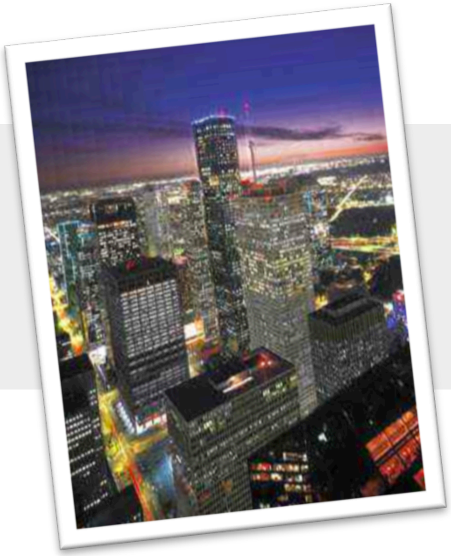
Enabling R on Hadoop

July 11, 2013

Your Presenters



Ravi Mutyala

Systems Architect

Paul Codding

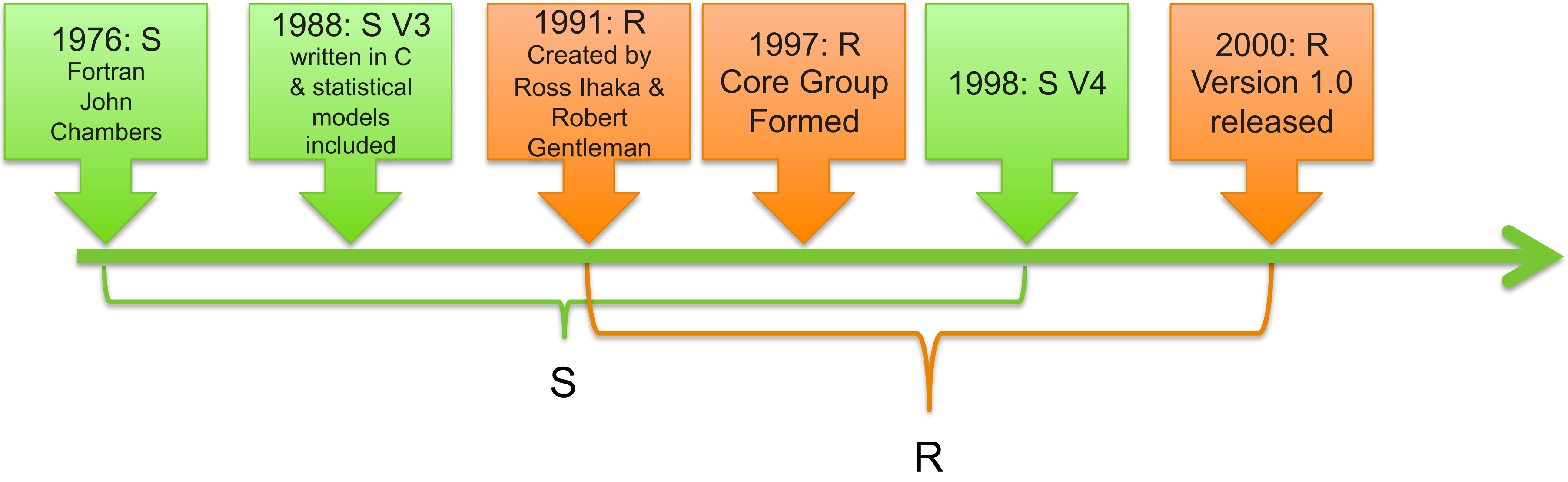
Solutions Engineer

Agenda

- **A Brief History of R**
- **How R is Typically Used**
- **How R is Used with Hadoop**
- **Getting Started**

