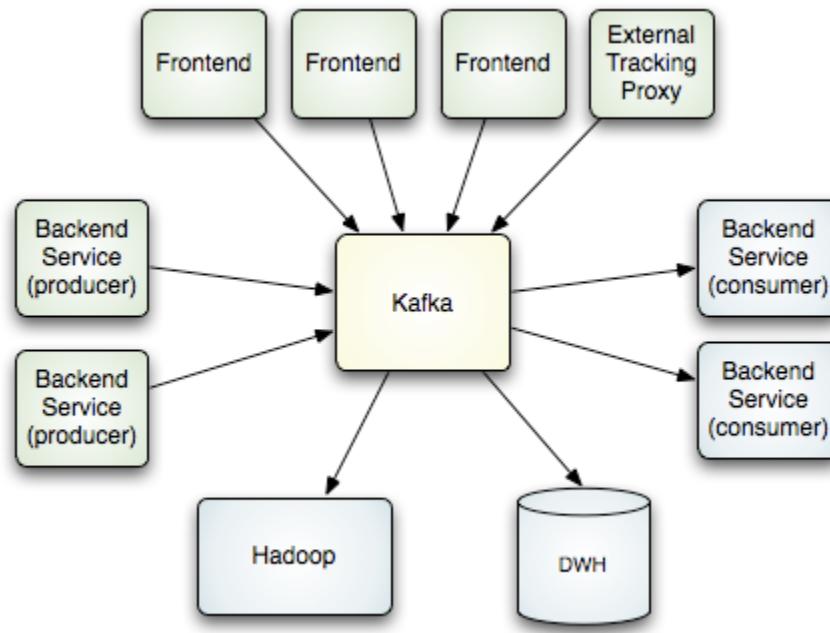
# Ambari

- The Apache Ambari project is aimed at making Hadoop management simpler by developing software for provisioning, managing, and monitoring Apache Hadoop clusters. Ambari provides an intuitive, easy-to-use Hadoop management web UI backed by its RESTful APIs.
- Ambari enables System Administrators to:
  - Provision a Hadoop Cluster
    - Ambari provides a step-by-step wizard for installing Hadoop services across any number of hosts.
    - Ambari handles configuration of Hadoop services for the cluster.
  - Manage a Hadoop Cluster
    - Ambari provides central management for starting, stopping, and reconfiguring Hadoop services across the entire cluster.
  - Monitor a Hadoop Cluster
    - Ambari provides a dashboard for monitoring health and status of the Hadoop cluster.
- Ambari leverages
  - Ganglia for metrics collections and [Nagios for system alerting and will send emails when your attention is needed \(e.g., a node goes down, remaining disk space is low, etc\).](#)

# Ambari Architecture

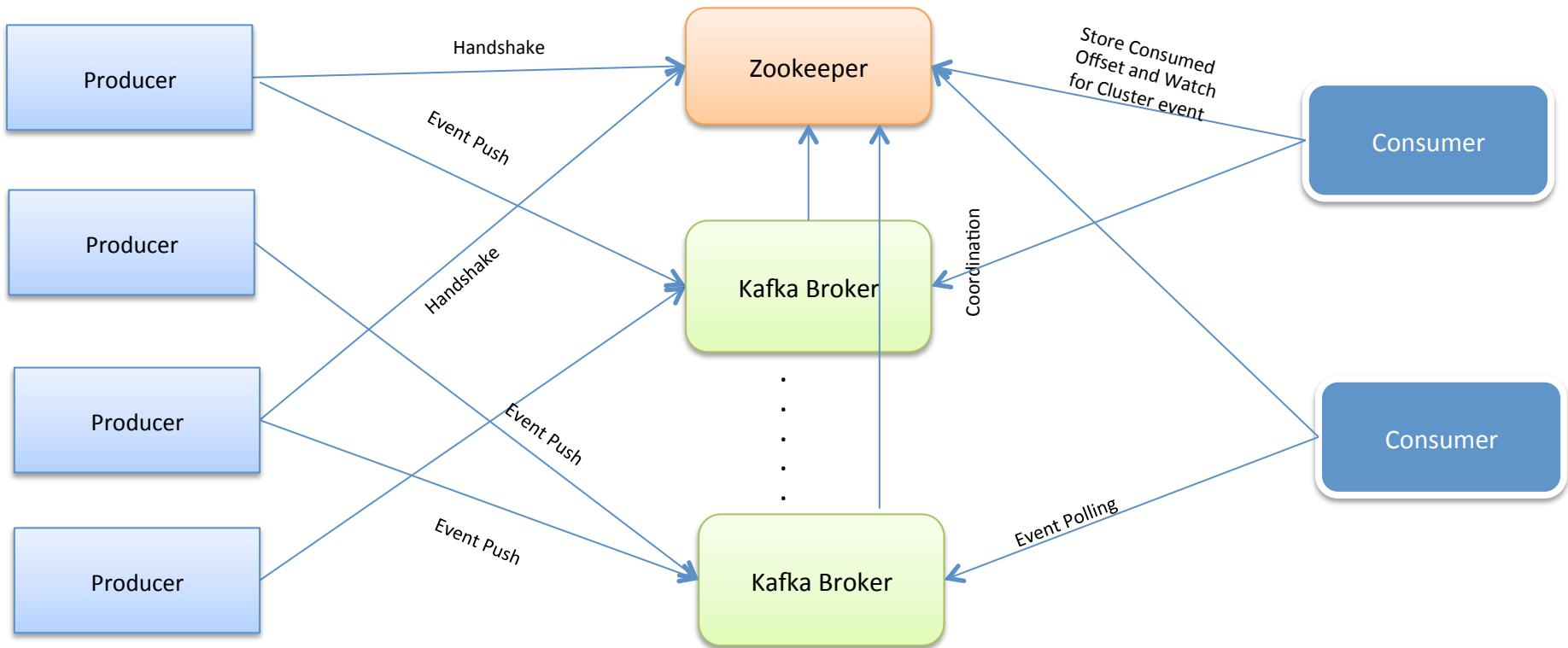


# Kafka - How it works

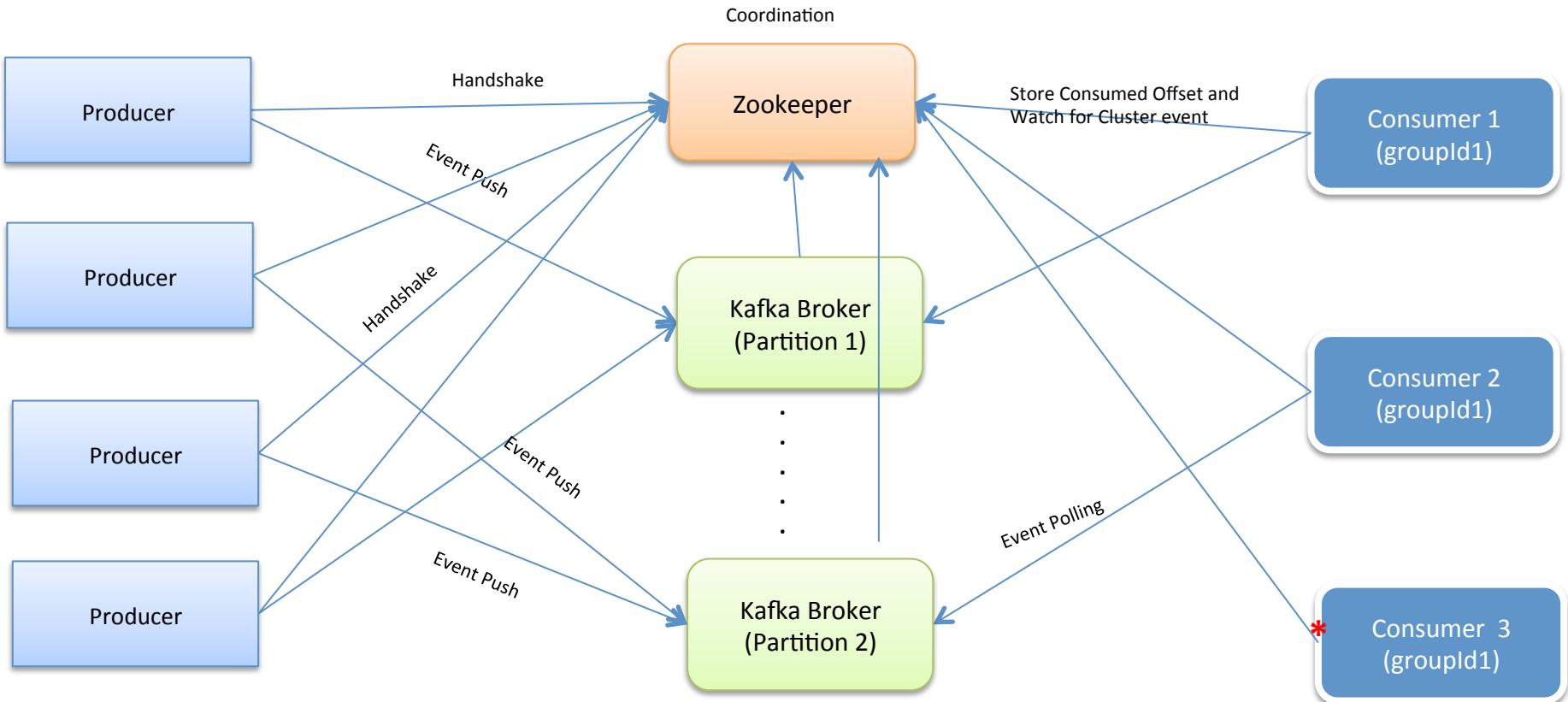


Credit : <http://kafka.apache.org/design.html>  
copyright @Sixth Sense Advisors Inc

# Kafka - How it works

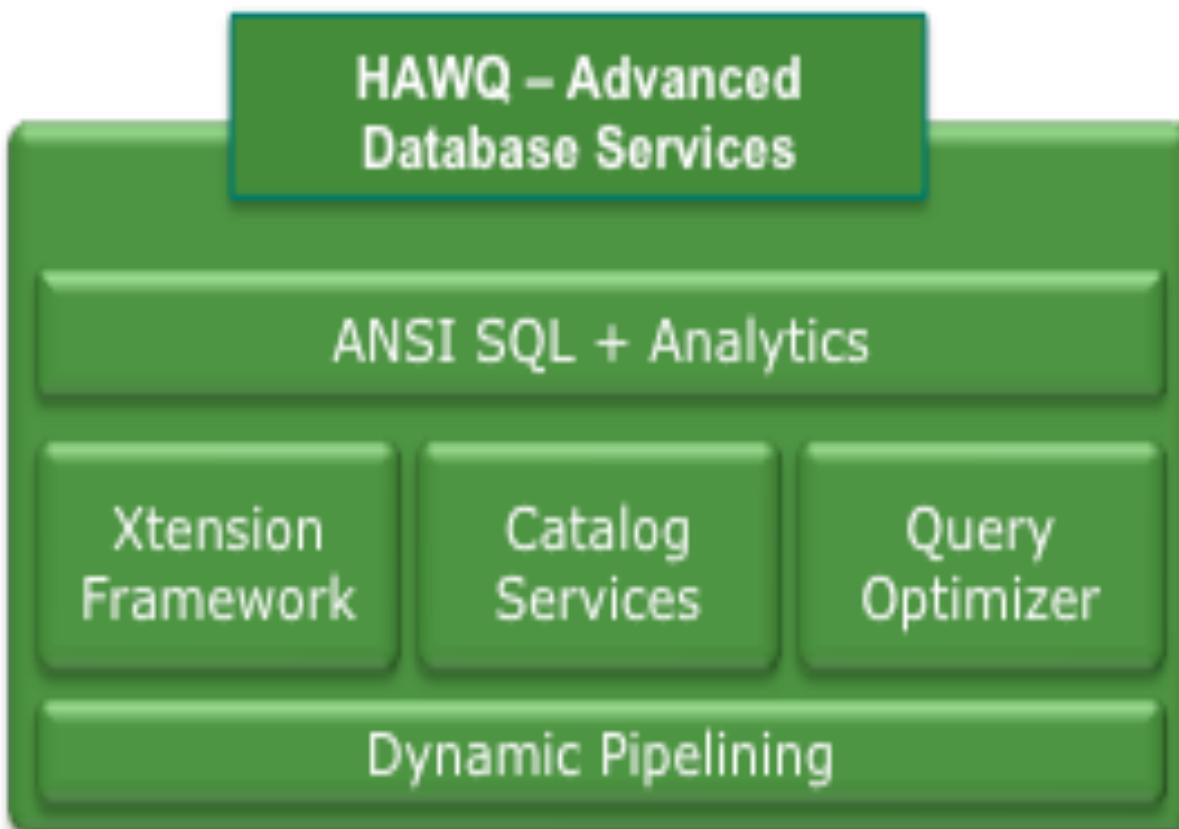


# Kafka - How it works (Queue)

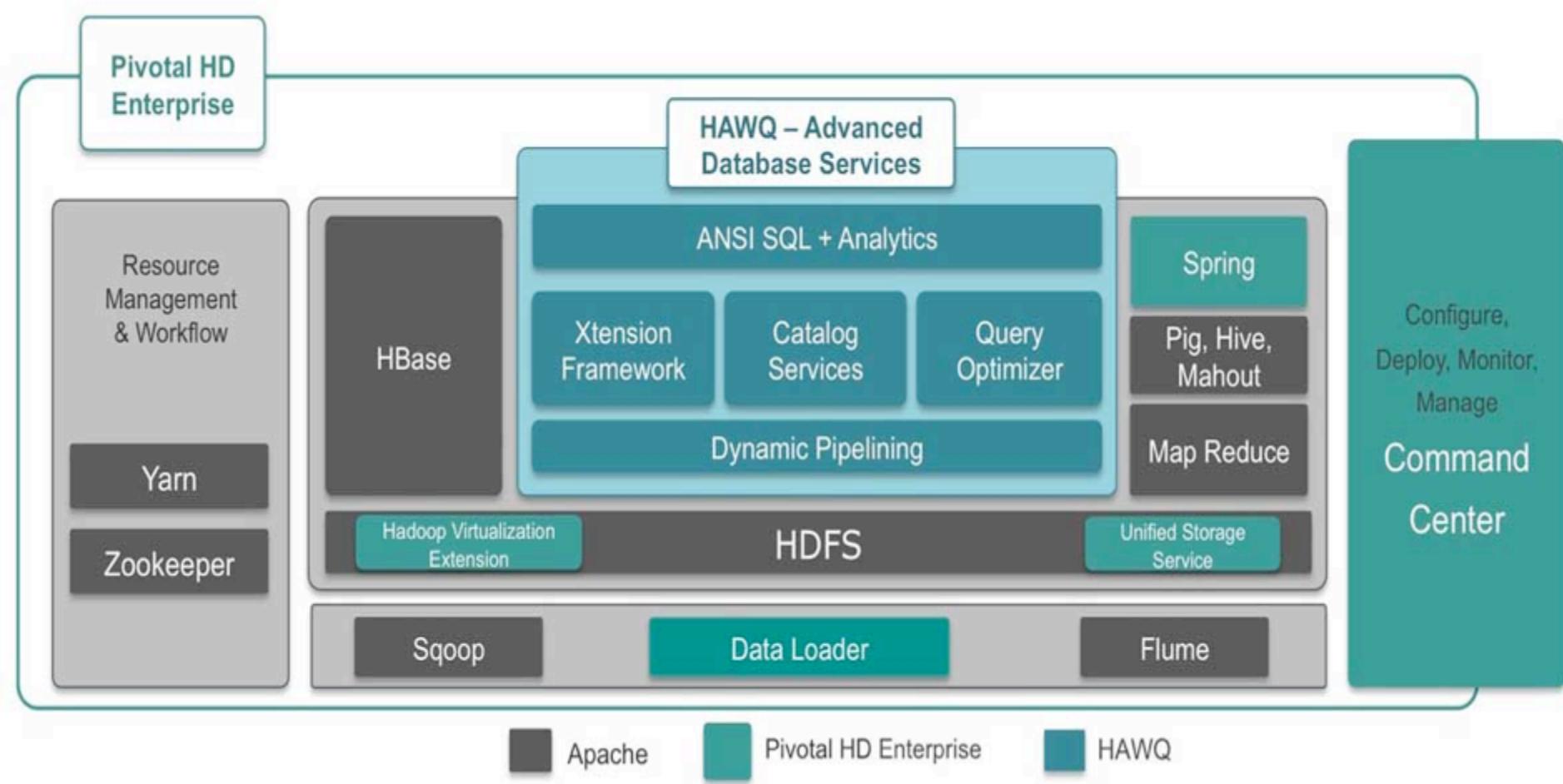


\* Consumer 3 would not receive any data, as number of consumers are more than number of partitions.

