SparkR

Advance Analytics for Big Data

A workshop with the Spark-Meetup
Tuesday 17th Nov 2015

Agenda

- INTORDUCTION
  - SPARK OVERVIEW
    - DATAFRAMES OVERVIEW
      - SPARKR
        - DEMO: MACHINE LEARNING

WHO AM I?

SAMUEL SHAMIRI

PhD STATISTICS + MSc ECONMETRICS

Senior Analyst

Samuel.Shamiri@veda.com.au

https://au.linkedin.com/pub/samuel-shamiri

http://sshamiri.blogspot.com/

veda applied intelligence is a data analytics business providing information and analytic services to businesses to assist them in making decisions and managing risks.

Veda holds data on more than **16.4 million** credit-active individuals, **3.6 million** on companies and businesses and **3.4 million** on Sole Traders throughout Australia, providing customers with the ability to make more informed decisions.

SPARK USERS in production by over 500 organizations

Telecom

Telefónica

verizon

Media

NETFLIX

Spotify

NBCUniversal

YAHOO!

Retail

阿里巴巴 Alibaba.com

ebay

Tencent 腾讯

Pharma

NOVARTIS

Investment

Goldman Sachs

Research

HHMI janelia farm research campus

Distributors

databricks

MAPR

IBM

cloudera

SAP

NTT DATA

ORACLE

DATASTAX

Hortonworks

other

intel

CISCO

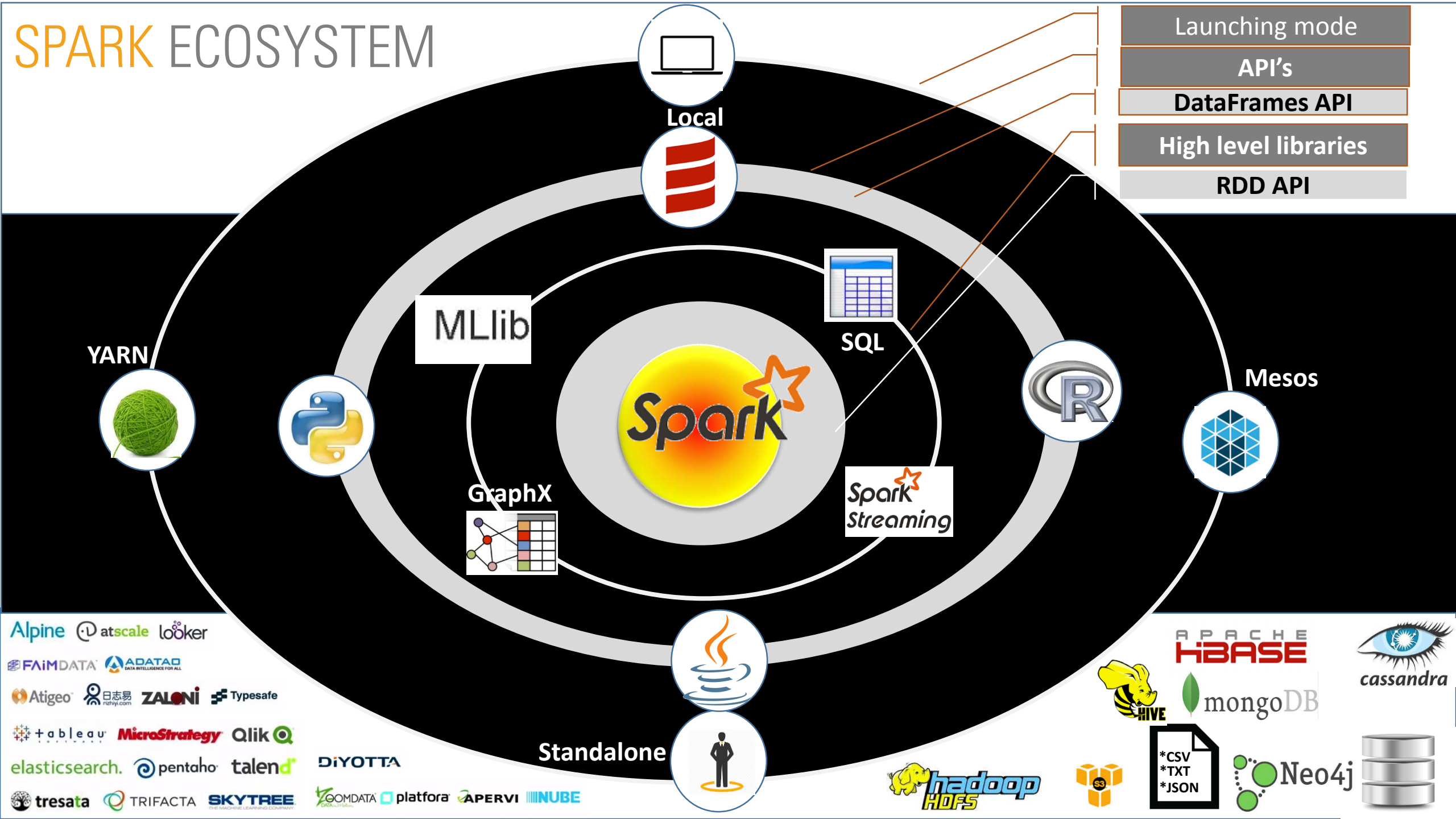
amazon web services

redhat

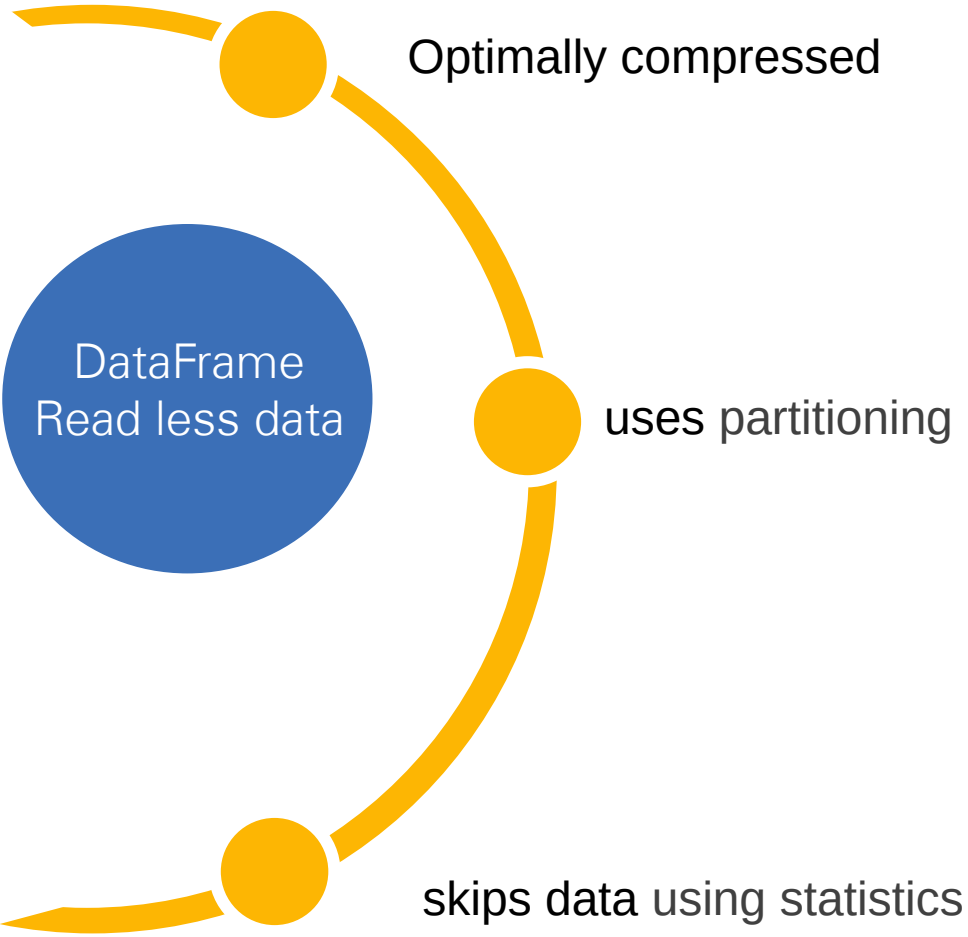
Space

NASA

SPARK ECOSYSTEM



INITIALIZE SPARK



Spark Context (sc)

The window to the world of Spark

sqlContext

The window to the world of DataFrames

Transformation (lazy)