A Brief History of R

History of R



1976: S
Fortran
John
Chambers

1988: S V3
written in C
& statistical
models
included

1991: R
Created by
Ross Ihaka &
Robert
Gentleman

1997: R
Core Group
Formed

1998: S V4

2000: R
Version 1.0
released

S

R

Hortonworks

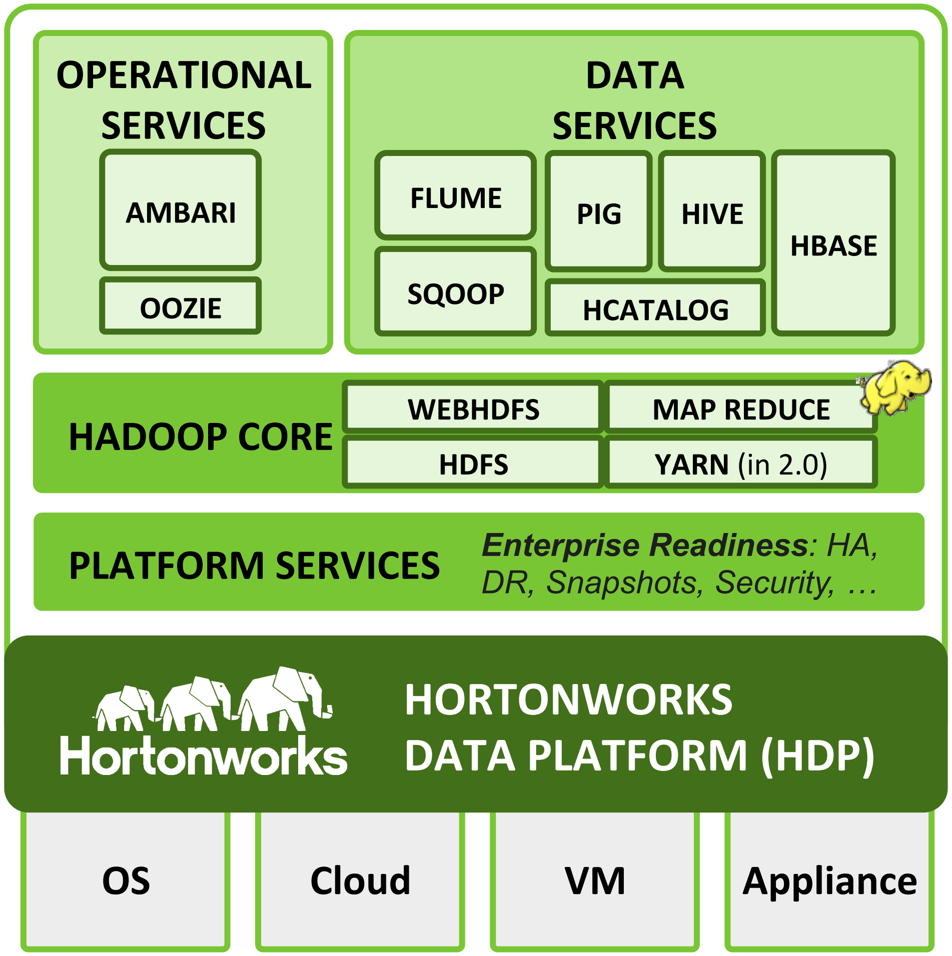
How R is Typically Used

Main Uses of R

- **Statistical Analysis & Modeling**
  - Classification
  - Scoring
  - Ranking
  - Clustering
  - Finding relationships
  - Characterization
- **Common Uses**
  - Interactive Data Analysis
  - General Purpose Statistics
  - Predictive Modeling

How R is Used with Hadoop

Hadoop Components



OPERATIONAL SERVICES

AMBARI

OOZIE

DATA SERVICES

FLUME

SQOOP

PIG

HIVE

HCATALOG

HBASE

HADOOP CORE

WEBHDFS

MAP REDUCE

HDFS

YARN (in 2.0)

PLATFORM SERVICES

***Enterprise Readiness**: HA, DR, Snapshots, Security, …*

Hortonworks

HORTONWORKS DATA PLATFORM (HDP)

