

# Force multiplier for data science: Introduction to H2O



Hank Roark  
@hankroark  
hank@h2o.ai

# **DEMONSTRATION**

# WHO AM I

Lead, Customer Data Science @ H2O.ai

John Deere: Research, Software Product Development, High Tech Ventures  
Lots of time dealing with data off of machines, equipment, satellites, weather, radar,  
hand sampled, and on.  
Geospatial, temporal / time series data almost all from sensors.  
Previously at startups and consulting (Red Sky Interactive, Nuforia, NetExplorer, Perot  
Systems, a few of my own)

Engineering & Management MIT  
Physics Georgia Tech

hank@h2oai.com  
@hankroark  
<https://www.linkedin.com/in/hankroark>

# WHAT IS H2O?

Math Platform

Open source in-memory prediction engine

- Parallelized and distributed algorithms making the most use out of multithreaded systems
- GLM, Random Forest, GBM, Deep Learning, etc.

API

Easy to use and adopt

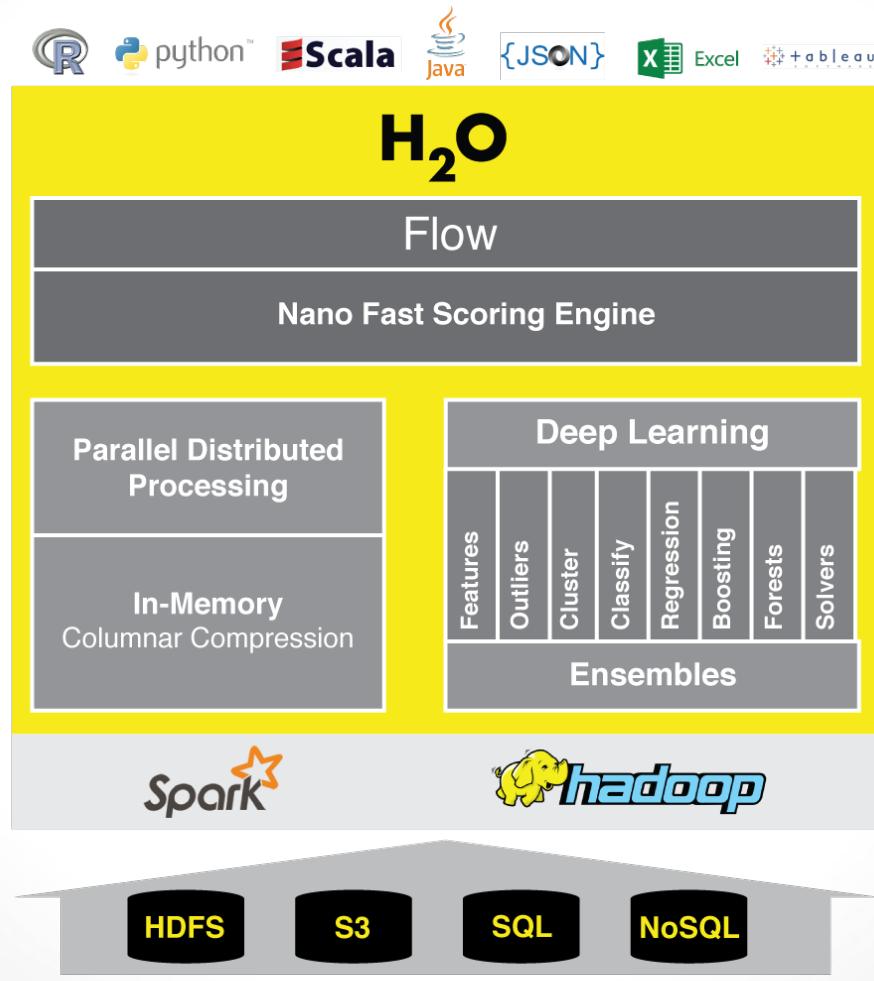
- Written in Java – perfect for Java Programmers
- REST API (JSON) – drives H2O from Browser UI, R, Python, Tableau

Big Data

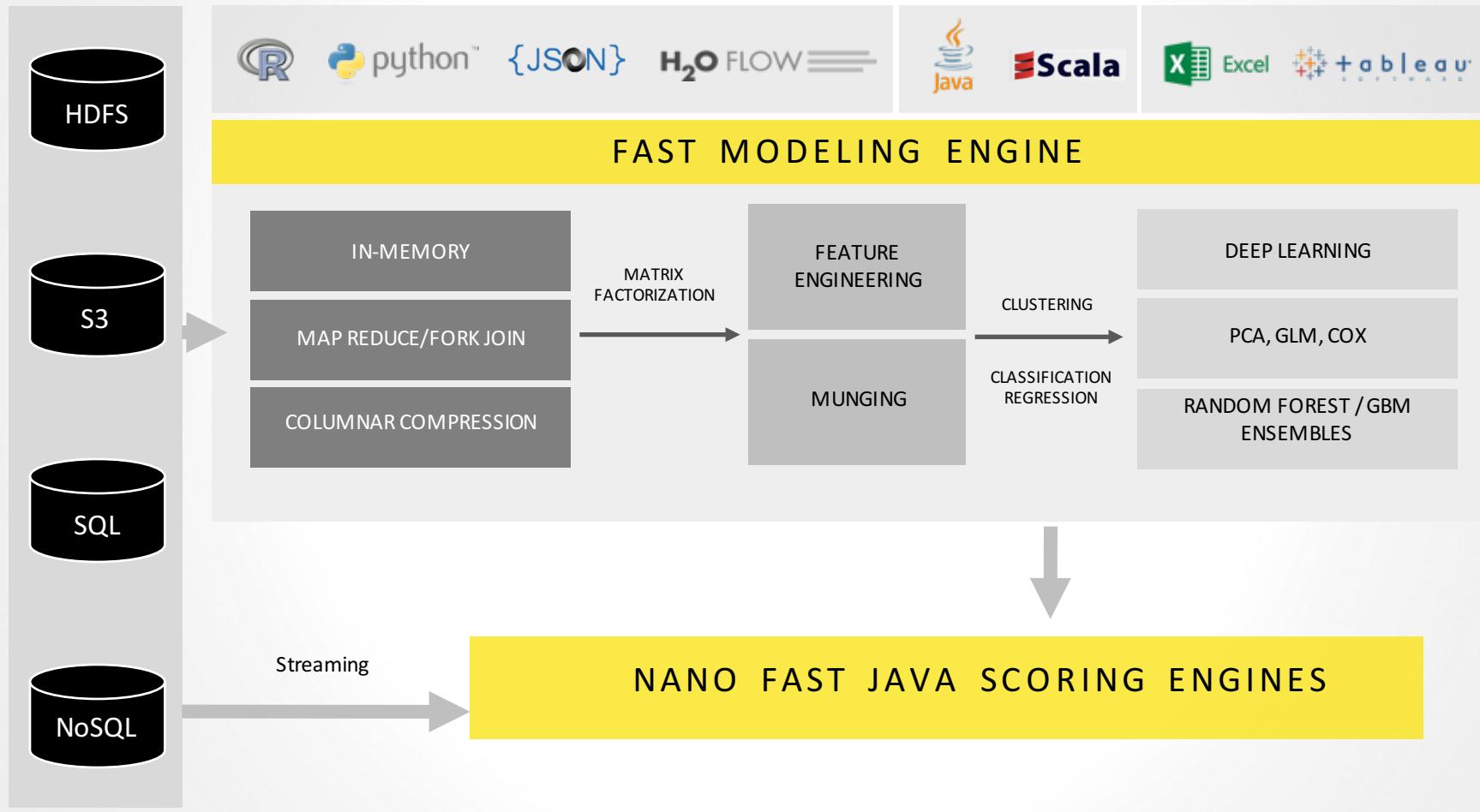
More data? Or better models? BOTH

- Use all of your data – model without down sampling
- Run a simple GLM or a more complex GBM to find the best fit for the data
- More Data + Better Models = Better Predictions

