Microsoft

Microsoft R Server

ADVANCED ANALYTICS EDITION

Sanket R.Dhurandhar
Global Black Belt – Advanced Analytics

Introducing Open Source R

Microsoft

R: What is it?

A language platform…

A Procedural Language optimized for Statistics and Data Science
A Data Visualization Framework
Provided as Open Source

A community…

3 million-plus Statistical Analysis and Machine Learning Users
Taught in Most University Statistics Programs
Active User Groups Across the World

An ecosystem

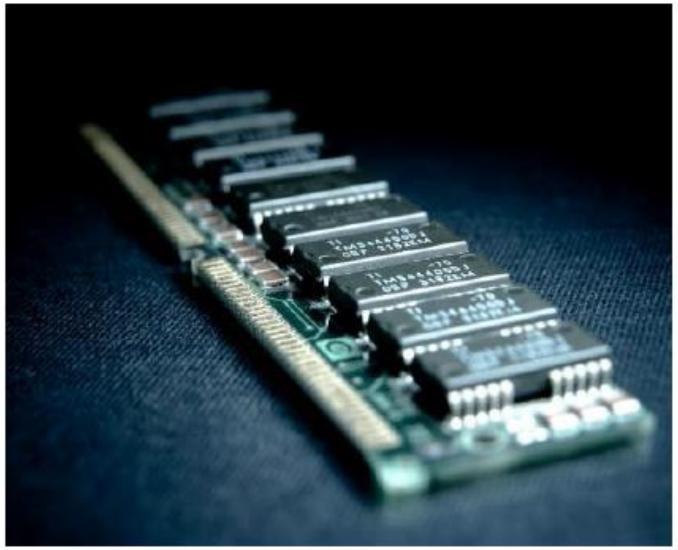
CRAN: 7,500-plus Freely Available Algorithms, Test Data and Evaluations
Many Applicable to Big Data If Scaled

R features

- R holds all data in-memory
- R’s memory management has to managed explicitly
- **Implicitly** parallel computation mechanism
- Requires data movement prior to analysis
- Mainly driven by community support
- Innovation is very fast to flow to developers

Why Microsoft R Server?

Open source R has some limitations for Enterprises



R needs data in memory to start a computation*

R is single threaded*

R requires skilled resource to scale out computations across a cluster and needs re-coding for R map-reduce in Hadoop

Open source R is supported by the community

Microsoft R Server solves these problems!

*Open source R work-arounds are available for some of these problems but do not work in all cases

Introducing Microsoft R Server

Microsoft

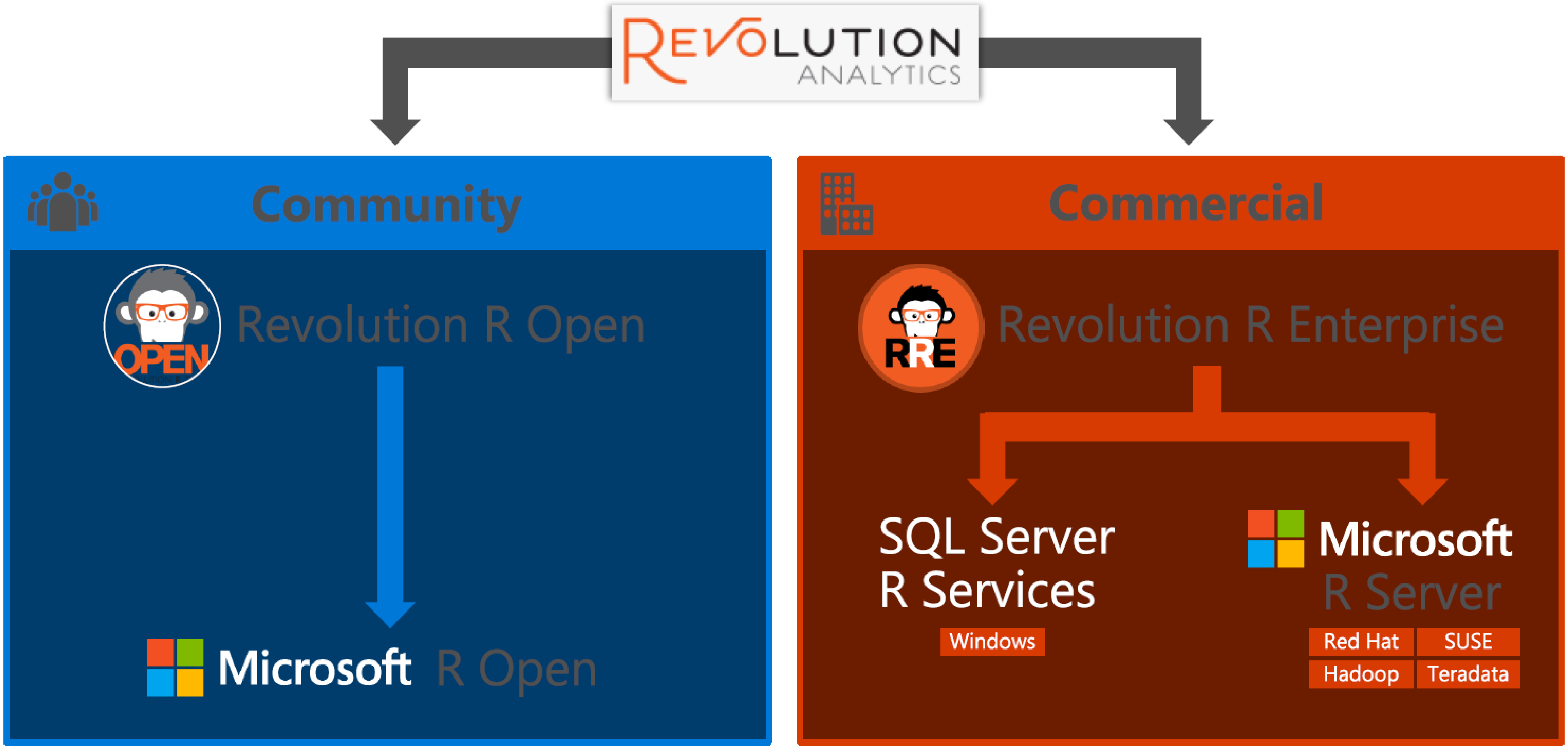
What is Microsoft R Server?

Microsoft R Server is an enterprise-class big data analytics platform for R .

Supporting a variety of big data statistics, predictive modeling and machine learning capabilities, R Server supports the full range of analytics – exploration, analysis, visualization and modeling based on Open Source R.

By leveraging and extending open source R, R Server is fully compatible with R scripts, functions and CRAN packages, while extending R to analyze data at enterprise scale.