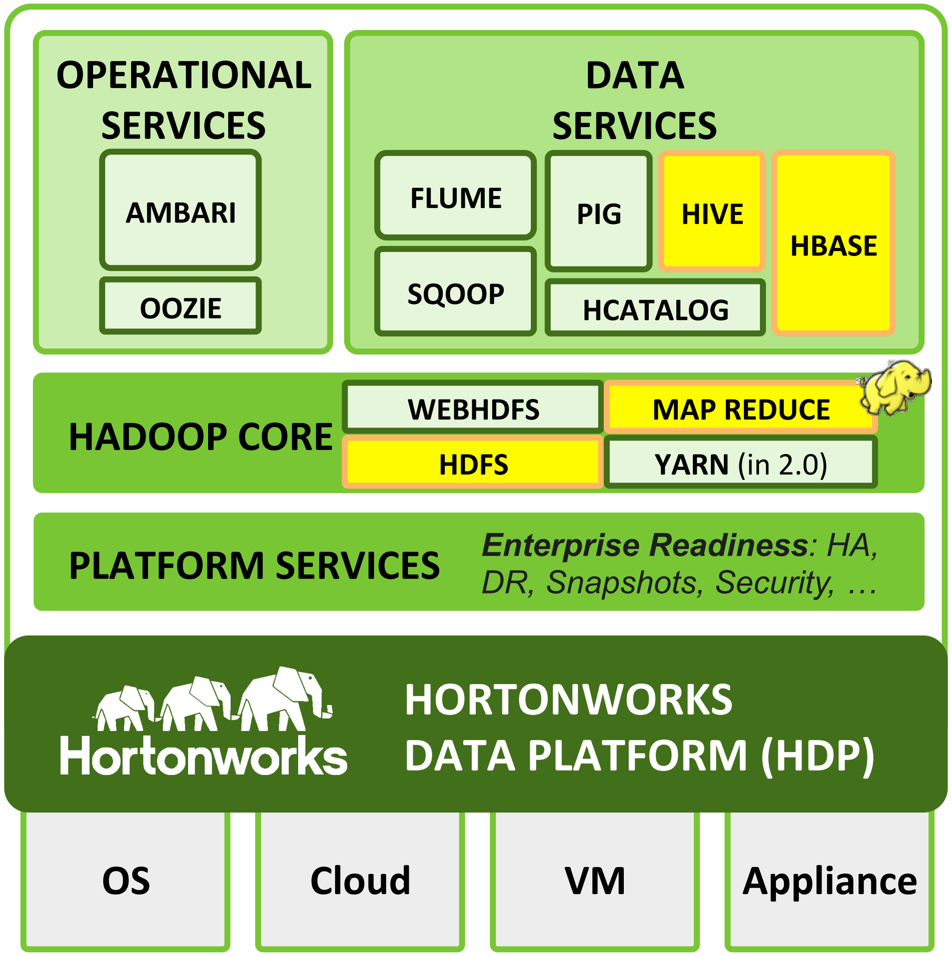
OS

Cloud

VM

Appliance

Hadoop Components & R



Data Service Components

- Hive
- HBase

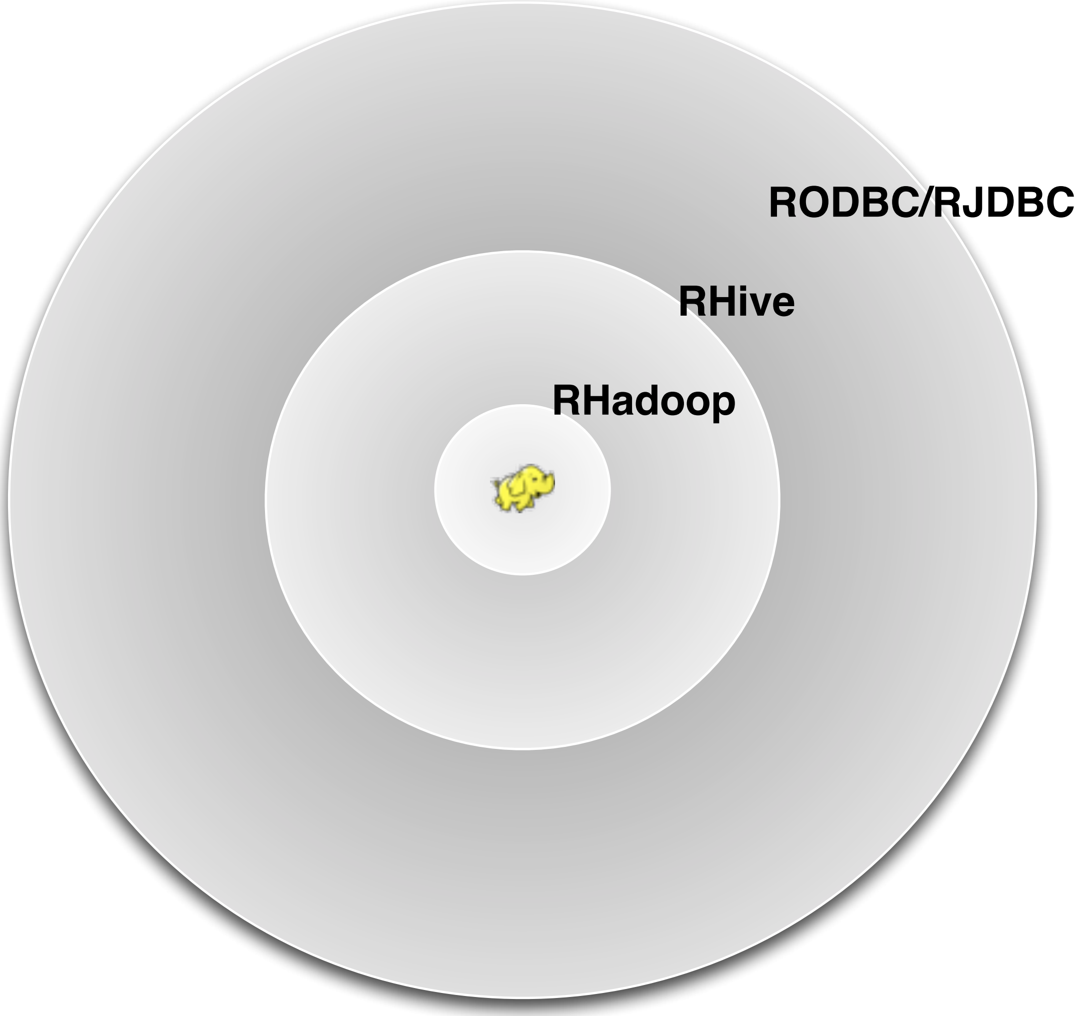
Hadoop Core

- Map Reduce
