From R Script to Production using rsparkling

Navdeep Gill, Hacker & Data Scientist @ H2O.ai

Agenda

- What/who is H2O?
- H2O Platform
- H2O Sparkling Water
- Sparklyr
- Rsparkling
- Demo

H2O.ai

H2O.ai

H2O Company

- Team : 65. Founded in 2012, Mountain View, CA
- Stanford Math & Systems Engineers

H2O Software

- Open Source Software (https://github.com/h2oai/h2o-3)
- Ease of Use via Web Interface (H2O Flow)
- R, Python, Scala, Spark, and Hadoop Interfaces
- Distributed Algorithms Scale to Big Data

Current Algorithm Overview

Statistical Analysis

- Linear Models (GLM)
- Naïve Bayes

Ensembles

- Random Forest
- Distributed Trees
- Gradient Boosting Machine
- Super Learner Ensembles

Deep Neural Networks

- Multi-layer Feed-Forward Neural Network
- Auto-encoder
- Anomaly Detection
- Deep Features

Clustering

- K-Means

Dimension Reduction

- Principal Component Analysis
- Generalized Low Rank Models

Solvers & Optimization

- Generalized ADMM Solver
- L-BFGS (Quasi Newton Method)
- Ordinary Least-Square Solver
- Stochastic Gradient Descent

Data Munging

- Scalable Data Frames
- Sort ,Slice, Log Transform

H2O.ai

H2O Components

H2O Cluster

- Multi-node cluster with share memory model
- All computations are in memory
- Each node only sees some rows of the data
- No limit on cluster size

Distributed Key Value Store

- Objects in the H2O cluster such as data frames, models and results are all reference by key
- Any node in the cluster can access any object in the cluster by key.

H2O Frame

- Distributed data frames (collection of vectors).
- Columns are distributed (across nodes) arrays
- Each node must be able to see the entire dataset (achieved by HDFS, S3, or multiple copies of the data if it is a CSV file).

H2O.ai

H2O in Spark

H2O.ai

H2O Sparkling Water

Spark Integration

- Sparkling Water is a transparent integration of H2O into the Spark ecosystem.
- H2O runs inside of the Spark Executor JVM.

Benefits

- Provides advanced machine learning algorithms to Spark workflows.
- Alternative to default Mllib library in Spark.

Sparkling Shell

- Sparkling Shell is just a standard Spark shell with addition Sparkling Water classes.
- Export MASTER="local-cluster[3,2,1024]"
- Spark-shell –jars sparkling-water.jar

https://github.com/h2oai/sparkling-water

H2O.ai

Sparkling Water Ecosystem

Scala: Sparkling Water

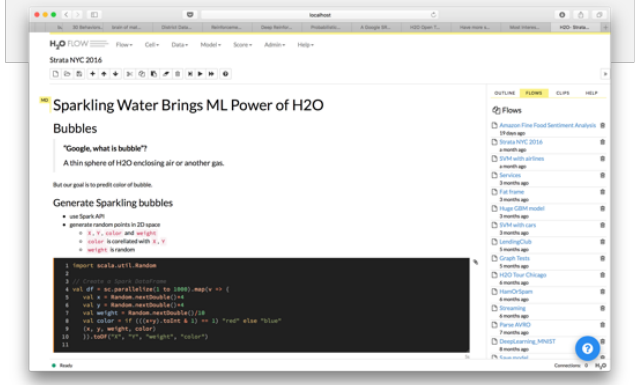
Spark

```
val sc = SparkContext.getOrCreate(...)

val df = sc.parallelize(1 to 10).toDF
```

```
val h2oContext =
H2OContext.getOrCreate(sc)

val hf = h2oContext.asH2OFrame(df)
```