Revolution Analytics product integration



Revolution Analytics

Community

Revolution R Open

Microsoft R Open

Commercial

Revolution R Enterprise

SQL Server R Services

Windows

Microsoft R Server

Red Hat

SUSE

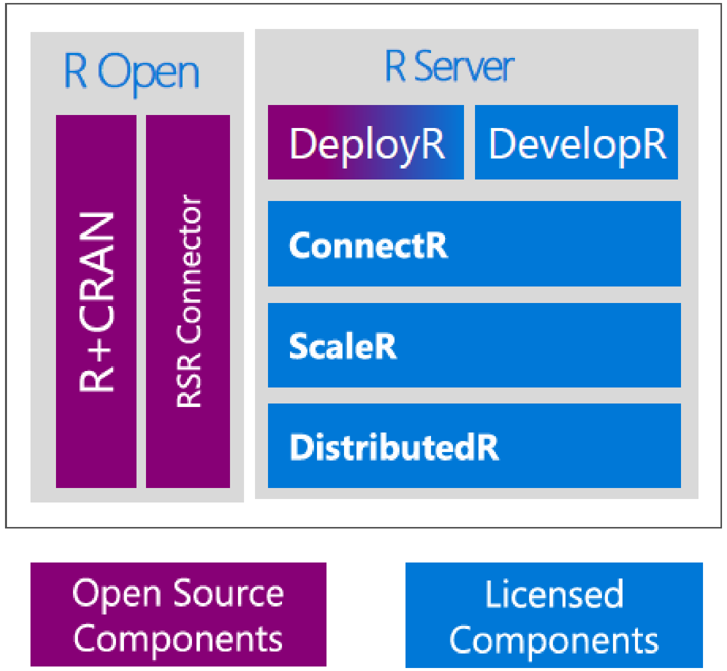
Hadoop

Teradata

Microsoft R Server
(previously Revolution R Enterprise (RRE))

High-performance open source R plus:

- Data source connectivity to big-data objects
- Big-data advanced analytics
- Multi-platform environment support
- In-Hadoop and in-Teradata predictive modeling
- Development and production environment support
  - IDE for data scientist developers
  - Secure, Scalable R Deployment
- Technical support, training and services



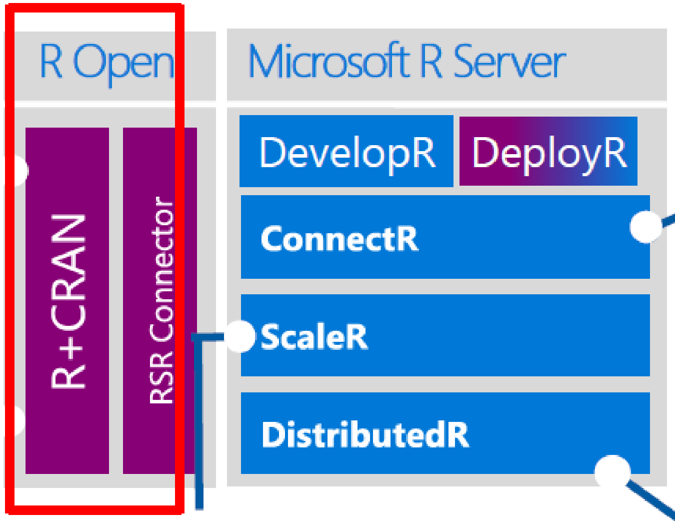
The Microsoft R Server Platform

R+CRAN

- Open source R interpreter
  - R 3.1.2
- Freely-available huge range of R algorithms
- Algorithms callable by RevoR
- Embeddable in R scripts
- 100% Compatible with existing R scripts, functions and packages

RevoR

- Performance enhanced R interpreter
- Based on open source R
- Adds high-performance math library to speed up linear algebra functions



ScaleR

- Ready-to-Use high-performance big data big analytics
- Fully-parallelized analytics
- Data prep & data distillation
- Descriptive statistics & statistical tests
- Range of predictive functions
- User tools for distributing customized R algorithms across nodes
- Wide data sets supported – thousands of variables

ConnectR

- High-speed & direct connectors

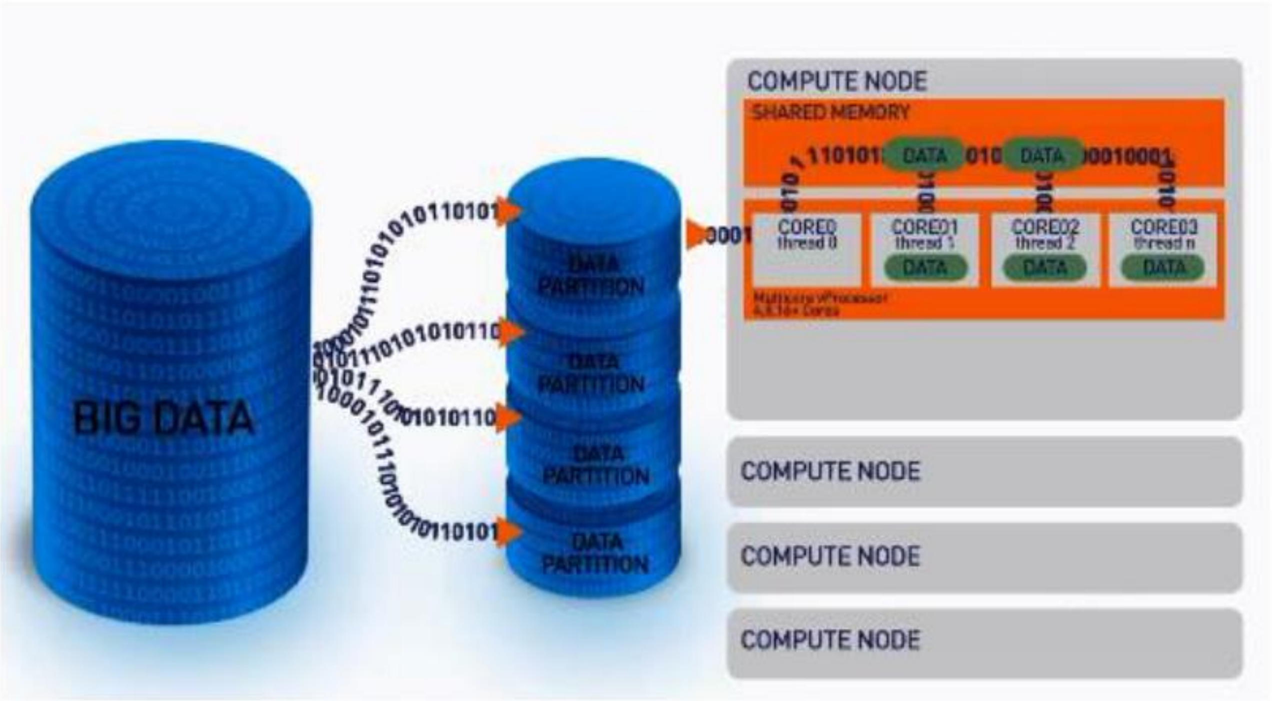
Available for:

- High-performance XDF
- SAS, SPSS, delimited & fixed format text data files
- Hadoop HDFS (text & XDF)
- Teradata Database & Aster
- EDWs and ADWs
- ODBC