# HAWQ



# Pivotal Big Data Platform

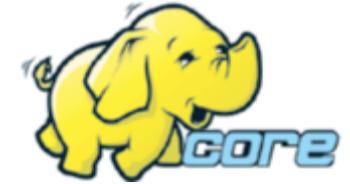


# History of Mahout

- Summer 2007
  - Developers needed scalable ML
  - Mailing list formed
- Community formed
  - Apache contributors
  - Academia & industry
  - Lots of initial interest
- Project formed under Apache Lucene
  - January 25, 2008



# Current Situation

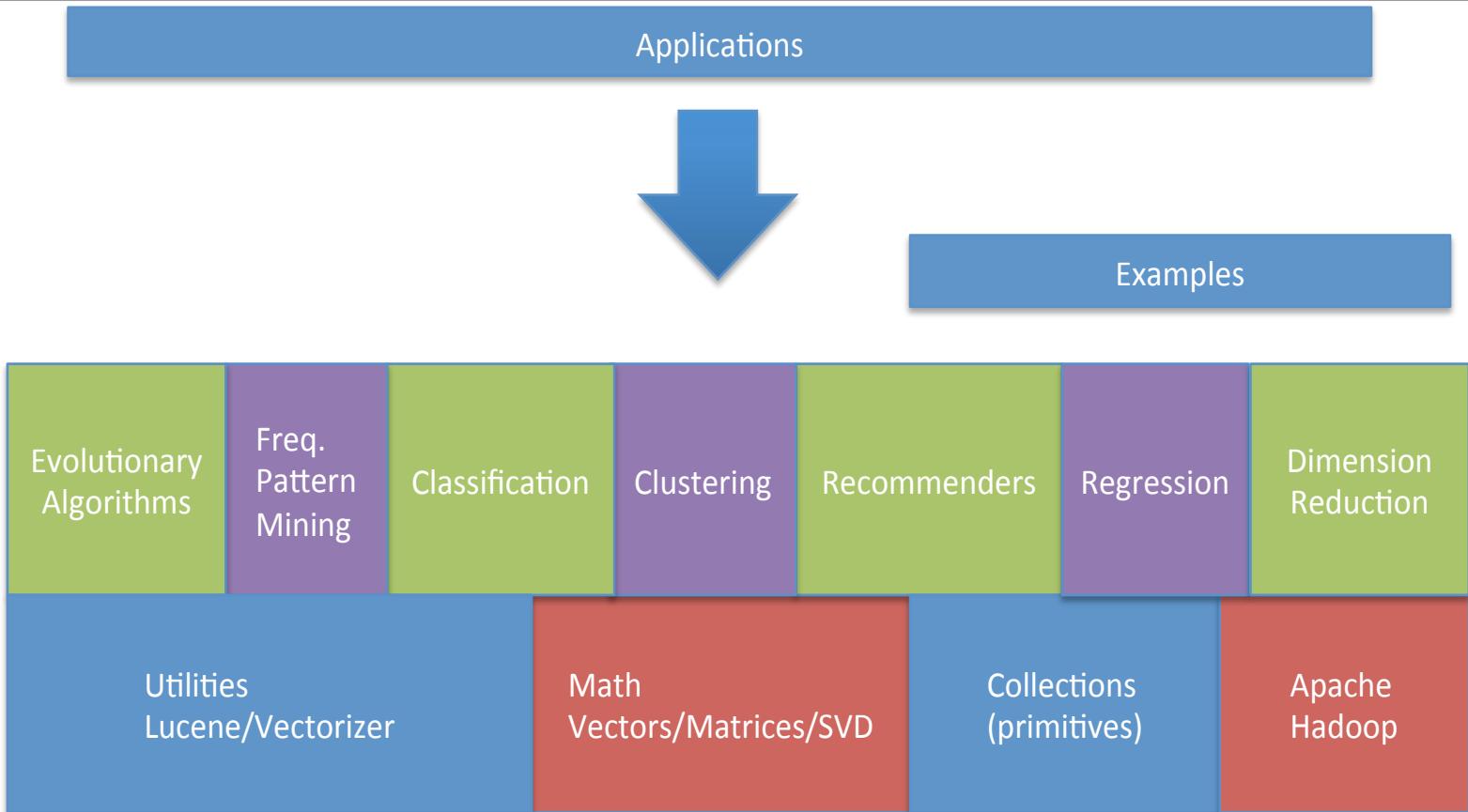


- Large volumes of data are now available
- Platforms now exist to run computations over large datasets (Hadoop, HBase)
- Sophisticated analytics are needed to turn data into information people can use
- Active research community and proprietary implementations of “machine learning” algorithms
- The world needs scalable implementations of ML under open license - ASF

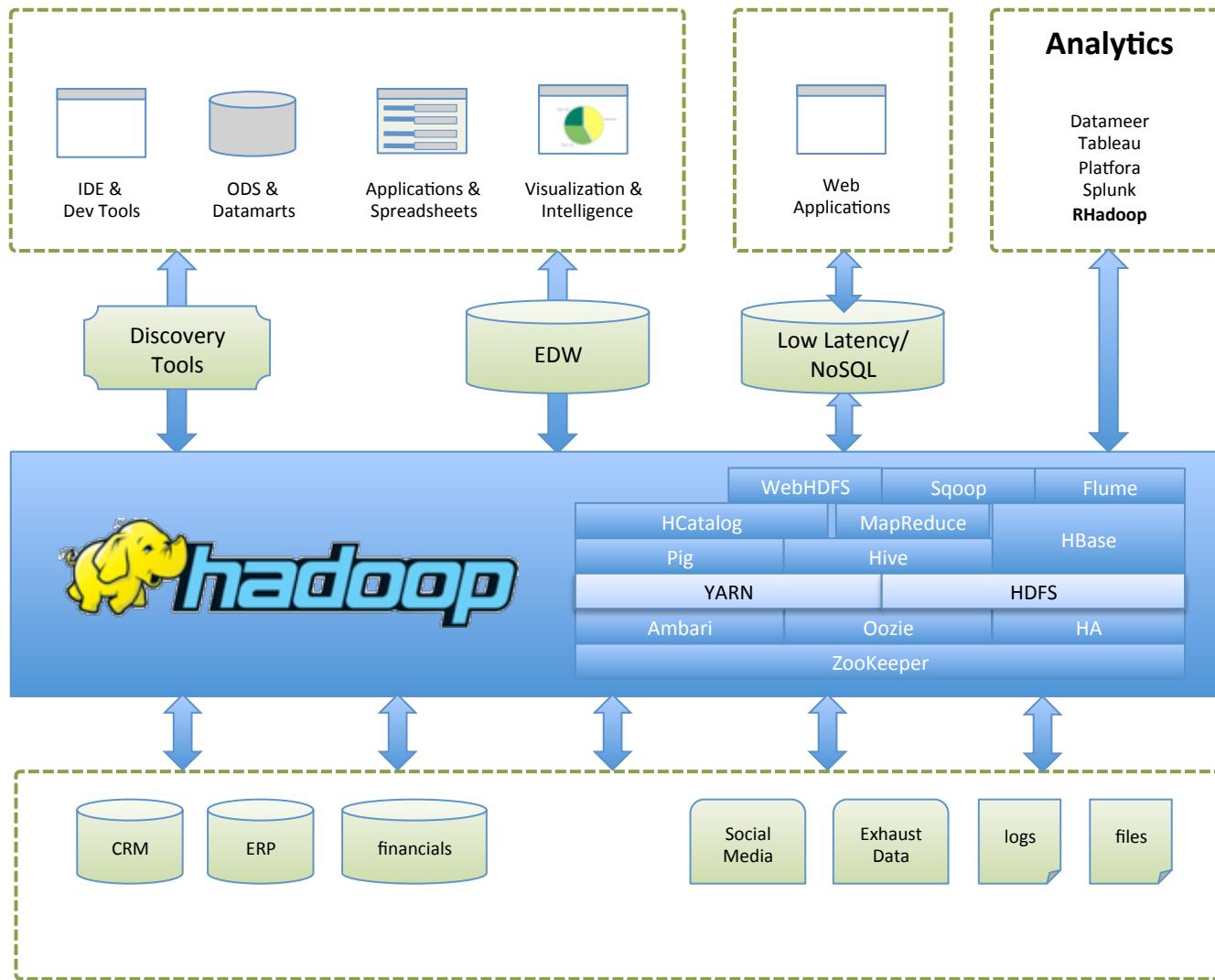
# Where is ML Used Today

- Internet search clustering
- Knowledge management systems
- Social network mapping
- Taxonomy transformations
- Marketing analytics
- Recommendation systems
- Log analysis & event filtering
- Fraud detection

# Mahout Architecture



# Hadoop Technical Architecture



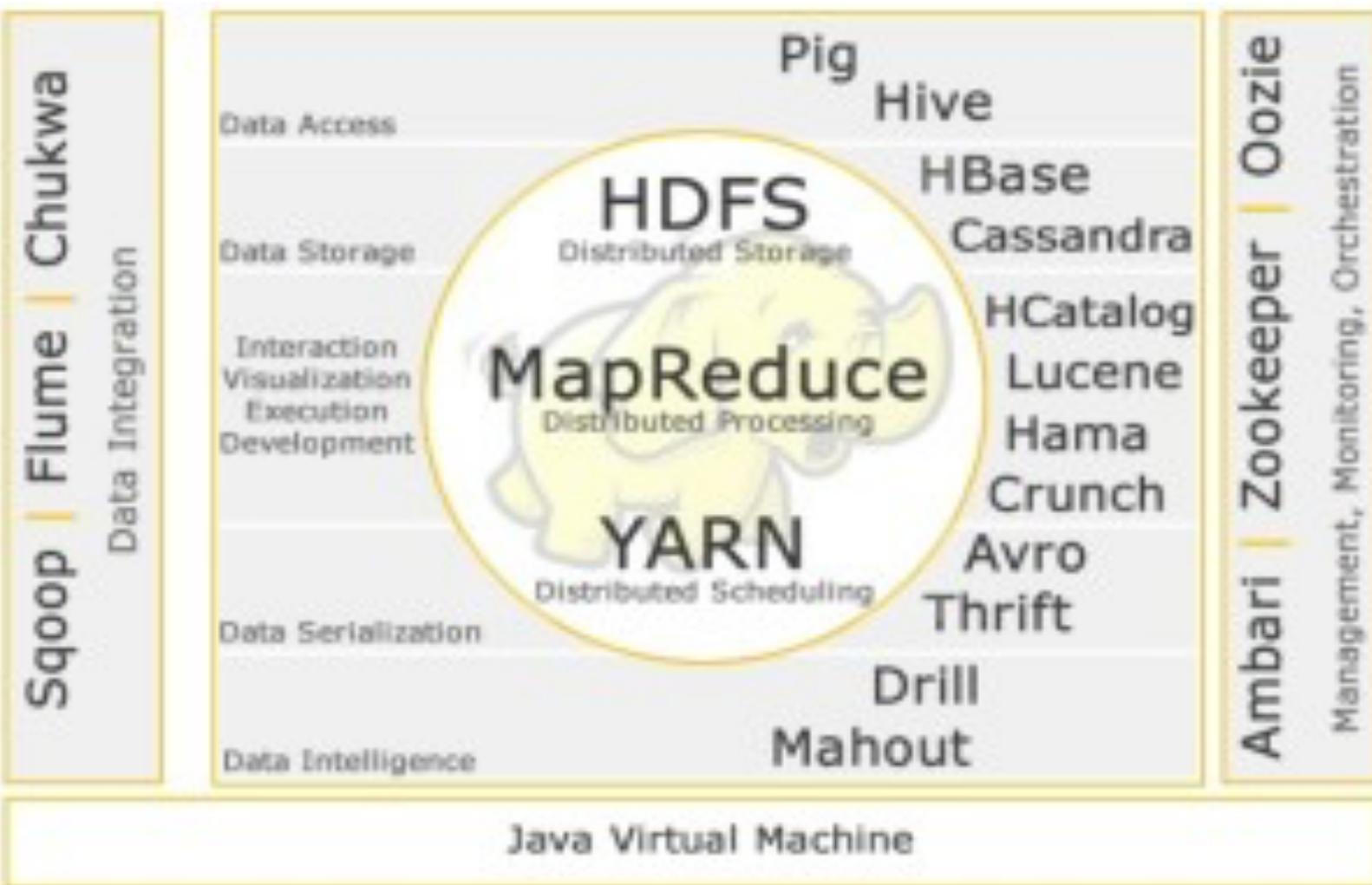
# SQL on Hadoop

- Apache Hive
- Impala
- Presto (Facebook)
- Shark
- Apache Drill
- EMC/Pivotal HAWQ
- BigSQL by IBM
- Apache Phoenix (for HBase)
- Apache Tajo

# Technical Architecture Best Practices

- Be ready to create a heterogeneous architecture
- Data strategy will be a key to establishing the success in the land of Hadoop
- IT should facilitate and govern the Hadoop environment
- Configuration files for YARN, HDFS, HIVE, HBASE and MapReduce need to be managed with attention to the infrastructure
- Do not expect the infrastructure to create speed of processing, this is depending on configuration and deployment of the ecosystem in its entirety