Takes an RDD/DataFrame and returns a new RDD/DataFrame

Action

Causes an RDD to be evaluated (often storing the result)

mapPartitions()

ShuffledRDD

ZipPartitions()

How can I read this? Compute the average with…

```
first_name,last_name,gender,age
Erin,Shannon,F,42
Norman,Lockwood,M,81
Miguel,Ruiz,M,64
Rosalita,Ramirez,F,14
Ally,Garcia,F,39
Claire,McBride,F,23
Abigail,Cottrell,F,75
José,Rivera,M,59
Ravi,Dasgupta,M,25
…
```

RDD

DataFrame

WRITE LESS CODE, BETTER READABILITY

```
private IntWritable one =
  new IntWritable(1)
private IntWritable output =
  new IntWritable()
proctected void map(
    LongWritable key,
    Text value,
    Context context) {
  String[] fields = value.split("\t")
  output.set(Integer.parseInt(fields[1]))
  context.write(one, output)
}

IntWritable one = new IntWritable(1)
DoubleWritable average = new DoubleWritable()

protected void reduce(
    IntWritable key,
    Iterable<IntWritable> values,
    Context context) {
  int sum = 0
  int count = 0
  for(IntWritable value : values) {
     sum += value.get()
     count++
    }
  average.set(sum / (double) count)
  context.Write(key, average)
}
```

RDD

```
peopleRDD <- textFile(sc, “people.txt”)
lines <- flatMap(peopleRDD,
function(line) {
strsplit(line, ", ")
})
ageInt <- lapply(lines,
function(line) {
as.numeric(line[2])
})
sum <- reduce(ageInt,function(x,y) {x+y})
avg <- sum / count(peopleRDD)
```

DataFrame

```
df <- read.df(sqlCtx, “people.json”, “json”)
avg <- select(df, avg(df$age))
```

Super awesome distributed, in-memory collections
Schemas == metadata, structure and declarative

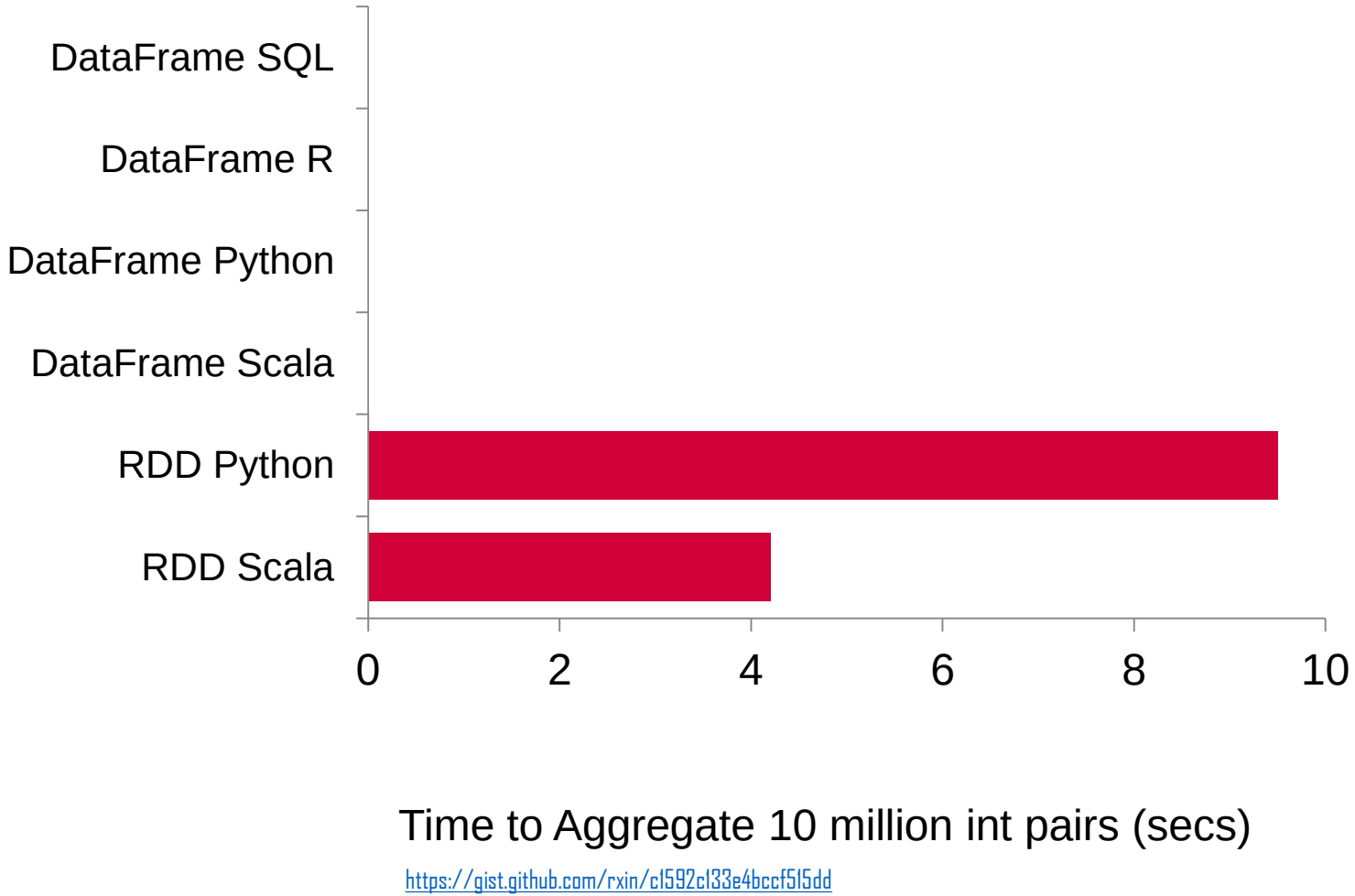
NOT R v PYTHON v SCALA, IT'S R/PYTHON/SCALA + SPARK