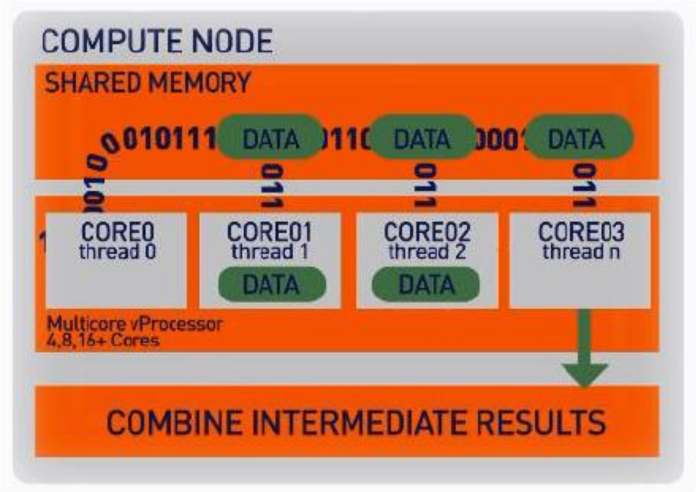
DistributedR

- Distributed computing framework
- Delivers cross-platform portability

ScaleR – Parallel + "Big Data"



Our ScaleR algorithms work inside multiple cores / nodes in parallel at high speed



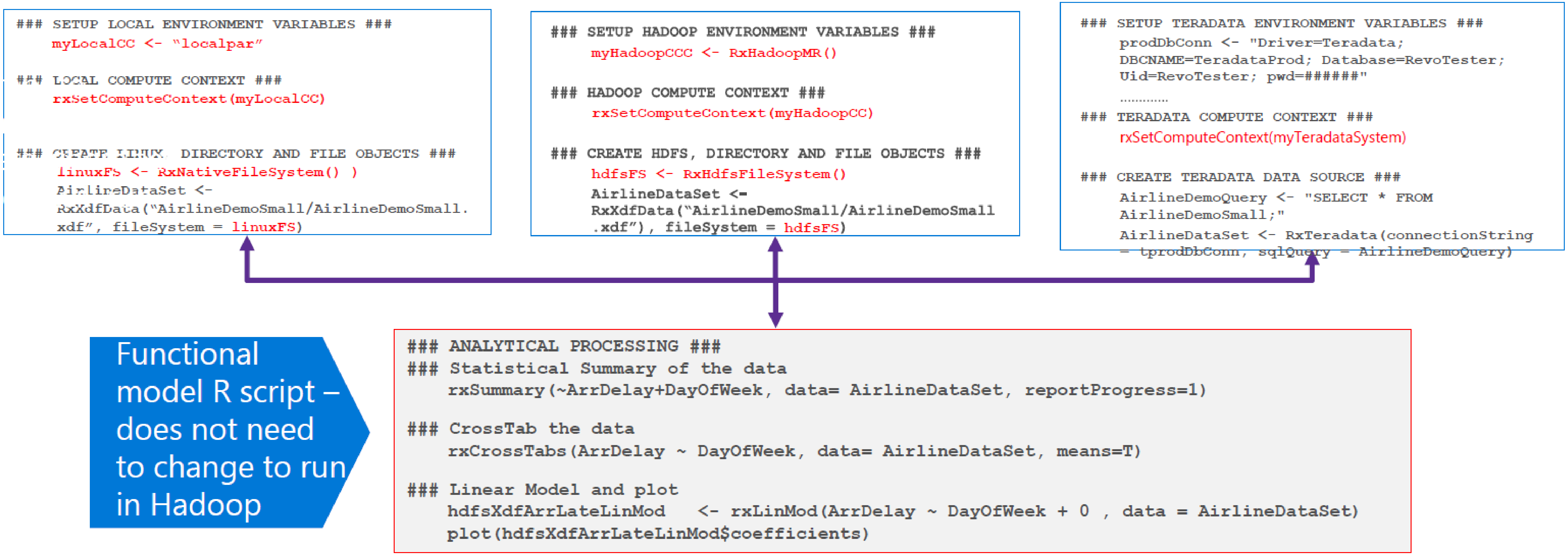
Interim results are collected and combined analytically to produce the output on the entire data set

Stream data in to RAM in blocks. "Big Data" can be any data size. We handle Megabytes to Gigabytes to Terabytes…

XDF file format is optimised to work with the ScaleR library and significantly speeds up iterative algorithm processing.

DistributedR - Code Portability

ScaleR models can be deployed from a server or edge node to run in Hadoop or in Teradata/SQL without any functional R model re-coding



Local Parallel – **Linux or Windows**

```
### SETUP LOCAL ENVIRONMENT VARIABLES ###
    myLocalCC <- "localpar"

### LOCAL COMPUTE CONTEXT ###
    rxSetComputeContext(myLocalCC)


### CREATE LINUX  DIRECTORY AND FILE OBJECTS ###
     linuxFS <- RxNativeFileSystem() )
     AirlineDataSet <-
     RxXdfData("AirlineDemoSmall/AirlineDemoSmall.
     xdf", fileSystem = linuxFS)
```

In – **Hadoop**

```
### SETUP HADOOP ENVIRONMENT VARIABLES ###
    myHadoopCCC <- RxHadoopMR()

### HADOOP COMPUTE CONTEXT ###
    rxSetComputeContext(myHadoopCC)

### CREATE HDFS, DIRECTORY AND FILE OBJECTS ###
    hdfsFS <- RxHdfsFileSystem()
    AirlineDataSet <-
    RxXdfData("AirlineDemoSmall/AirlineDemoSmall
    .xdf"), fileSystem = hdfsFS)
```