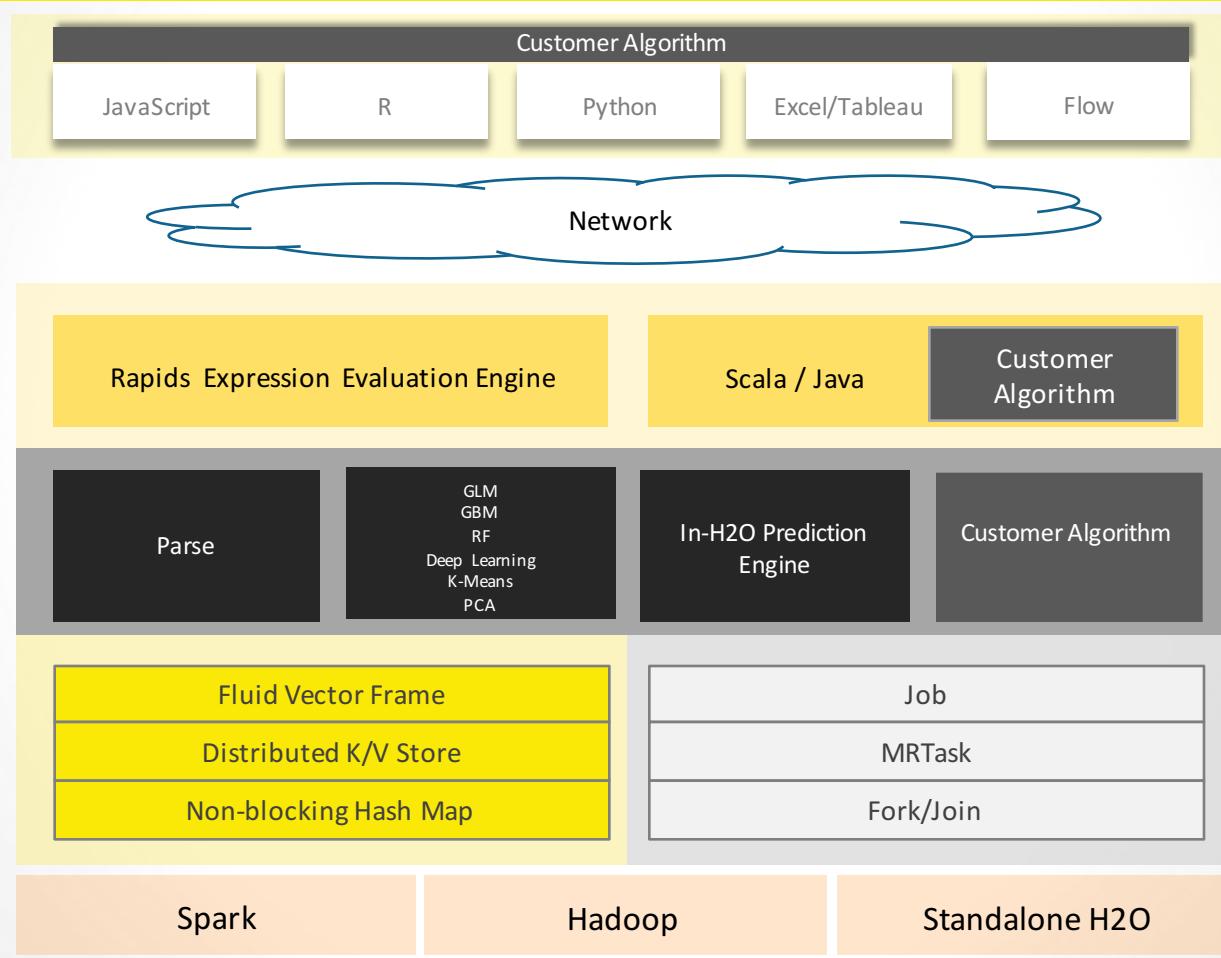
# ACCURACY WITH SPEED AND SCALE



# ACCURACY WITH SPEED AND SCALE



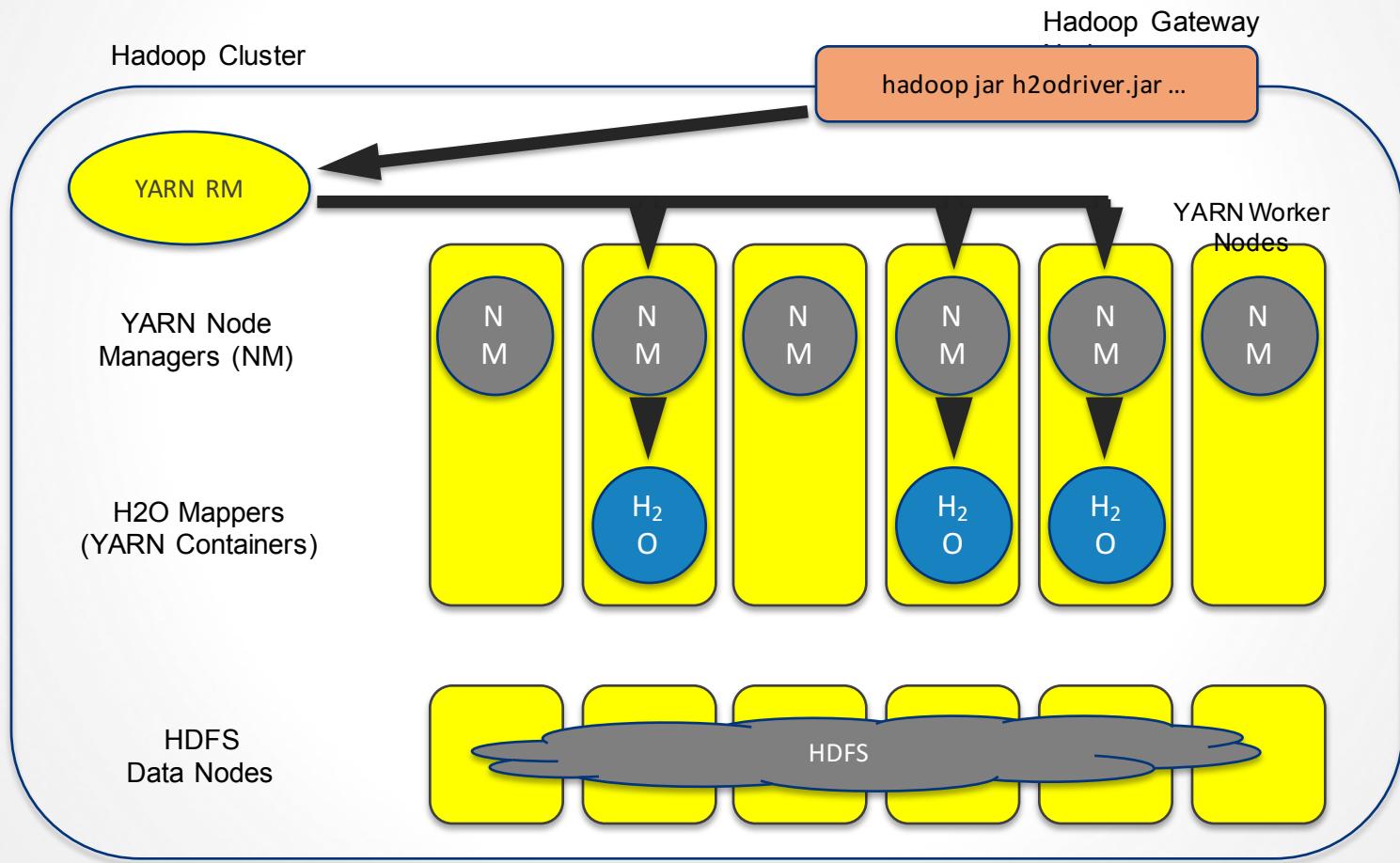
# H2O SOFTWARE STACK



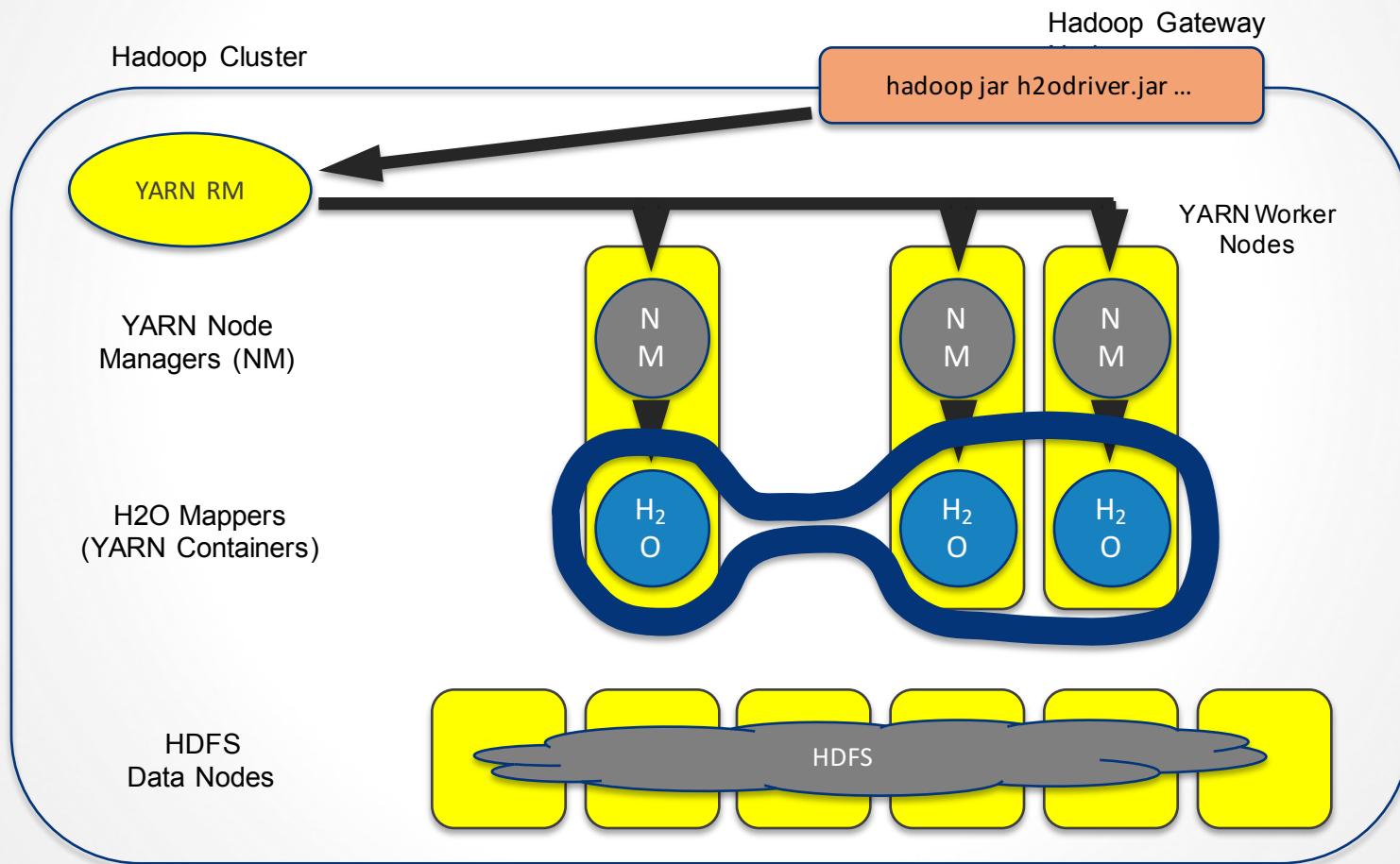
# Hadoop (and YARN)

- You can launch H2O directly on Hadoop:  
`$ hadoop jar h2odriver.jar ... -nodes 3 -mapperXmx 50g`
- H2O uses Hadoop MapReduce to get CPU and Memory on the cluster, not to manage work
  - H2O mappers stay at 0% progress forever
    - Until you shut down the H2O job yourself
  - All mappers (3 in this case) must be running at the same time
  - The mappers communicate with each other
    - Form an H2O cluster on-the-spot within your Hadoop environment
  - No Hadoop reducers(!)
- Special YARN memory settings for large mappers
  - yarn.nodemanager.resource.memory-mb
  - yarn.scheduler.maximum-allocation-mb
- CPU resources controlled via –nthreads h2o command line argument

