Introduction to Hadoop Tools

Timothy Spann

2017 Future of Data – Princeton Meetup
May 16, 2017

HORTONWORKS®
POWERING THE FUTURE OF DATA™

Agenda

- 6:00pm – 6:45pm Registration and Food
- 6:45pm – 7:00pm Introduction and Welcome
- 7:00pm – 7:45 pm Install and Cloud by Milind Pandit
- 7:45pm – 8:30 pm Tools by Tim Spann
- 8:30pm – 8:45pm Questions

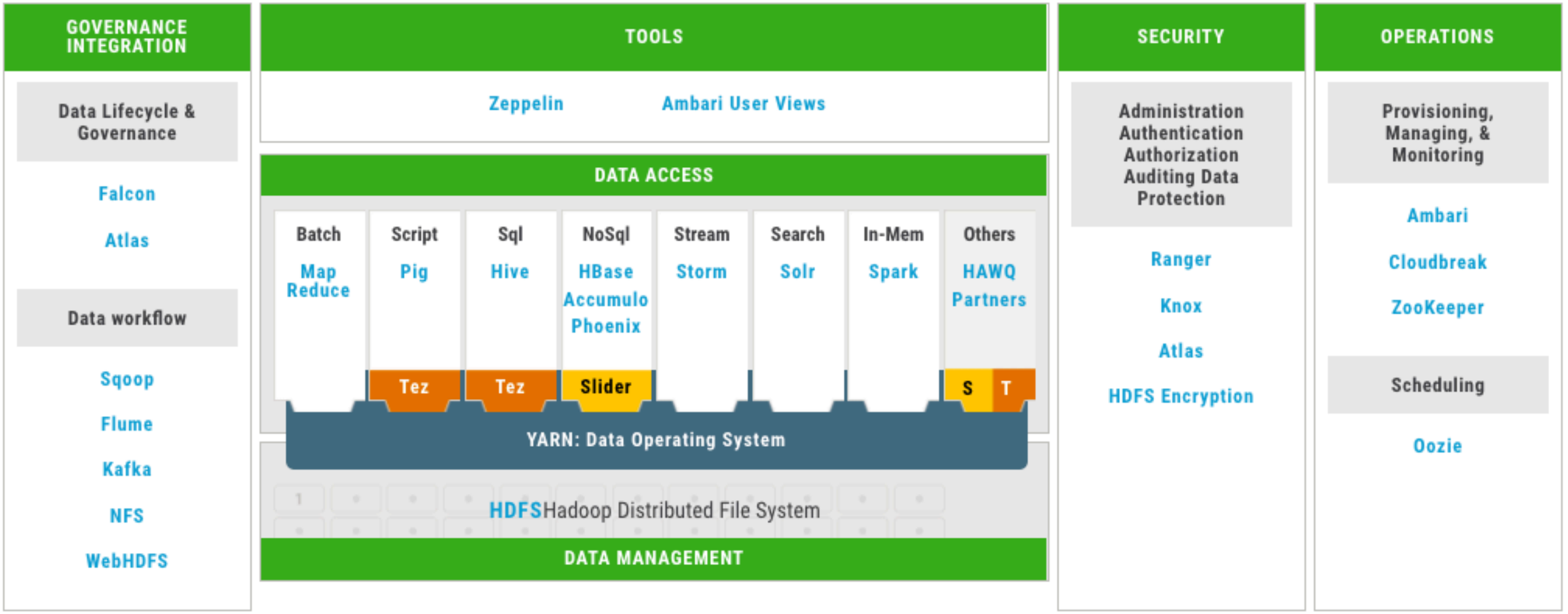
HORTONWORKS

All Your Data Are Belong to Hadoop

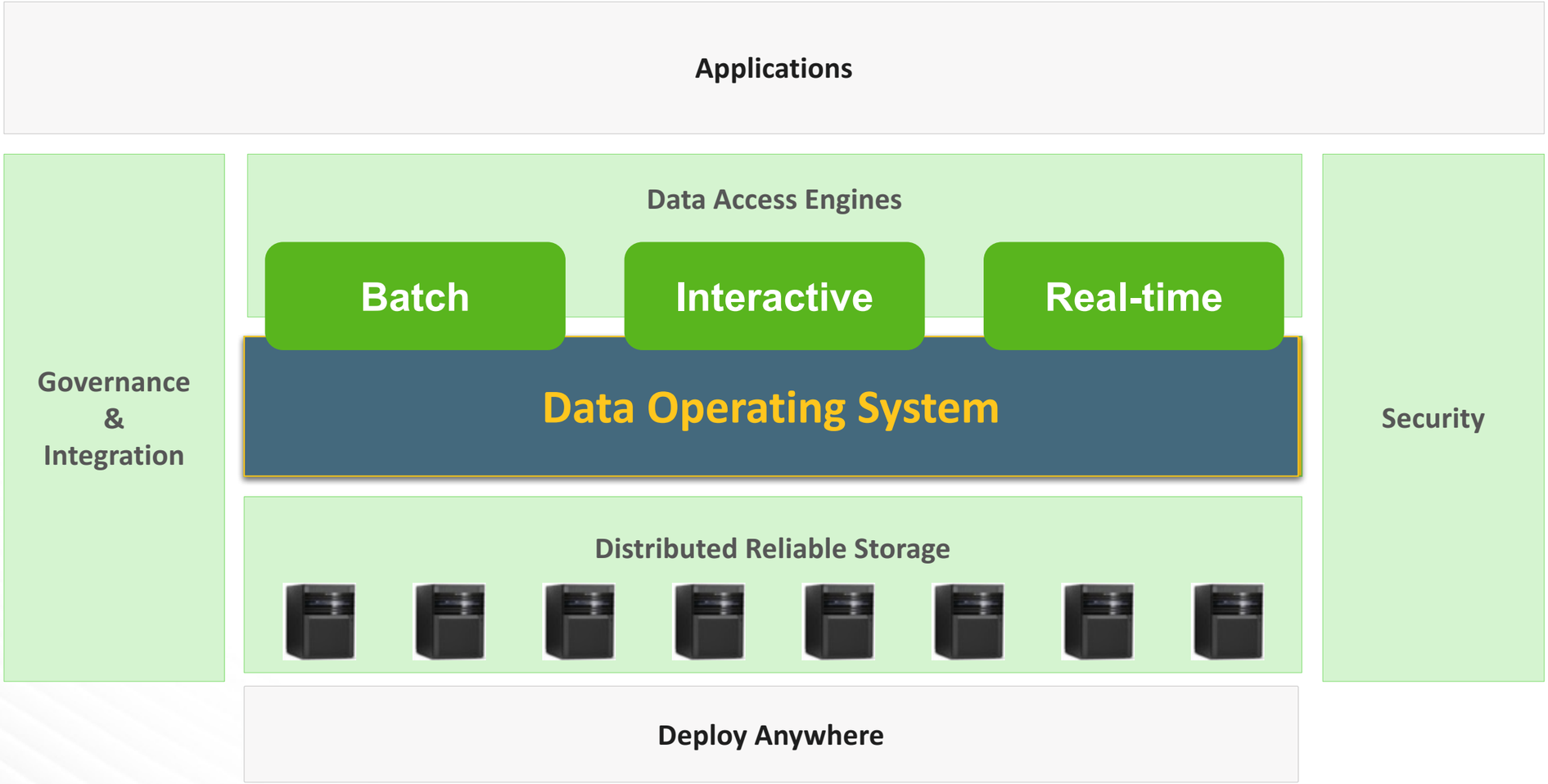
- CSV
- JSON
- TSV
- TEXT
- PDF
- XML
- HTML
- AVRO
- PARQUET
- ORC
- Sequence File
- HFile
- JPEG
- PNG
- MP4
- RTF
- Word
- Excel

HORTONWORKS

Apache Big Data Ecosystem

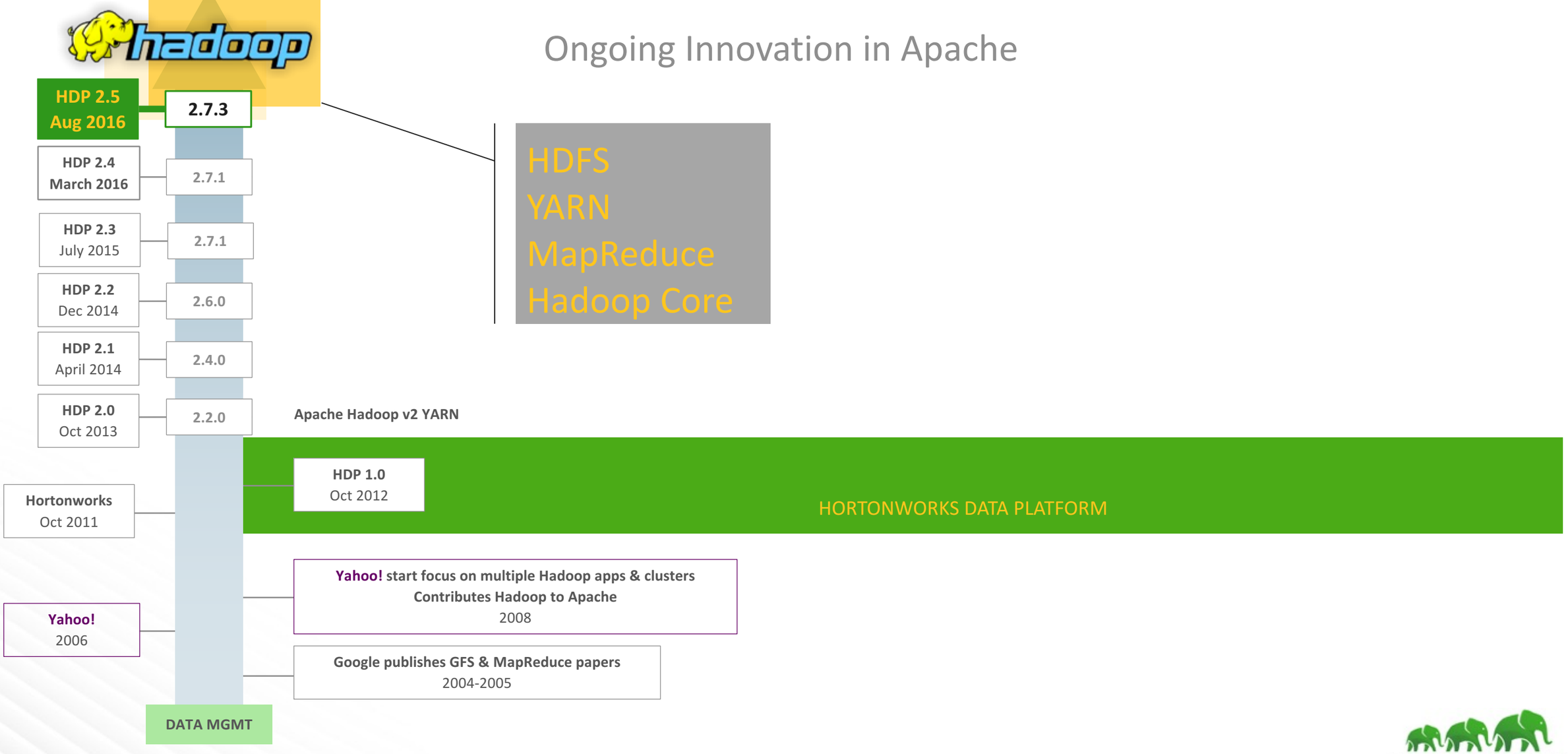


Hadoop Architecture



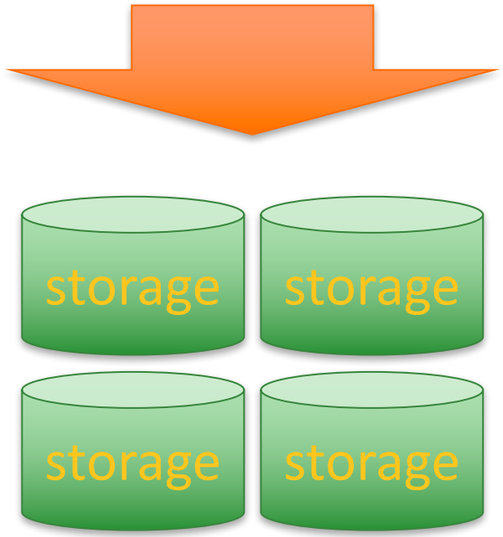
HORTONWORKS

What is Apache Hadoop?