# Distributions

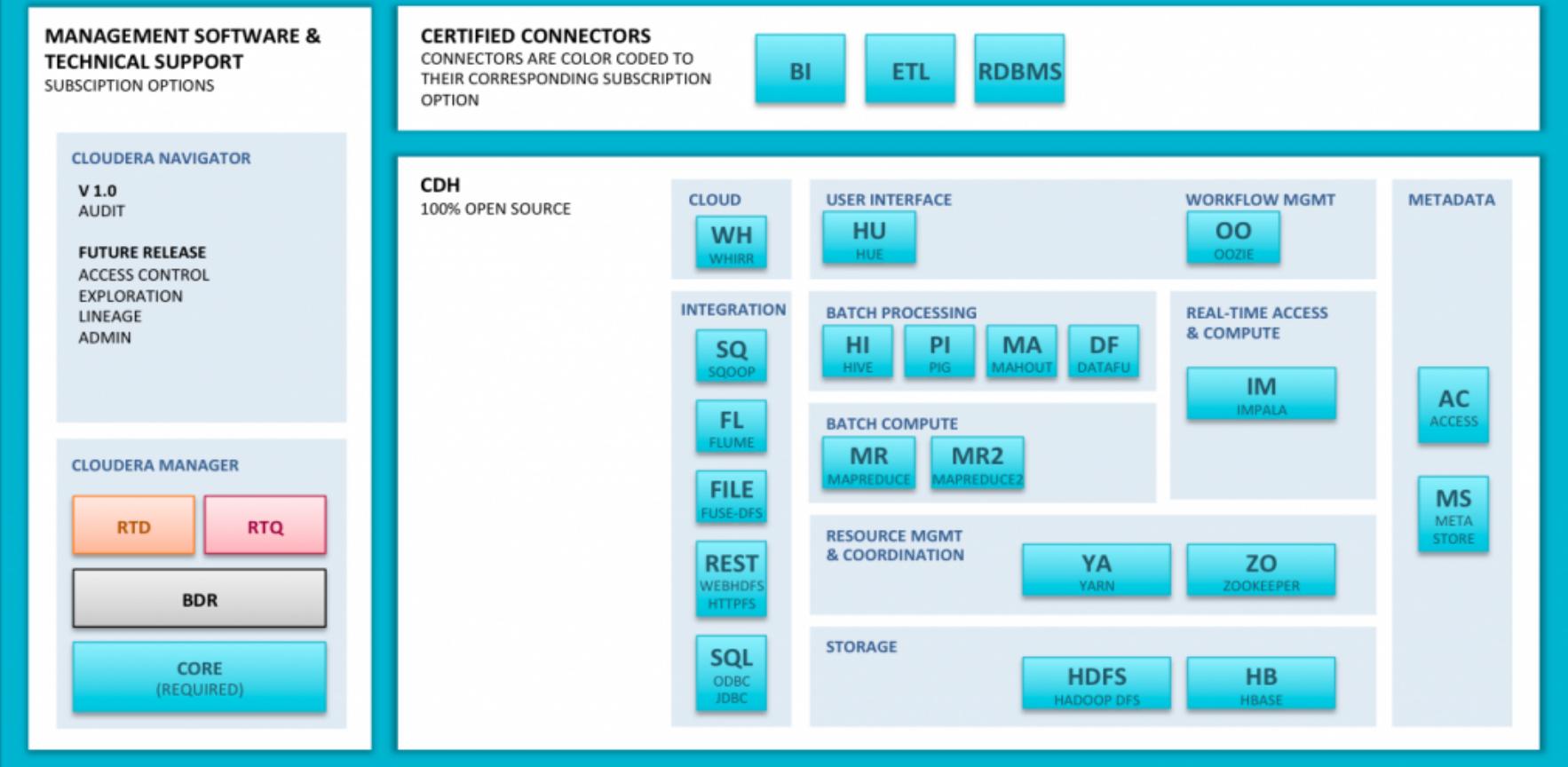


# Hortonworks



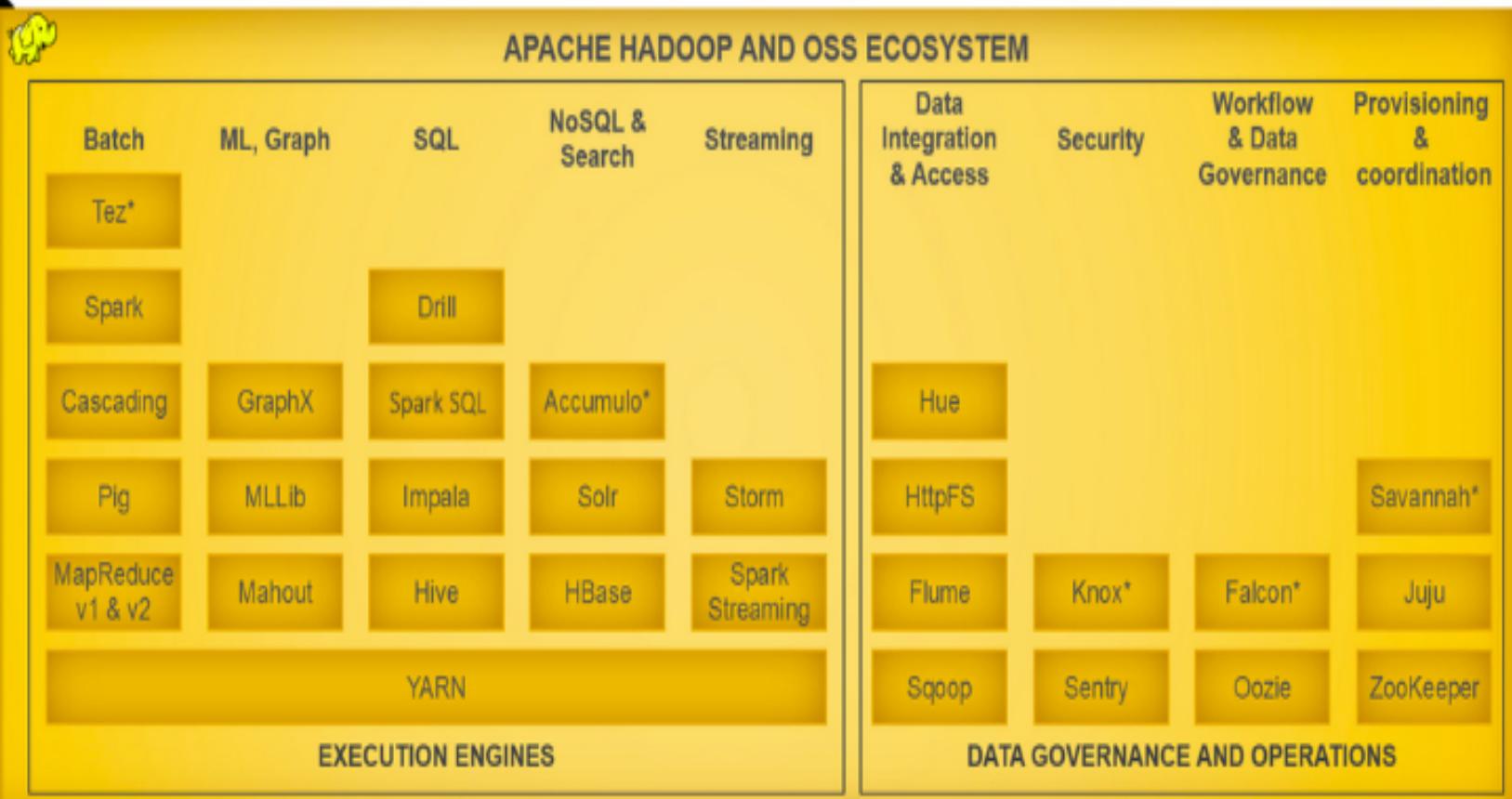
# Cloudera

CLOUDERA ENTERPRISE SUBSCRIPTION : MANAGEMENT AUTOMATION SOFTWARE + SUPPORT + OPEN SOURCE SOFTWARE



# MapR

Management

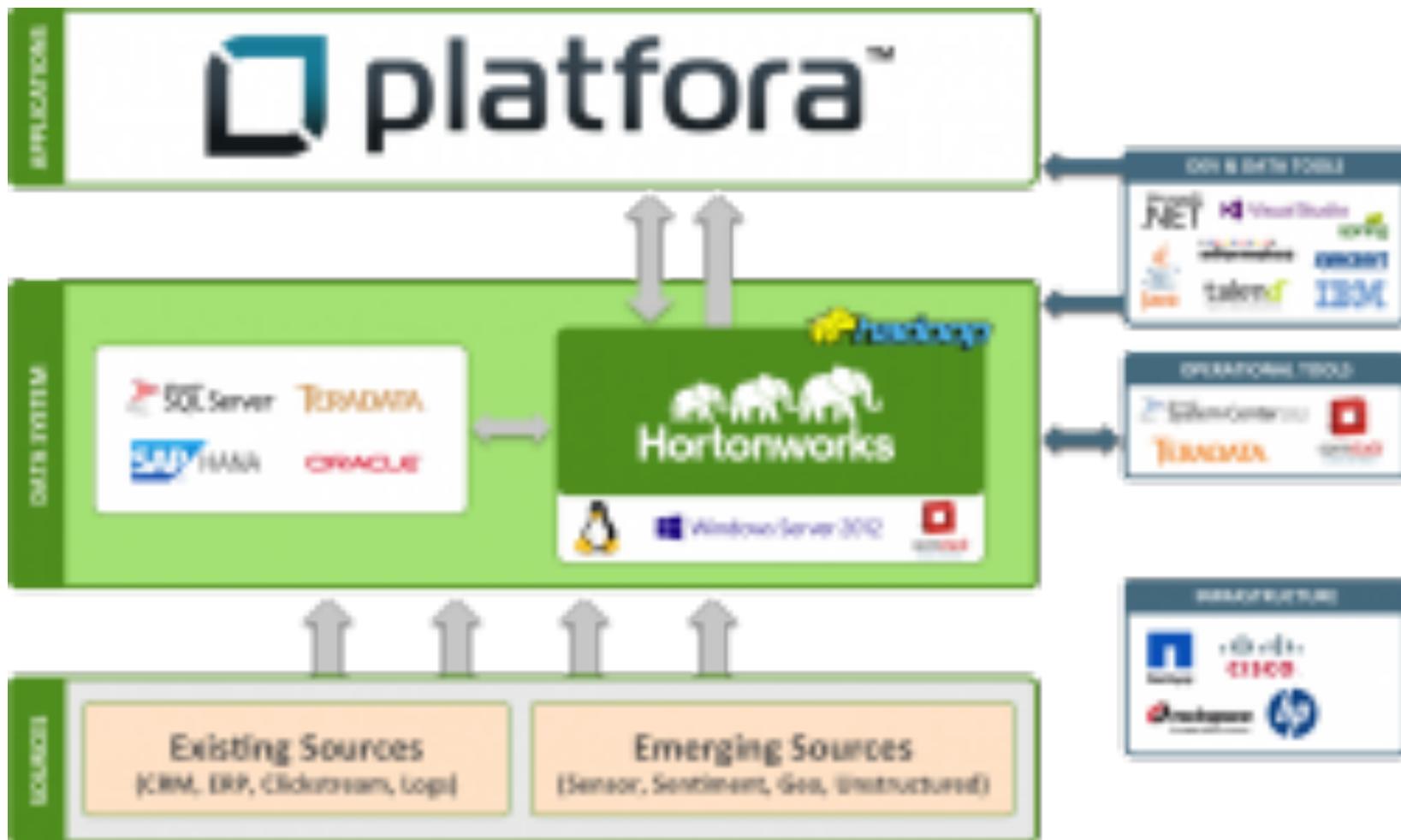


MapR-FS

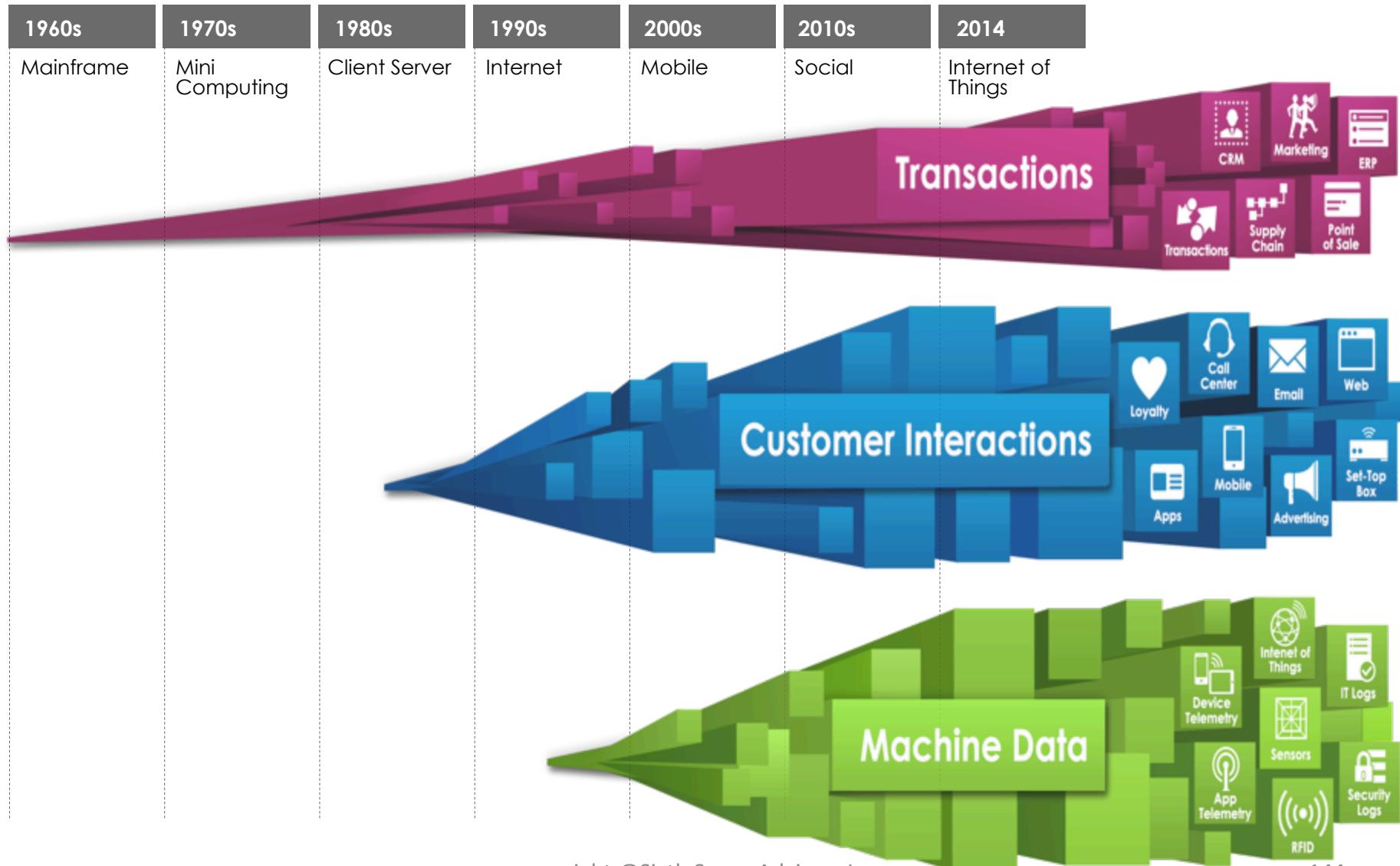
MapR Data Platform

MapR-DB

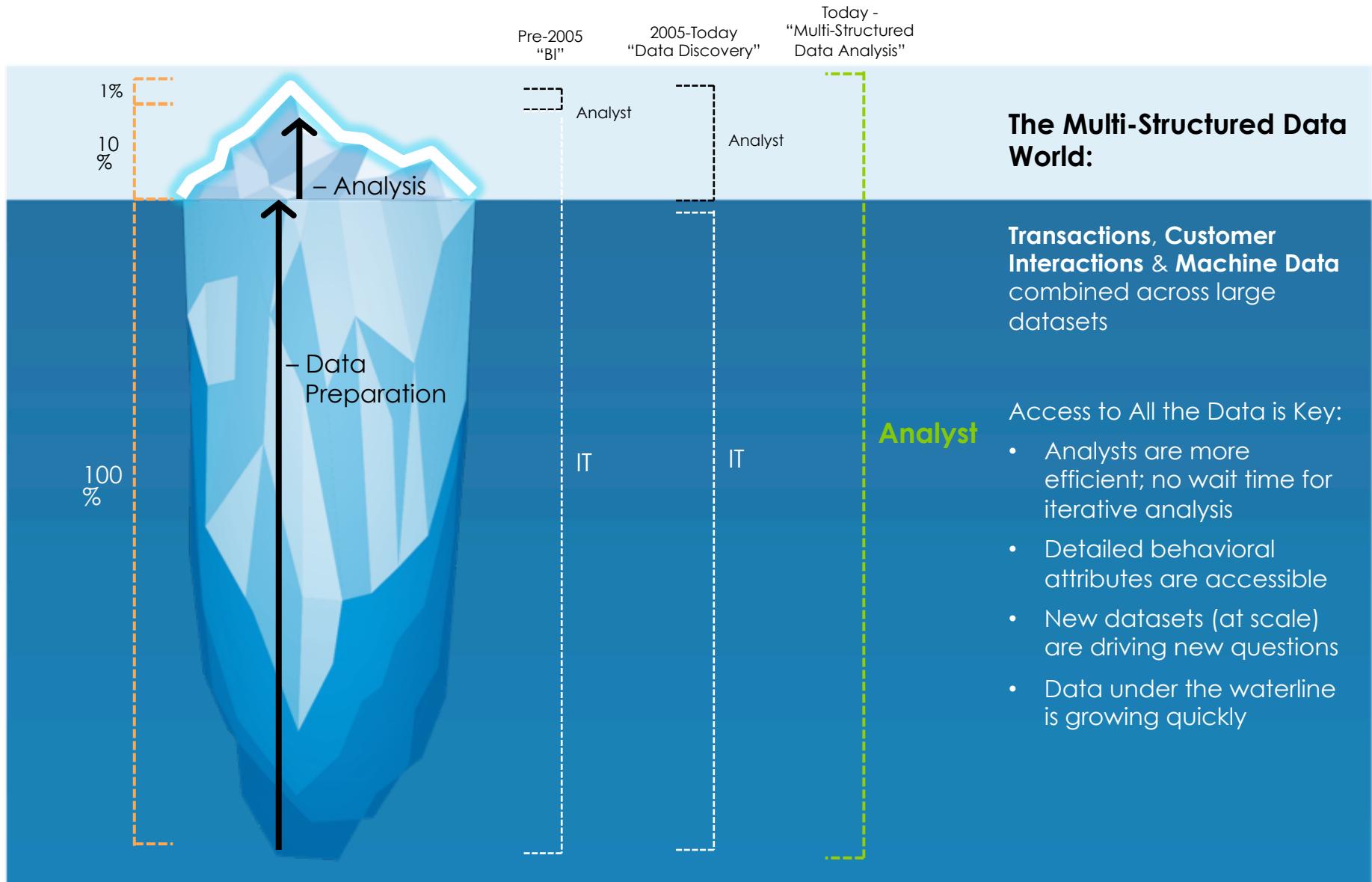
# Platfora



# The new world of data is Multi-Structured



# Evolution of Data Analyst Workflows



# Siloed marketing data makes it difficult to understand your audience

Forcing you to analyze a subset of your data



"Most companies estimate they're analyzing a mere 12% of the data they have...missing out on data-driven insights hidden inside the 88% of data they're ignoring."



# Understand your audience like never before

## Connections at every touchpoint

The screenshot shows the plafotra platform interface with the following details:

- Header:** Home, Data Catalog, Vibboards, System, Courtney Johnson, Help.
- Left Sidebar:** PAGES (1, 2, 3), Event Se..., Call Act..., BUILDER.
- Central Area:**
  - Dashboard Title:** Money Tree Dashboard
  - Left Panel (Builder):** Shows a funnel configuration for "Money Tree Customer Events". It includes stages: Enter Application, General Data, SSN / Dob, Complete, and Loan Approved. Each stage has an associated event type (Page View) and a condition (e.g., "is in list page1").
  - Right Panel (Visualizations):** A "Funnel by Customer Rating" visualization. The legend indicates "No stage selected". The funnel shows the count and percentage of total for each stage across five customer rating categories: Fallen\_Angel, Prime\_Holder, Rising\_Star, Subprime\_Holder, and Total. Each stage is represented by a blue bar chart with a red vertical line indicating the end of the funnel.
- Bottom:** A footer bar with the text "3.6.1-build.39 (2014.08.06)".

## Segmentation Analysis

Understand a wide variety of your audience's characteristics

## Behavioral Analysis

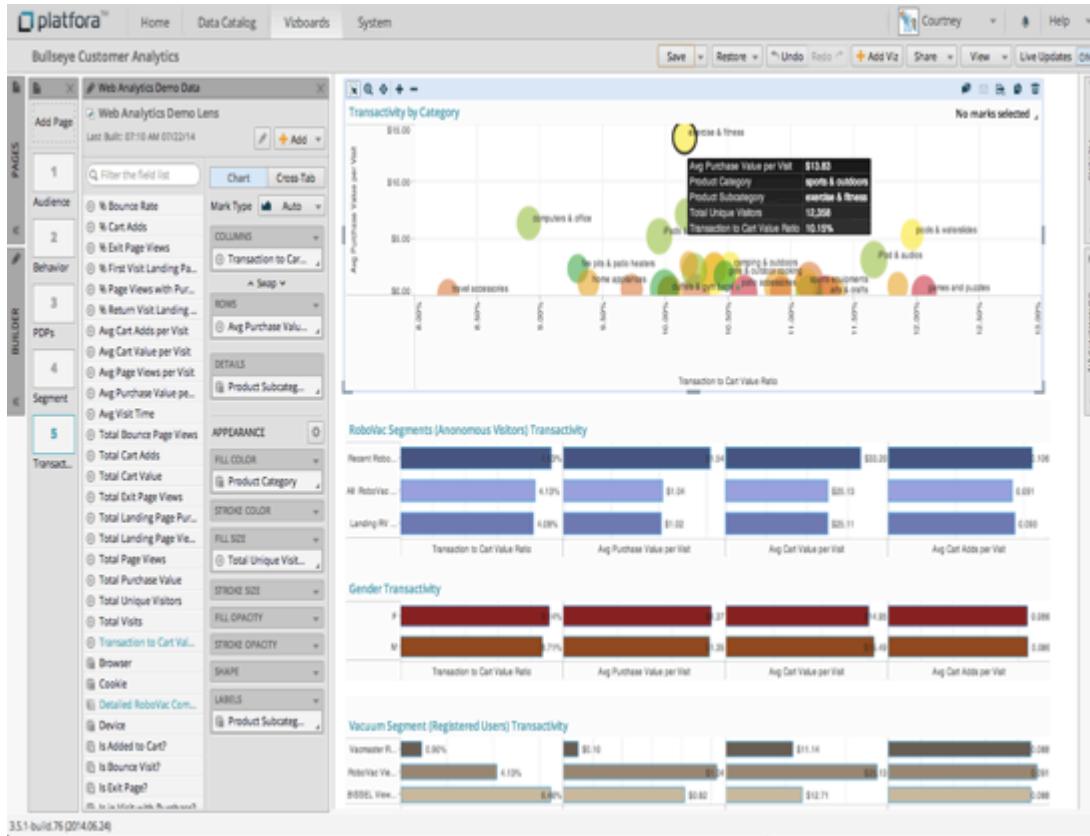
## Improve your targeting with robust behavioral measures