Python: PySparkling Water

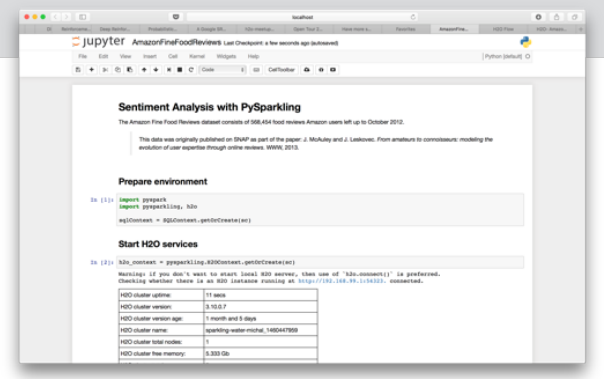
PySpark

```
sc = SparkContext(...)

df = sc.parallelize(range(1,11))
       .toDF("int")
```

```
h2o_context =
H2OContext.getOrCreate(sc)

hf = h2o_context.as_h2o_frame(df)
```



R: RSparkling Water

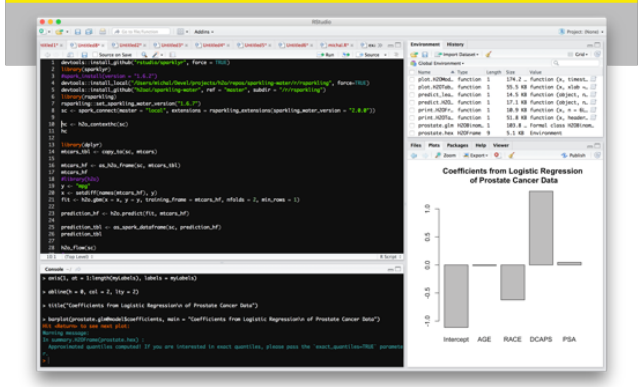
sparklyr

```
sc <- spark_connect(...)

tbl <- data_frame(c(1:10))
df <- copy_to(sc, tbl)
```

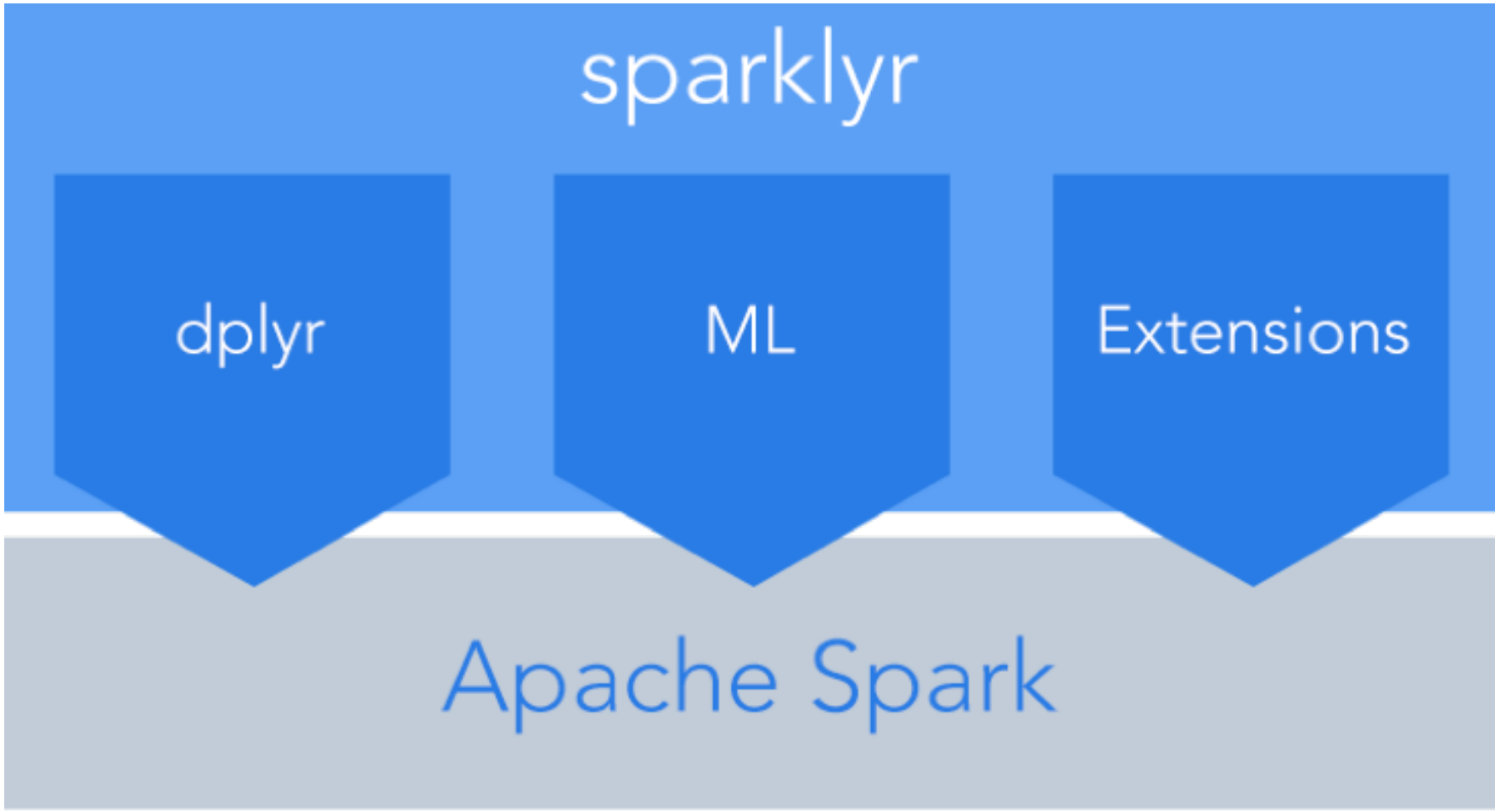
```
hc <- h2o_context(sc)

hf <- as_h2o_frame(sc, df)
```