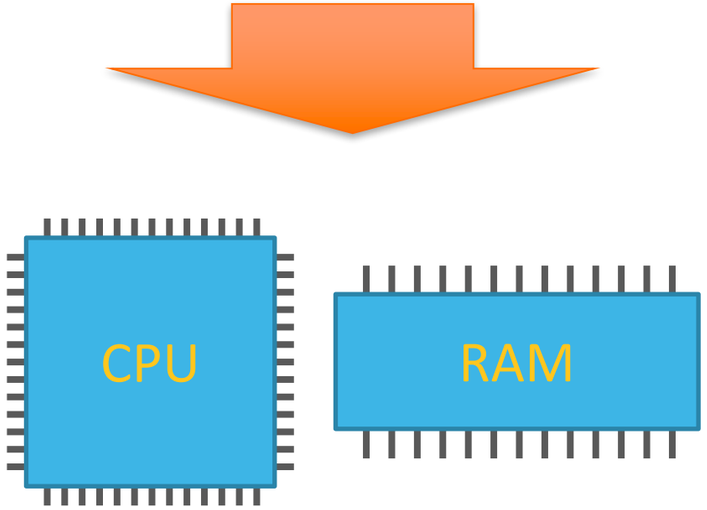


Apache Hadoop = Storage + Compute

Hadoop Distributed File System (HDFS)

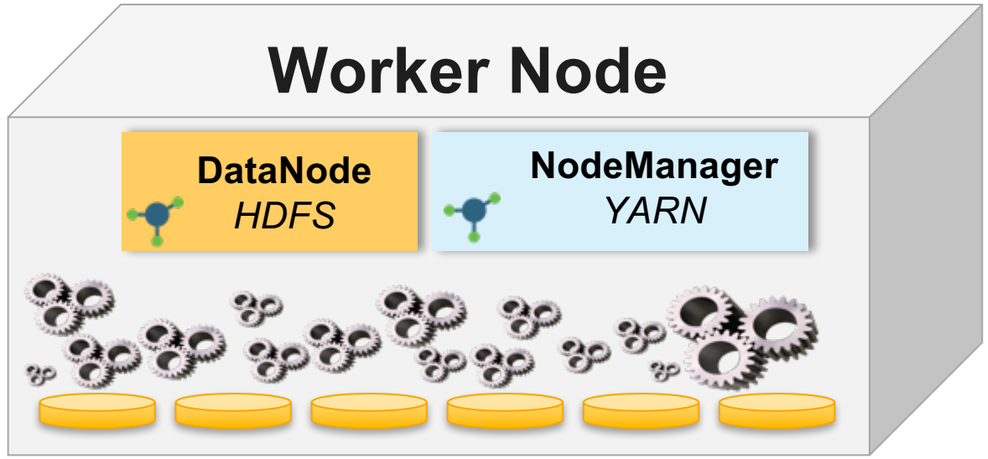


Yet Another Resource Negotiator (YARN)



HORTONWORKS

Core hadoop



Disk, CPU, Memory

+

NameNode
HDFS

`/directory/structure/in/memory.txt`

ResourceManager
YARN

Resource management + scheduling

Hadoop daemon

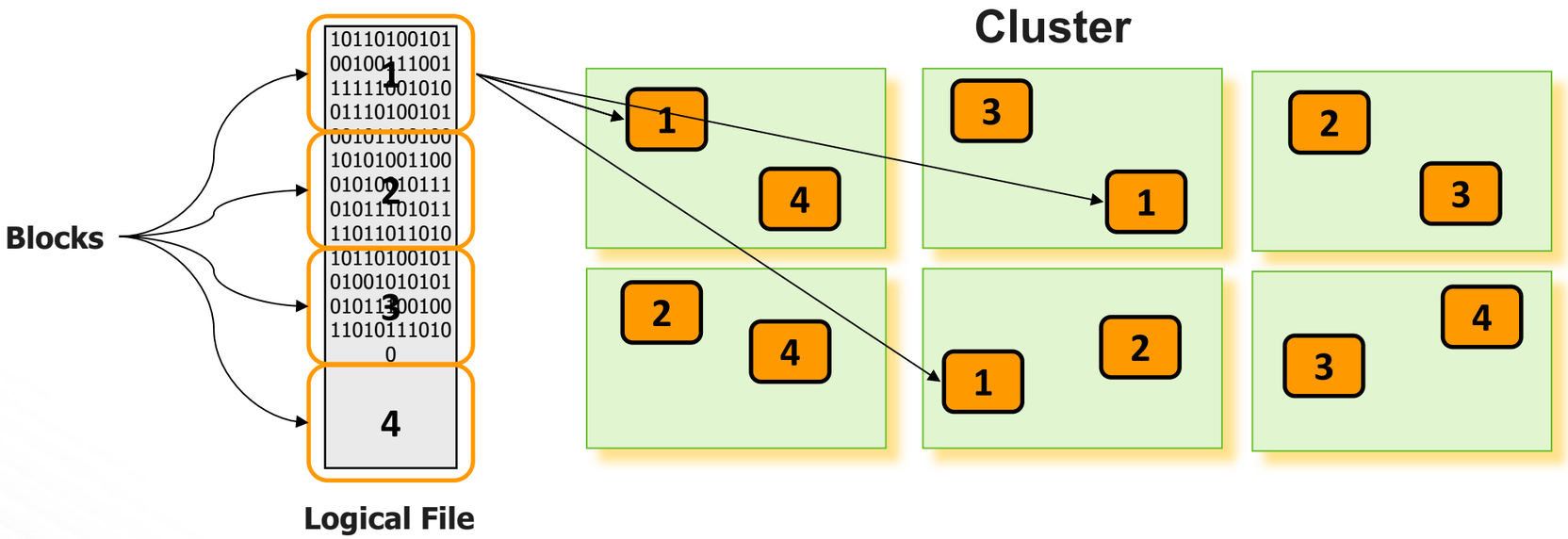
User application

HORTONWORKS

Hadoop Distributed File System (HDFS)

Fault Tolerant Distributed Storage

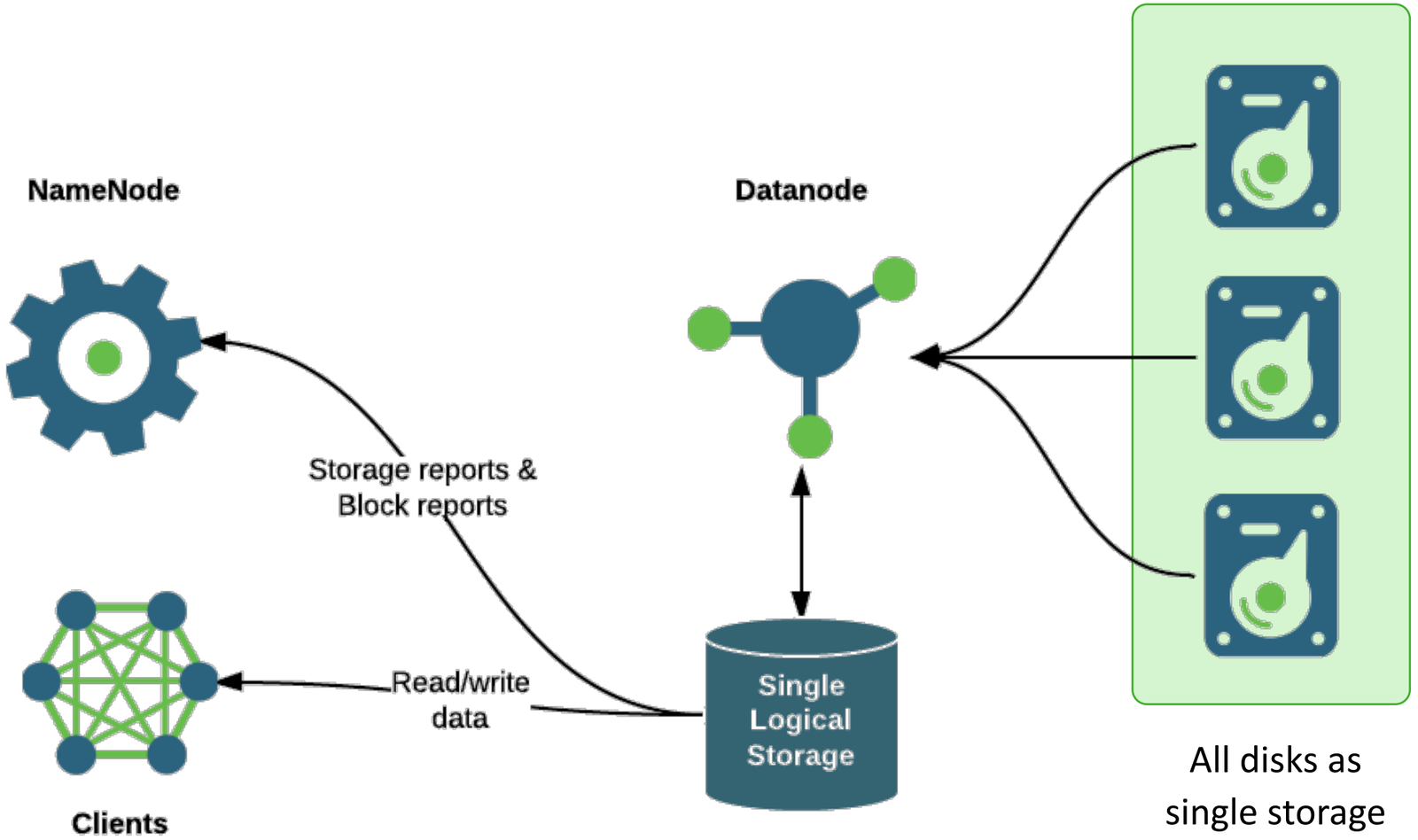
- Divide files into big blocks and distribute 3 copies **randomly** across the cluster
- Processing Data Locality
  - Not Just storage but computation



HDFS Storage Architecture - Before

Before

- DataNode is a single storage
- Storage is uniform - Only storage type Disk
- Storage types hidden from the file system