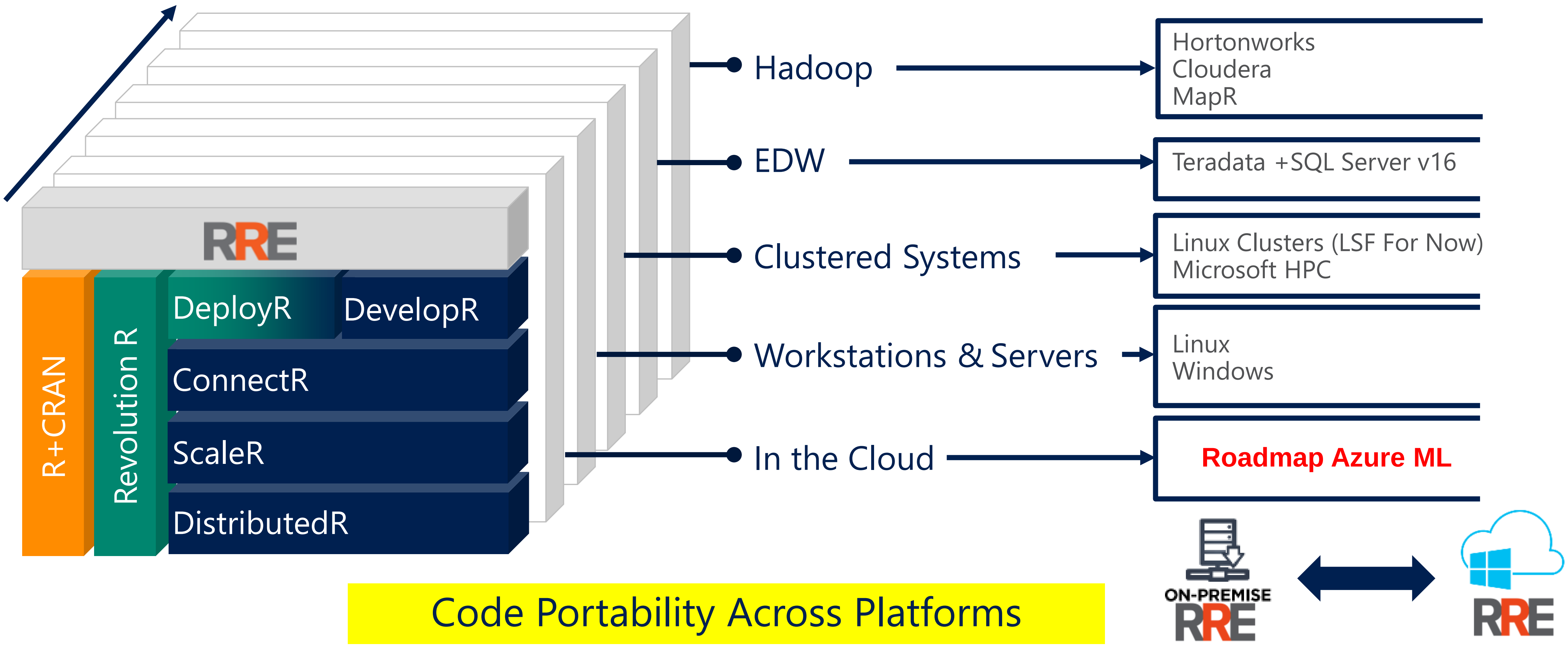
In – **Teradata**

```
### SETUP TERADATA ENVIRONMENT VARIABLES ###
    prodDbConn <- "Driver=Teradata;
    DBCNAME=TeradataProd; Database=RevoTester;
    Uid=RevoTester; pwd=######"
    ………….
### TERADATA COMPUTE CONTEXT ###
    rxSetComputeContext(myTeradataSystem)

### CREATE TERADATA DATA SOURCE ###
    AirlineDemoQuery <- "SELECT * FROM
    AirlineDemoSmall;"
    AirlineDataSet <- RxTeradata(connectionString
    = tprodDbConn, sqlQuery = AirlineDemoQuery)
```

Functional model R script – does not need to change to run in Hadoop

```
### ANALYTICAL PROCESSING ###
### Statistical Summary of the data
    rxSummary(~ArrDelay+DayOfWeek, data= AirlineDataSet, reportProgress=1)

### CrossTab the data
    rxCrossTabs(ArrDelay ~ DayOfWeek, data= AirlineDataSet, means=T)

### Linear Model and plot
    hdfsXdfArrLateLinMod    <- rxLinMod(ArrDelay ~ DayOfWeek + 0 , data = AirlineDataSet)
    plot(hdfsXdfArrLateLinMod$coefficients)
```

2# Model development and model compute choice: “Write Once. Deploy Anywhere.”



RRE

R+CRAN

Revolution R

DeployR

DevelopR

ConnectR

ScaleR

DistributedR

Hadoop → Hortonworks
Cloudera
MapR

EDW → Teradata +SQL Server v16

Clustered Systems → Linux Clusters (LSF For Now)
Microsoft HPC

Workstations & Servers → Linux
Windows

In the Cloud → **Roadmap Azure ML**

ON-PREMISE RRE ↔ RRE

Code Portability Across Platforms

Revolution R Enterprise Parallelized Algorithms

Data Step

- Data import – Delimited, Fixed, SAS, SPSS, OBDC
- Variable creation & transformation
- Recode variables
- Factor variables
- Missing value handling
- Sort, Merge, Split
- Aggregate by category (means, sums)

Descriptive Statistics

- Min / Max, Mean, Median (approx.)
- Quantiles (approx.)
- Standard Deviation
- Variance
- Correlation
- Covariance
- Sum of Squares (cross product matrix for set variables)
- Pairwise Cross tabs
- Risk Ratio & Odds Ratio
- Cross-Tabulation of Data (standard tables & long form)
- Marginal Summaries of Cross Tabulations

Statistical Tests

- Chi Square Test
- Kendall Rank Correlation
- Fisher's Exact Test
- Student's t-Test

Sampling

- Subsample (observations & variables)
- Random Sampling

Predictive Models

- Sum of Squares (cross product matrix for set variables)
- Multiple Linear Regression
- Generalized Linear Models (GLM) exponential family distributions: binomial, Gaussian, inverse Gaussian, Poisson, Tweedie. Standard link functions: cauchit, identity, log, logit, probit. User defined distributions & link functions.
- Covariance & Correlation Matrices
- Logistic Regression
- Classification & Regression Trees
- **NEW** Naïve Bayes Classification
- Predictions/scoring for models
- Residuals for all models

Variable Selection

- Stepwise Regression Linear, Logistic and GLM
- Stepwise Coefficient Tracking Wide datasets(40k vars)