# Event Series Analysis

## Unite and make sense of raw data overtime

# Ask unlimited questions about your data

## Self-service, iterative, and interactive



Add new datasets instantly

Quickly add new datasets and analyze interesting data in Hadoop immediately

Zero in on what matters

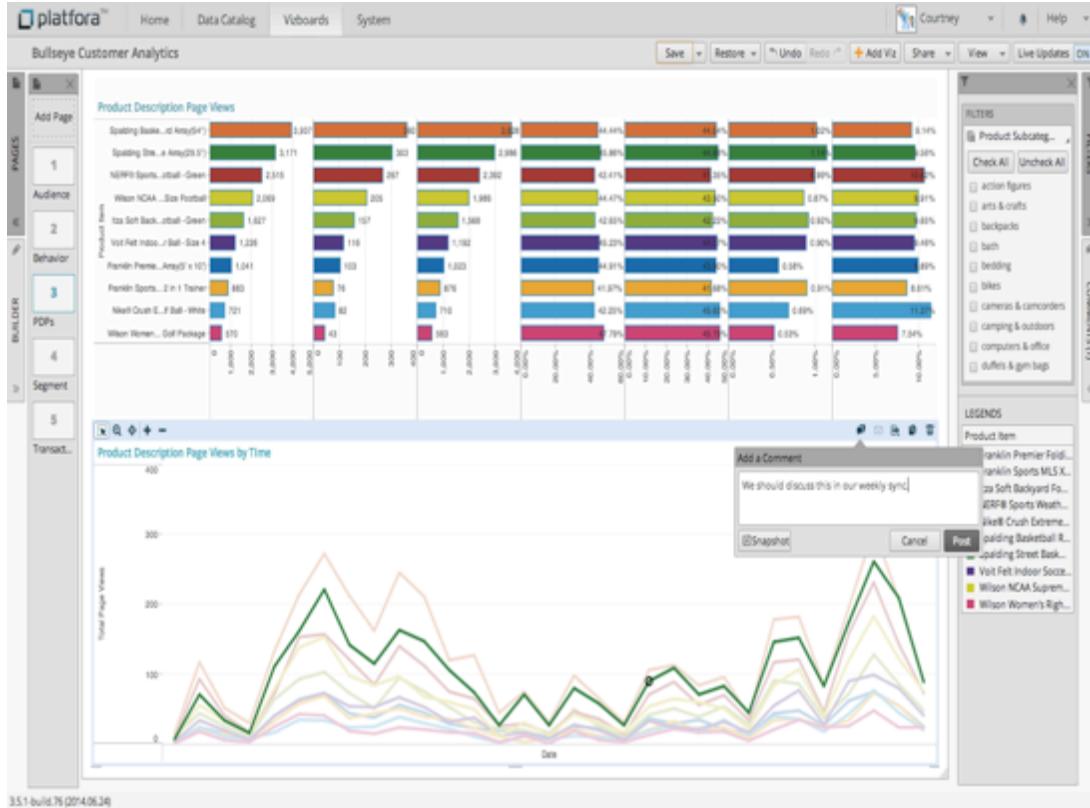
Create Lenses to analyze the data that is most valuable to your business

No code required

Drag and drop functionality eliminates the need for code

# Benefit from economies of scale

## Easy to use, scalable, and affordable



### Get up & running easily

Visualize insights from raw data in a day

### Analyze 100% of your data

Have confidence in your decisions by analyzing more accurate trends over a longer time period

### Democratize your data

Easily share insights with your coworkers regardless of their technical skill level