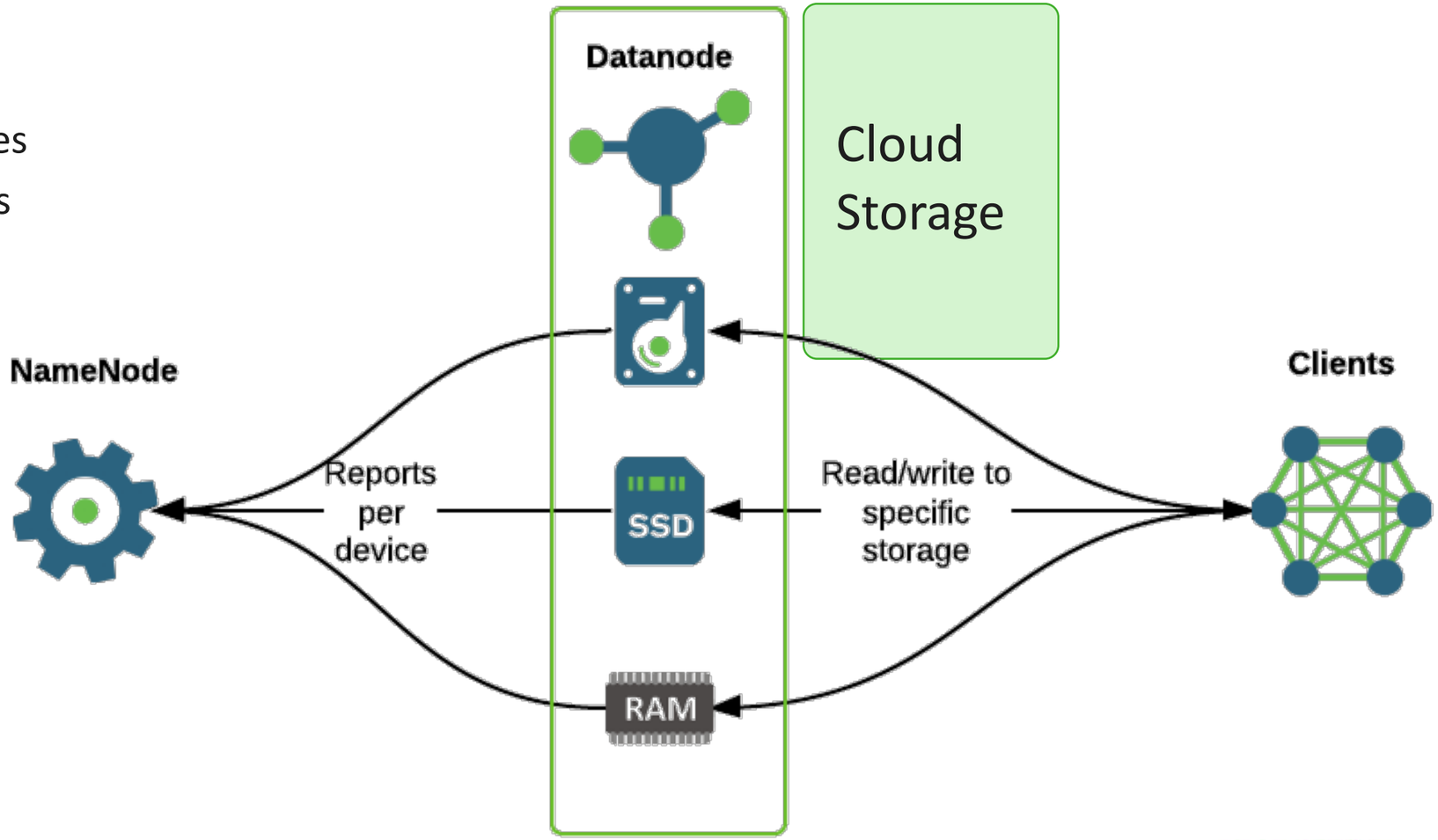
HDFS Storage Architecture - Now

New Architecture

- DataNode is a collection of storages
- Support different types of storages
  - Disk, SSDs, Memory

Block Storage Policies

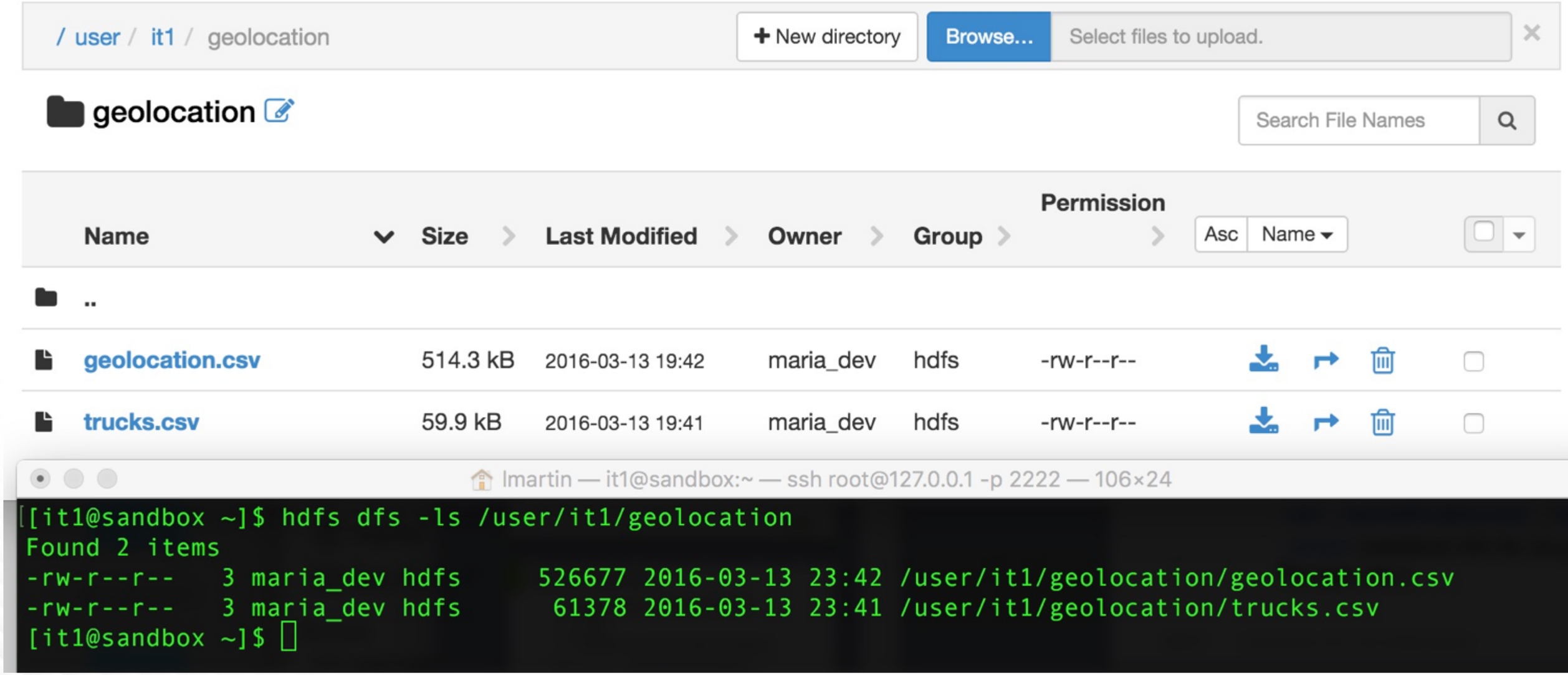
- Describes how to store data blocks in HDFS



Collection of tier storage

HORTONWORKS

It Looks Like a File System



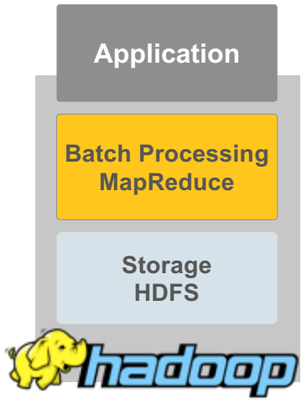
WebHDFS

http://princeton.server.com:50070/webhdfs/v1/demo/clickstream/weblogs/clickstream-feed-generated.tsv?OP=open

https://community.hortonworks.com/articles/60480/using-images-stored-in-hdfs-for-web-pages.html

HORTONWORKS

Hadoop emerged as foundation of new data architecture



Apache Hadoop is an open source data platform for managing large volumes of high velocity and variety of data

- Built by Yahoo! to be the heartbeat of its ad & search business
- Donated to Apache Software Foundation in 2005 with rapid adoption by large web properties & early adopter enterprises
- Incredibly disruptive to current platform economics

Traditional Hadoop Advantages

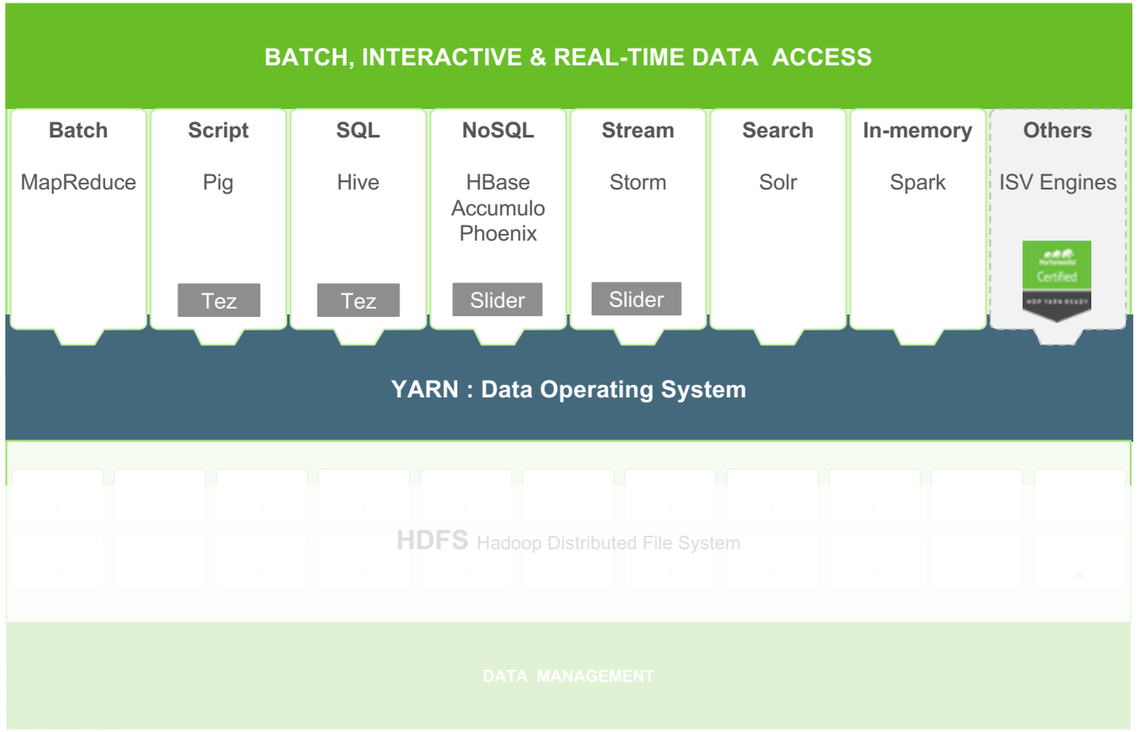
- ✓ Manages new data paradigm
- ✓ Handles data at scale
- ✓ Cost effective
- ✓ Open source

Traditional Hadoop Had Limitations

- ✘ Batch-only architecture
- ✘ Single purpose clusters, specific data sets
- ✘ Difficult to integrate with existing investments
- ✘ Not enterprise-grade

HORTONWORKS

YARN extends Hadoop into data center leaders



YARN
The Architectural Center of Hadoop

- Common data platform, many applications
- Support multi-tenant access & processing
- Batch, interactive & real-time use cases
- Supports 3rd-party ISV tools

  (ex. SAS, Syncsort, Actian, etc.)

YARN Ready Applications

Facilitates ongoing innovation and enterprise adoption via ecosystem of new and existing “YARN Ready” solutions

Overview of SQL on Hadoop Solutions

	Apache Hive is a data warehouse infrastructure built on top of Hadoop for providing data summarization, query, and analysis.
Spark SQL	Spark's module for working with structured data. Run SQL queries alongside complex analytic algorithms.
APACHE PHOENIX	High performance relational database layer over HBase for low latency applications.
Traditional MPP on Hadoop 	Many traditionally architected MPP solutions have been ported to Hadoop and some new ones have been developed from scratch.

HORTONWORKS

SQL on Hadoop: Vitals

Project	First GA Release	Lines of Code (June 2015) (*)	Most Typical Use
Apache Hive	April, 2009 (7 Years)	1 Million	EDW / ETL Offload
SparkSQL	March, 2015 (4 Months)	56.6k	Exploratory Analytics
Apache Phoenix	March, 2014 (2 Year)	200k	Low-Latency Dashboards

HORTONWORKS

Apache Hive: Fast Facts

Most Queries Per Hour

100,000 Queries Per Hour
(Yahoo Japan)

Analytics Performance

100 Million rows/s Per Node
(with Hive LLAP)

Largest Hive Warehouse

300+ PB Raw Storage
(Facebook)

Largest Cluster

4,500+ Nodes
(Yahoo)

HORTONWORKS

Phoenix and HBase: Fast Facts

Largest Database

5 Petabytes
(Flurry)

Best Known App

Facebook Messages
(Facebook)

Fastest Ingestion

10 Million Events/s
(Yahoo)

Biggest SQL App

Real-Time SQL on 140m+ Records
(PubMatic)

HORTONWORKS

Apache Hive: Journey to SQL:2011 Analytics

Data Types
Numeric
FLOAT/DOUBLE
DECIMAL
INT/TINYINT/SMALLINT/BIGINT
BOOLEAN
String
CHAR / VARCHAR
STRING
BINARY
Date, Time
DATE
TIMESTAMP
Interval Types
Complex Types
ARRAY
MAP
STRUCT
UNION

SQL Features
Core SQL Features
Date, Time and Arithmetical Functions
INNER, OUTER, CROSS and SEMI Joins
Derived Table Subqueries
Correlated and Uncorrelated Subqueries
UNION ALL
UDFs, UDAFs, UDTFs
Common Table Expressions
UNION DISTINCT
INTERSECT, EXCEPT
Non-Equality Joins
Advanced Analytics
OLAP and Windowing Functions
CUBE and Grouping Sets
Nested Data Analytics
Nested Data Traversal
Lateral Views
ACID Transactions
INSERT / UPDATE / DELETE
MERGE

File Formats
Columnar
ORCFile
Parquet
Text
CSV
Logfile
Nested / Complex
Avro
JSON
XML
Custom Formats
Other Features
XPath Analytics
Procedural Extensions (PL/Hive)

Legend

- Hive 1.0
- Hive 1.2
- Future

HORTONWORKS

Apache Hive facilitates **querying and managing large datasets.**

Hive provides SQL on Hadoop.