Simulation

- Monte Carlo
- Parallel Random Number Generation

Cluster Analysis

- K-Means

Classification

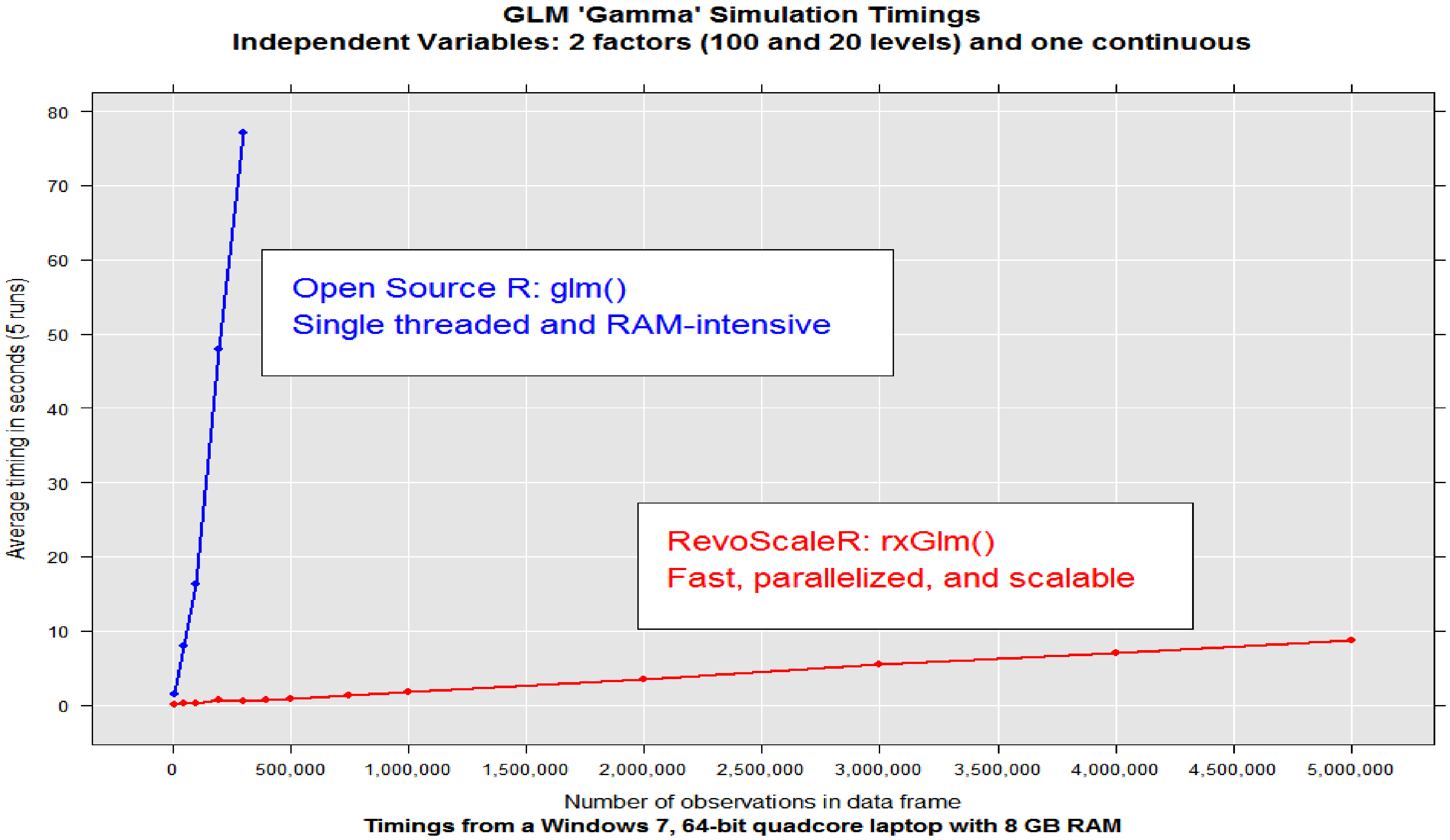
- Decision Trees
- Decision Forests
- Stochastic Gradient Boosted Decision Tree

Combination

- Using Revolution rxDataStep and rxExec functions to combine open source R with Revolution R
- **NEW** PEMA API

works with Open Source R v3

1# Improve Dramatically on Performance and Capacity



Example: performance assessment of SAS, R, Hadoop, Revolution (Strata/Hadoop World, October 2012) at AllState Insurance

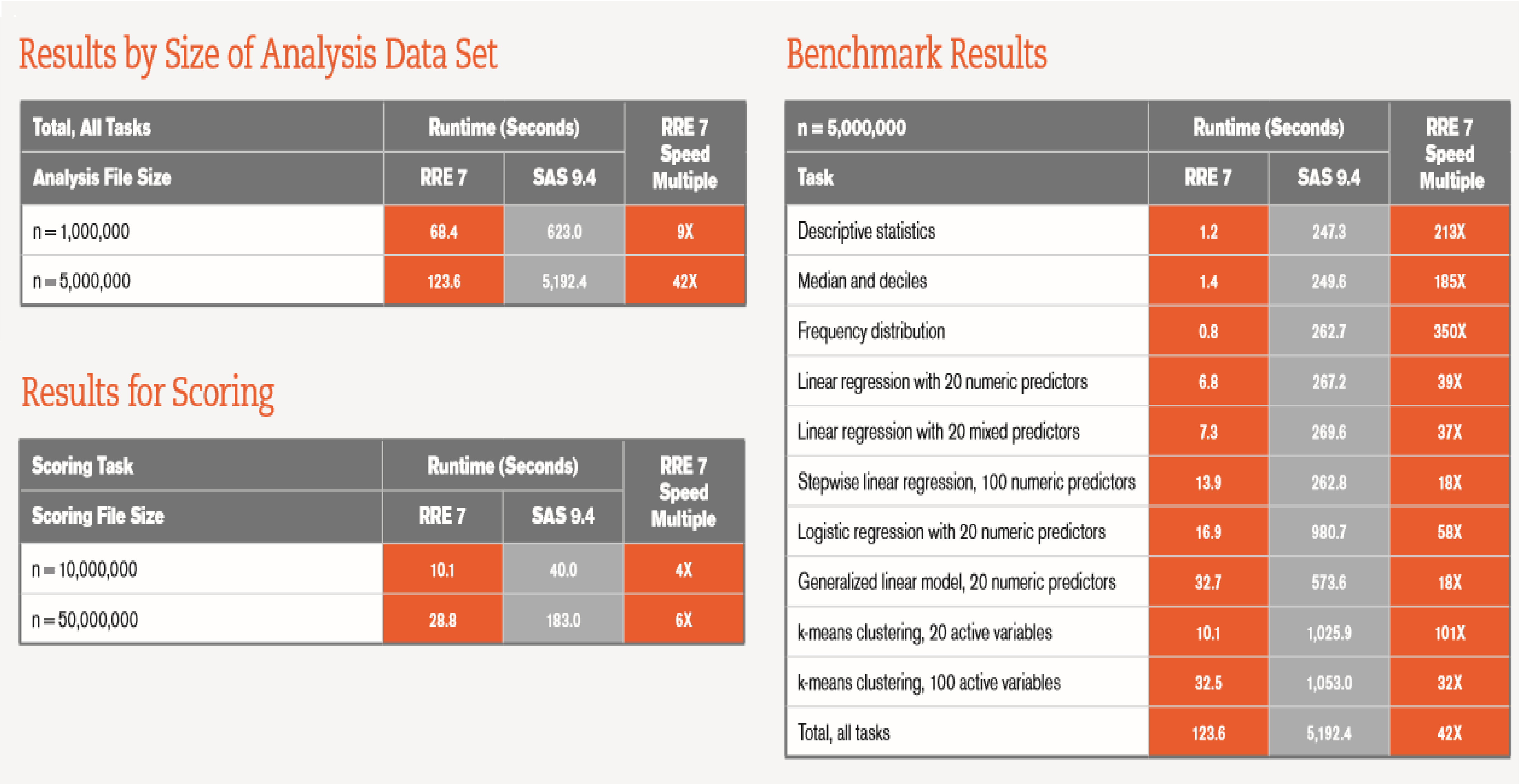
- Steve Yun, Principal Predictive Modeller at Allstate Research and Planning Centre benchmarked SAS, R and Hadoop. “Data is our competitive advantage”.
- *Generalised Linear Model* for 150 million observations of insurance data and 70 degrees of freedom.