- HDFS

Options for R on Hadoop

- **Options**
  - RODBC/RJDBC
  - RHive
  - RHadoop

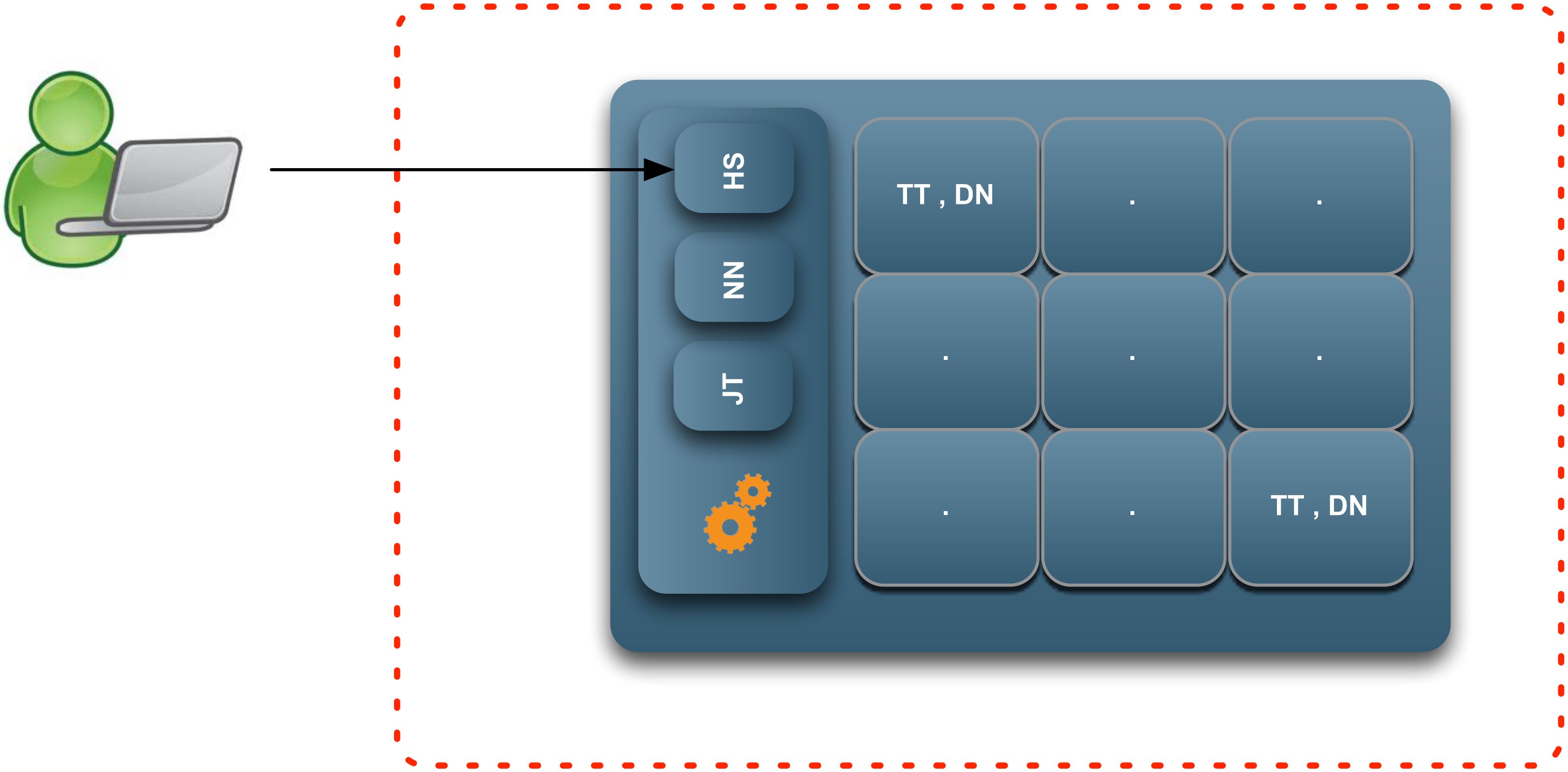
- **Analysis**
  - Focus
  - Integration Ease
  - Benefits
  - Limitations