Data analysts use Hive to **explore, structure and analyze** that **data** using the familiar comfortable SQL syntax they are used to.

Hive also comes with **HCatalog**; a global metadata management layer that exposes **Hive table metadata** to all other Hadoop applications.

HORTONWORKS

Apache Zeppelin is a Web-based notebook that enables interactive data analytics.

Data Scientists and End-Users alike can make beautiful data-driven, interactive and collaborative documents with SparkSQL, Scala, Python, JDBC connections, Files, and more.

Notebooks contain code samples, source data, descriptive markup, result sets, and rich visualizations.

Zeppelin Notebook Interpreter Configuration

Search in your notebooks

Connected

Australian Dataset (SparkSQL example)

default

Register RDD as table

FINISHED

```
case class Health (year:  String, state: String, category:String, funding_src1: String, funding_scr2: String, spending: Integer)
val health = dataset.map(k=>k.split(",")).map(
    k => Health(k(0),k(1),k(2),k(3), k(4), k(5).toInt)
    )
// toDF() works only in spark 1.3.0.
// For spark 1.1.x and spark 1.2.x,
// use below instead:
// health.registerTempTable("health_table")
health.toDF().registerTempTable("health_table")
```

```
defined class Health
health: org.apache.spark.rdd.RDD[Health] = MapPartitionsRDD[7] at map at <console>:33
```

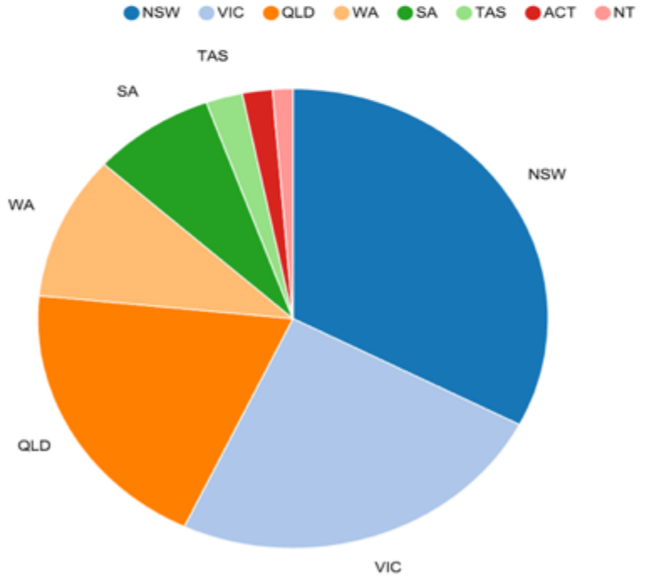
Took 3 seconds

Spending (in billions) by state

FINISHED

```
%sql
select state, sum(spending)/1000 SpendinginBillions
from health_table
group by state
order by SpendinginBillions desc
```

settings

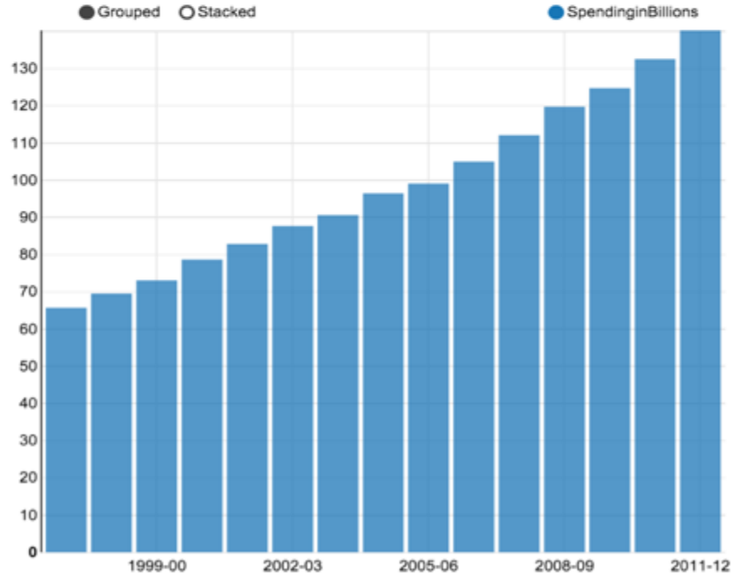


Spending (In Billions) By Year

FINISHED

```
%sql
select year,sum(spending)/1000 SpendinginBillions
from health_table
group by year
order by SpendinginBillions
```

settings



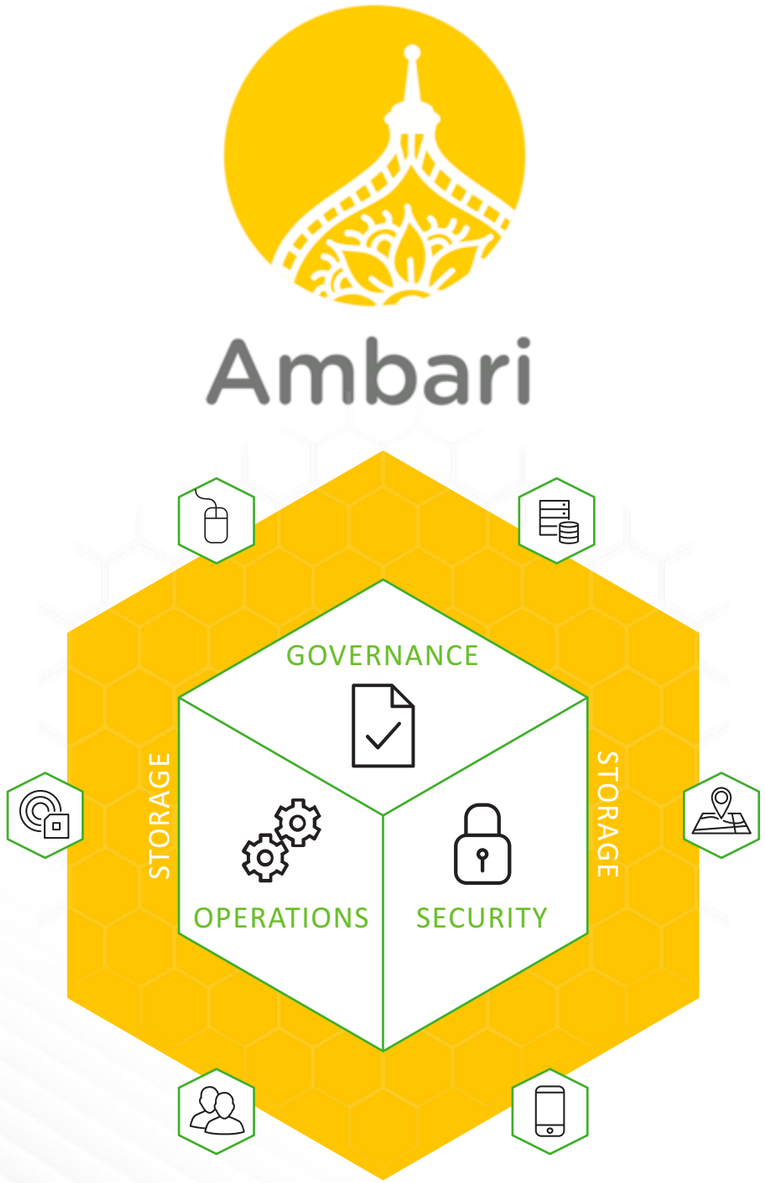
Spending (in billions) by area

FINISHED

```
%sql
select category, sum(spending)/1000 SpendinginBillions
from health_table
group by category
order by SpendinginBillions desc
```

category	SpendinginBillions
Public hospitals	445.845
Medical services	272.507
Private hospitals	121.022
Benefit-paid pharmaceuticals	104.221
Dental services	90.786
Community health	75.765
Capital expenditure	72.698
All other medications	70.508
Other health practitioners	51.382
Administration	41.029
Research	40.074
Aids and appliances	37.155
Patient transport services	28.174
Public health	27.072
Medical expense tax rebate	0.0