# Datameer

## Data Sources



Social Media



RSS/Web



RDBMS



Hive  
HBase  
Cassandra



Web Logs/  
Application Logs



Java

## Modeling



## Virtualization



## Security



LDAP

## Monitoring

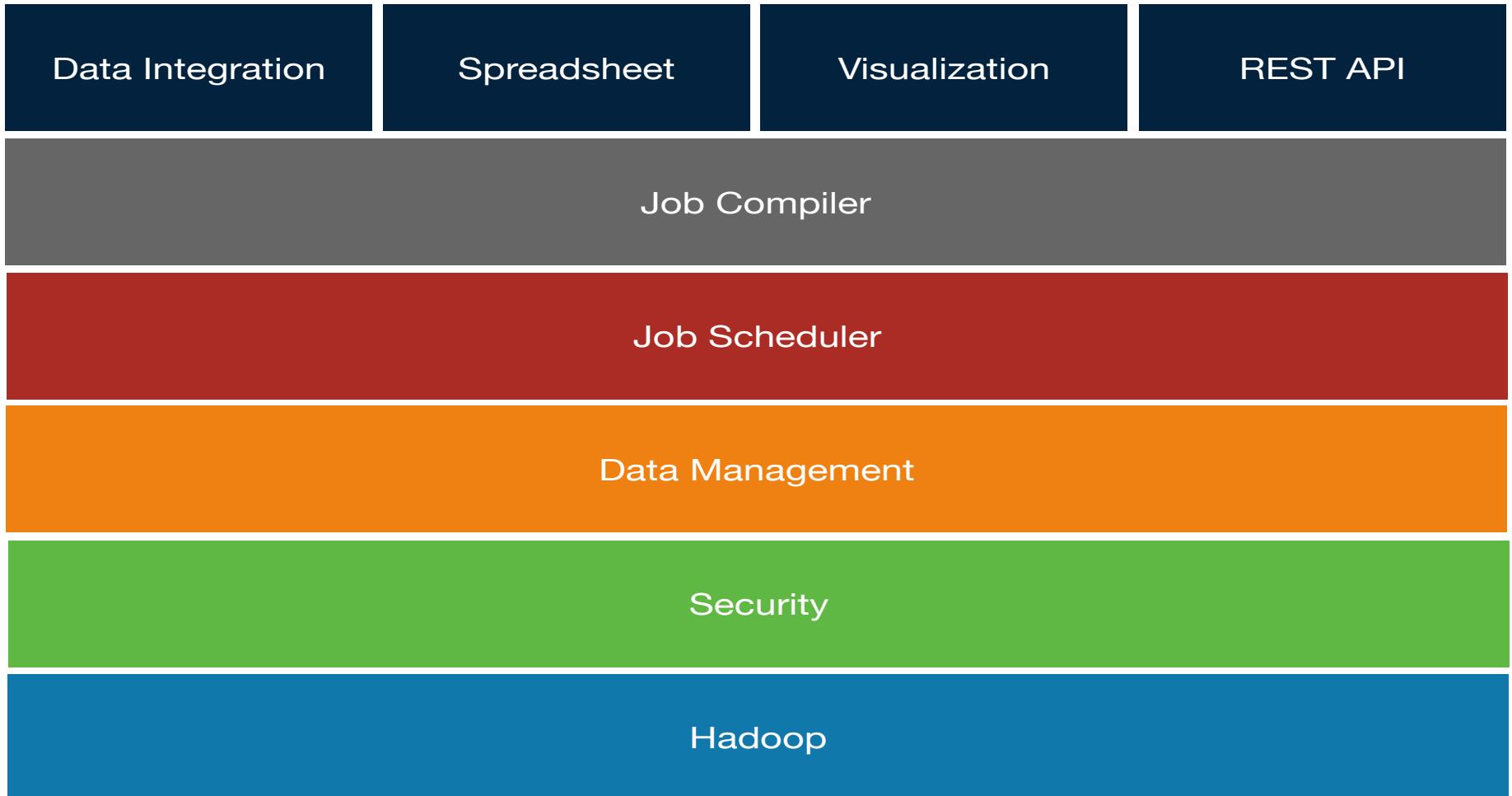
Nagios JMX

## Applications

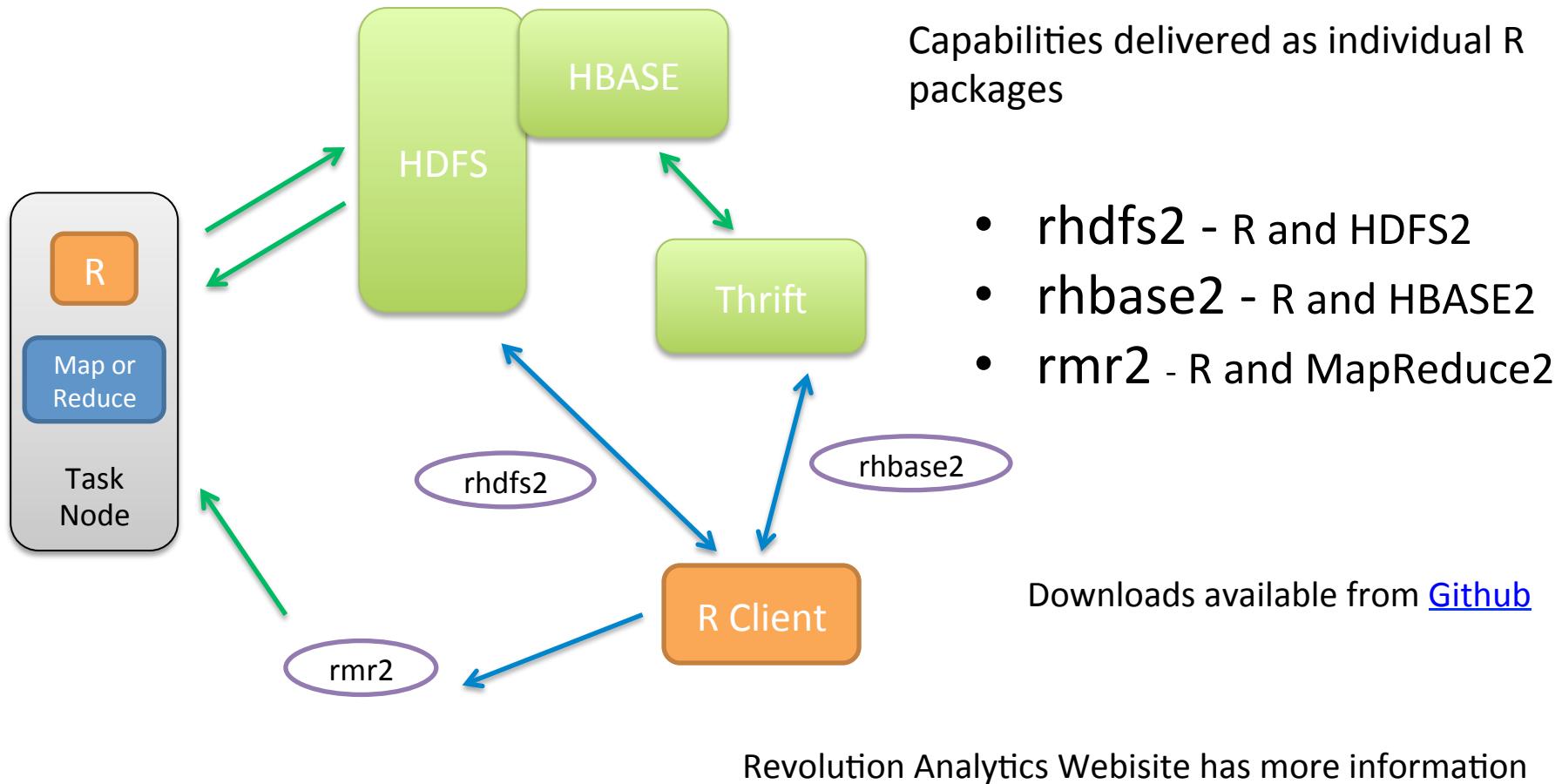


REST

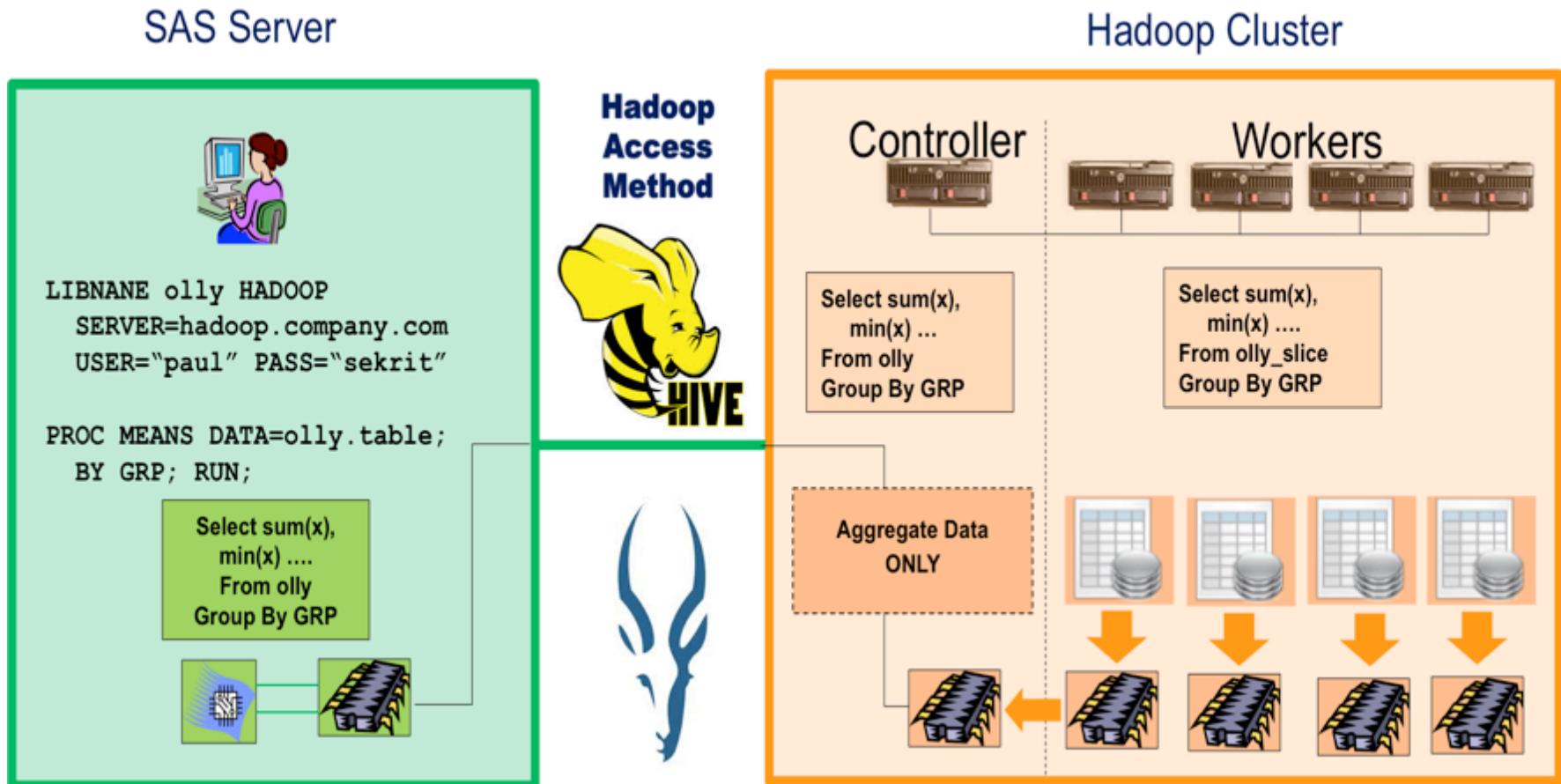
# Datameer Architecture



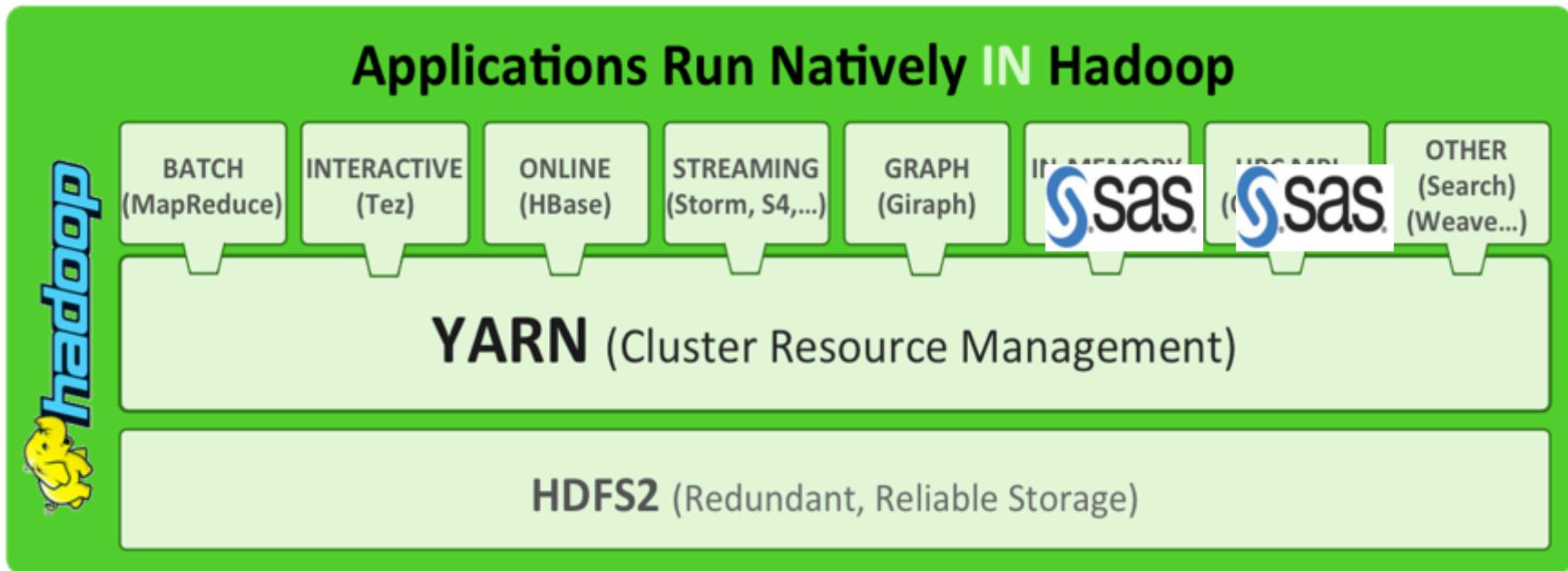
# RHadoop



# SAS-Hadoop Initial Integration



# SAS-Hadoop Integration (2015)

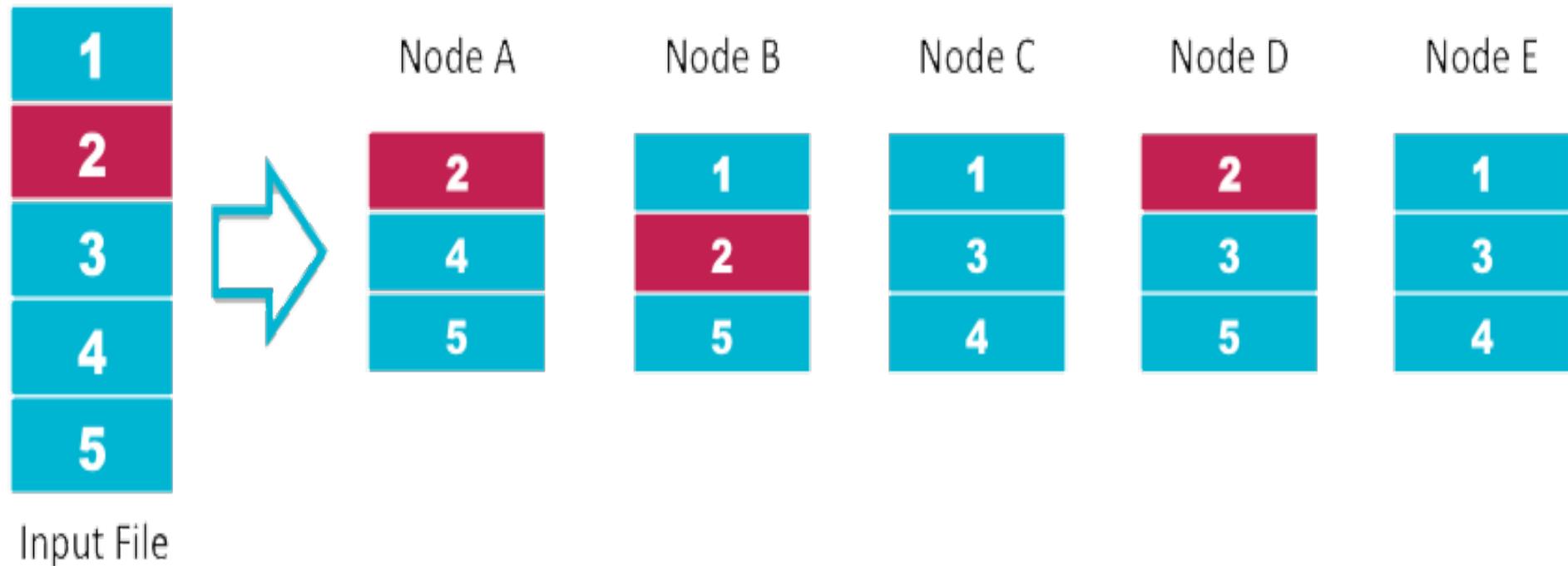


# Module 4

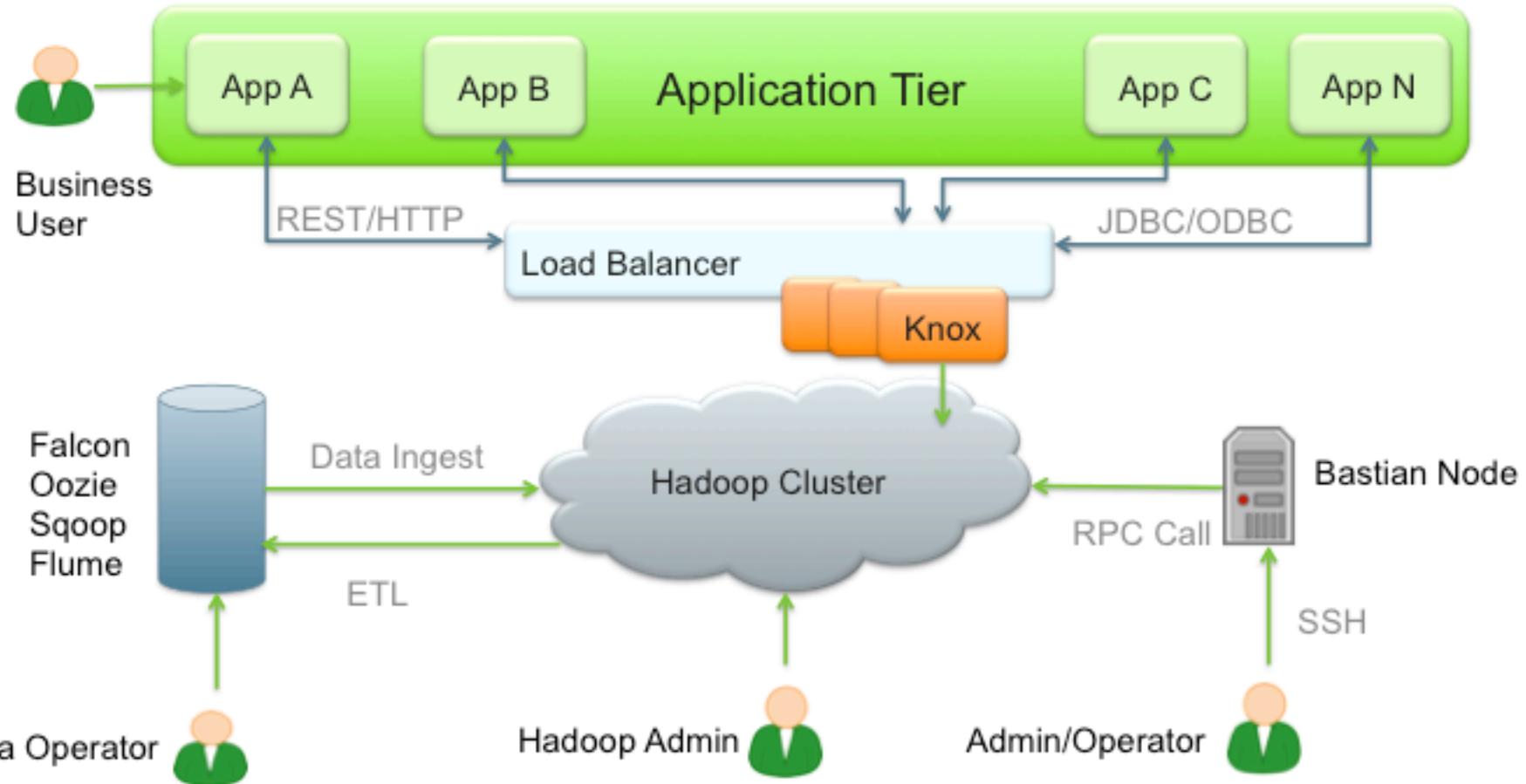
- Hadoop Configuration & Management
- Installation Planning
- Nodes
  - Multi-Distribution
  - tition Strategy
  - Security
  - Data Lifecycle
- Configuration & Setup
  - Planning Installation
  - Setup Specifics
  - Parameters to Check
  - Data Distribution Strategies
- Performance Tuning Basics
- Compression & Storage
- Replication
- Node Based Workload Management
- Management
- Users
- Data
- Nodes
- Security
- Governance

# Replication

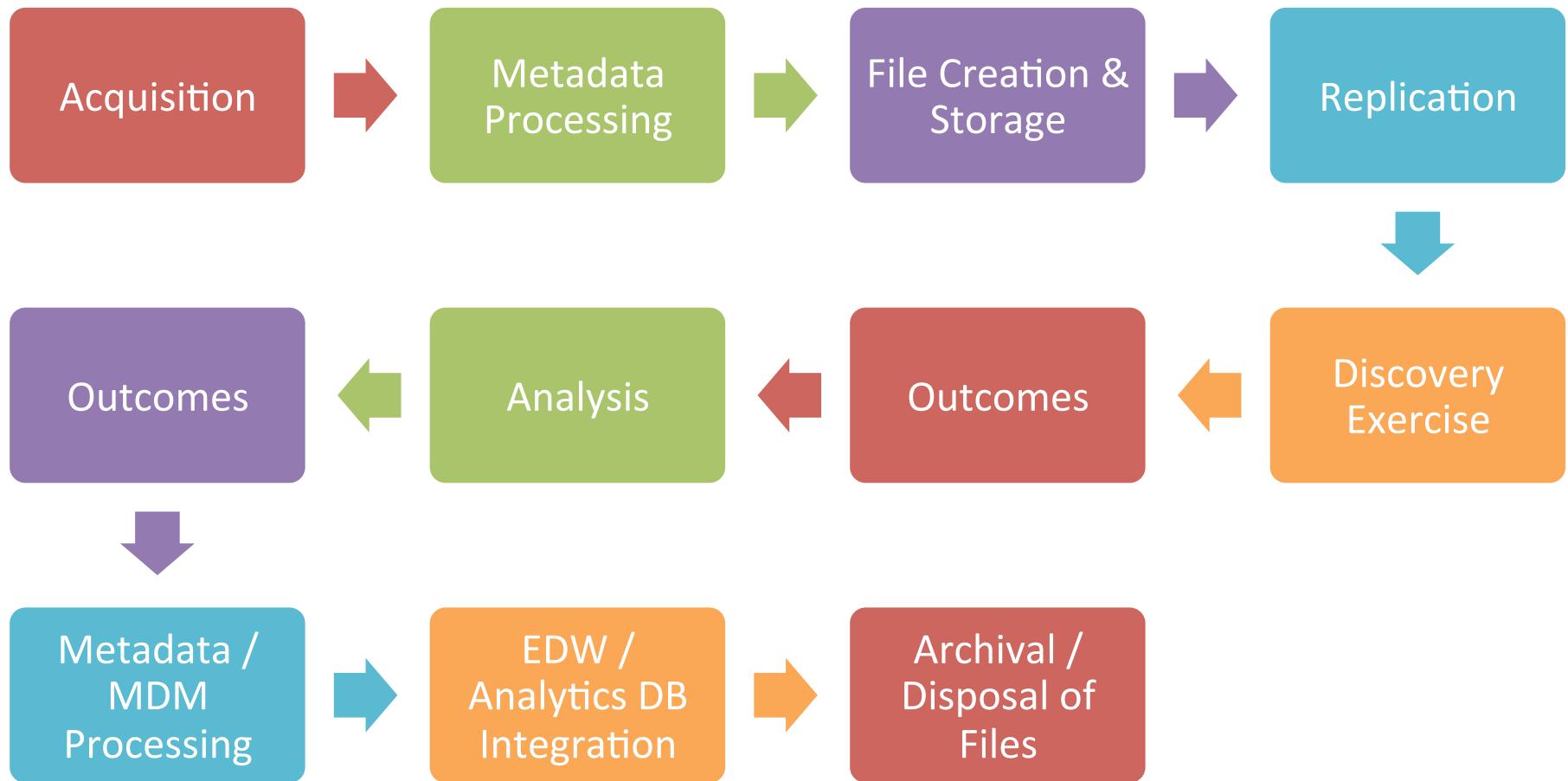
## HDFS Data Distribution



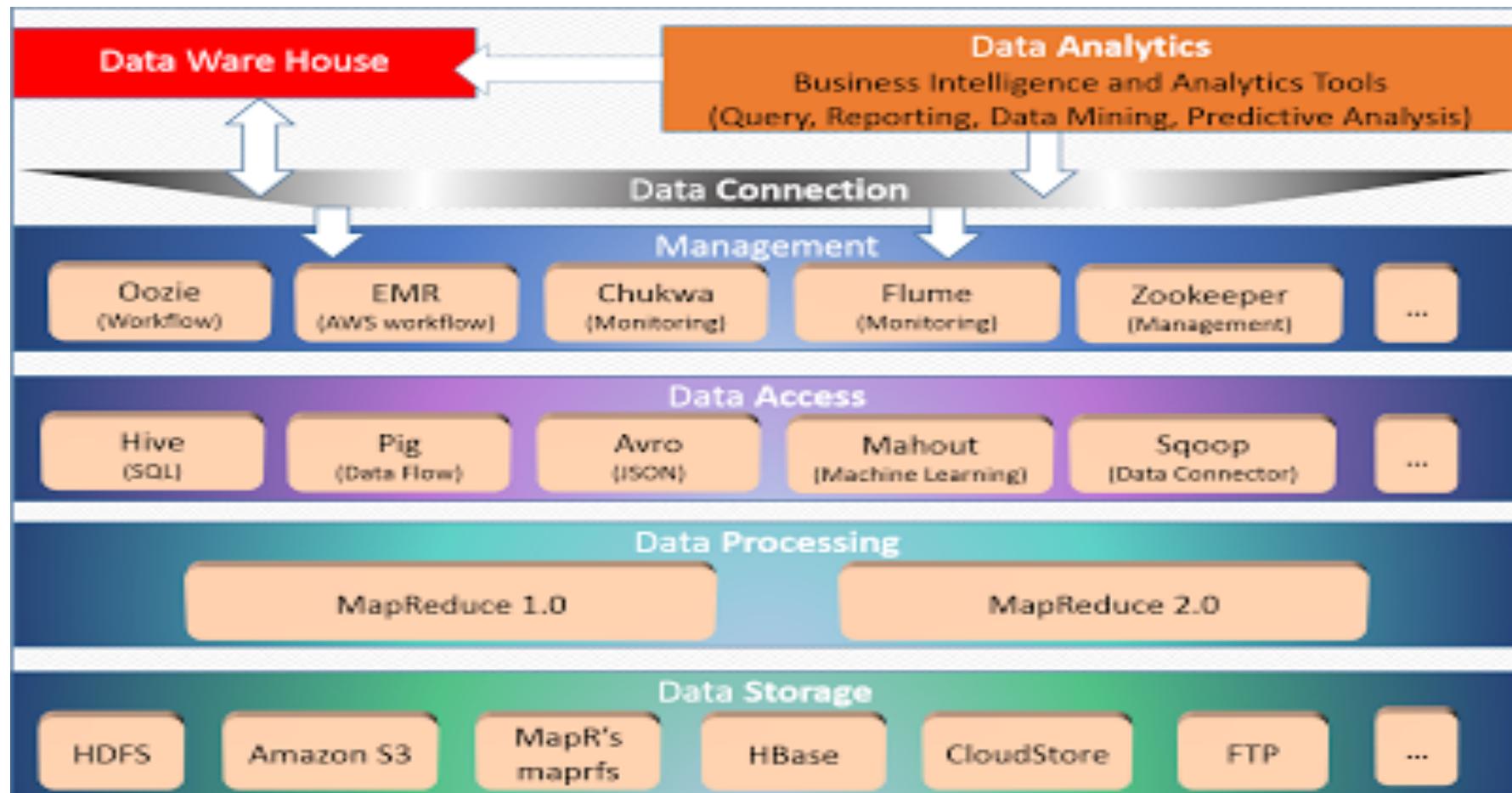
# Security



# Data Lifecycle



# Planning Installation



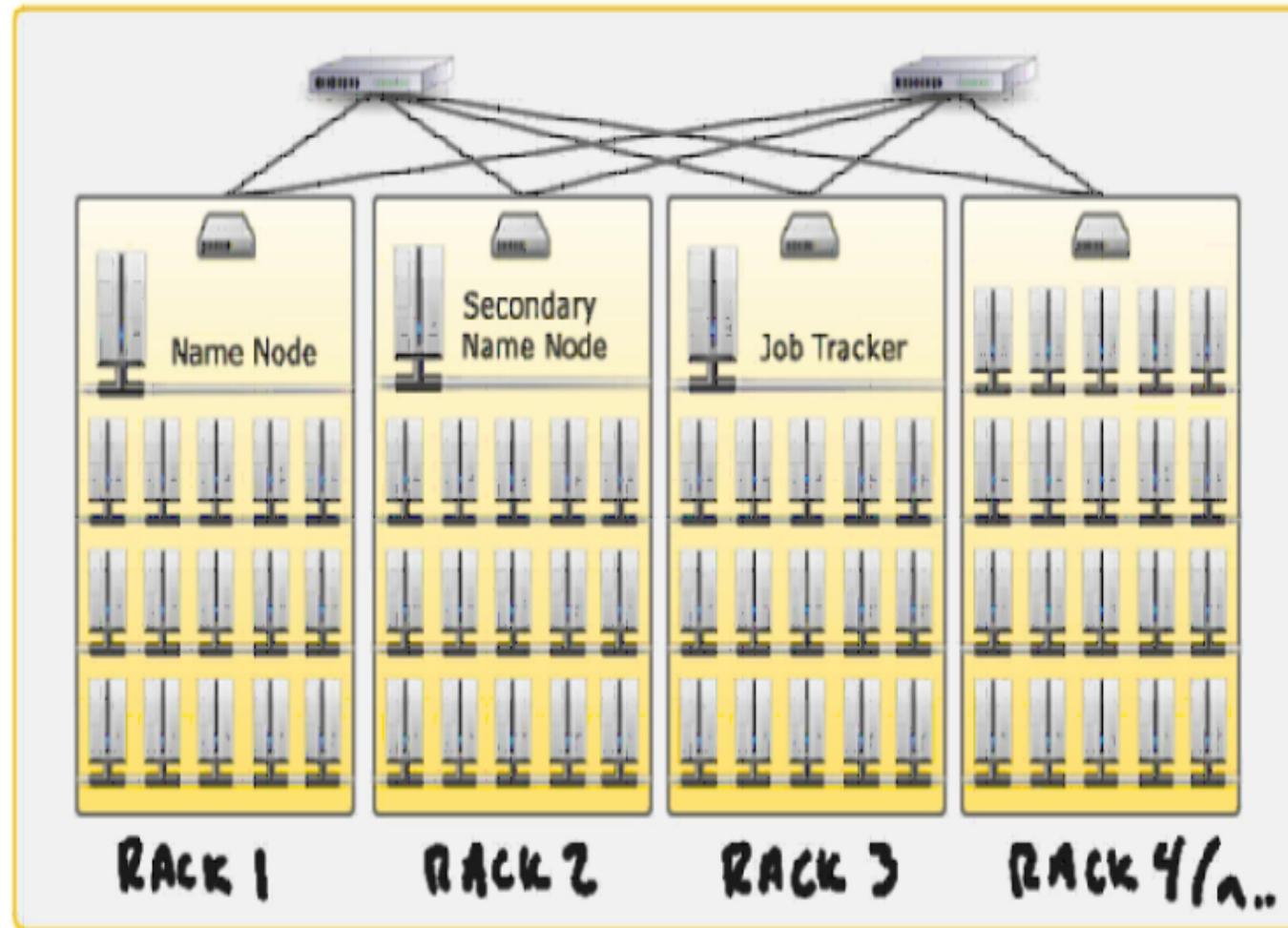
# Setup Specifics

Machine Type	Workload Pattern/ Cluster Type	Storage	Processor (# of Cores)	Memory (GB)	Network
Slaves	Balanced workload	Twelve 2-3 TB disks	8	128-256	1 GB onboard, 2x10 GBE mezzanine/external
	Compute-intensive workload	Twelve 1-2 TB disks	10	128-256	1 GB onboard, 2x10 GBE mezzanine/external
	Storage-heavy workload	Twelve 4+ TB disks	8	128-256	1 GB onboard, 2x10 GBE mezzanine/external
NameNode	Balanced workload	Four or more 2-3 TB RAID 10 with spares	8	128-256	1 GB onboard, 2x10 GBE mezzanine/external
ResourceManager	Balanced workload	Four or more 2-3 TB RAID 10 with spares	8	128-256	1 GB onboard, 2x10 GBE mezzanine/external