RODBC/RJDBC

- **Focus**
  - SQL Access from R
- **Integration Ease**
  - Install Hortonworks Hive ODBC Driver
  - Install Hive libraries
- **Benefits**
  - Low impact on existing R scripts leveraging other DB packages
  - Not required to install Hadoop configuration/binaries on client machines
- **Limitations**
  - Parallelism limited to Hive
  - Result set size

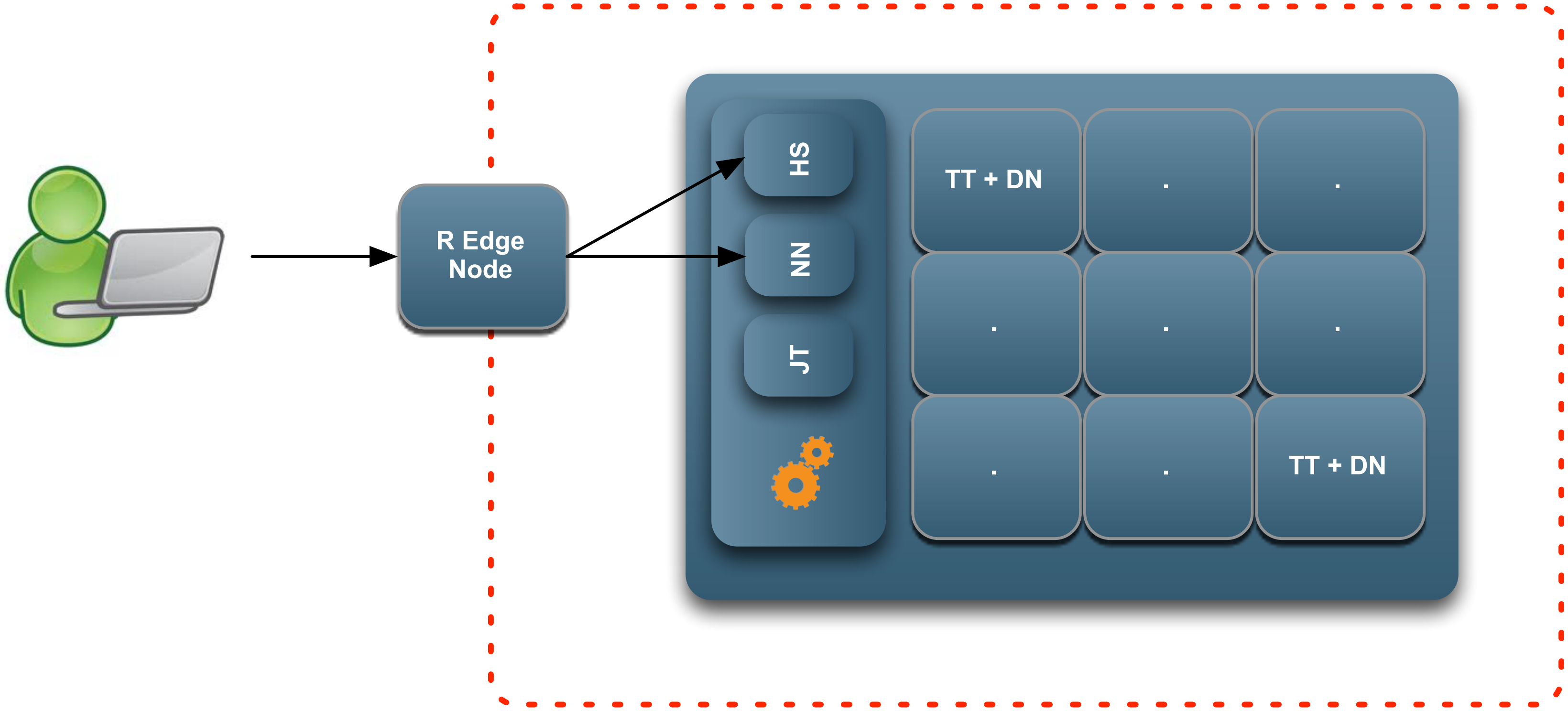
Deployment Considerations



RHive

- **Focus**
  - Broad access to Hive and HDFS
- **Integration Ease**
  - Requires Hadoop binaries, libraries, and configuration files on client machines
  - Uses Java DFS Client and HiveServer
- **Benefits**
  - Wide range of features expressed through HQL
    - rhive-apply *R Distributed apply function using HQL*
- **Limitations**
  - Requires heavy client deployment
  - Dependent on HiveServer, and can't be used with HiveServer2