HORTONWORKS

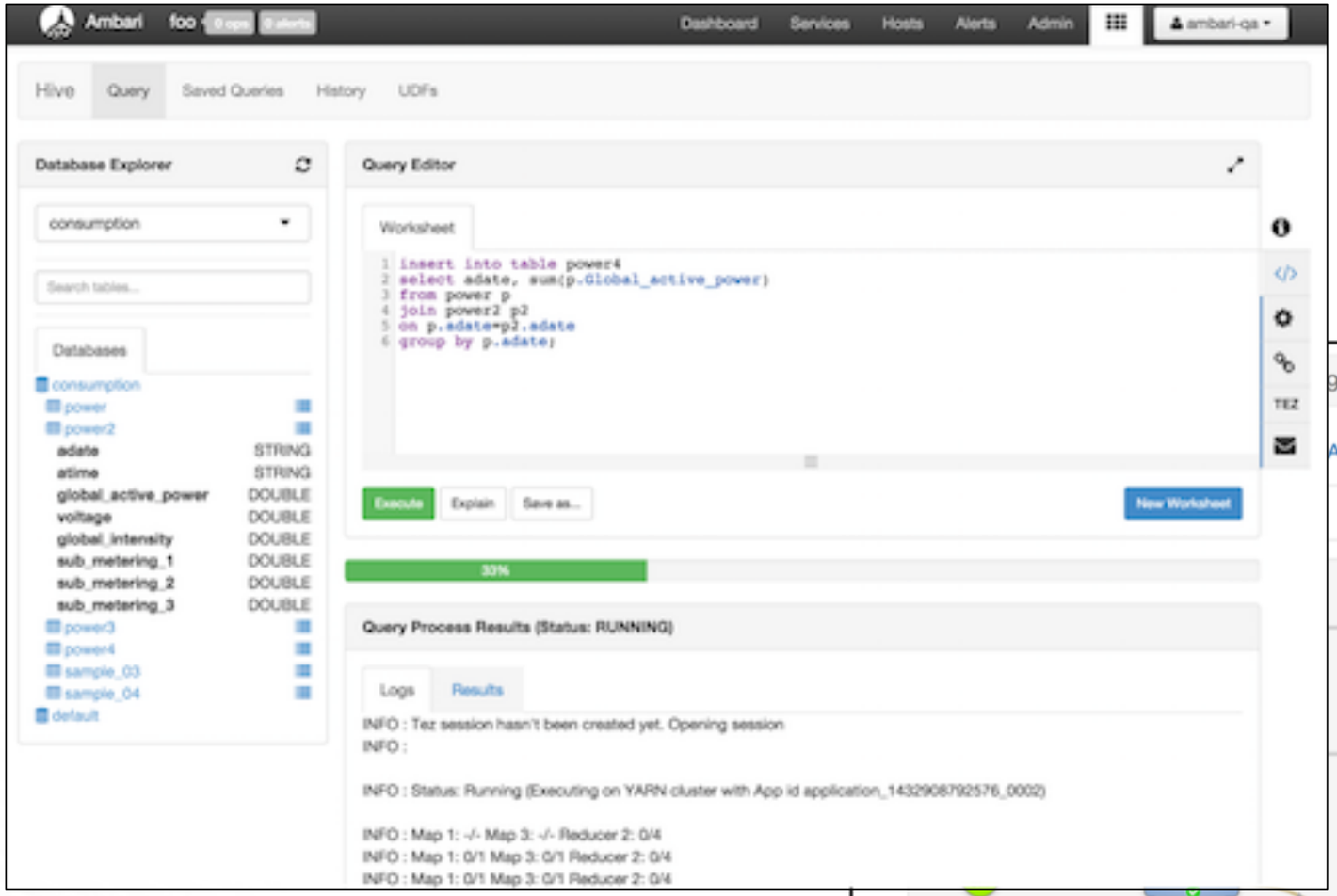


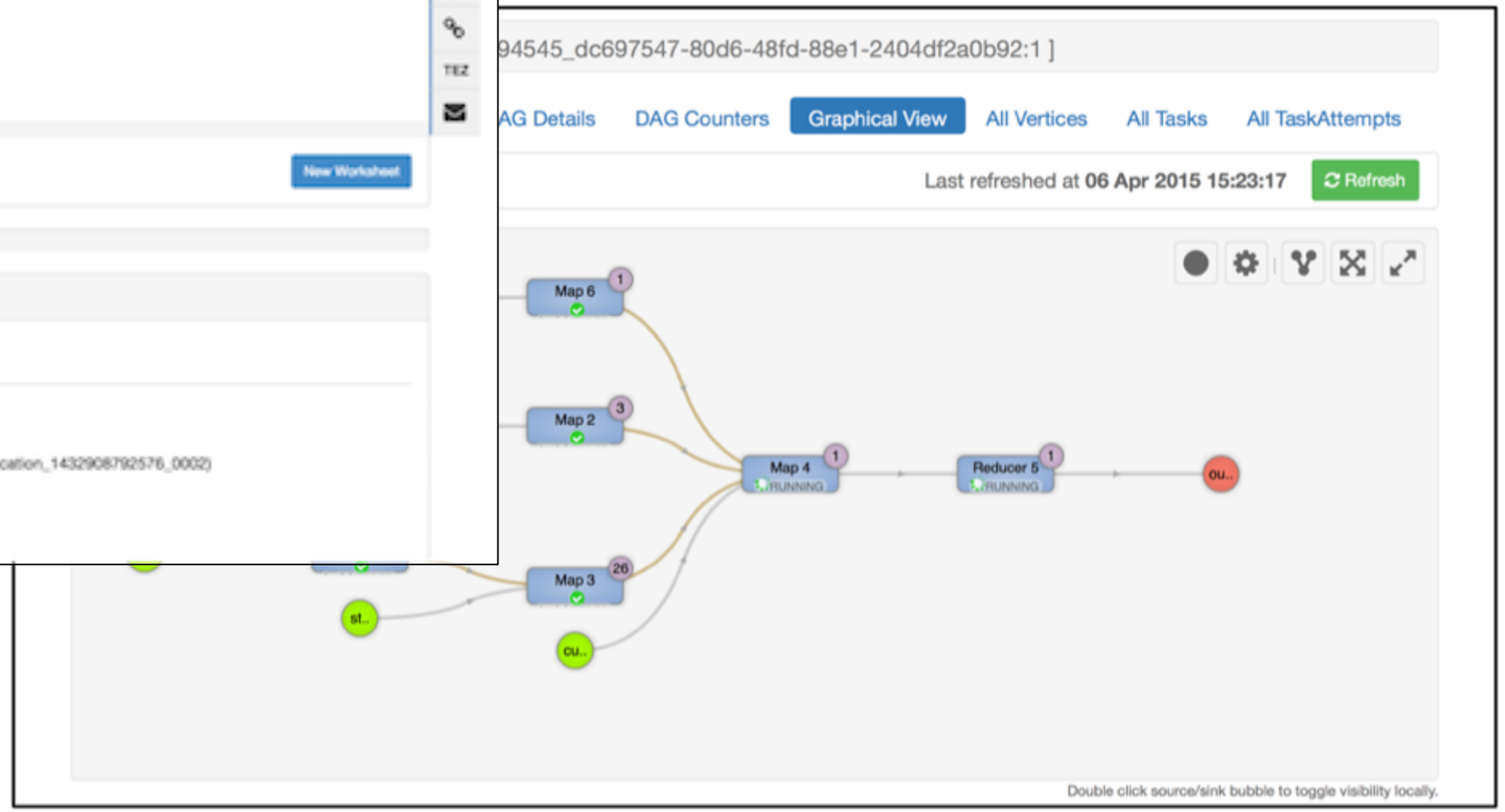
Ambari Views are a built-in set of Views that are pre-deployed for you to use with your cluster.

These GUI components increase ease-of-use to end users. Current Ambari Views include Hive, Pig, Tez, Capacity Scheduler, File, HDFS.

The **Ambari Views Framework** allow developers to create new user interface components that plug into the Ambari web interface.

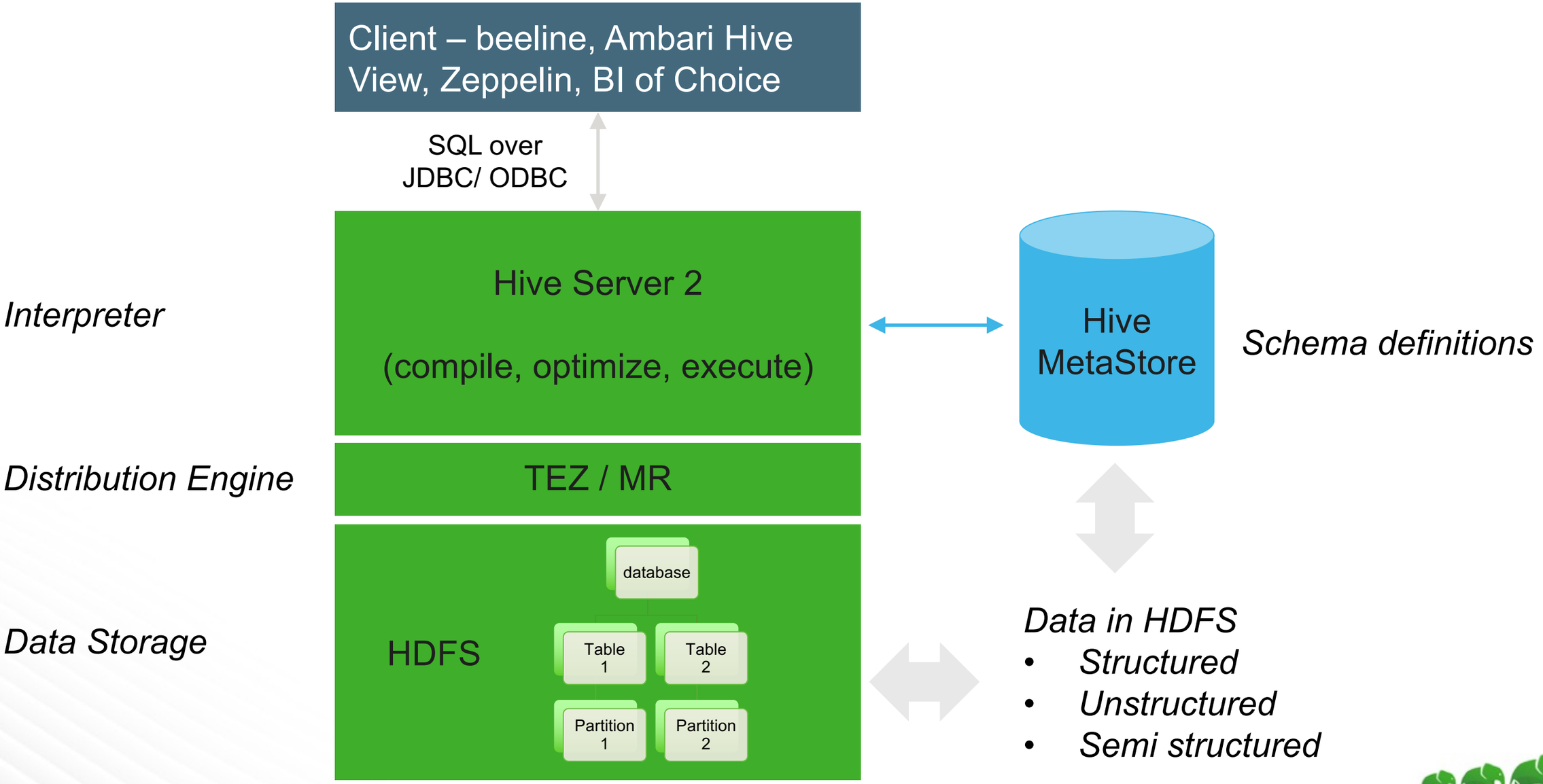
AMBARI VIEWS
TOOLS





HORTONWORKS

Apache Hive Architecture



Submitting Hive Queries

- Hive CLI
  - Traditional Hive client that connects to a HiveServer instance
  - ```
    $ hive
    hive>
    ```
- Beeline
  - A new command line client that connects to a HiveServer2 instance
  - ```
    $ beeline
    beeline> !connect
    jdbc:hive2://hostname:10000 username password
    org.apache.hive.jdbc.HiveDriver
    ```


HORTONWORKS

Best Practices for Data Loading - Create External Table, Create ORC Table

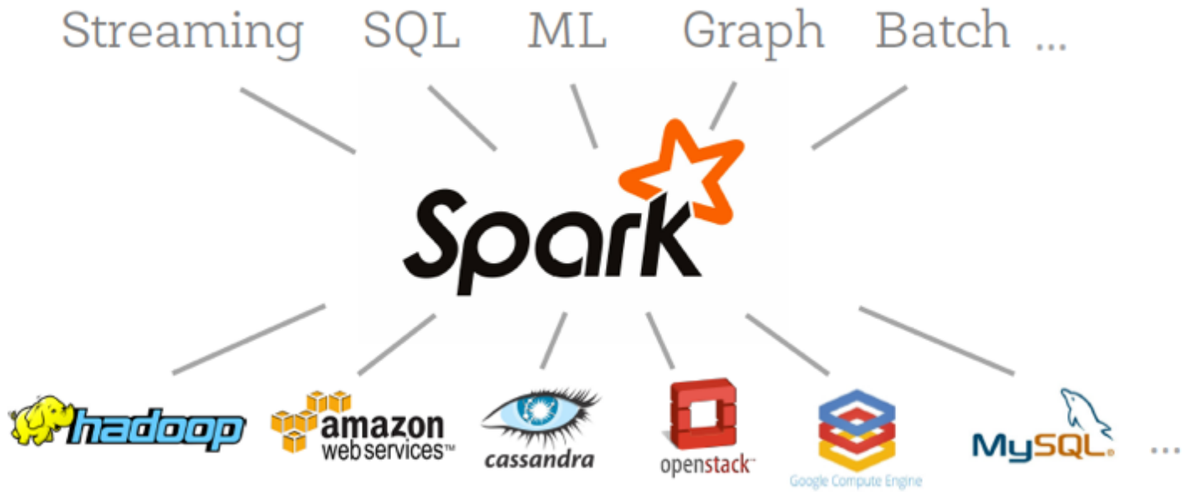
```
hdfs dfs -copyFromLocal cars.csv /visdata

CREATE EXTERNAL TABLE IF NOT EXISTS Cars(Name STRING, Miles_per_Gallon INT,  Cylinders
INT, Displacement INT, Horsepower INT, Weight_in_lbs INT, Acceleration DECIMAL, Year DATE,
Origin CHAR(1)) COMMENT 'Data'    ROW FORMAT DELIMITED    FIELDS TERMINATED BY ','
STORED AS TEXTFILE  location ’/visdata';

CREATE TABLE IF NOT EXISTS mycars( Name STRING, Miles_per_Gallon INT, Cylinders INT,
Displacement INT, Horsepower INT, Weight_in_lbs INT, Acceleration DECIMAL, Year DATE, Origin
CHAR(1)) COMMENT 'Data' ROW FORMAT DELIMITED FIELDS TERMINATED BY ',' STORED AS
ORC;

INSERT OVERWRITE TABLE mycars SELECT * FROM cars;
```

Data Formats

HORTONWORKS

What Is Apache Spark?

- **Apache open source project** originally developed at AMPLab (University of California Berkeley)
- **Unified data processing engine** that operates across varied data workloads and platforms



HORTONWORKS

Why Apache Spark?