Software	Platform	Comments	Time to fit
SAS (current tool)	16-core Sun Server	Proc GENMOD	5 hours
rmr / map-reduce	10-node (8 cores / node) Hadoop cluster	Lot of coding, prep and error investigation. Possible to improve time?	> 9 hours processing
Open source R	250-GB Server	Full data set and sampling. Sampling quicker but not acceptable to business.	Impossible (> 3 days)
Revolution ScaleR	5-node (4 cores / node) LSF cluster	90 minutes to load full data set	5.7 minutes

Allstate’s conclusion:

- SAS works, but is slow.
- The data is too big for open-source R, even on a very large server.
- Hadoop is not a right fit
- Revolution ScaleR gets the same results as SAS, but much faster and on cheaper kit

SAS Comparison



Results by Size of Analysis Data Set

Total, All Tasks	Runtime (Seconds)		RRE 7 Speed Multiple
Analysis File Size	RRE 7	SAS 9.4	
n = 1,000,000	68.4	623.0	9X
n = 5,000,000	123.6	5,192.4	42X

Results for Scoring

Scoring Task	Runtime (Seconds)		RRE 7 Speed Multiple
Scoring File Size	RRE 7	SAS 9.4	
n = 10,000,000	10.1	40.0	4X
n = 50,000,000	28.8	183.0	6X

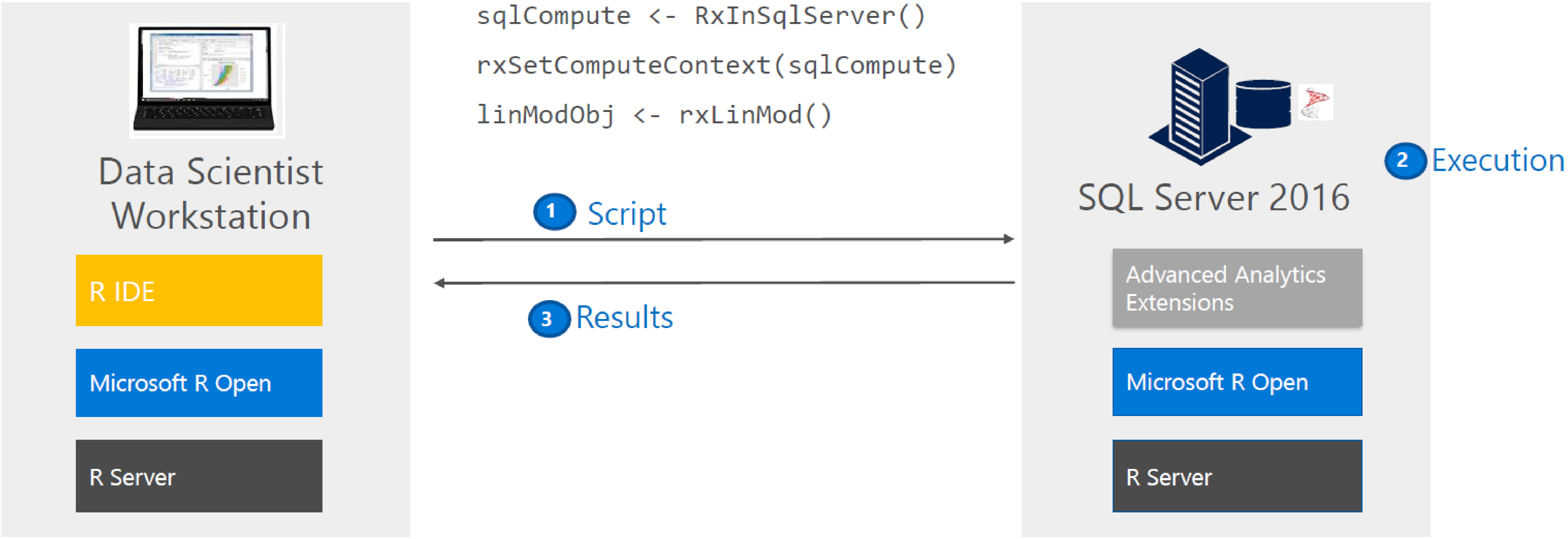
Benchmark Results

n = 5,000,000	Runtime (Seconds)		RRE 7 Speed Multiple
Task	RRE 7	SAS 9.4	
Descriptive statistics	1.2	247.3	213X
Median and deciles	1.4	249.6	185X
Frequency distribution	0.8	262.7	350X
Linear regression with 20 numeric predictors	6.8	267.2	39X
Linear regression with 20 mixed predictors	7.3	269.6	37X
Stepwise linear regression, 100 numeric predictors	13.9	262.8	18X
Logistic regression with 20 numeric predictors	16.9	980.7	58X
Generalized linear model, 20 numeric predictors	32.7	573.6	18X
k-means clustering, 20 active variables	10.1	1,025.9	101X
k-means clustering, 100 active variables	32.5	1,053.0	32X
Total, all tasks	123.6	5,192.4	42X

SQL Server R Services

Model Development (R Users)

Working from R IDE on a local workstation, execute an R script that runs in-database on remote SQL server, and get the results back.



```
sqlCompute <- RxInSqlServer()

rxSetComputeContext(sqlCompute)

linModObj <- rxLinMod()
```