Easier to program

Significantly fewer Lines of Code

Improved performance

via intelligent optimizations and code-generation

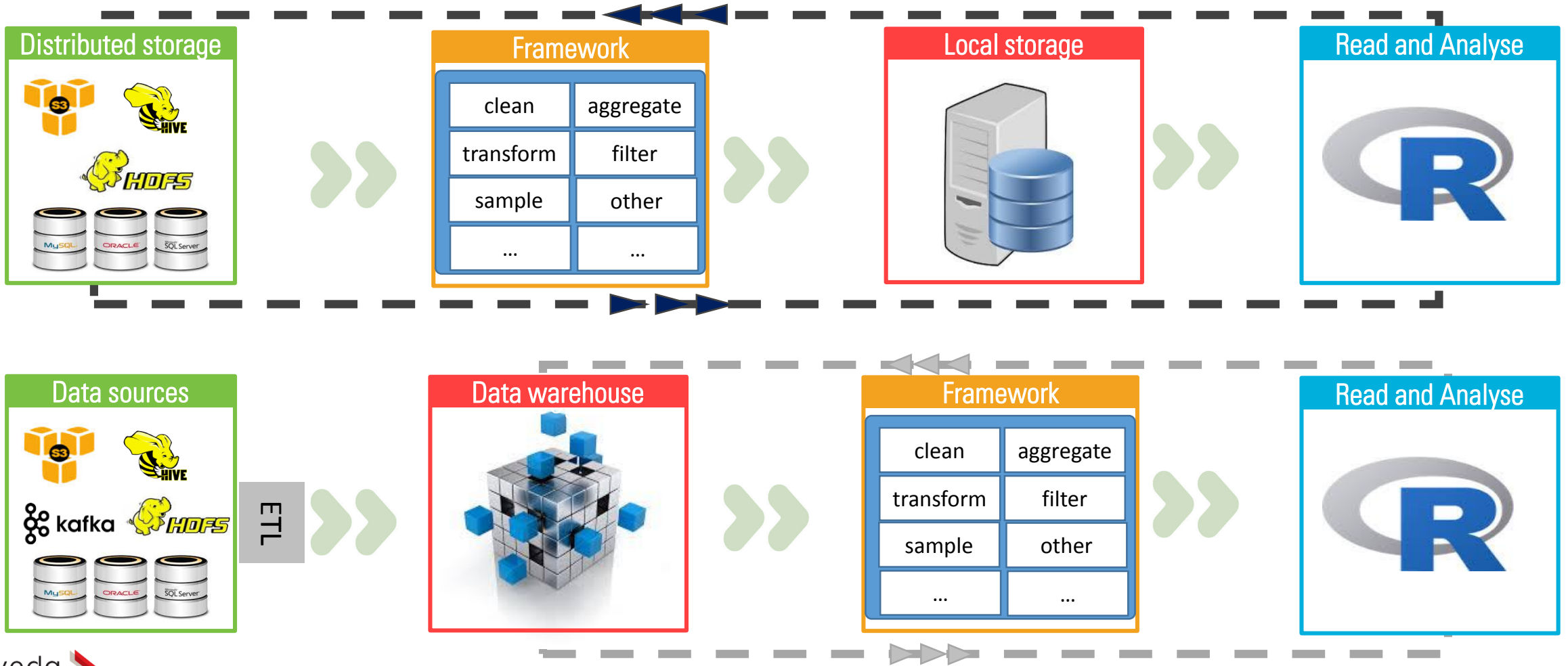


https://gist.github.com/rxin/c1592c133e4bccf515dd

veda
applied intelligence

LIMITATION - COMPLICATION:R with other frameworks

R dynamic design imposes performance problem on runtime (single threaded, fit all in memory). Data scientists uses R in conjunction with other frameworks as