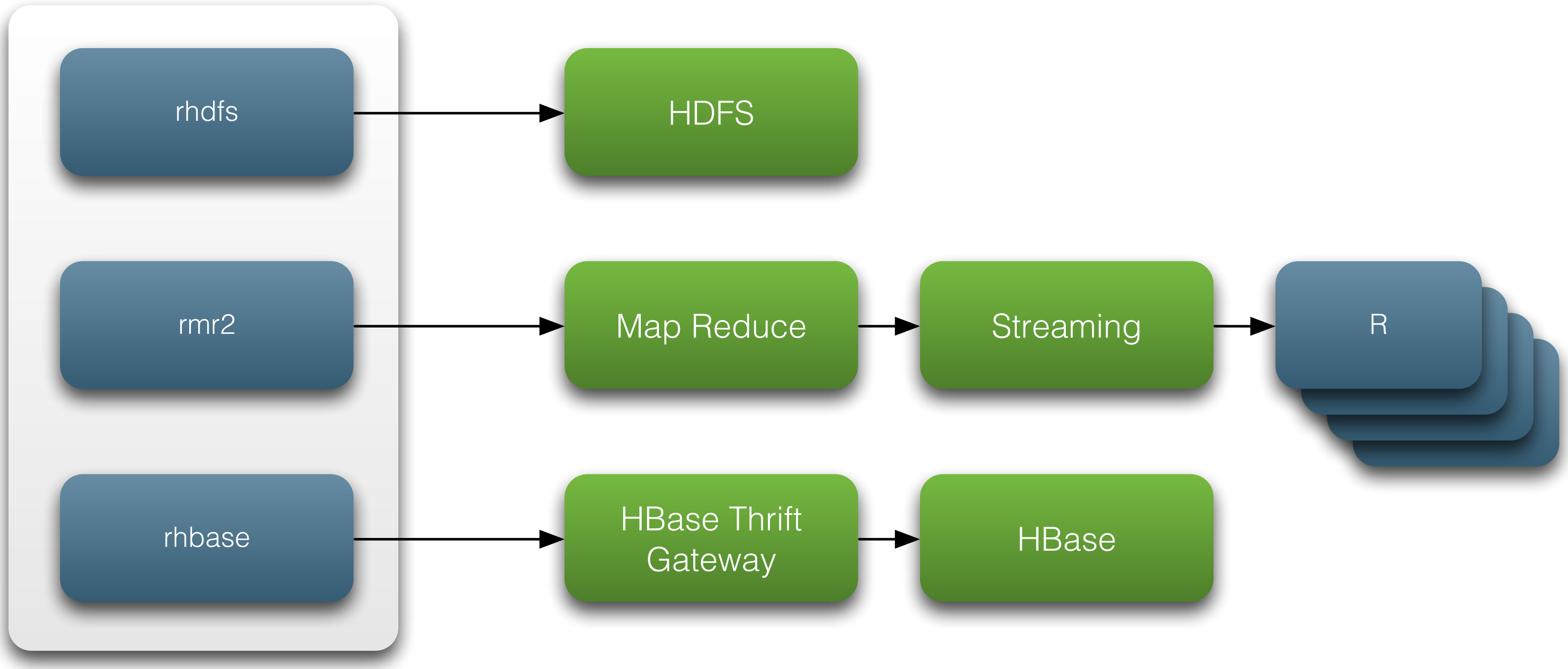
Deployment Considerations



RHadoop

- **Focus**
  - Tight integration with core Hadoop components
- **Benefit**
  - Ability to run R on a massively distributed system
  - Ability to work with full data sets instead of sample sets
- **Additional Information**
  - https://github.com/RevolutionAnalytics/RHadoop/wiki

RHadoop Architecture



rhdfs

- **Access HDFS from R**
- **Read from HDFS to R dataframe**
- **Write from R dataframe to HDFS**
- **1.0.6 adds support for Windows (using HDP)**

rhdfs

- **Hadoop CLI Commands & rhdfs equivalent**
- hadoop fs –ls /
  - hdfs.ls(“/”)
- hadoop fs –mkdir /user/rhdfs/ppt
  - hdfs.mkdir(“/user/rhdfs/ppt”)
- hadoop fs –put 1.txt /user/rhfds/ppt/
  - localData <- system.file(file.path("unitTestData", ”1.txt"), package="rhdfs”)
  - hdfs.put(localData, ”/user/rhdfs/ppt/1.txt”)
- hadoop fs –get /user/rhdfs/ppt/1.txt 1.txt
  - hdfs.get(”/user/rhdfs/ppt/1.txt”,”test”)
- hadoop fs –rm /user/rhdfs/ppt/1.txt
  - hdfs.delete(“/user/rhdfs/ppt/1.txt”)