# Parameters To Check

- Daemons
- Memory Allocation
- Replicas
- Node Health
- Hadoop Startup
- Hadoop Shutdown
- Special Requirements

# Data Distribution



# Compression Options in Hadoop (1/2)

Format	Algorithm	Strategy	Emphasis	Comments
zlib	Uses DEFLATE (LZ77 and Huffman coding)	Dictionary-based, API	Compression ratio	Default codec
gzip	Wrapper around zlib	Dictionary-based, standard compression utility	Same as zlib, codec operates on and produces standard gzip files	For data interchange on and off Hadoop
bzip2	Burrows-Wheeler transform	Transform-based, block-oriented	Higher compression ratios than zlib	Common for Pig
LZO	Variant of LZ77	Dictionary-based, block-oriented, API	High compression speeds	Common for intermediate compression, HBase tables
LZ4	Simplified variant of LZ77	Fast scan, API	Very high compression speeds	Available in newer Hadoop distributions
Snappy	LZ77	Block-oriented, API	Very high compression speeds	Came out of Google, previously known as Zippy

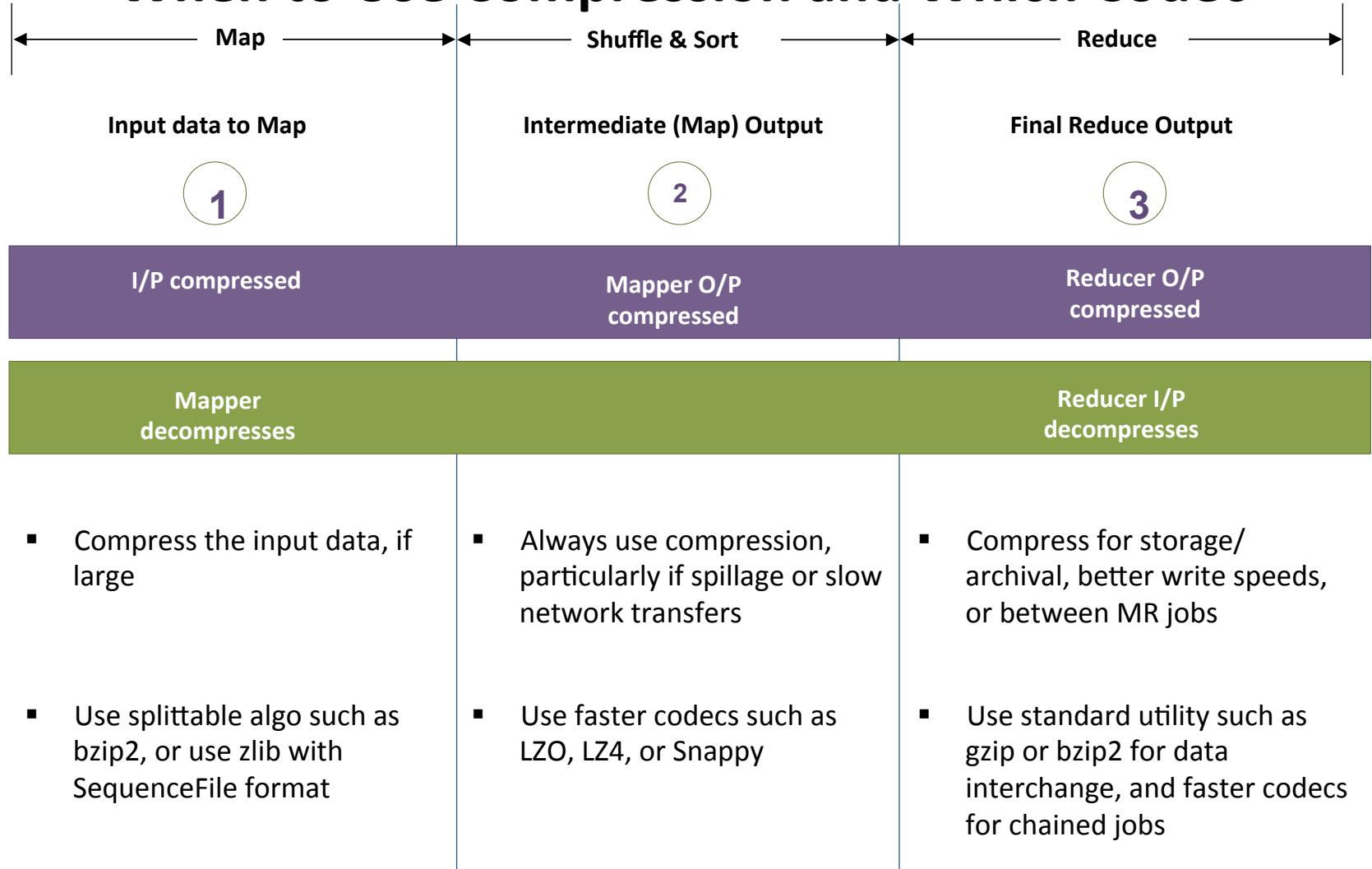
# Compression Options in Hadoop (2/2)

Format	Codec (Defined in <code>io.compression.codecs</code> )	File Extn.	Splittable	Java/ Native
zlib/ DEFLATE (default)	org.apache.hadoop.io.compress.DefaultCodec	.deflate	N	Y/ Y
gzip	org.apache.hadoop.io.compress.GzipCodec	.gz	N	Y/ Y
bzip2	org.apache.hadoop.io.compress.BZip2Codec	.bz2	Y	Y/ Y
LZO (download separately)	com.hadoop.compression.lzo.LzoCodec	.lzo	N	N/ Y
LZ4	org.apache.hadoop.io.compress.Lz4Codec	.lz4	N	N/ Y
Snappy	org.apache.hadoop.io.compress.SnappyCodec	.snappy	N	N/ Y

## NOTES:

- **Splittability** – Bzip2 is “splittable”, can be decompressed in parallel by multiple MapReduce tasks. Other algorithms require all blocks together for decompression with a single MapReduce task.
- **LZO** – Removed from Hadoop because the LZO libraries are licensed under the GNU GPL. LZO format is still supported and the codec can be downloaded separately and enabled manually.
- **Native bzip2 codec** – added by Yahoo! as part of this work in Hadoop 0.23

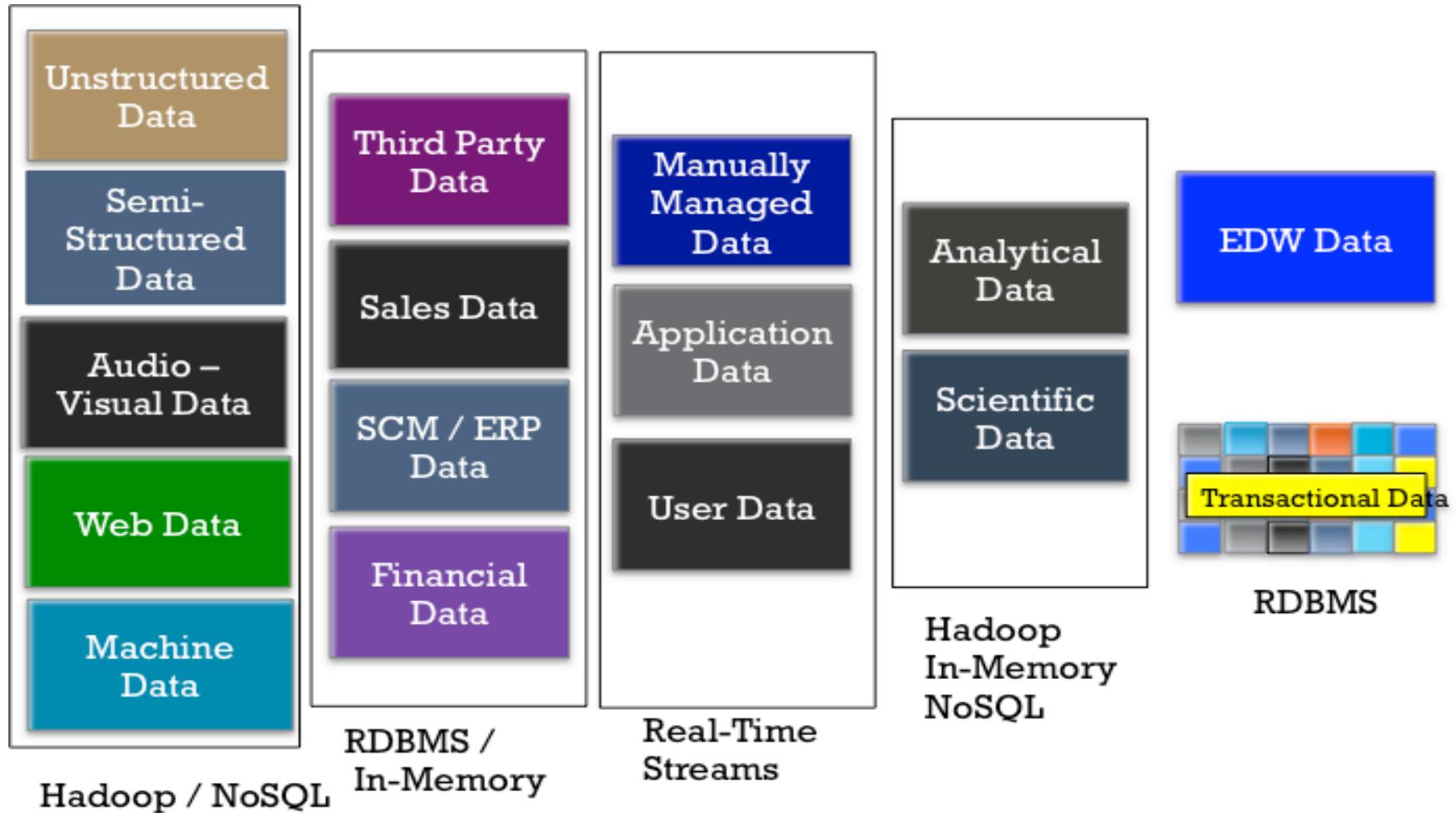
# When to Use Compression and Which Codec