- **Elegant Developer APIs**
  - Single environment for data munging, data wrangling, and Machine Learning (ML)
- **In-memory computation model – Fast!**
  - Effective for iterative computations and ML
- **Machine Learning**
  - Implementation of distributed ML algorithms
  - Pipeline API (Spark ML)

HORTONWORKS

Spark SQL
Structured Data

Spark Streaming
Near Real-time

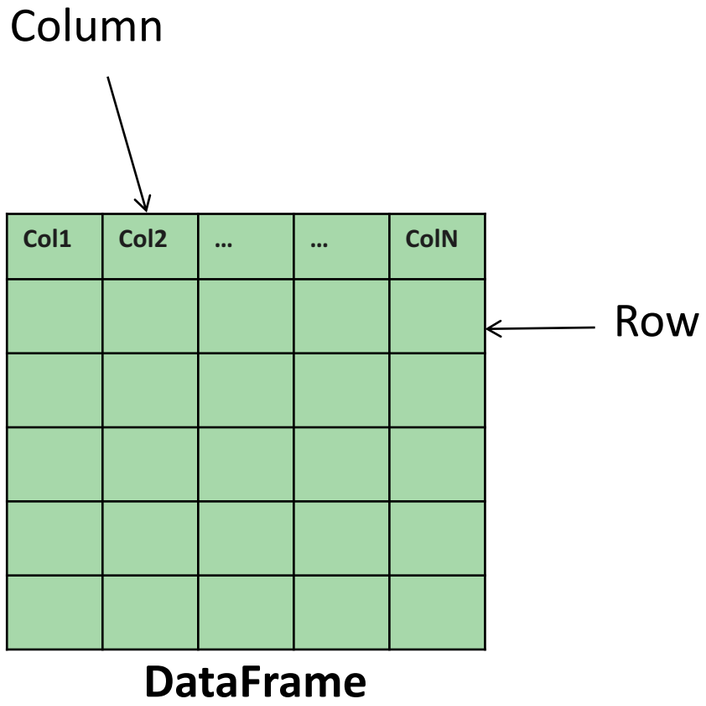
Spark MLlib
Machine Learning

GraphX
Graph Analysis

HORTONWORKS

DataFrames

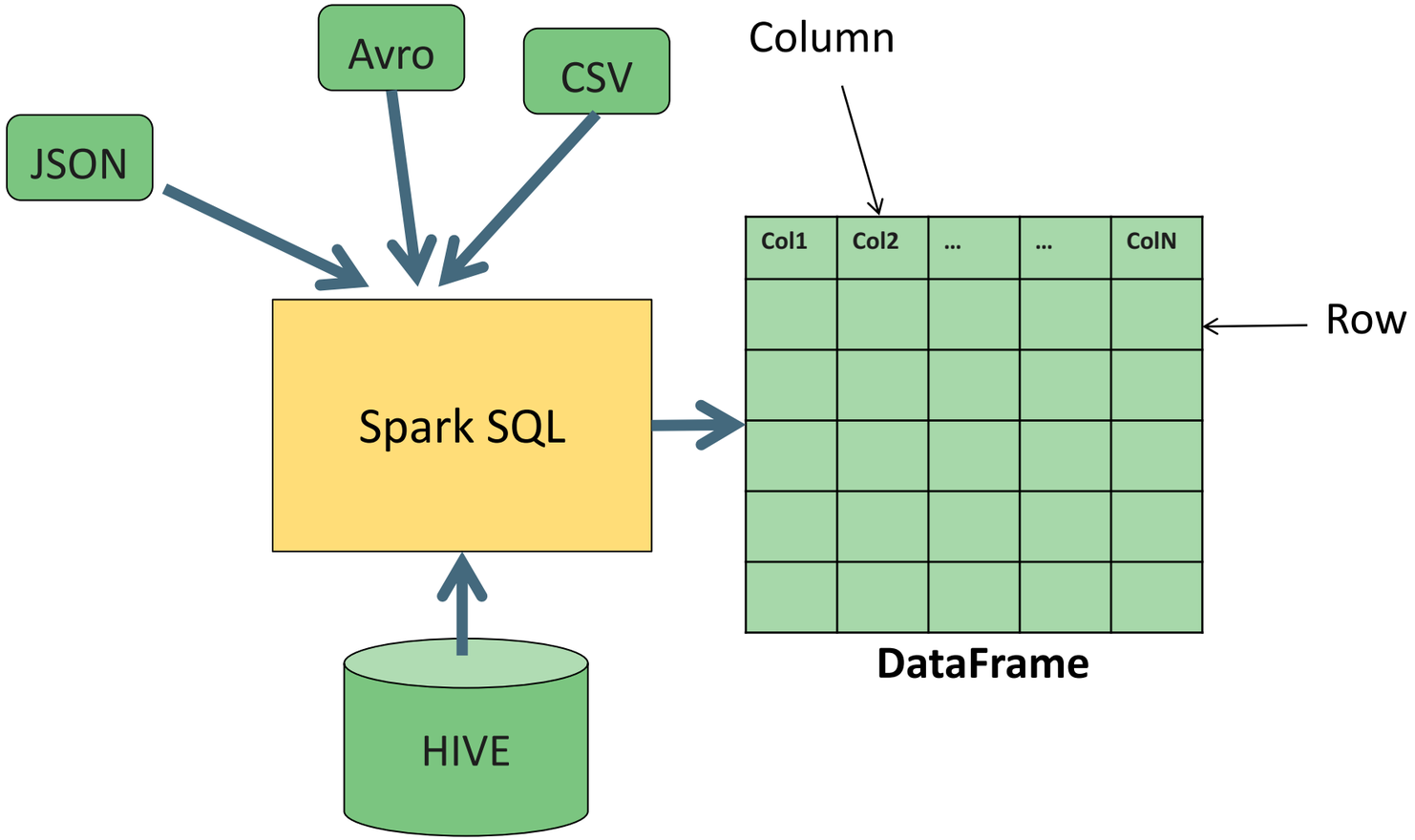
- **Distributed collection** of **data organized** into ***named columns***
- Conceptually equivalent to a table in **relational DB** or a **data frame in R/Python**
- API available in Scala, Java, Python, and R



Data is described as a DataFrame with **rows, columns,** and a **schema**

HORTONWORKS

Sources



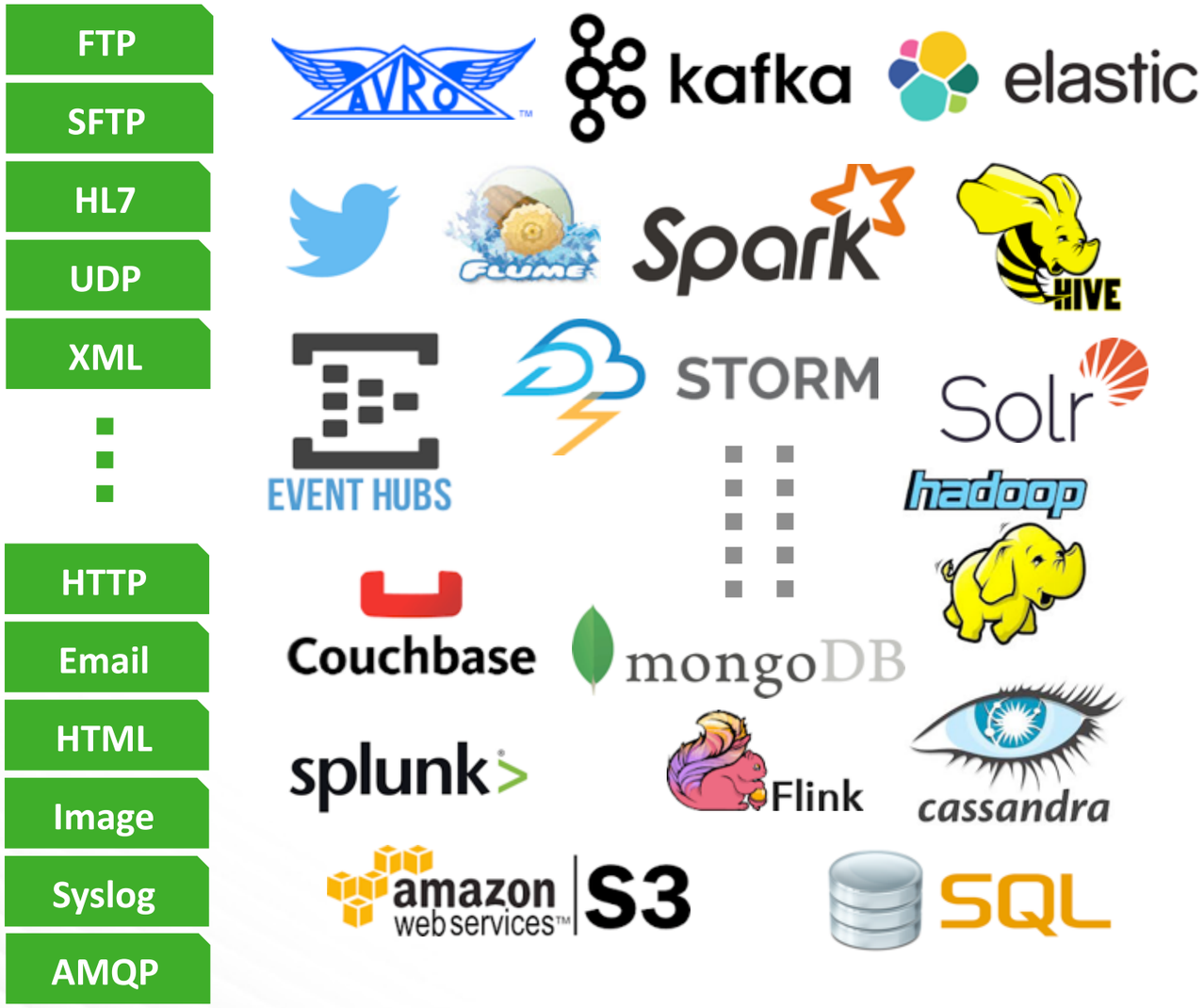
Create a DataFrame

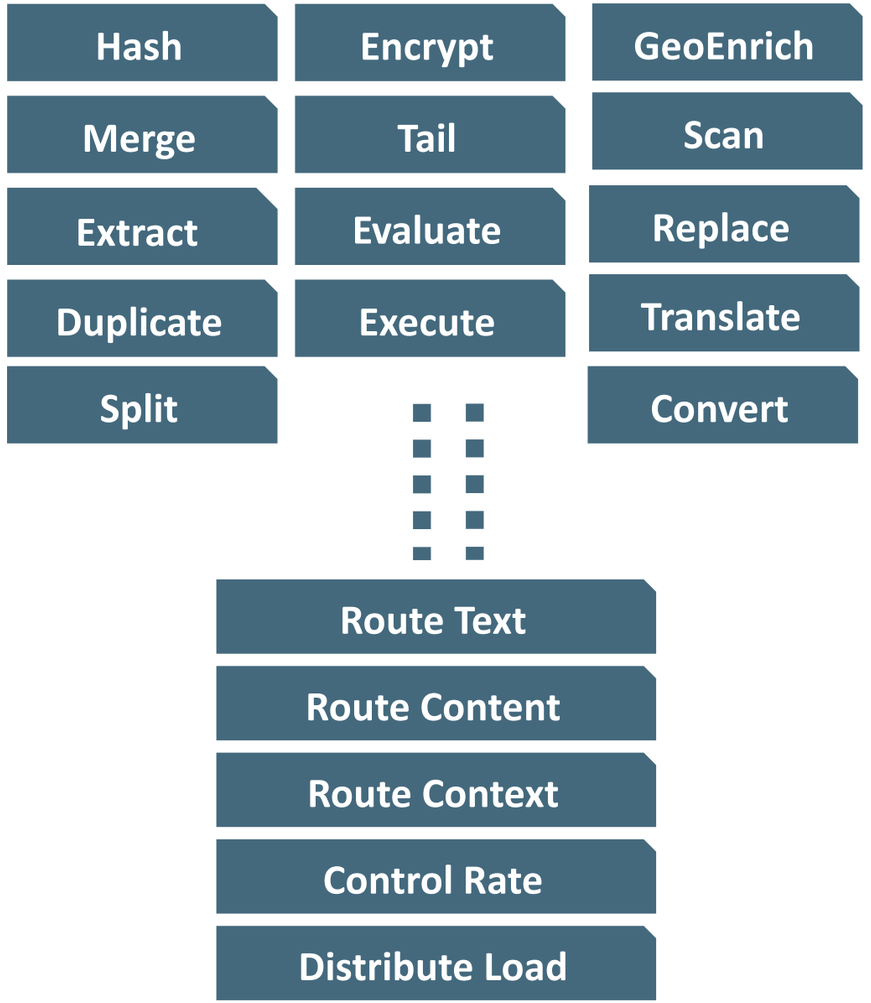
Example

```
val path = "examples/flights.json"
val flights = spark.read.json(path)
```