veda
applied intelligence

USE SPARK'S DISTRIBUTED, PARRLLEL IN MEMORY COLLECTION

Distributed storage/Data source

S3 HIVE kafka HDFS MySQL ORACLE SQL Server

Spark

Read and Analyse

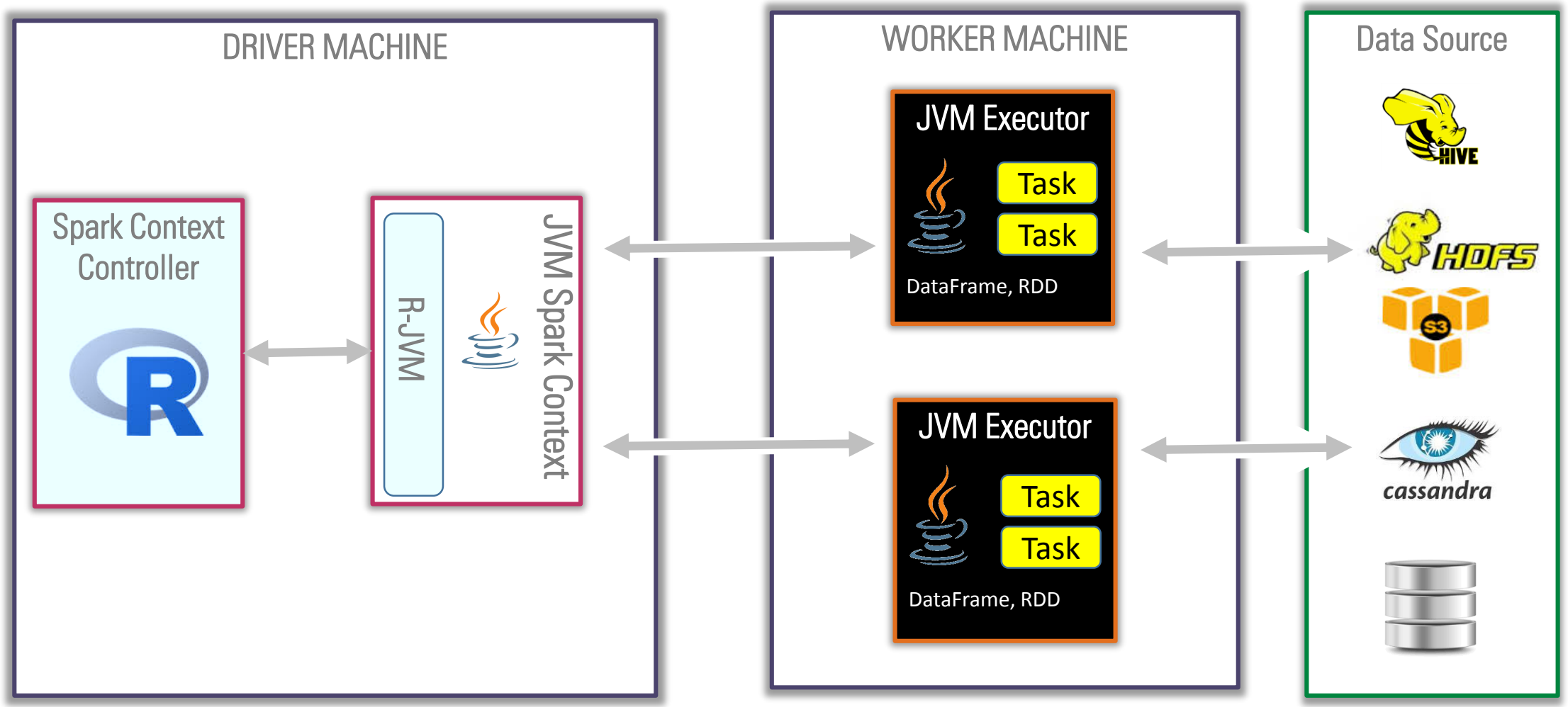