Compress

Decompress

# Workload Isolation



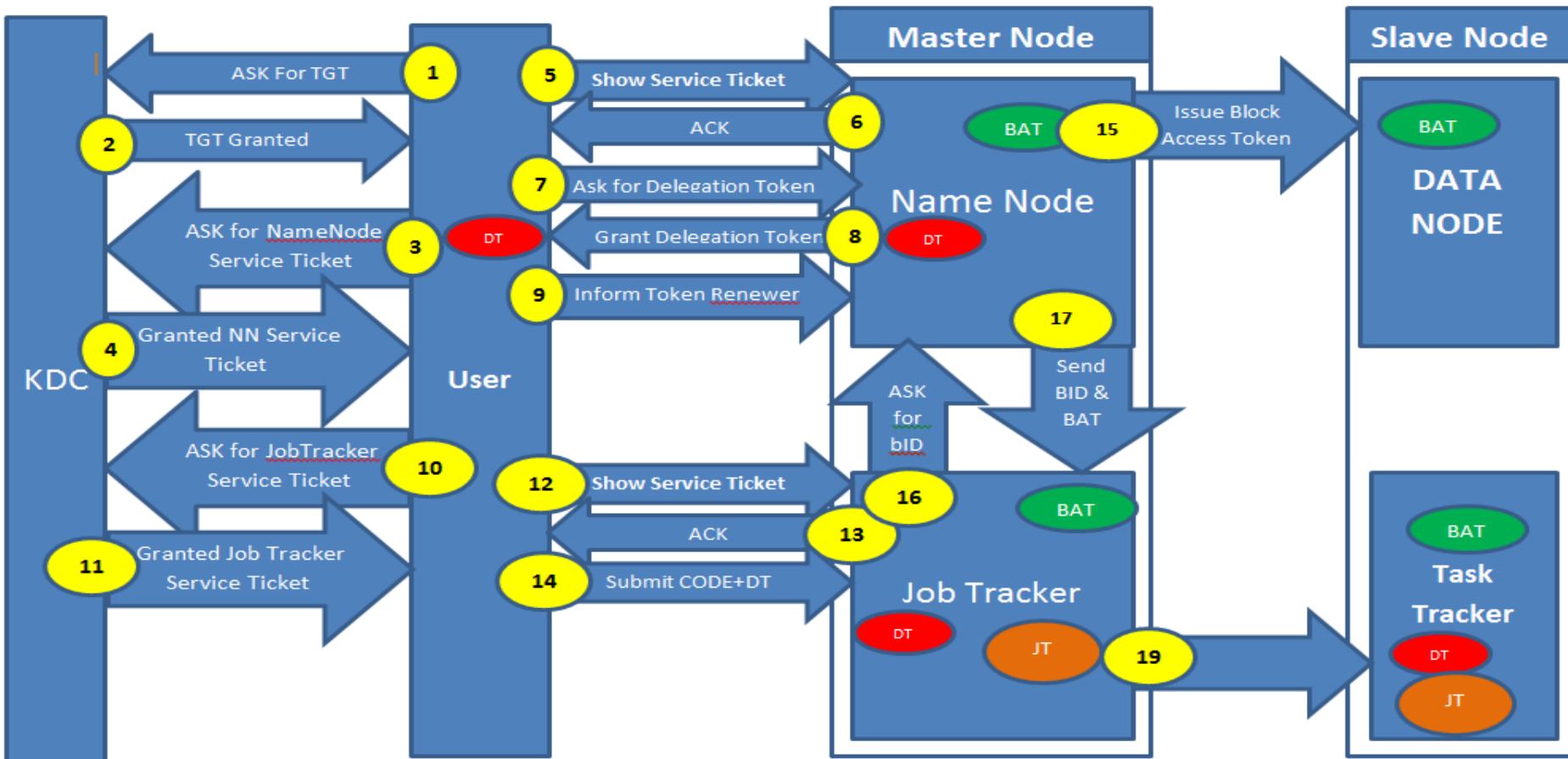
# Users



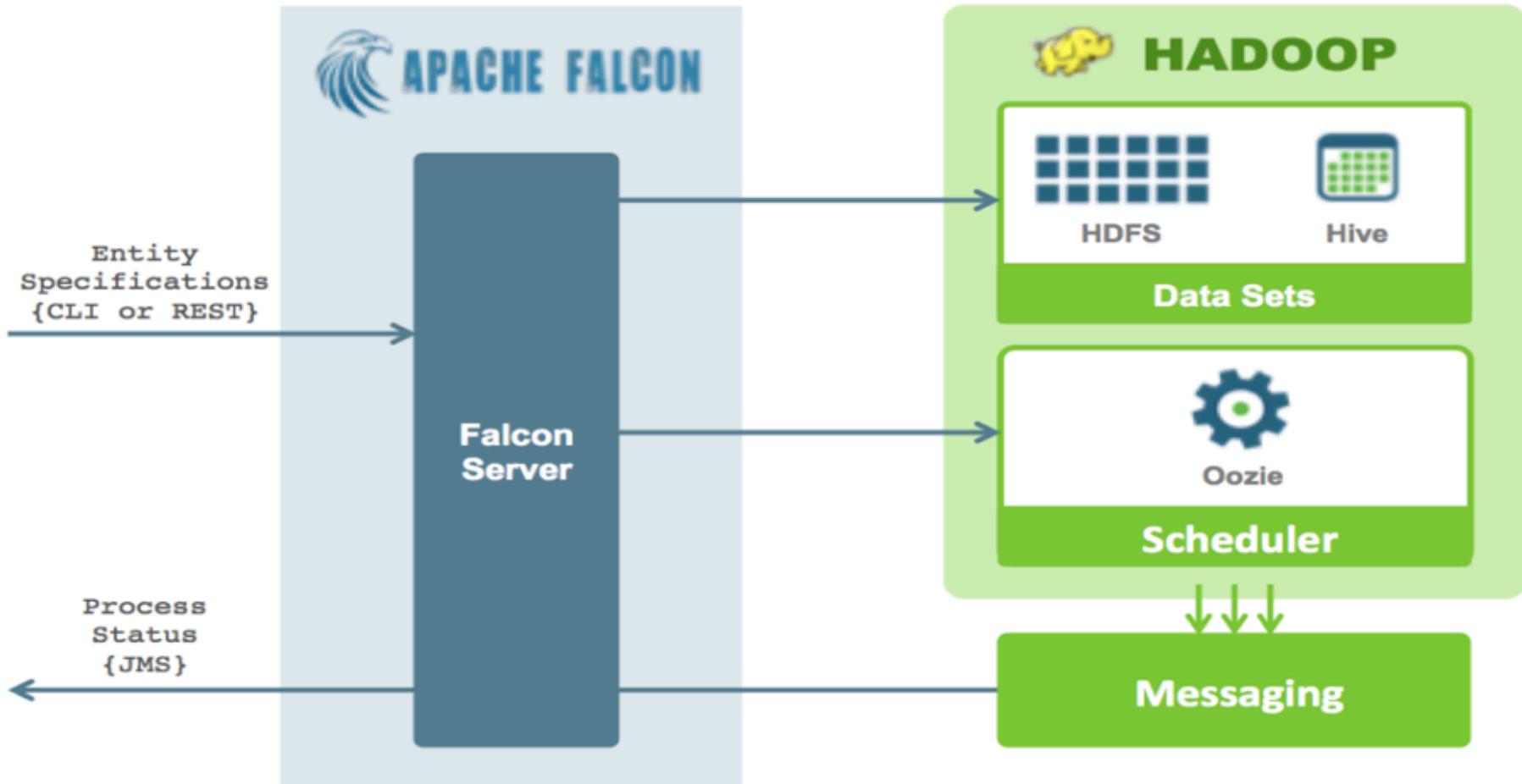
# Data



# Hadoop Security

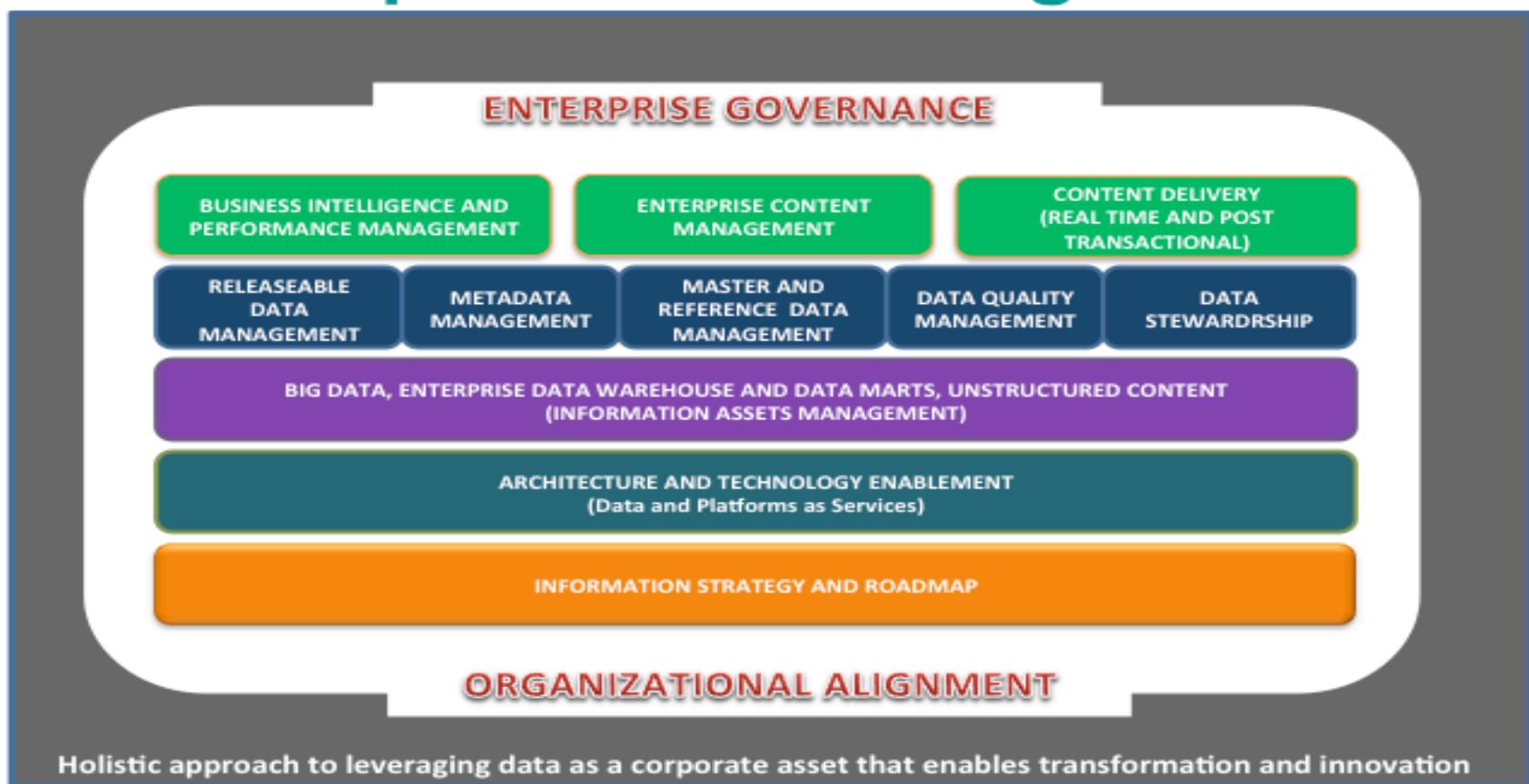


# Governance



# Data Governance

## Great Data by Design: Enterprise Data Management



# Module 5

Examples & Case Study

Initial Success Stories

Healthcare

Financial Services

Internet Companies

Retail

Current State Usage & Application

Best Practices Approach

Pitfalls to Avoid

Government Usage & Success

Data.Gov from US Government

UK Government

Implementation Pain Points

A Lifecycle Case Study

# Value from Hadoop

1

**Create transparency**



2

**Expose variability and enable experimentation**



3

**Segment populations to customize actions**



4

**Replace/support human decision-making  
with automated algorithms**

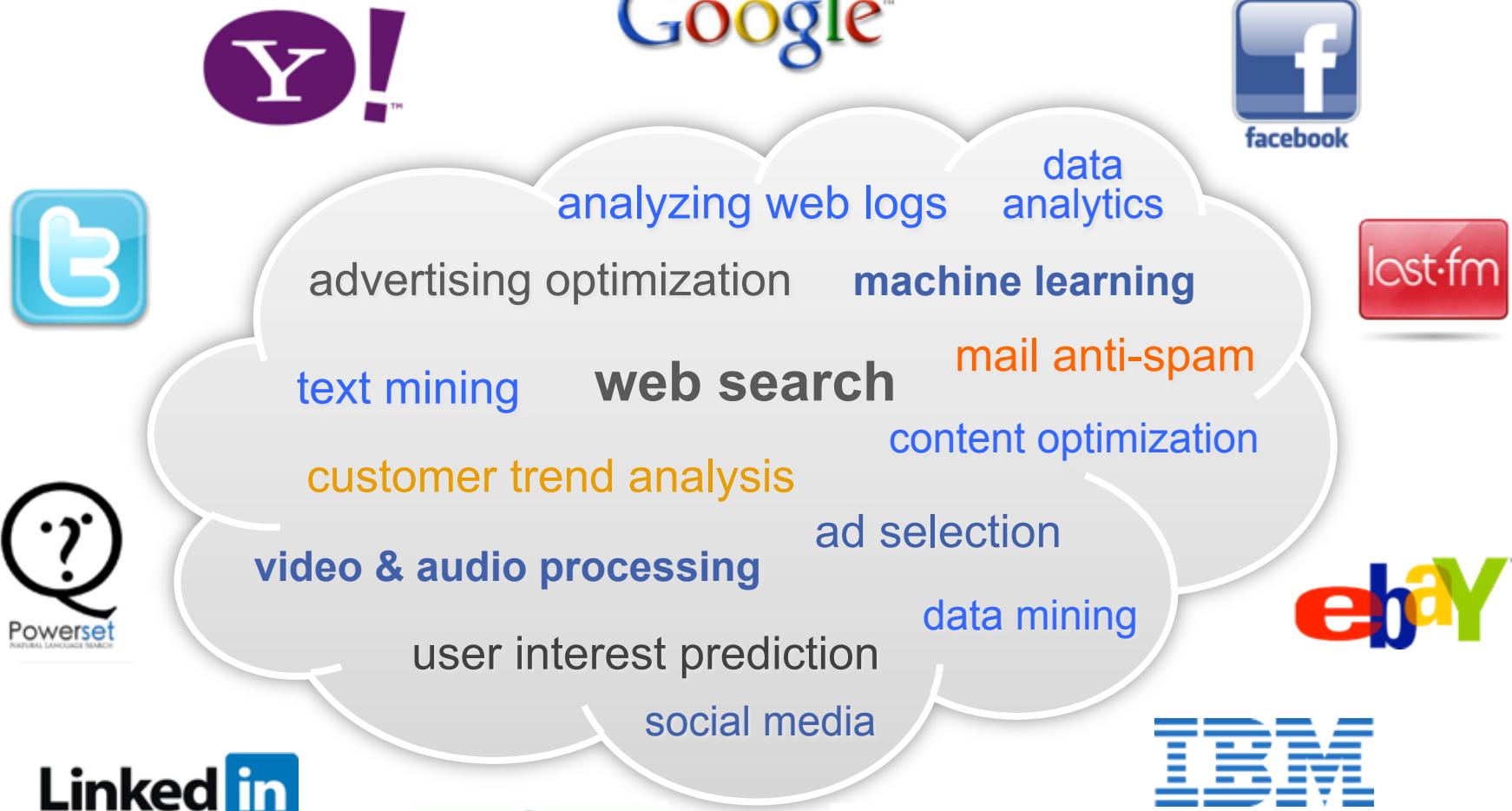


5

**Innovate new business models, products,  
and services**



# Early Adopters



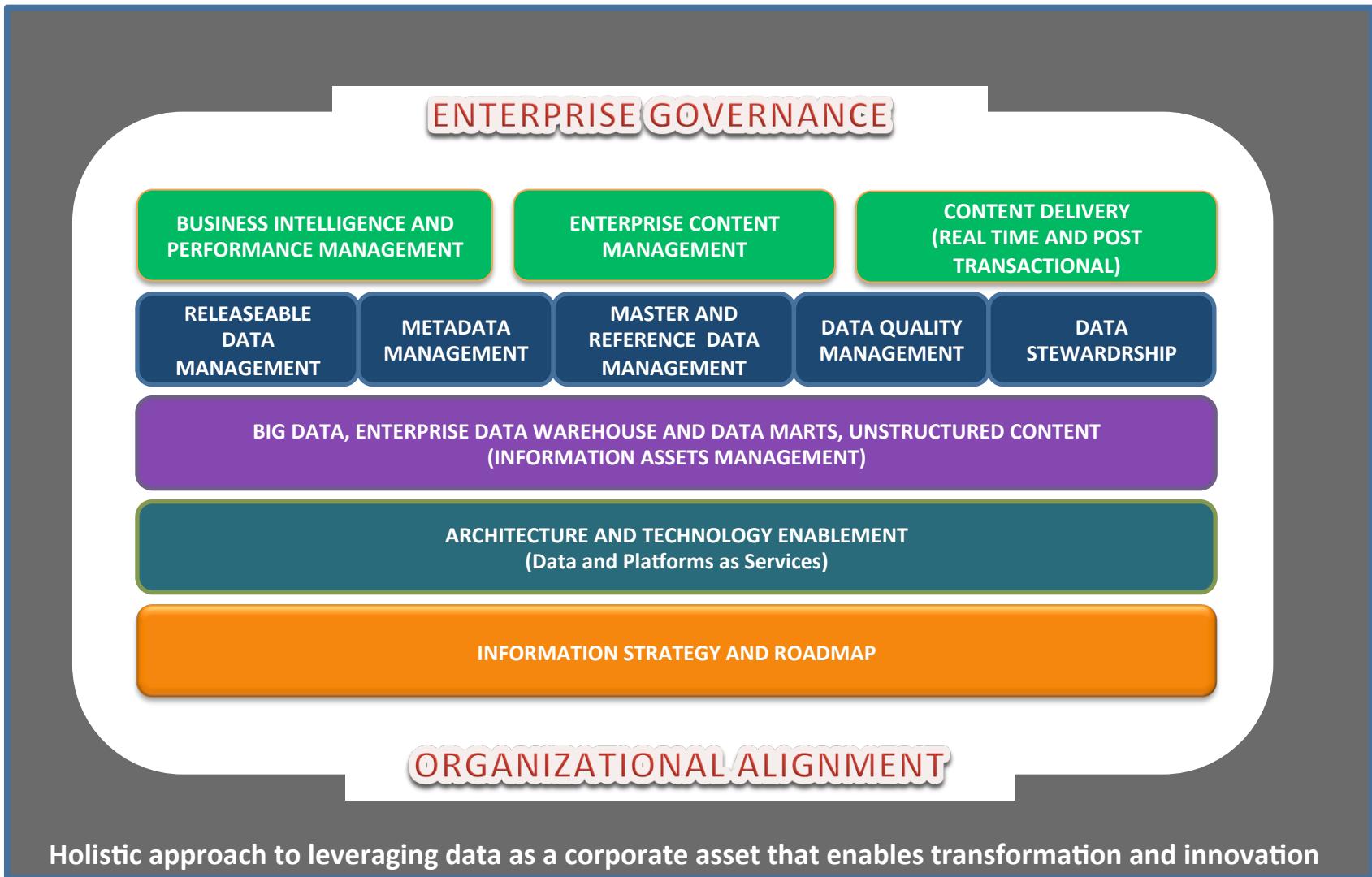
# Enterprise Data Management - Strategy

***Goal: To anticipate the needs of our patients, clients and operations and advance the integration of our information assets, business policies, development processes and tools that enable and support emerging health care delivery models.***

## **Plan:**

- ✓ ***Establish a formalized governance and stewardship structure around policies and priorities***
- ✓ ***Establish a formalized structure around data priorities, tools and analytic needs***
- ✓ ***Create an enablement roadmap*** that aligns our people, process and technology
- ✓ ***Build an information foundation*** that is integrated, managed, governed, trusted and sustainable
- ✓ ***Equip the clients*** with the right information and unified tools to drive meaningful action

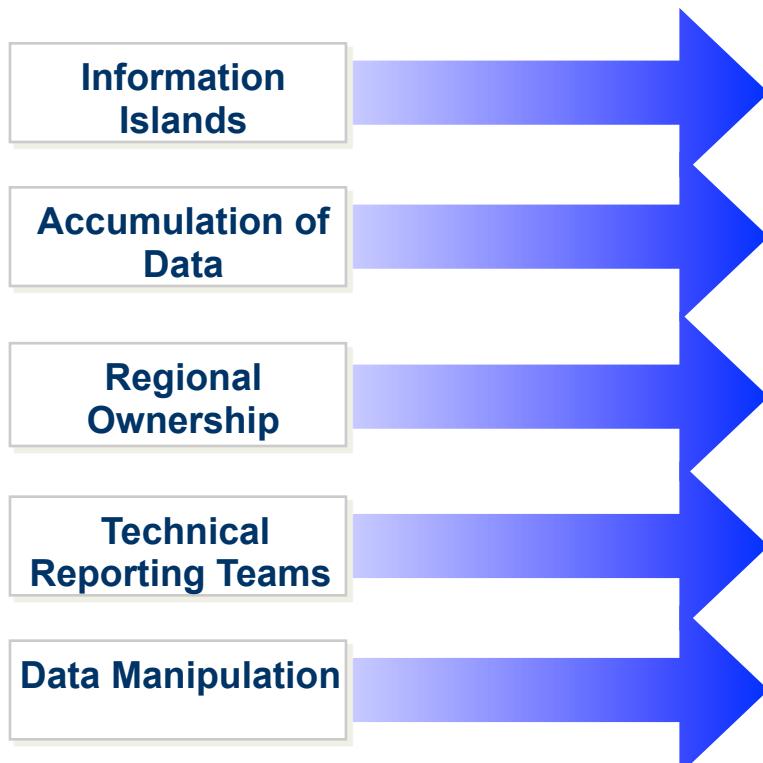
# Great Data by Design: Enterprise Data Management



# Enterprise Data Management - Benefits

*Execution will position us to exceed customer needs and become a industry leader*

## Information Independence



## Information Inter-dependence

*Shared information supported by common data definitions, tools, and access methods*