Data Scientist Workstation

R IDE

Microsoft R Open

R Server

1 Script

3 Results

SQL Server 2016

2 Execution

Advanced Analytics Extensions

Microsoft R Open

R Server

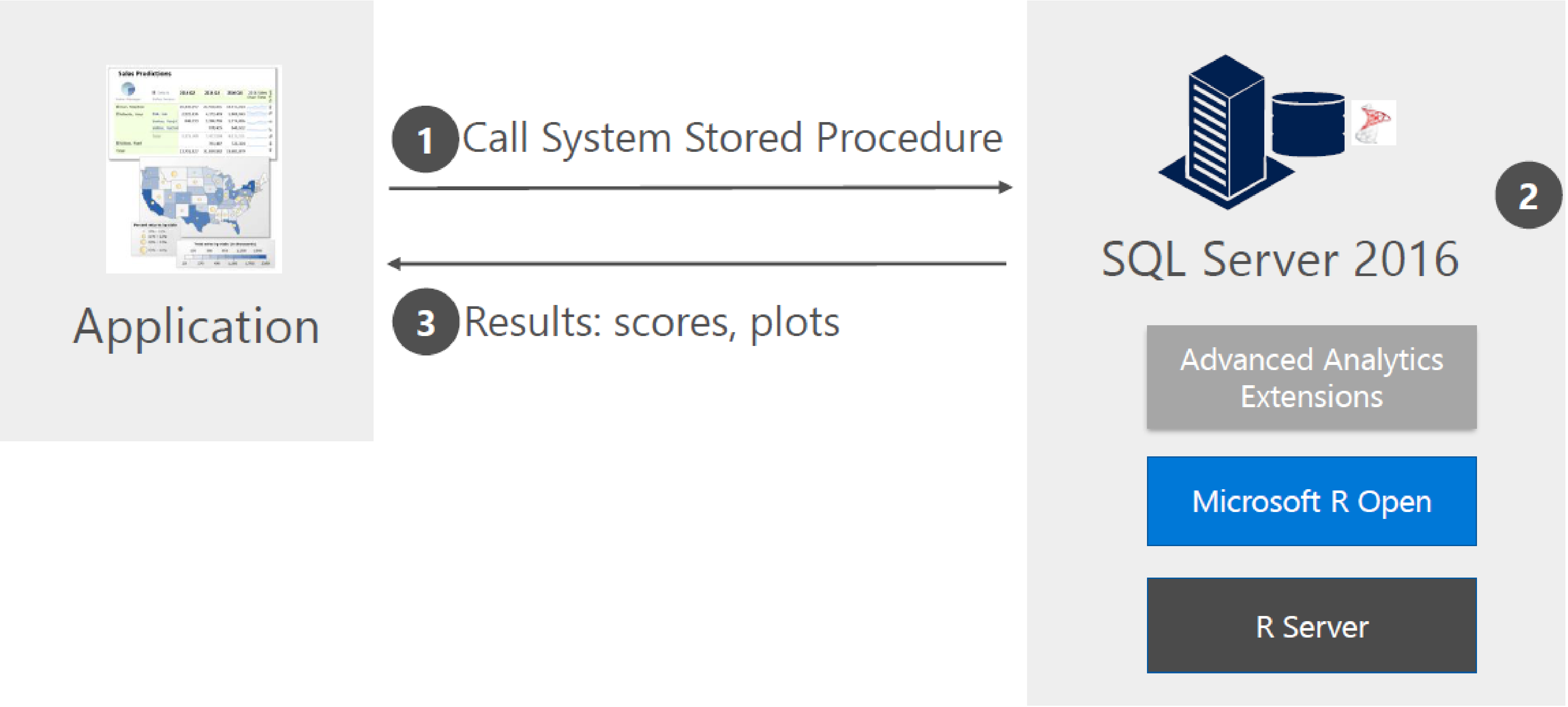
SQL Server R Services

Model Operationalization

(R Code->T-SQL Stored Proc.)

Call a T-SQL System Stored Procedure to generate features and train (or retrain) the model

Call a T-SQL System Stored Procedure from my application and have it trigger R script execution in-database to predict on new dataset. Results are then returned to my application (predictions, plots).



Application

1 Call System Stored Procedure

3 Results: scores, plots

SQL Server 2016

Advanced Analytics Extensions

Microsoft R Open

R Server

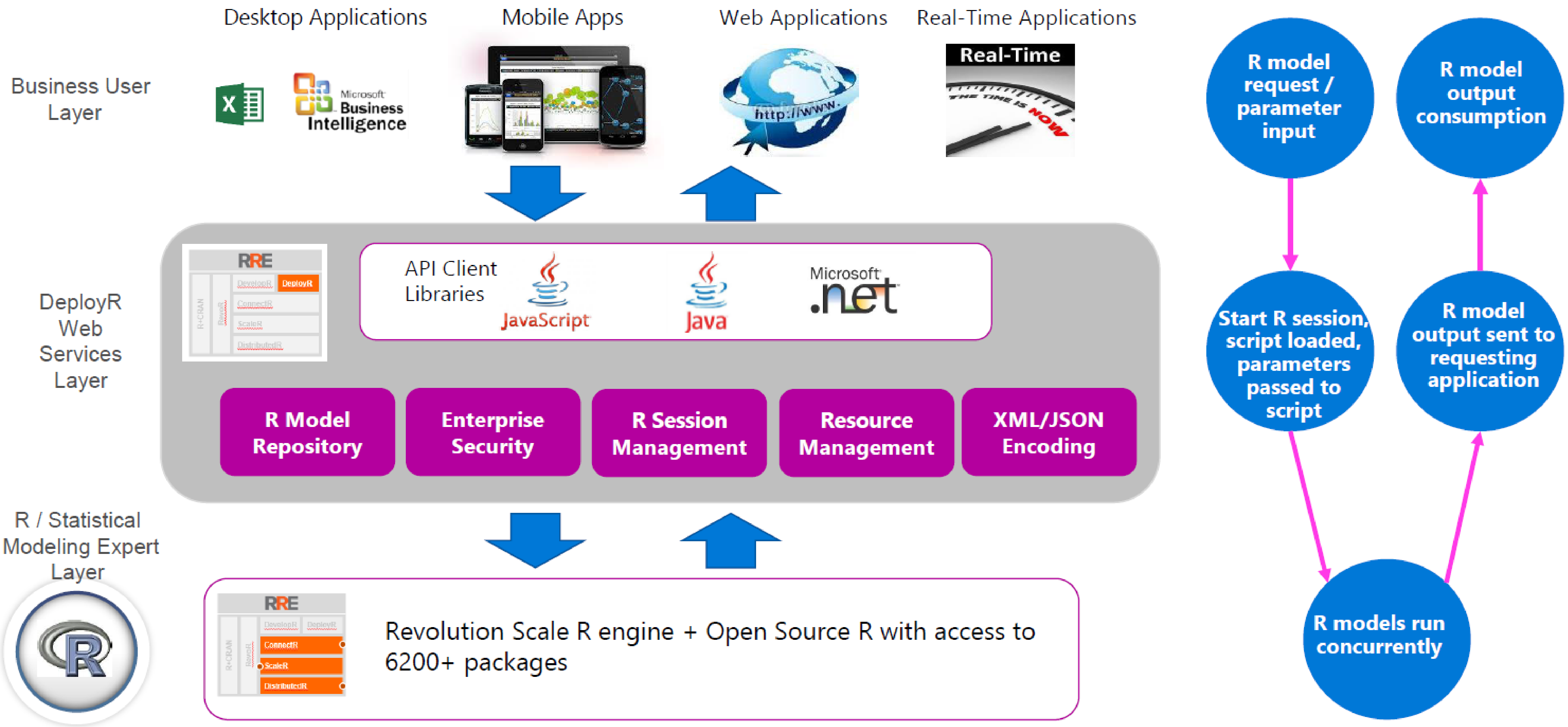
2 The stored procedure contains R code and executes in-database

```
exec sp_execute_external_script

    @language = 'R'
,   @script =
-- R code --
```