R

distributed/robust processing, off-memory data structures for interactive analysis at speed

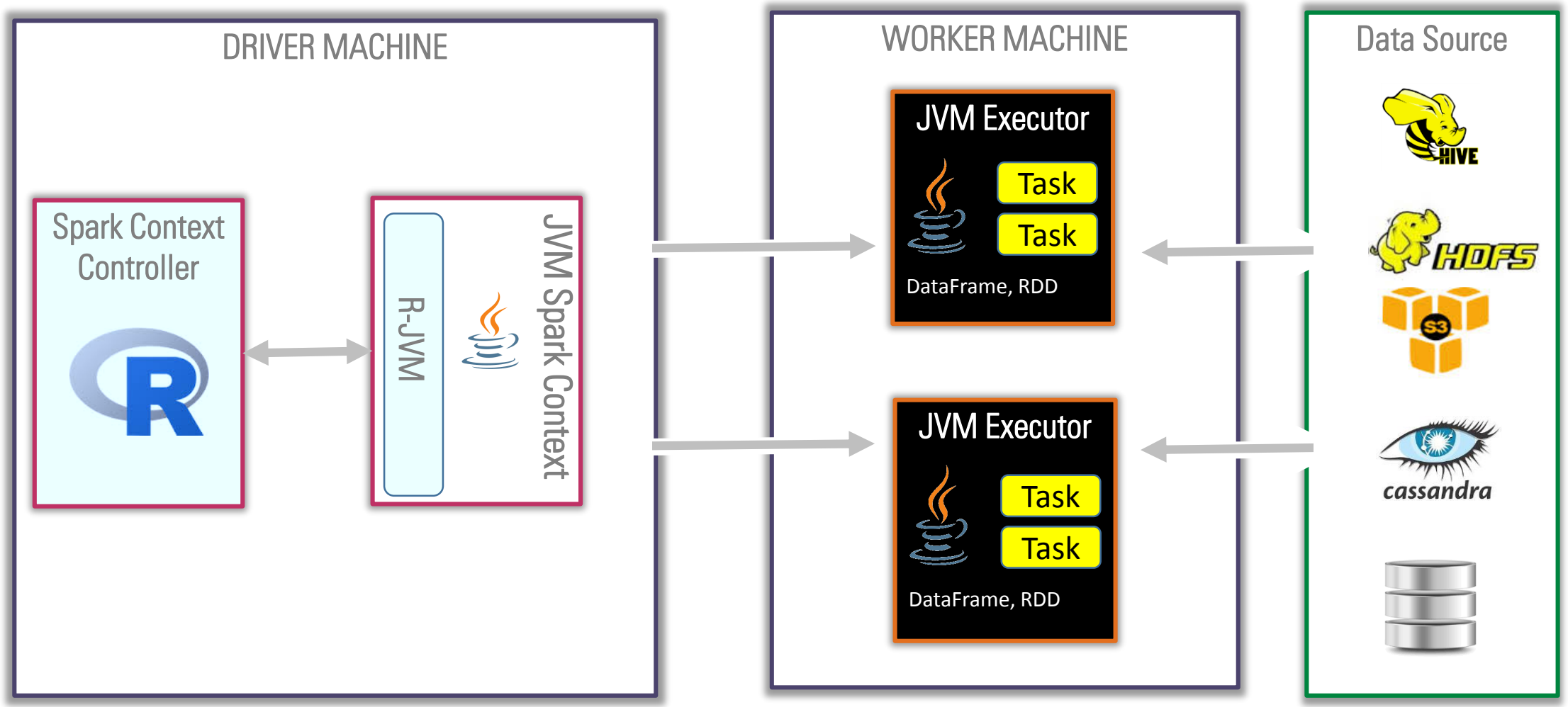
Dynamic environment, interactvity, packages, visualizaJon