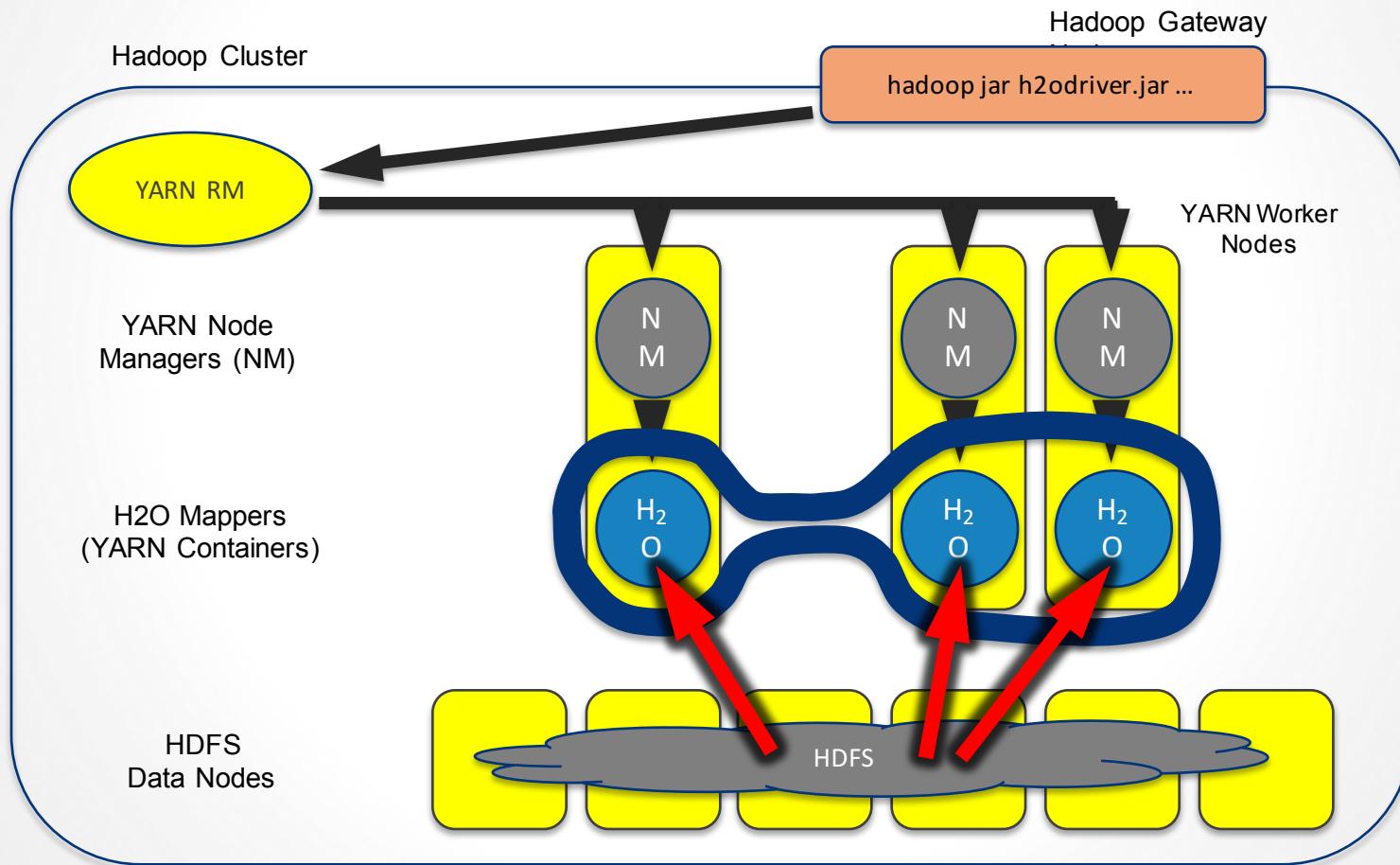
# H2O on YARN Deployment



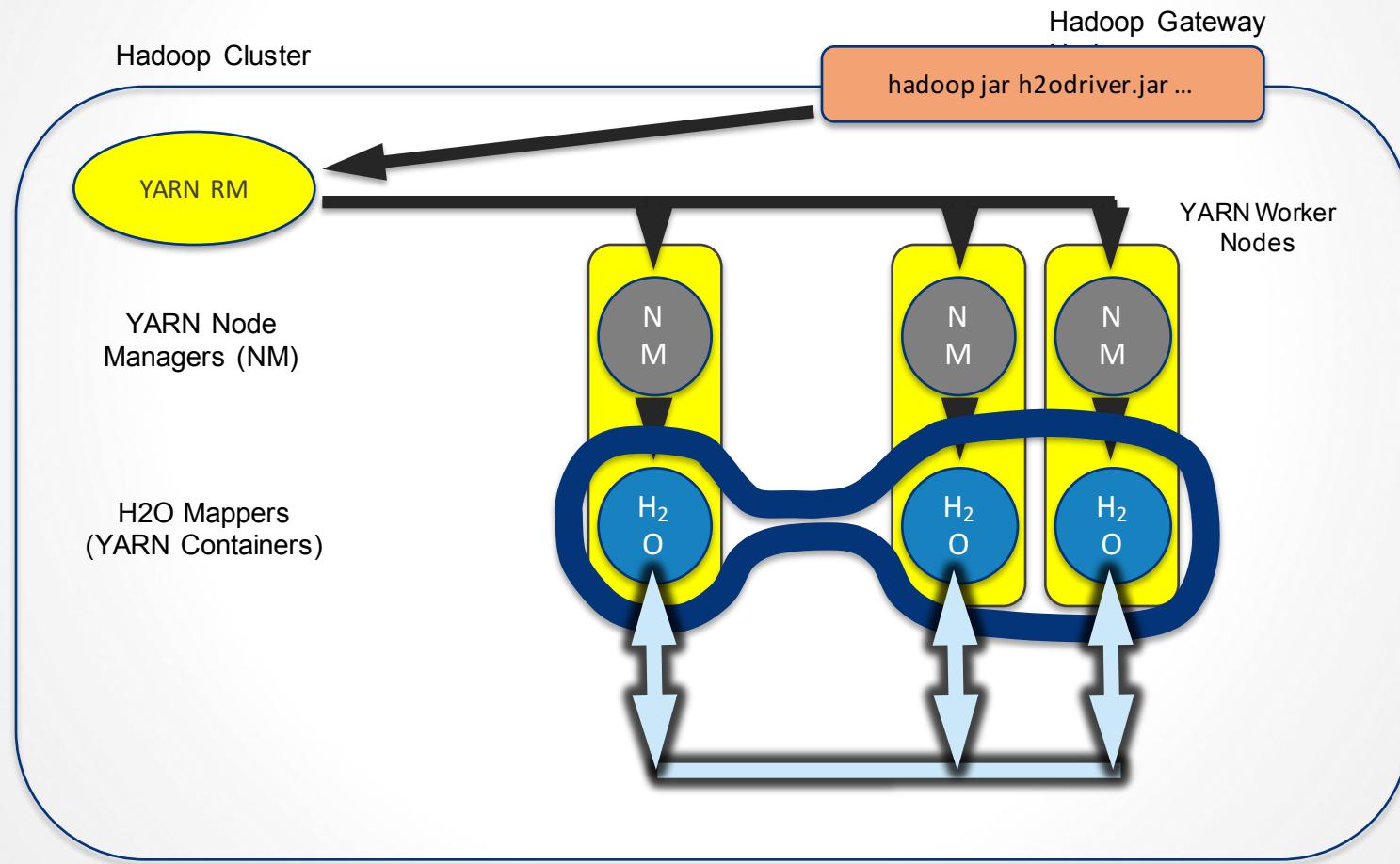
# Now You Have an H2O Cluster



# Read Data from HDFS *Once*



# Build Models *in-Memory*



# PYTHON AND R OBJECTS ARE PROXIES FOR BIG DATA

## STEP 1



```
→ h2o_df = h2o.import_file("hdfs://path/to/data.csv")
```

Python user

# PYTHON AND R OBJECTS ARE PROXIES FOR BIG DATA

## STEP 2

Python



2.1

Python function call

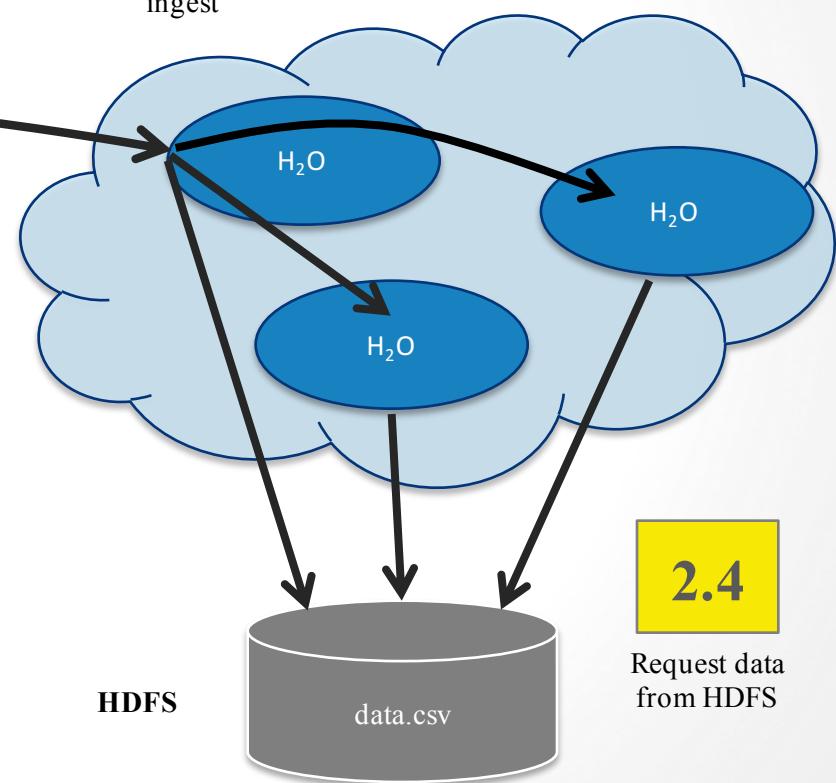