*Business owners will be notified of unfavorable trends or conditions*

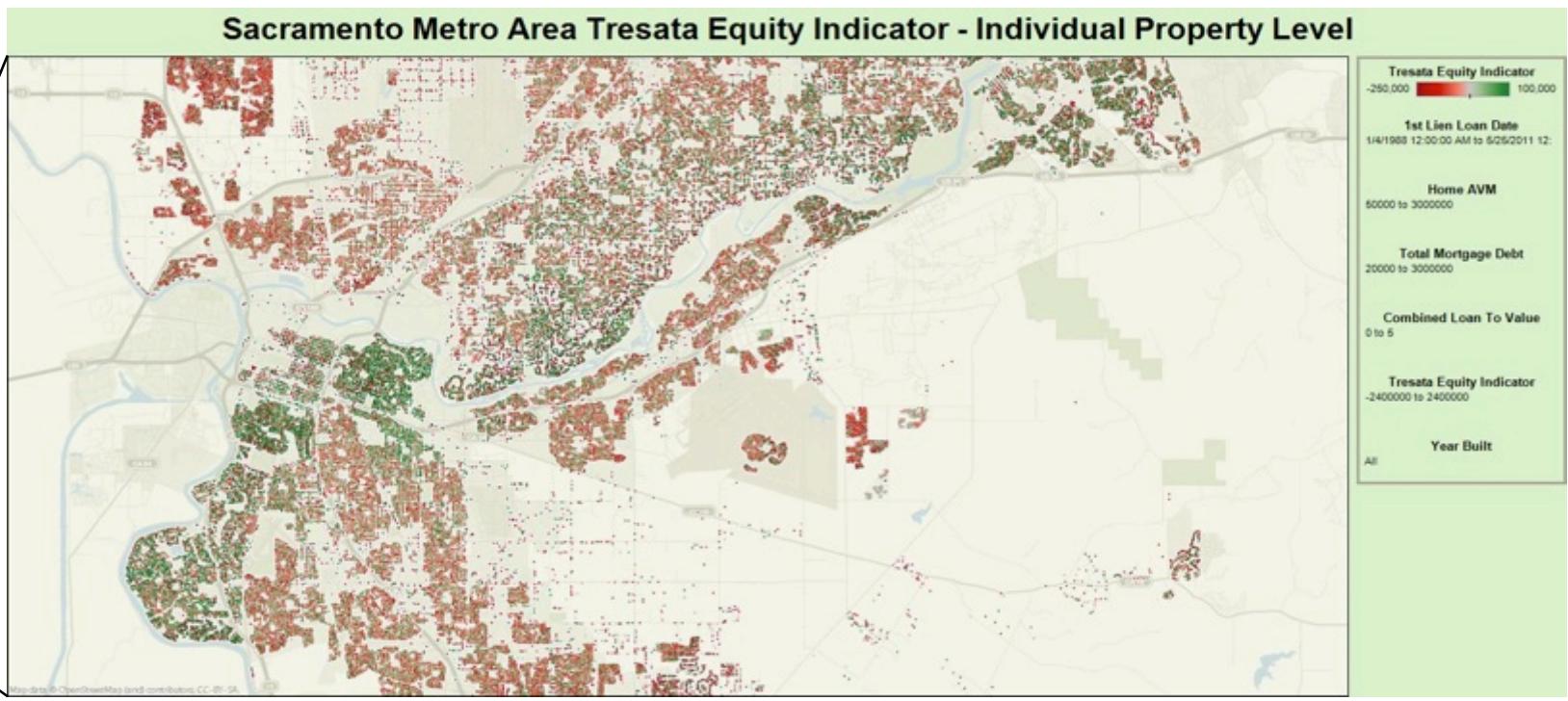
*New projects follow an established set of guiding principles*

*Next generation tools enabling forward looking insight and modeling*

*Business users receive timely information that is tailored to their needs*

# Banking Story

## Tresata Analytics



# Tresata Big Data Application

Welcome Nitin Kak   My Profile   Logout

FACTORY   EXECUTE   EXPLORE   ANALYZE   VISUALIZE   MONITOR   ADMINISTER

Credit Score Analysis		
Tresata AL	Status	Time
Load	●	0 Min
De-Dup	●	0 Min
Clean ID	●	0 Min
Match	●	0 Min
Join	●	0 Min
Analyze	●	0 Min
Score	●	0 Min

Total Time: 0 Min   Execute Job

Customer Loan Analysis		
Tresata AL	Status	Time
Load	●	10 Min
De-Dup	●	200 Min
Clean ID	●	9 Min
Match	●	21 Min
Join	●	20 Min
Analyze	●	20 Min
Score	●	20 Min

Total Time: 290 Min   Kill Job

Loan Deposit		
Tresata AL	Status	Time
Load	●	20 Min
De-Dup	●	100 Min
Clean ID	●	30 Min
Match	●	30 Min
Join	●	40 Min
Analyze	●	50 Min
Score	●	60 Min

Total Time: 330 Min   Kill Job

Loan Recovery		
Tresata AL	Status	Time
Load	●	10 Min
De-Dup	●	20 Min
Clean ID	●	10 Min
Match	●	0 Min
Join	●	0 Min
Analyze	●	100 Min
Score	●	0 Min

Total Time: 140 Min   Finished

Customer Loan Analysis		
Tresata AL	Status	Time
Load	●	10 Min
De-Dup	●	200 Min
Clean ID	●	9 Min
Match	●	21 Min
Join	●	0 Min
Analyze	●	0 Min
Score	●	0 Min

Total Time: 240 Min   Failed

Loan Deposit		
Tresata AL	Status	Time
Load	●	20 Min
De-Dup	●	100 Min
Clean ID	●	30 Min
Match	●	30 Min
Join	●	40 Min
Analyze	●	50 Min
Score	●	60 Min

Total Time: 330 Min   Kill Job

© tresata. All Rights Reserved

# Banking Case Study

## A. Client Business Problem

- i. Problem – Process data and score for **>30 MM** client applications
- ii. Data Sources – **23** separate data sources, **multiple time series**
- iii. Raw Variables – **100 variables** per client per data source
- iv. Current state - **expensive** legacy platform, algorithms developed on **sub-samples, unable to scale** algorithms to full data set

## B. Tresata Solution

- i. Data Engine – **automated** data import, cleaning, matching, scoring
- ii. Compute Engine – algorithms process & score **>30MM** in **minutes**
- iii. Integration – work with existing tools and processes
- iv. Scalable deployment – **Big Data as a Service** delivery

# Internet Companies



# YAHOO!

# Retail

- Early Adopters
- Large analytics on pricing and demand fulfillment
- Social CRM
- Real-Time Campaigns
- In-Store promotions



# US Government

## DATA.GOV

[HOME](#) | [CATALOG](#) | [ABOUT](#) | [FAQ](#) | [CONTACT US](#) | [SUGGEST OTHER DATASETS](#) |

DISCOVER.  
PARTICIPATE.  
ENGAGE.

Search Data.gov catalog by category, agency, or both

[All Categories](#)

[All Agencies](#)

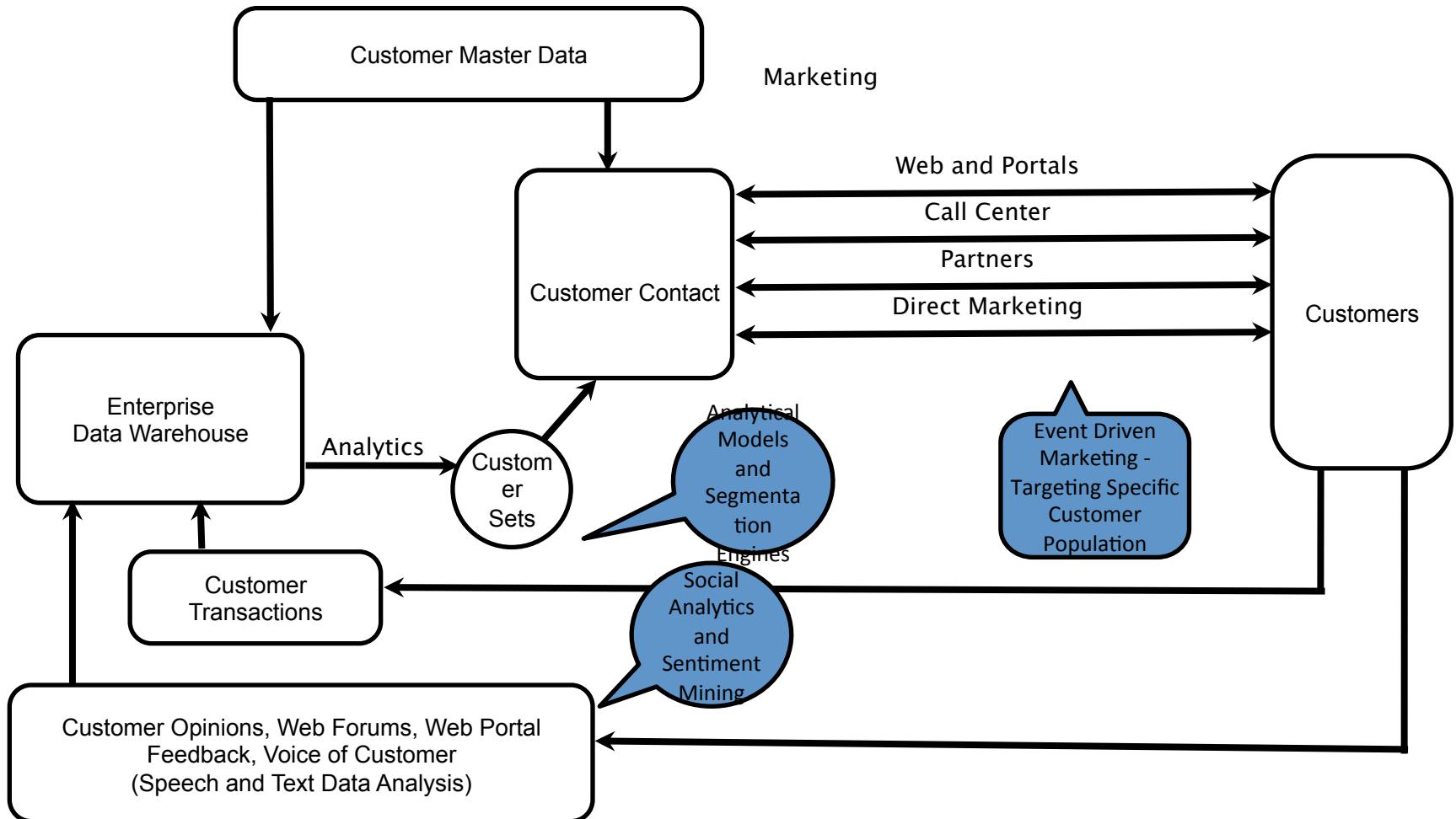
FEATURED DATA

ENERGY INFO

Residential Energy Co



# Implementation Case Study



# Key Points

- **HADOOP**
  - Is the infrastructure for the next generation data management in the enterprise
  - A lot of projects are in-process for the ecosystem for making Hadoop enterprise ready
  - Data governance is a key project to implement in Hadoop
  - Semantic data and workflows need to be implemented for each project
  - Infrastructure planning needs to be completed for the program from the end of the POC cycle
  - Replication strategy is a critical program
  - Learn to adopt open source strategies on sizing and implementation

# First Project

- People
  - Executive Sponsorship
    - Budget
    - Vision
    - Scope
    - Success Criteria
  - Business SME
    - Participation
    - Data Discovery
    - Analytics
    - KPI
    - Statistics
  - IT
    - Infrastructure Configuration
    - Security
    - Network
    - Support
  - Data Teams
    - Management of Data
    - Acquisition
- Discovery
  - Analysis
  - Outcomes
- Process
  - Data
    - Acquisition
    - Metadata
    - Semantic Tagging
    - Contextualization
    - Extraction
    - Analytics
    - Archival
    - Disposal
  - Culture
    - Governance
    - Stewardship
    - Communications
- Technology
  - POC
    - Vendors Selection
    - Use Cases and Data
- Sharing
  - POC Execution
  - Results analysis
- Vendor Selection
  - POC Outcomes
  - Procurement
  - Installation
  - Configuration
  - Security Rules
  - Users
  - Development Environment Readiness

# Comparison of Vendors

	Hortonworks	Cloudera	MapR
<b>Performance and Scalability</b>			
Data Ingest	Batch	Batch	Batch and streaming writes
Metadata Architecture	Centralized	Centralized	Distributed
HBase Performance	Latency spikes	Latency spikes	Consistent low latency
NoSQL Applications	Mainly batch applications	Mainly batch applications	Batch and online/real-time applications
<b>Dependability</b>			
High Availability	Single failure recovery	Single failure recovery	Self healing across multiple failures
MapReduce HA	Restart jobs	Restart jobs	Continuous without restart
Upgrading	Planned downtime	Rolling upgrades	Rolling upgrades
Replication	Data	Data	Data + metadata
Snapshots	Consistent only for closed files	Consistent only for closed files	Point-in-time consistency for all files and tables
Disaster Recovery	No	File copy scheduling (BDR)	Mirroring

# Comparison of Vendors

Manageability			
Management Tools	Ambari	Cloudera Manager	MapR Control System
Volume Support	No	No	Yes
Heat map, Alarms, Alerts	Yes	Yes	Yes
Integration with REST API	Yes	Yes	Yes
Data and Job Placement Control	No	No	Yes
Data Access			
File System Access	HDFS, read-only NFS	HDFS, read-only NFS	HDFS, read/write NFS (POSIX)
File I/O	Append only	Append only	Read/write
Security: ACLs	Yes	Yes	Yes
Wire-level Authentication	Kerberos	Kerberos	Kerberos, Native

# Roadmap Components

Scope

Goals

Data

Technology

Governance

Projects

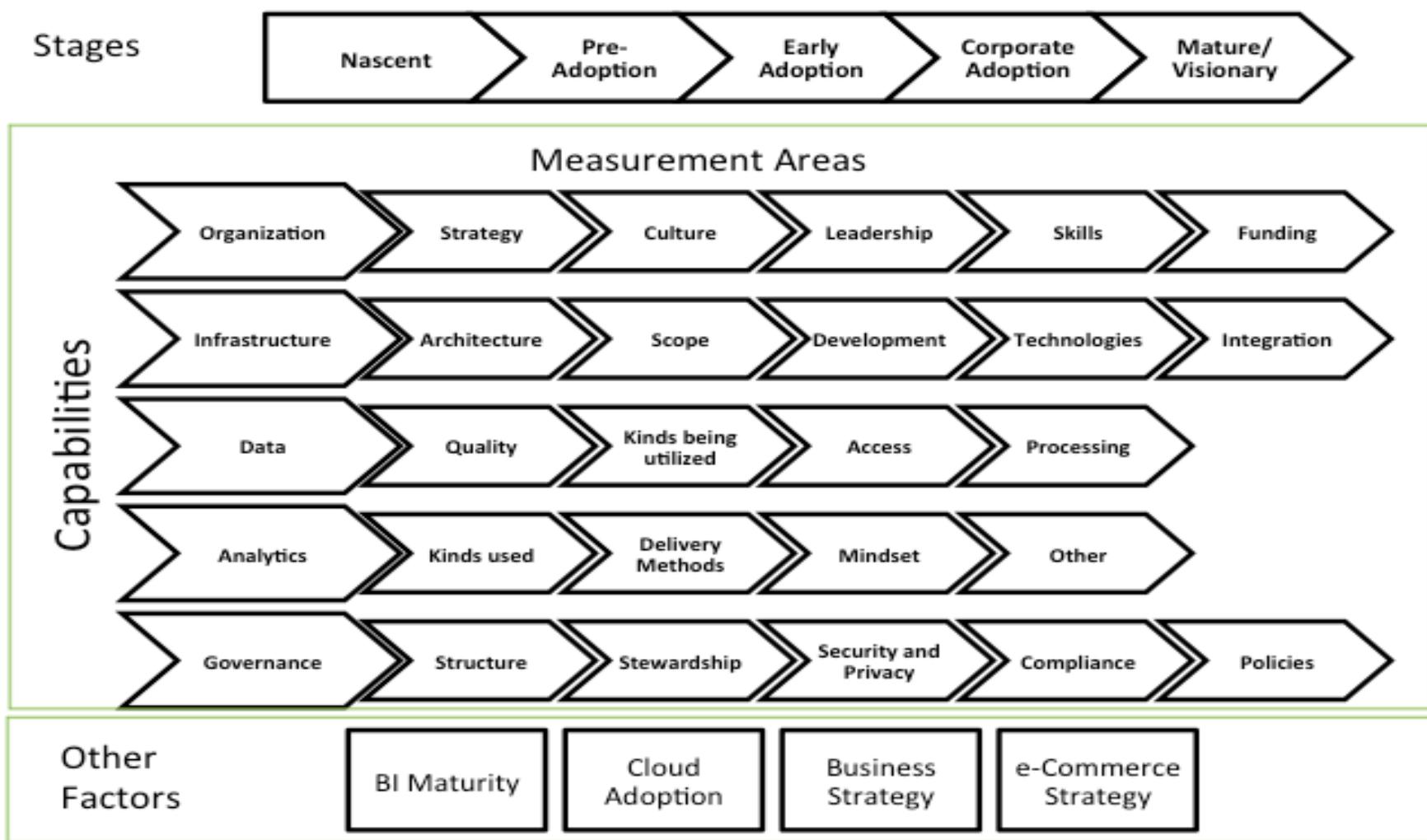
Steering  
Committee

Budget – For  
Programs

Rollout

Maintenance

# TDWI Big Data Analytics Maturity Model Framework



# **REFERENCE**

# Whitepapers, Articles & Plus

- What is NASA doing with Big Data today? – Nick Skytland
- DARPA Calls for Advances in “Big Data” to Help the Warfighter
- 1000 Genomes Project data available on Amazon Cloud - National Human Genome Research Institute (NHGRI)
- Big Data : What\_It\_Is\_and\_Why\_You\_Should\_Care
- Big Data EDW - Big Data Analytics - Dr Ralph Kimball
- BigDataPrimer-IBM
- Capitalize on the Power of Big Data to Transform Marketing - Sponsored by IBM 2013-03-11
- Managing Data as a Strategic Corporate Resource - A 25 Year Retrospective and Directions for the Future
- The\_Forester\_Wave\_Big Data\_2012
- Keynote: Big Data - Are You Ready?

# Whitepapers, Articles & Plus

- MGI\_big\_data\_full\_report(2).pdf
- IDC\_AMD\_Big\_Data\_Whitepaper.pdf
- Opening Our Eyes in the Era of Big Data
- BarryHurd- Facebook – Big Data – Our Privacy
- ZD Net – Web Data is Big Data
- NY Times – Big Data troves stay forbidden to social scientist
- ComputerWorld – VC Funds pour into Big Data vendors
- Fast Co Design – How Companies like Amazon use big data to make you love them
- InfoWorld – Big Data in the Cloud, Its time to Experiment
- WSJ – What They Know

# Thanks To

- Google Papers and Articles on Spanner and Dremel
- UC Berkeley Articles on Spark and Shark
- MIT Innovation Labs
- Google Innovation Labs
- Big Data Vendors
  - Gartner
  - HortonWorks
  - Pivotal
  - Cloudera
  - MapR
  - Platfora
  - Datameer
- Forrester