HORTONWORKS

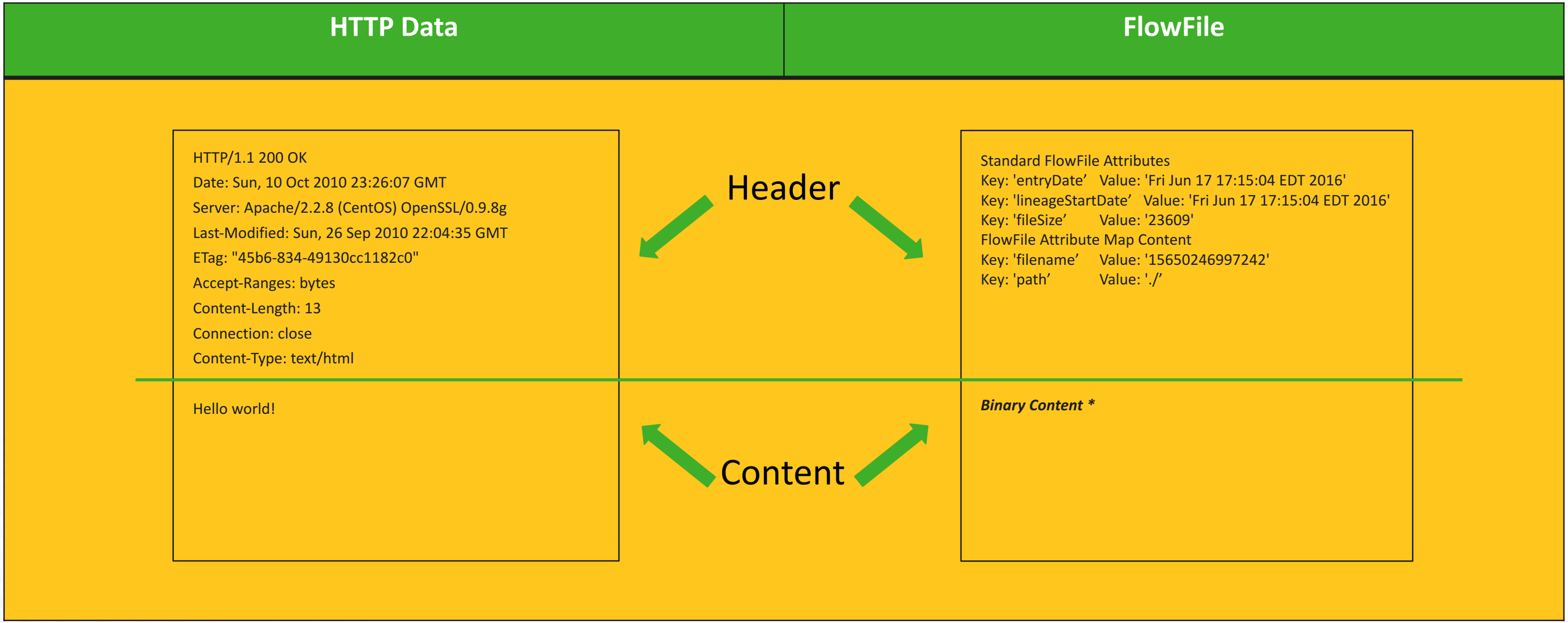
Rapid Ecosystem Adoption: 210+ Processors

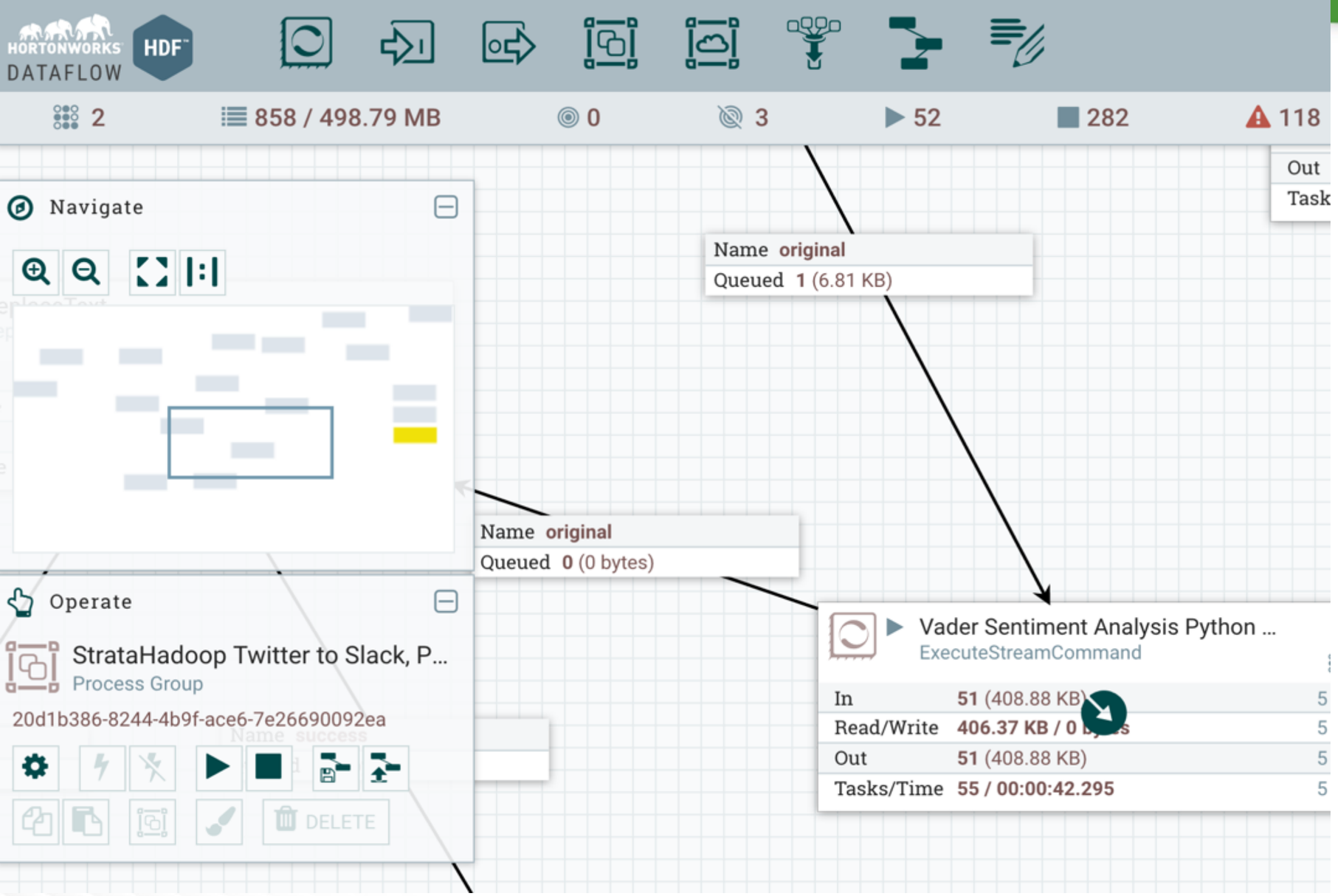




HORTONWORKS

FlowFiles are like HTTP data





http://hortonworks.com/blog/hdf-2-0-flow-processing-real-time-tweets-strata-hadoop-slack-tensorflow-phoenix-zeppelin/

HORTONWORKS

Quick Terms and Reference

FlowFile: Each piece of "User Data" (i.e., data that the user brings into NiFi for processing and distribution) is referred to as a FlowFile. A FlowFile is made up of two parts: Attributes and Content. The Content is the User Data itself. Attributes are key-value pairs that are associated with the User Data.

Processor: The Processor is the NiFi component that is responsible for creating, sending, receiving, transforming, routing, splitting, merging, and processing FlowFiles. It is the most important building block available to NiFi users to build their dataflows.

https://nifi.apache.org/docs/nifi-docs/html/getting-started.html
https://nifi.apache.org/docs/nifi-docs/html/overview.html
http://www.slideshare.net/aldrinpiri/apache-nifi-crash-course-san-jose-hadoop-summit-66967077
https://hortonworks.com/hadoop-tutorial/learning-ropes-apache-nifi/

Input

InvokeHttp

GetTwitter

HORTONWORKS

Processing

RouteOnAttribute

ExecuteStreamCommand

UpdateAttribute

HORTONWORKS

Output

PutHDFS: Have access to your Hadoop HDFS from the NIFI box and have this configuration: /etc/hadoop/conf/core-site.xml . Also create a directory to use like hdfs dfs –mkdir /nifi-place

PutSQL: Create a connection pool, know your JDBC information.

1. **Phoenix: URL**= jdbc:phoenix:clusterzookeeper:2181:/hbase-unsecure org.apache.phoenix.jdbc.PhoenixDriver file:///opt/demo/phoenix-client.jar
   User= root pool=2 You will need the JDBC JAR on the local file system.
2. **MySQL**: jdbc:mysql://tspanndev11.field.hortonworks.com:3306/datacom.mysql.jdbc.Driver/usr/share/java/mysql-connector-java.jar

PutSlack

Need to get your webhook URL from your slack site. You can go to slack.com and get your own free room to test with.
https://api.slack.com/incoming-webhooks

Local TensorFlow via Python or C++ Binary

python **classify_image.py** --image_file /opt/demo/dronedataold/Bebop2_20160920083655-0400.jpg
solar dish, solar collector, solar furnace (score = 0.98316)
window screen (score = 0.00196)
manhole cover (score = 0.00070)
radiator (score = 0.00041)
doormat, welcome mat (score = 0.00041)

bazel-bin/tensorflow/examples/label_image/label_image --
image=/opt/demo/dronedataold/Bebop2_20160920083655-0400.jpg
tensorflow/examples/label_image/main.cc:204] **solar dish** (577): 0.983162I
tensorflow/examples/label_image/main.cc:204] window screen (912): 0.00196204I
tensorflow/examples/label_image/main.cc:204] manhole cover (763): 0.000704005I
tensorflow/examples/label_image/main.cc:204] radiator (571): 0.000408321I
tensorflow/examples/label_image/main.cc:204] doormat (972): 0.000406186

Local Sentiment Analysis via Python

/opt/demo/sentiment/run.sh

```
        python /opt/demo/sentiment/sentiment.py "$@”
```

```
from nltk.sentiment.vader import SentimentIntensityAnalyzer
import sys
sid = SentimentIntensityAnalyzer()
ss = sid.polarity_scores(sys.argv[1])
print('Compound {0} Negative {1} Neutral {2} Positive {3} '.format(
        ss['compound'],ss['neg'],ss['neu'],ss['pos']))
```

or

```
if ss['compound'] == 0.00:     print('Neutral')
elif ss['compound'] < 0.00:    print ('Negative')
else:                          print('Positive')
```


HORTONWORKS

Installing NLTK for Python 2.7

https://pip.pypa.io/en/latest/installing/
http://www.nltk.org/install.html

wget https://bootstrap.pypa.io/get-pip.py
python get-pip.py
sudo pip install -U nltk
sudo pip install -U numpy