2.2

HTTP REST API  
request to H<sub>2</sub>O  
has HDFS path

2.3

Initiate distributed  
ingest

H<sub>2</sub>O Cluster



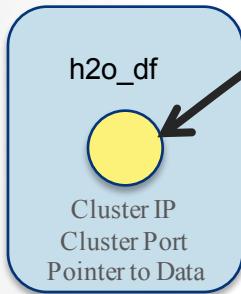
2.4

Request data  
from HDFS

# PYTHON AND R OBJECTS ARE PROXIES FOR BIG DATA

## STEP 3

Python



3.4

h2o\_df object created  
in Python

3.3

Return pointer to data  
in REST API JSON  
Response

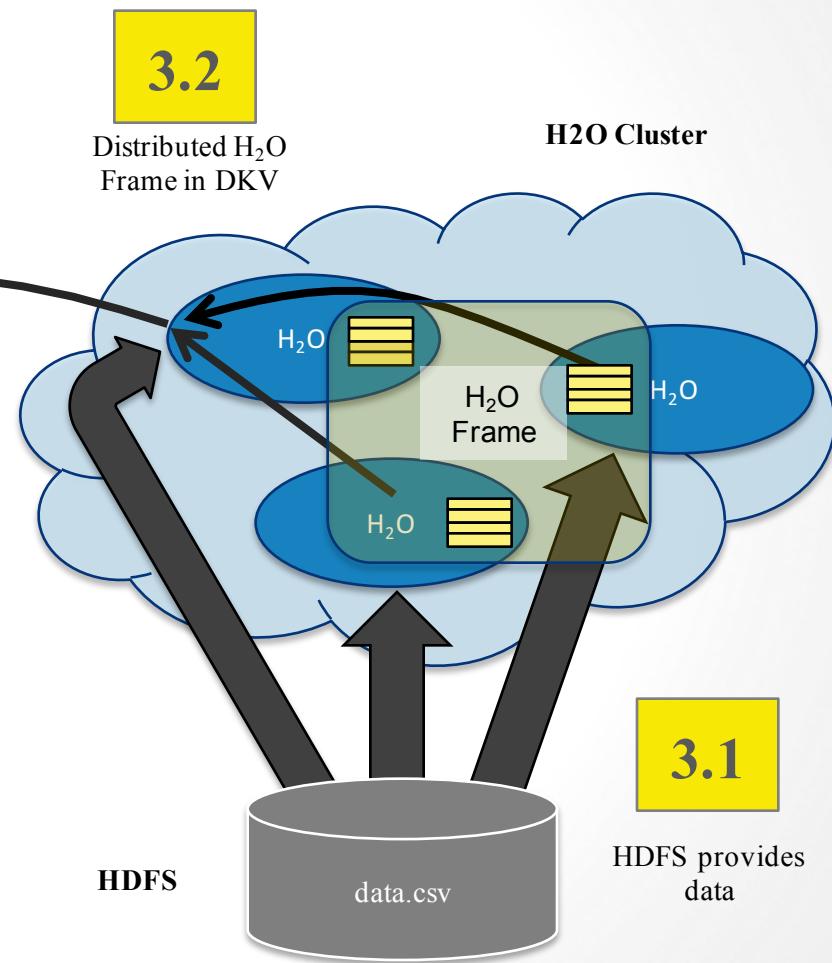
3.2

Distributed H<sub>2</sub>O  
Frame in DKV