H2O.ai

Sparklyr



H2O.ai

Sparklyr

- Connect to Spark from R.
- The sparklyr package provides a complete dplyr backend.
- Filter and aggregate Spark datasets then bring them into R for analysis and visualization.
- Use Spark's distributed machine learning library from R.
- Create extensions that call the full Spark API and provide interfaces to Spark packages.

```
library(sparklyr)
spark_install(version = "2.1.1")
sc <- spark_connect(master = "local")
my_tbl <- copy_to(sc,iris)
```

https://github.com/rstudio/sparklyr

H2O.ai

RSparkling

H2O.ai

RSparkling

- The rsparkling R package is an extension package for sparkapi / sparklyr that creates an R front-end for a Spark package (Sparkling Water from H2O) .
- This provides an interface to H2O's machine learning algorithms on Spark, using R.
- This package implements basic functionality (creating an H2OContext, showing the H2O Flow interface, and converting between Spark DataFrames and H2O Frames).

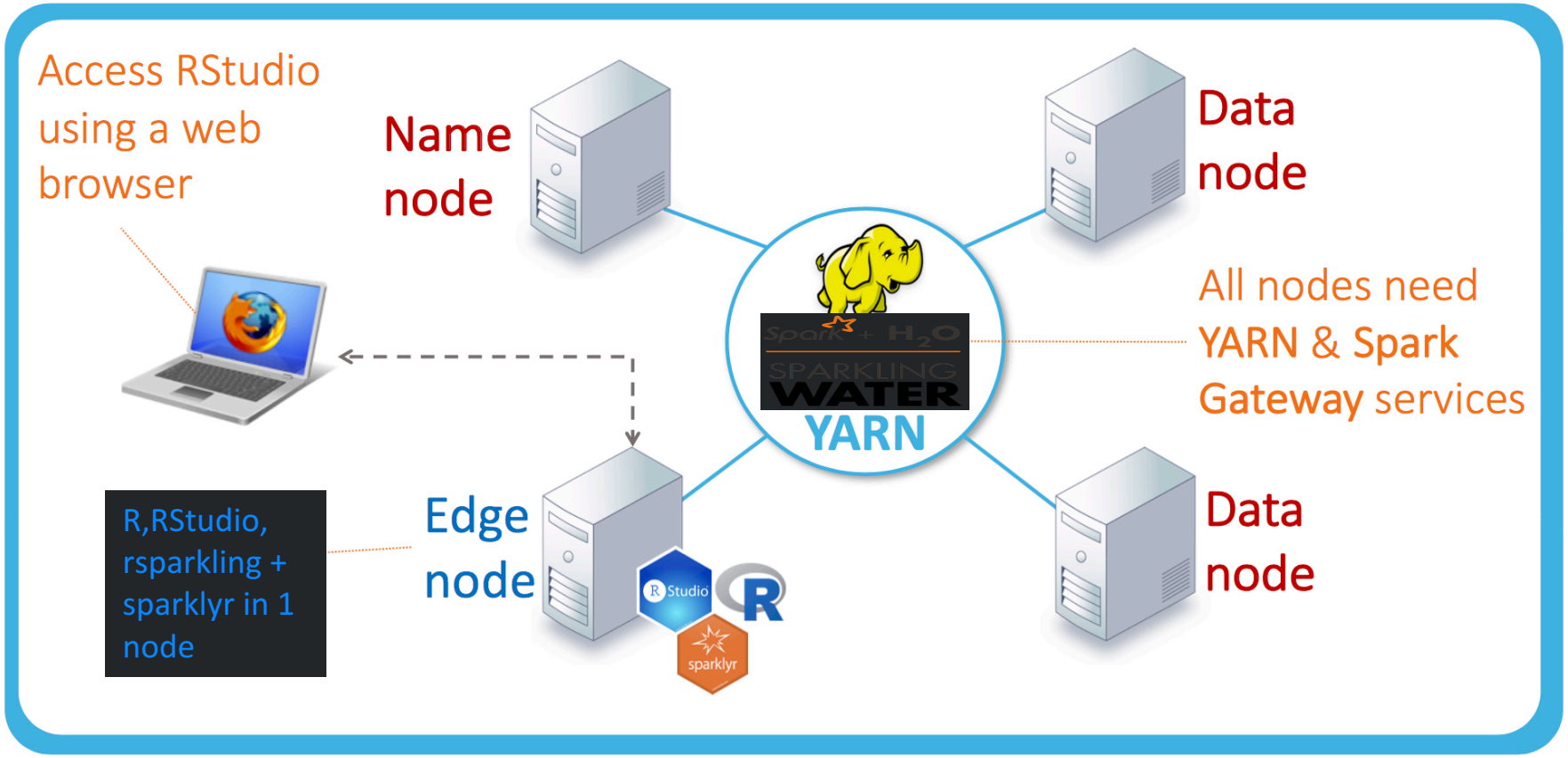
```
library(sparklyr)
spark_install(version = "2.0.0")
options(rsparkling.sparklingwater.version = "2.0.0")
library(rsparkling)
sc <- spark_connect(master = "local")
```