rhbase

- **Access and change data within HBase**
- **Uses Thrift API**
- **Command Examples**
  - hb.new.table
  - hb.insert
  - hb.scan.ex
  - hb.scan

rmr2

- Enables writing MapReduce jobs using R
- Ability to parallelize algorithms
- Ability to use big data sets without needing to sample data
- mapreduce(input, output, map, reduce, …)
- Reduces takes a key and a collection of values which could be vector, list, data frame or matrix
- 2.2.1 adds support for Windows (using HDP)

Sample code - wordcount

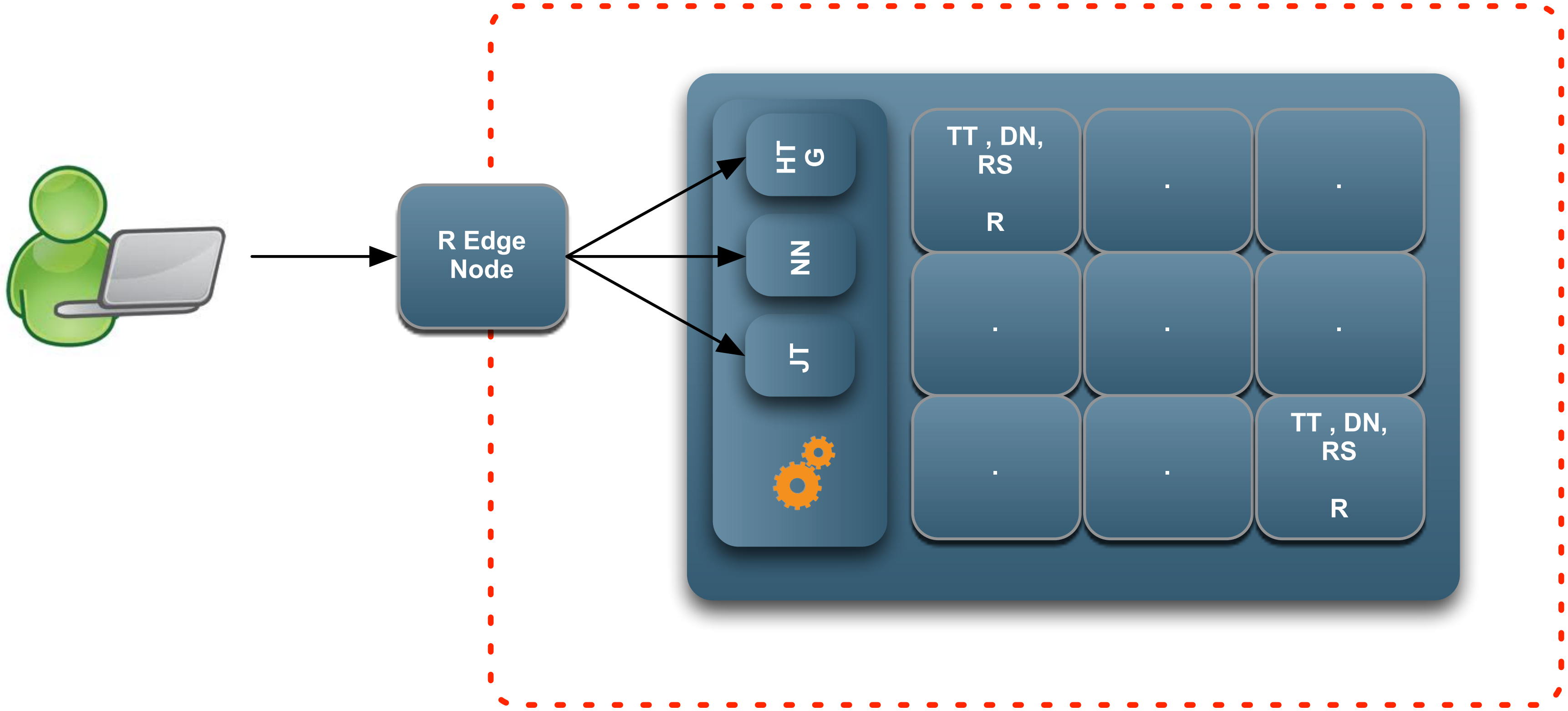
```
 wc.map =
      function(., lines) {
        keyval(
          unlist(
            strsplit(
              x = lines,
              split = pattern)),
          1))}
wc.reduce =
      function(word, counts ) {
        keyval(word, sum(counts))}

mapreduce(
      input = input ,
      output = output,
      input.format = "text",
      map = wc.map,
      reduce = wc.reduce,
      combine = T)}
```