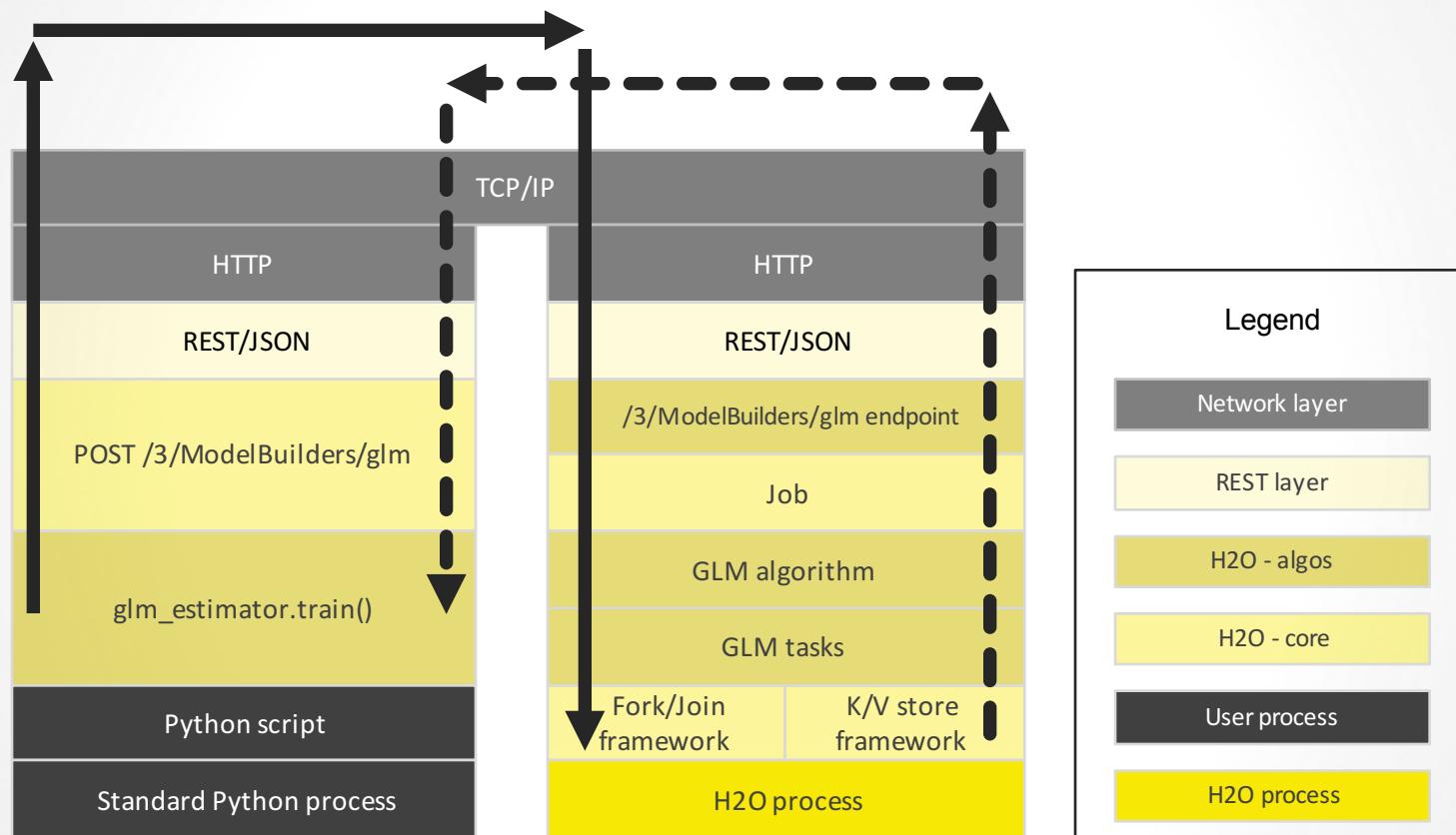
H<sub>2</sub>O Cluster

3.1

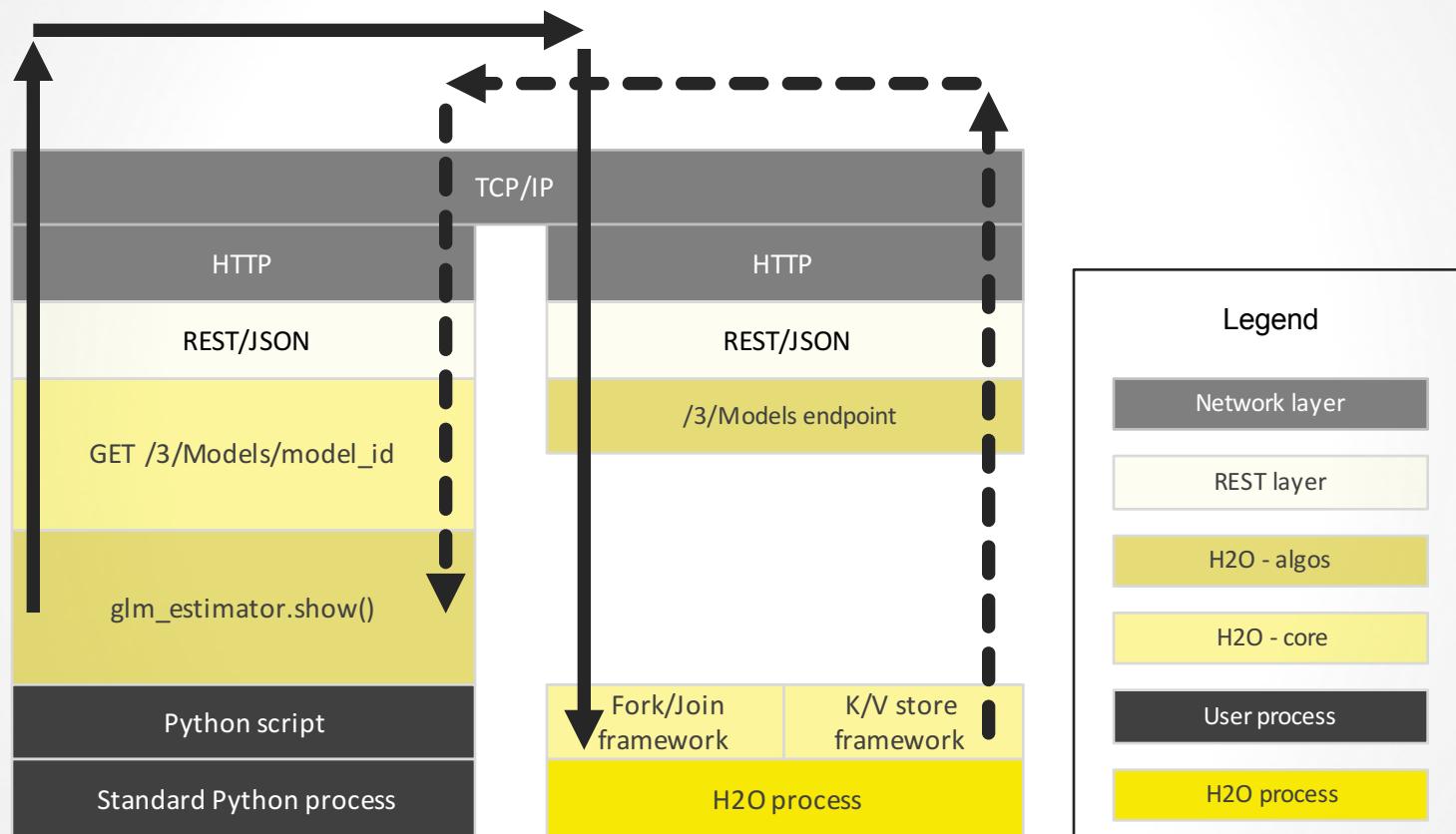
HDFS provides  
data



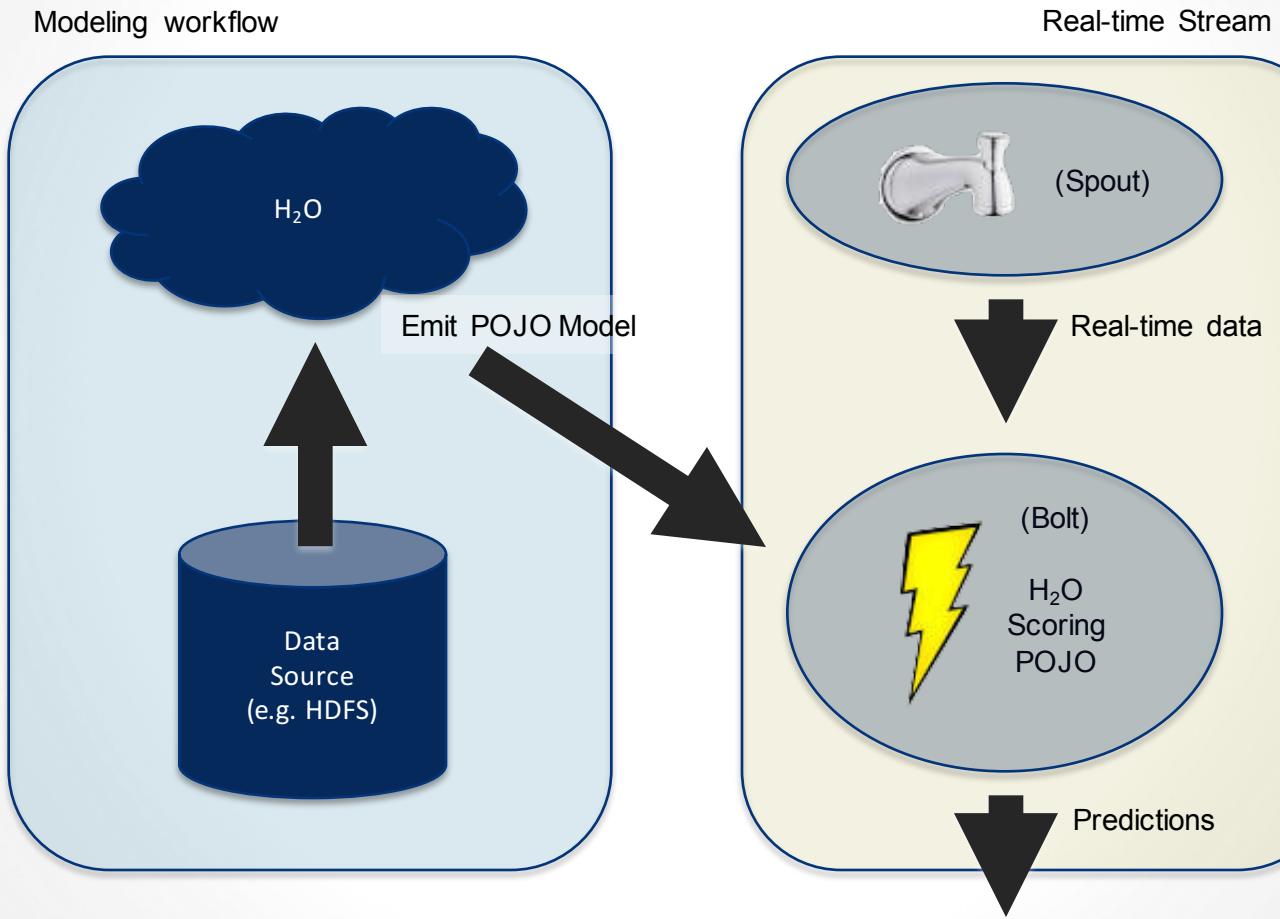
# PYTHON SCRIPT STARTING H2O GLM



# PYTHON SCRIPT STARTING H2O GLM

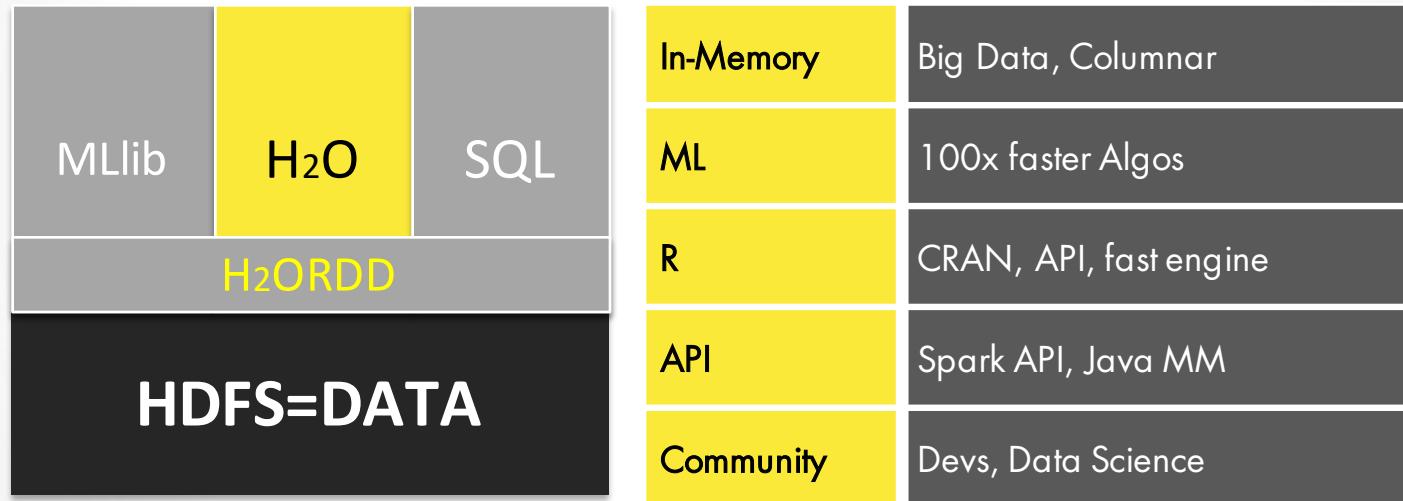


# H<sub>2</sub>O on Storm



# H<sub>2</sub>O – The Killer-App for Spark

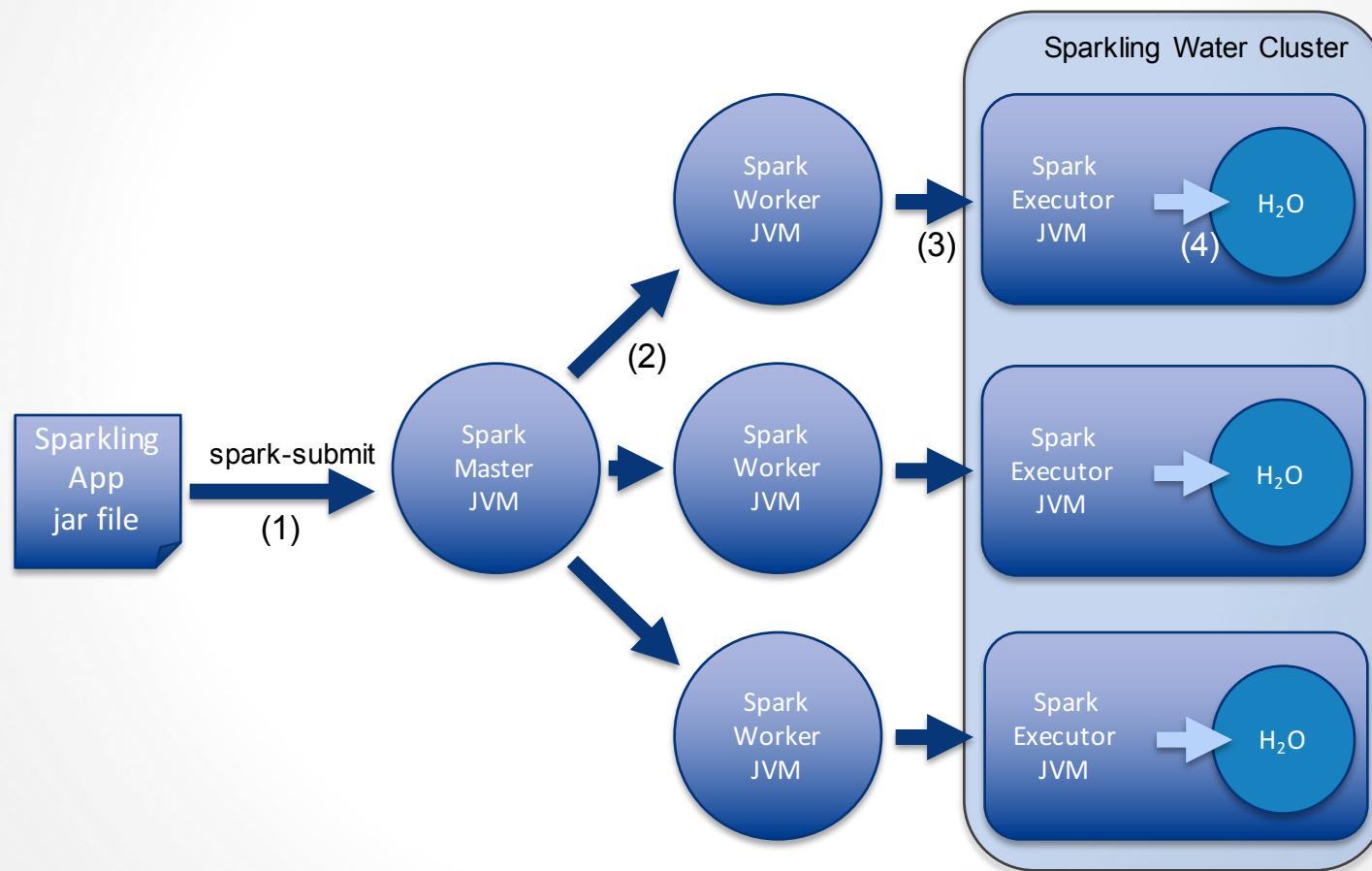
## H2O Sparkling Water



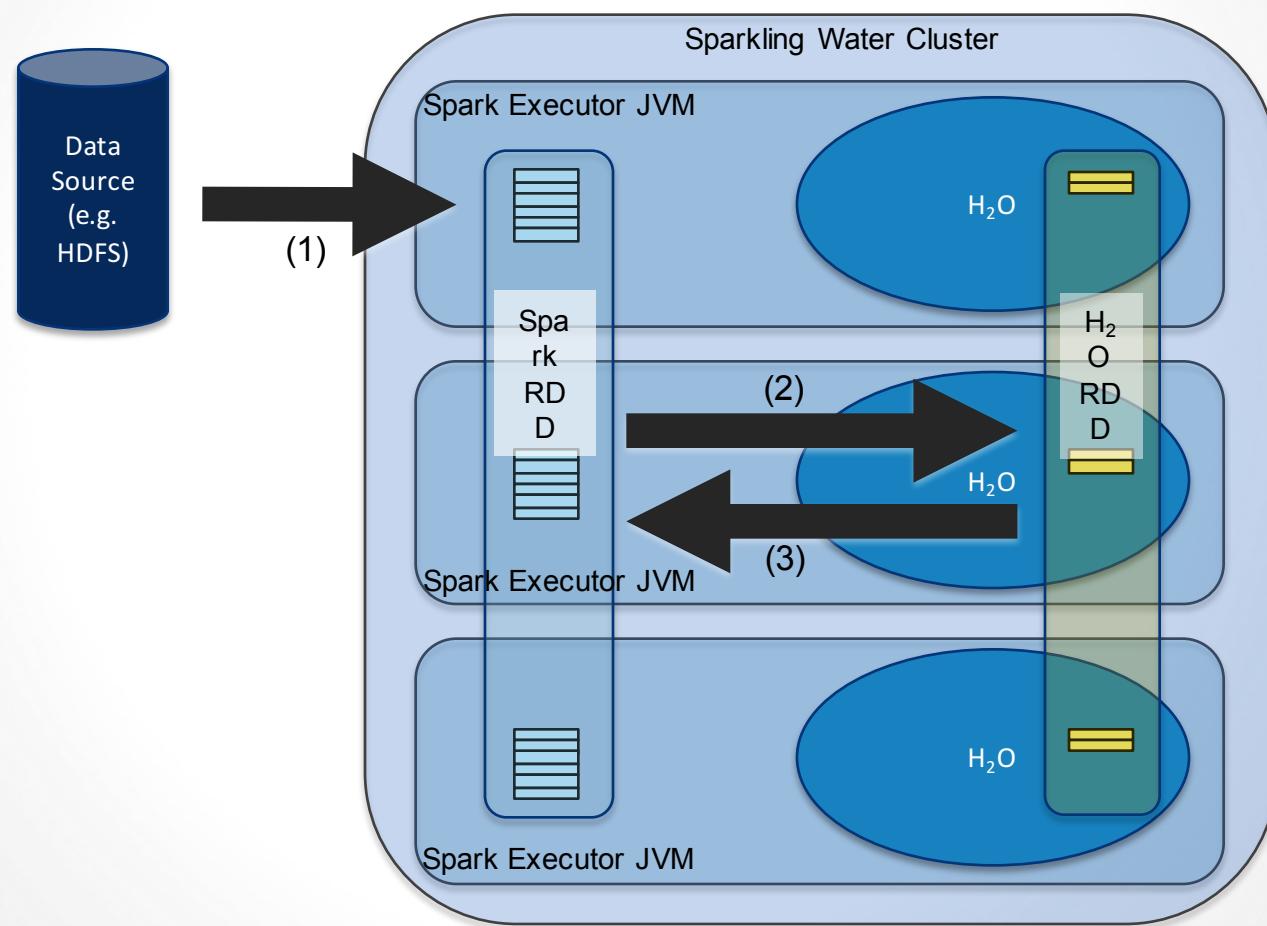