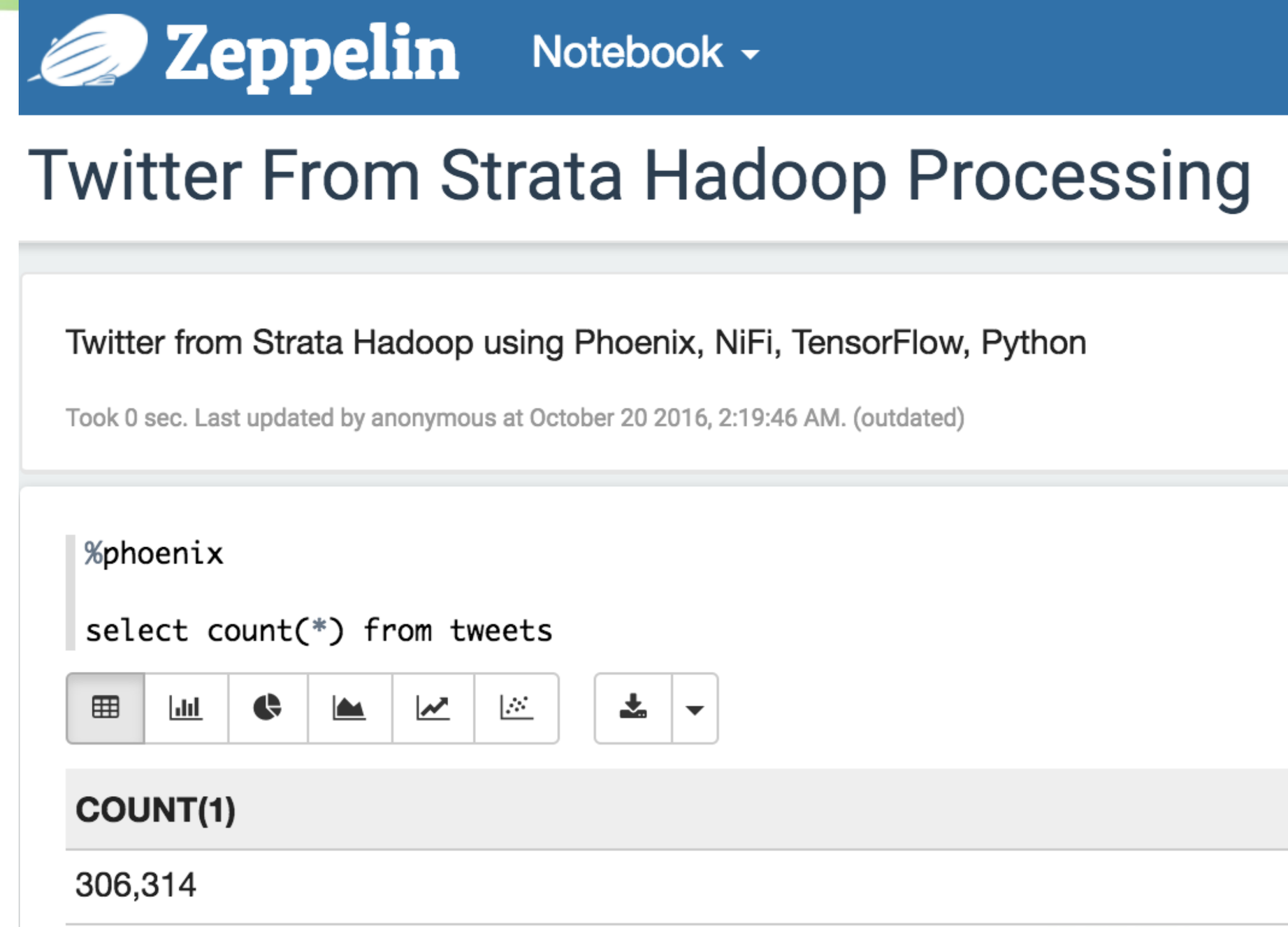
Installing TensorFlow for Python 2.7

Installing TensorFlow is a very difficult exercise, after getting NLTK you can start the process. You will need most of the development tools for Python, C, C++, Bezel, Pip and more. A beefy machine with a lot of RAM, CPUs and GPUs would be useful.

Check out my install article for a guide:
https://dzone.com/articles/deep-learning-resources

HORTONWORKS

Results



Results

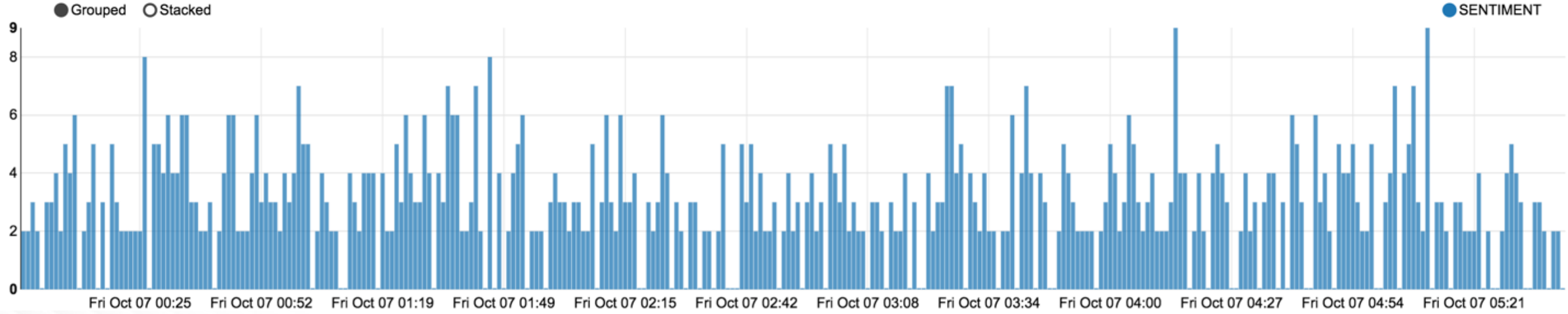
Zeppelin Notebook Search your Notebooks anonymous

Twitter From Strata Hadoop Processing default

```
%phoenix

select minute, sentiment, count(1) as cnt from
(
   select substr(time, 0, 16) as minute, handle,  sentiment, length(msg) as wordcount, msg  from tweets
   order by time desc
) sub1
group by minute, sentiment
order by minute
```

FINISHED

settings



HORTONWORKS

Installation

Download the binary from here:
http://hortonworks.com/downloads/#dataflow
Or here:
https://nifi.apache.org/download.html
Or on Mac:
brew install nifi

https://nifi.apache.org/docs/nifi-docs/html/getting-started.html#starting-nifi

bin/nifi.sh start
bin/nifi.sh install (now it's installed as a service on Linux)

Contact:

Timothy Spann @PaaSDeV

www.meetup.com/futureofdata-princeton

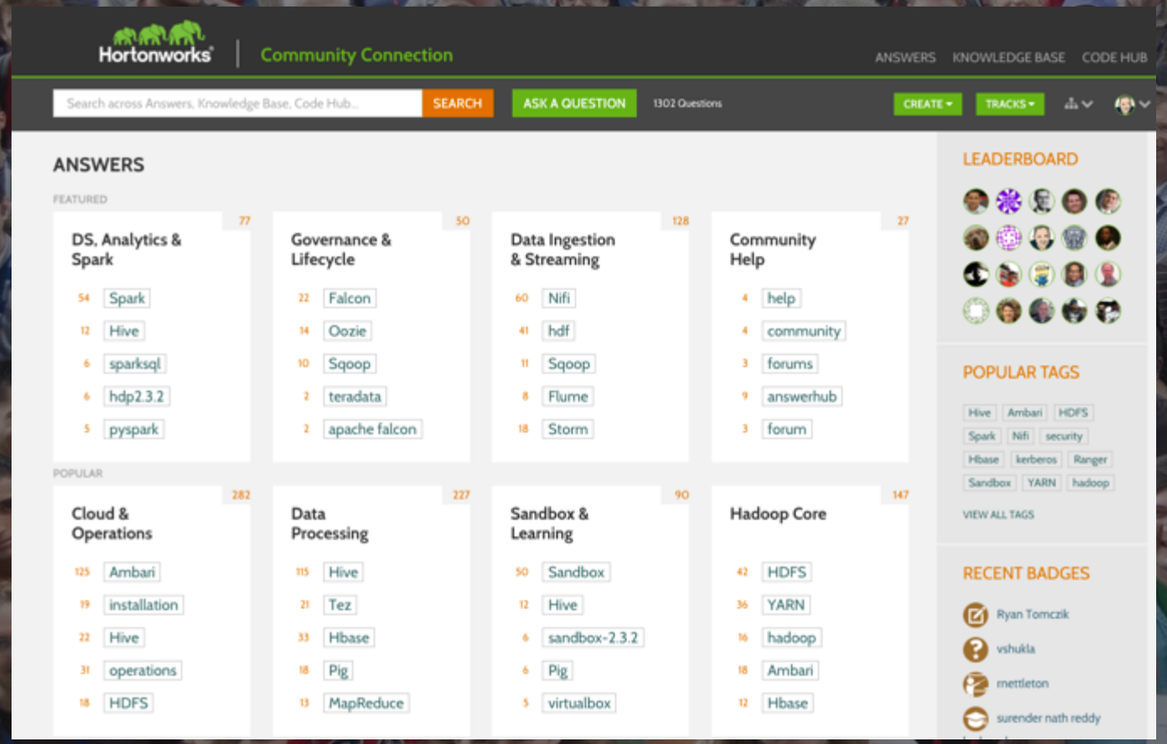
community.hortonworks.com/users/9304/tspann.html

http://www.coreservlets.com/hadoop-tutorial/

https://www.slideshare.net/HadoopSummit/hadoop-crash-course

HORTONWORKS

Hortonworks Community Connection



- Full Q&A Platform (like StackOverflow)
- Knowledge Base Articles
- Code Samples and Repositories

Read access for everyone, join to participate and be recognized

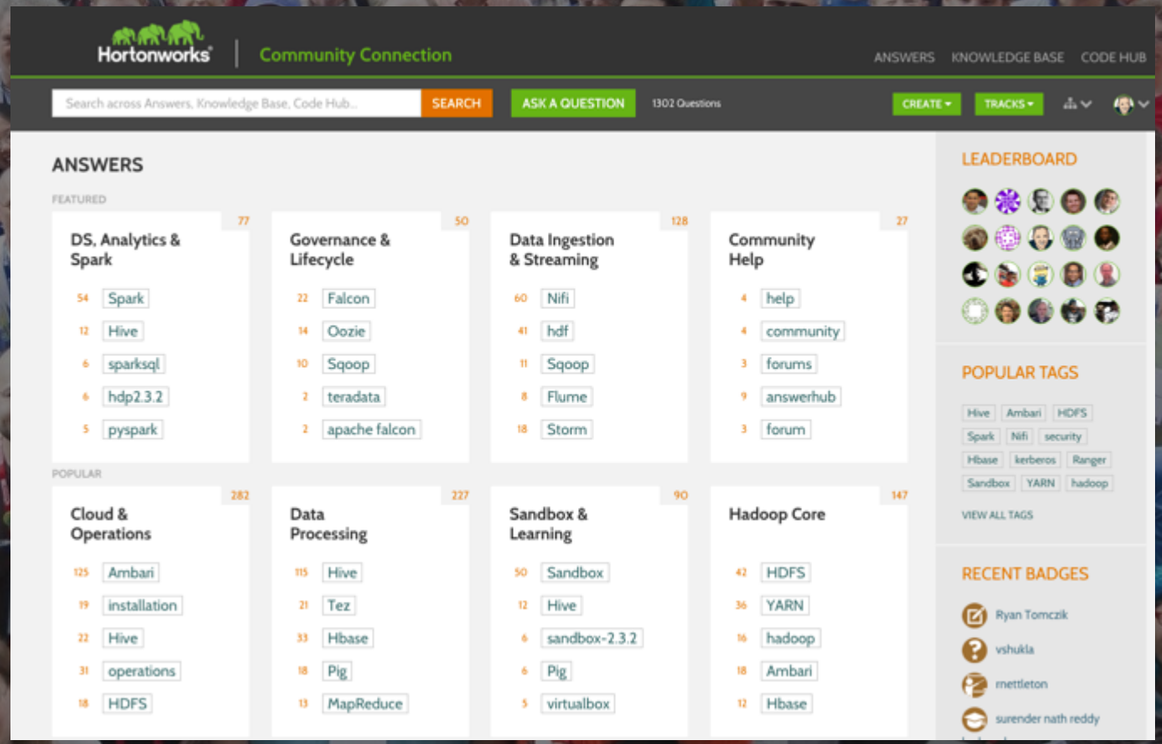
Community Engagement

4,000+
Registered Users

10,000+
Answers

15,000+
Technical Assets

One Website!



Participate now at: community.hortonworks.com