veda
applied intelligence

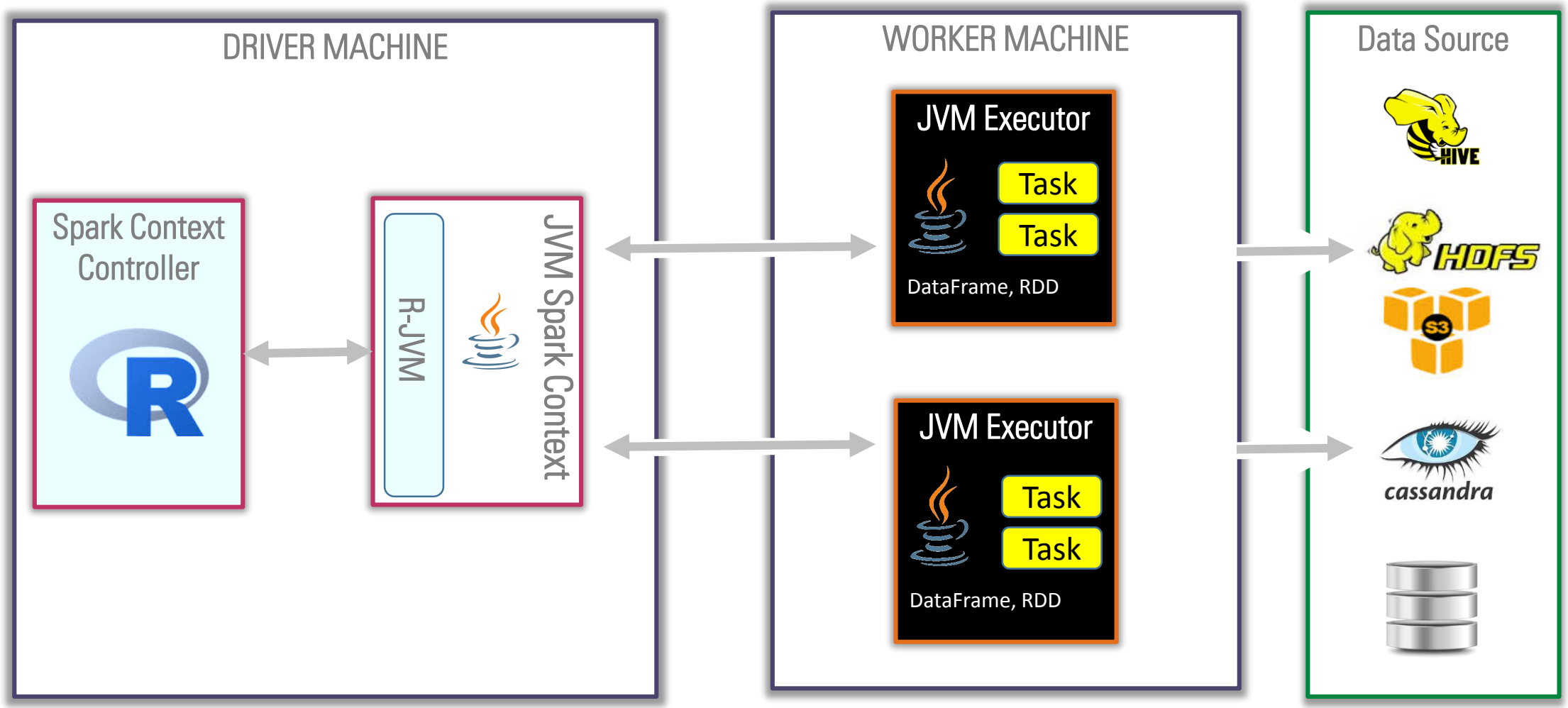
SPARKR ARCHITECTURE



SPARKR ARCHITECTURE



SPARKR ARCHITECTURE



Data wrangling and
Machine learning with SparkR

Questions

Slides, Demo, and Data available on GitHub at

@SamuelShamiri

https://github.com/SShamiri/SparkR