# Spark (and Sparkling Water)

- H2O runs as an application on a Spark cluster using spark-submit
  - Standard Spark 1.3+
  - Includes H2O on Spark on YARN
- H2O and Spark nodes share a JVM process
- H2ORDD facilitates easy data sharing between Spark (e.g. Spark SQL, MLlib) and H2O (e.g. Deep Learning)
- Scala & PySpark support

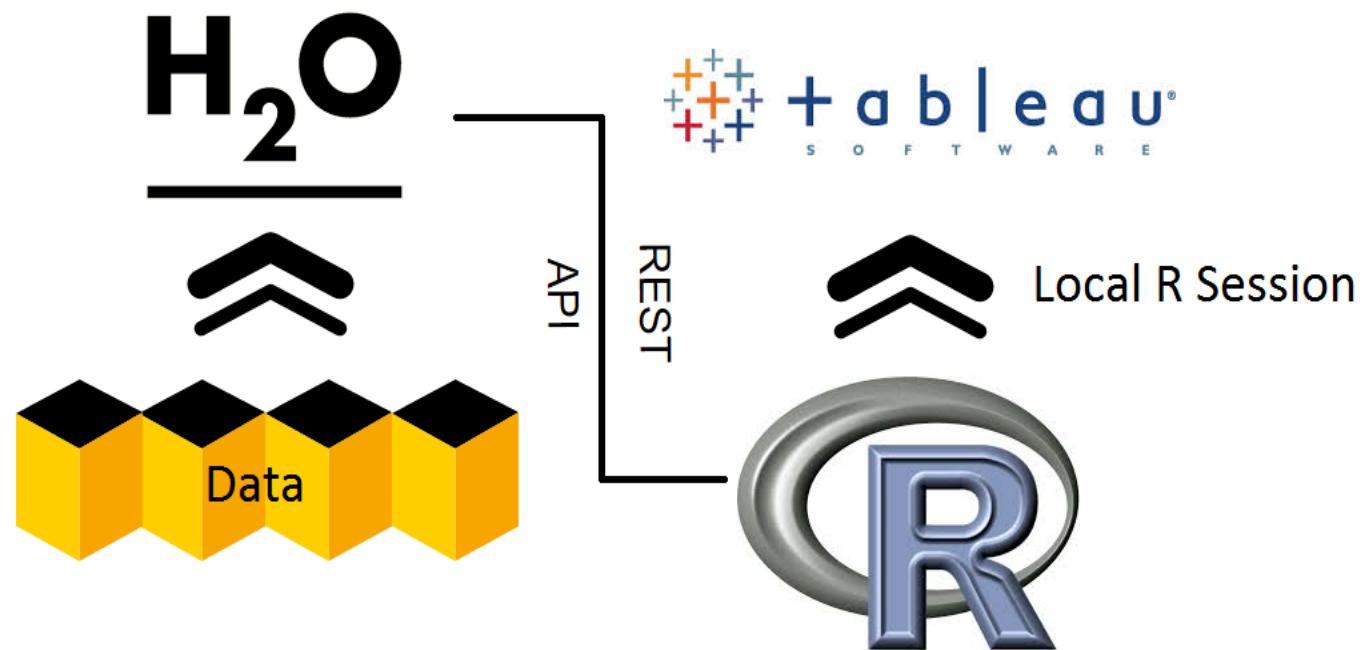
# Sparkling Water Application Life Cycle



# Sparkling Water Data Distribution



# OPEN FOR INTEGRATION



# RESOURCES

- Download and go: <http://www.h2o.ai/download>
- Documentation: <http://docs.h2o.ai/>
- Booklets, Datasheet: <http://www.h2o.ai/resources/>
- Github: <http://github.com/h2oai/>
- Training: <http://learn.h2o.ai/>

## GET INVOLVED

CONTRIBUTE: <http://github.com/h2oai>

COMMIT: [careers@h2o.ai](mailto:ccareers@h2o.ai)

(Algorithm engineers, software engineers,  
customer data scientists, solutions architects, ...)

# **THANK YOU**