https://github.com/h2oai/rsparkling

H2O.ai

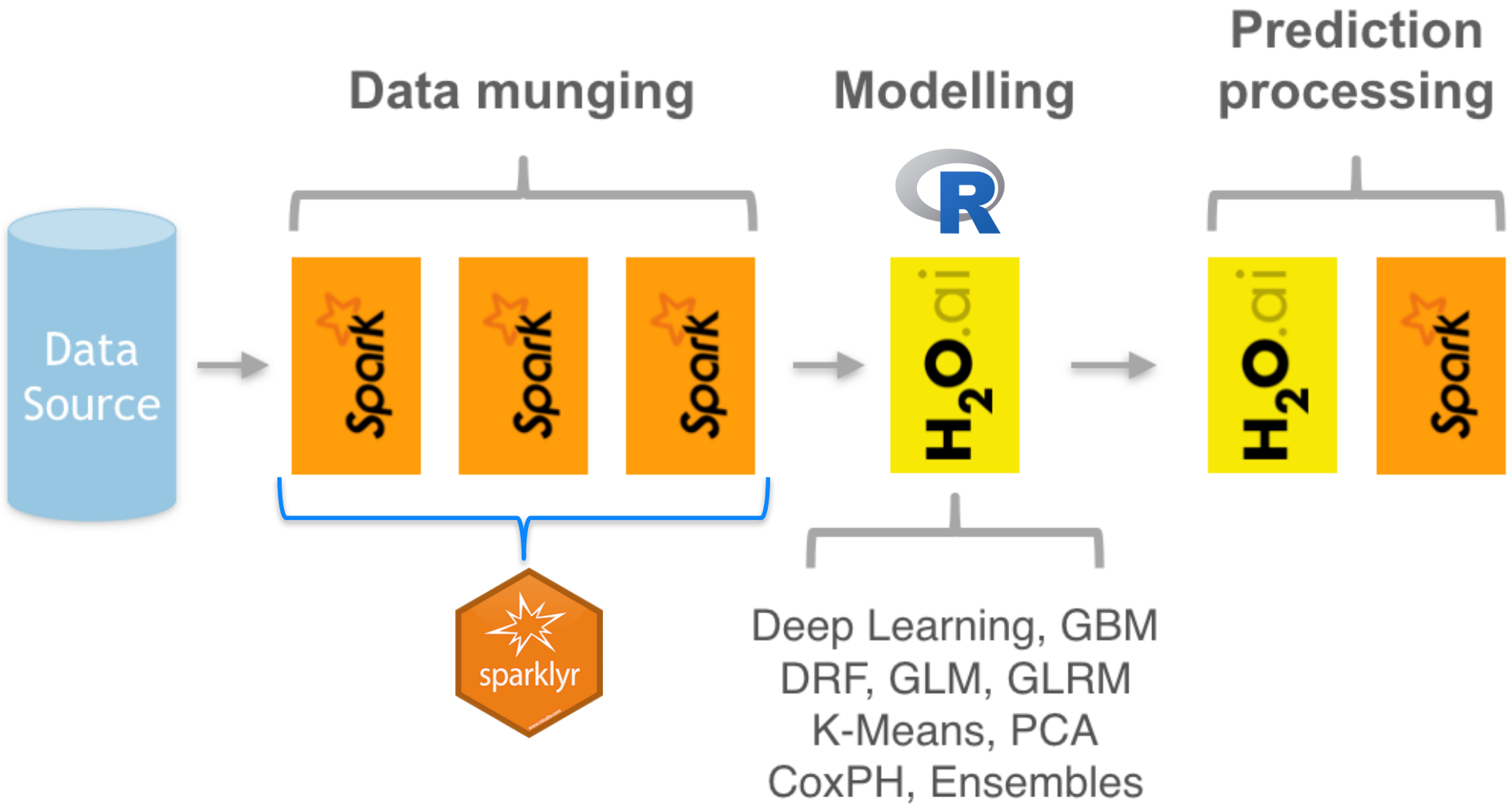
RSparkling

Cluster setup



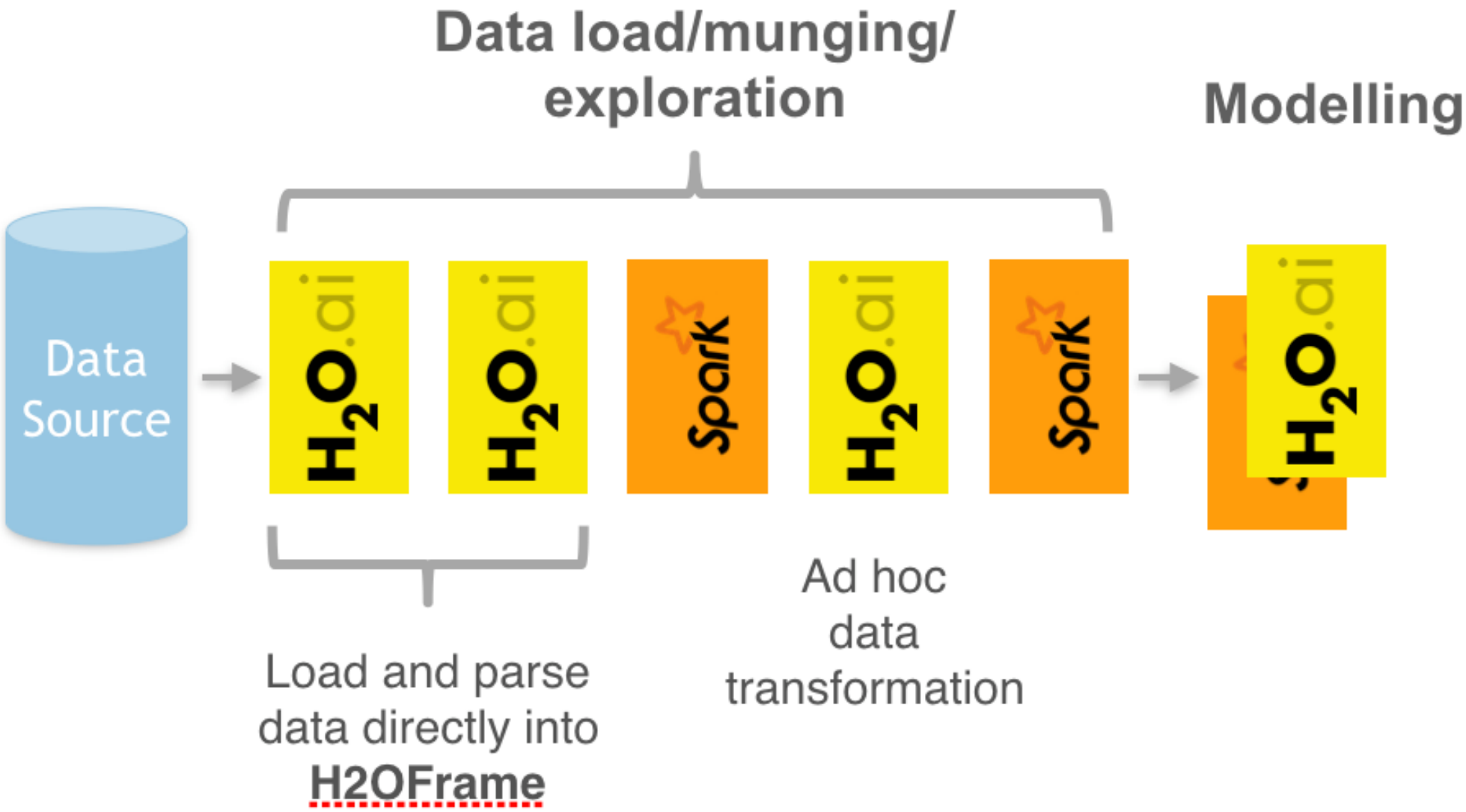
H2O.ai

Use Case



H2O.ai

Use Case


Data load/munging/
exploration
Modelling
Data
Source
H2O.ai
H2O.ai
Spark
H2O.ai
Spark
H2O.ai
Load and parse
data directly into
H2OFrame
Ad hoc
data
transformation

DEMO!

https://github.com/h2oai/rsparkling/blob/master/inst/examples/nycflights13.R

H2O.ai

Thank You.

@Navdeep_Gill_ on Twitter
navdeep-G on Github
navdeep@h2o.ai