More Sample Code

```
groups = rbinom(32, n = 50, prob = 0.4)
 tapply(groups, groups, length)
```

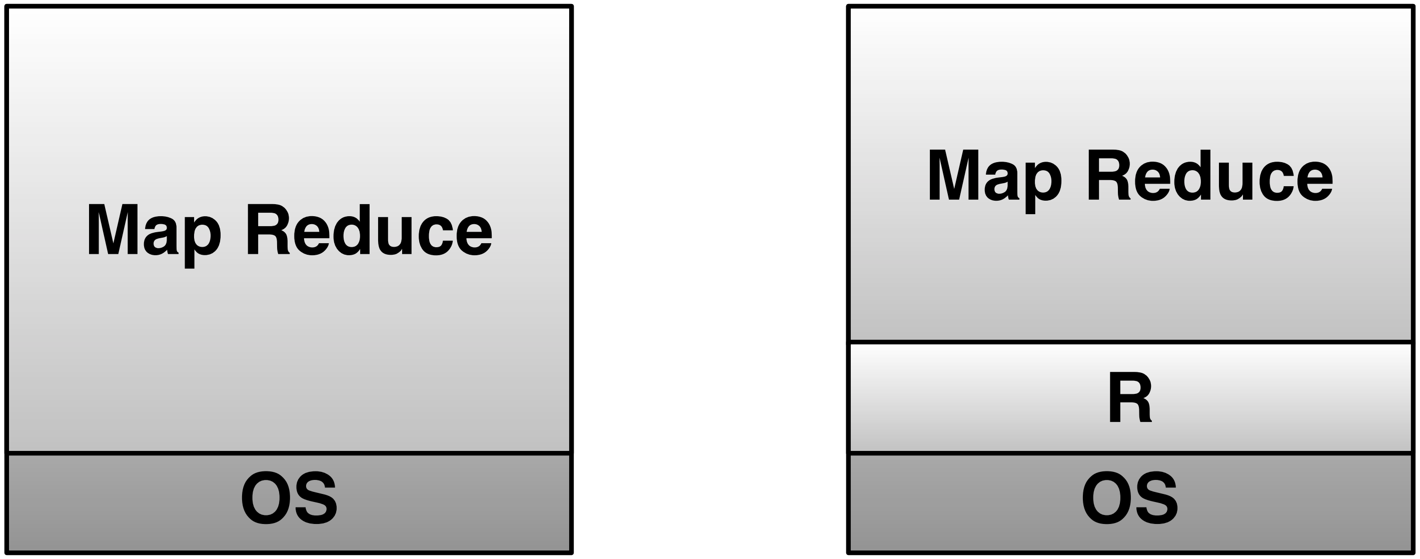
```
groups = to.dfs(groups)
 from.dfs(
   mapreduce(
     input = groups,
     map = function(., v) keyval(v, 1),
     reduce =
       function(k, vv)
         keyval(k, length(vv))))
```

Deployment Considerations



RHadoop

- **Limitations**
  - Requires installation of R on all TaskTracker nodes
  - Does not automatically parallelize algorithms
  - Different slot/memory configuration recommended to leave memory and CPU resources for R



Getting Started

Your Fastest On-ramp to Enterprise Hadoop™!

The Sandbox lets you experience Apache Hadoop from the convenience of your own laptop – no data center, no cloud and no internet connection needed!

The Hortonworks Sandbox is:

- A free download: http://hortonworks.com/products/hortonworks-sandbox/
- A complete, self contained virtual machine with Apache Hadoop pre-configured
- A personal, portable and standalone Hadoop environment
- A set of hands-on, step-by-step tutorials that allow you to learn and explore Hadoop

Installation

- **Install R on all nodes**
- **Install dependent packages**
  - RJSONIO
  - itertools
  - digest
  - Rcpp
  - rJava
  - functional
  - RCurl
  - httr
  - plyr

- **Download & Install RHadoop Packages**
  - rmr2
  - rhdfs
  - rhbase (requires Thrift)

Questions & Answers

TRY

Download HDP at hortonworks.com

LEARN

Applying Data Science using Apache Hadoop Training

FOLLOW

twitter: @hortonworks

Facebook: facebook.com/hortonworks

Further questions & comments:
paul@hortonworks.com
ravi@hortonworks.com