Introducing DeployR

Microsoft

DeployR: Framework for R as a service for BI / web apps



Business User Layer

Desktop Applications

Mobile Apps

Web Applications

Real-Time Applications

DeployR Web Services Layer

API Client Libraries

R Model Repository

Enterprise Security

R Session Management

Resource Management

XML/JSON Encoding

R / Statistical Modeling Expert Layer

Revolution Scale R engine + Open Source R with access to 6200+ packages

R model request / parameter input

Start R session, script loaded, parameters passed to script

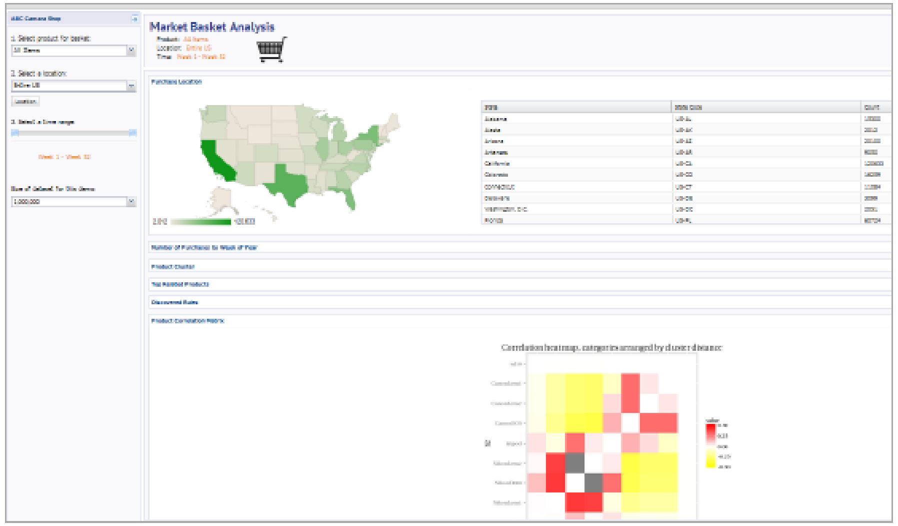
R models run concurrently

R model output sent to requesting application

R model output consumption

DeployR: example R as a service for BI / web apps

Example: Market Basket Analysis in HTML tool


Market Basket Analysis
Purchase Location
Correlation heatmap, categories arranged by cluster distance

Example: fraud analytics deployed to BI tool

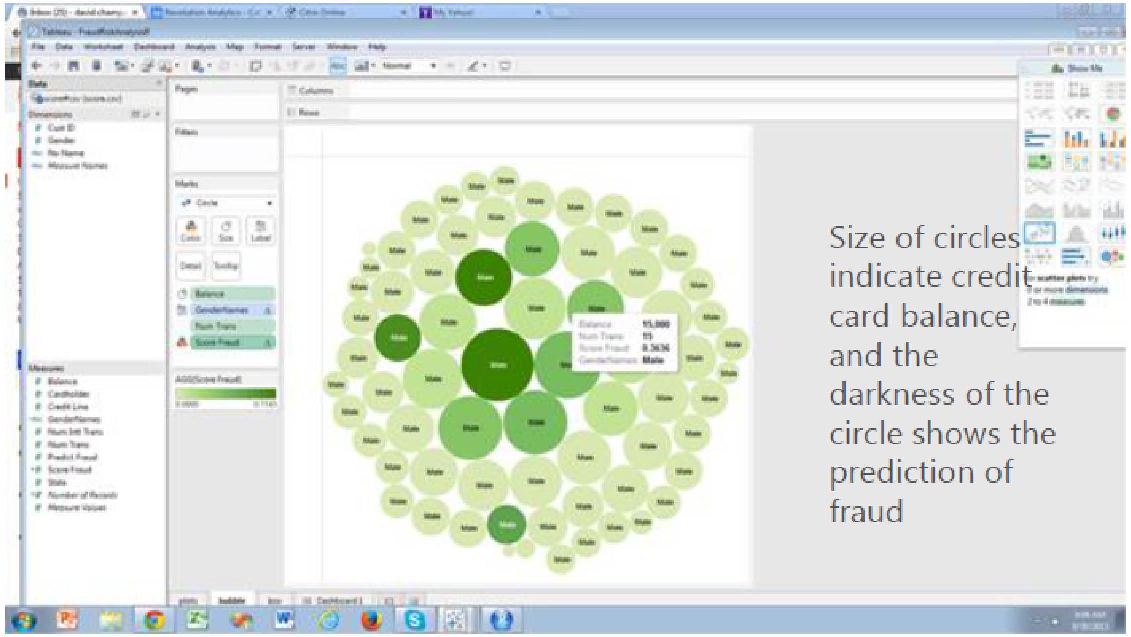
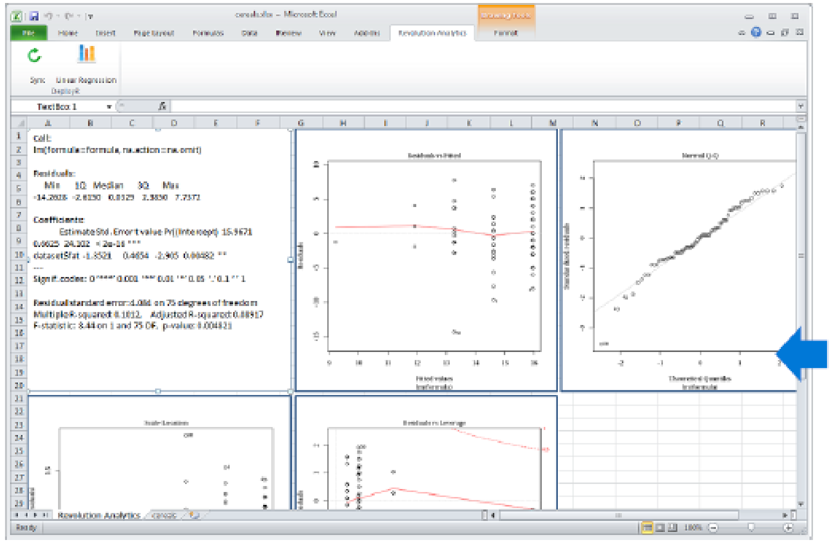


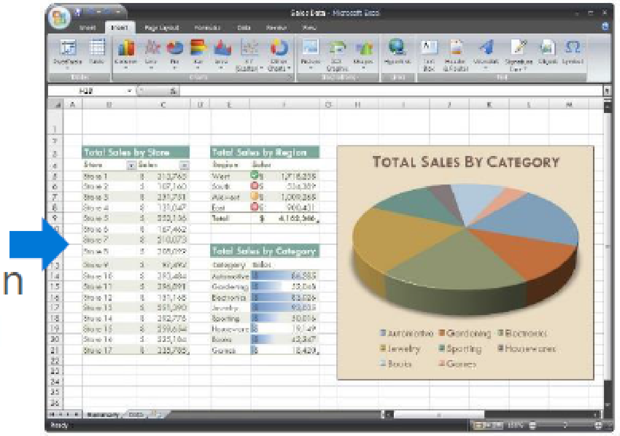
Tableau - FraudRiskAnalysis
File Data Worksheet Dashboard Analysis Map Format Server Window Help
Show Me

Size of circles indicate credit card balance, and the darkness of the circle shows the prediction of fraud

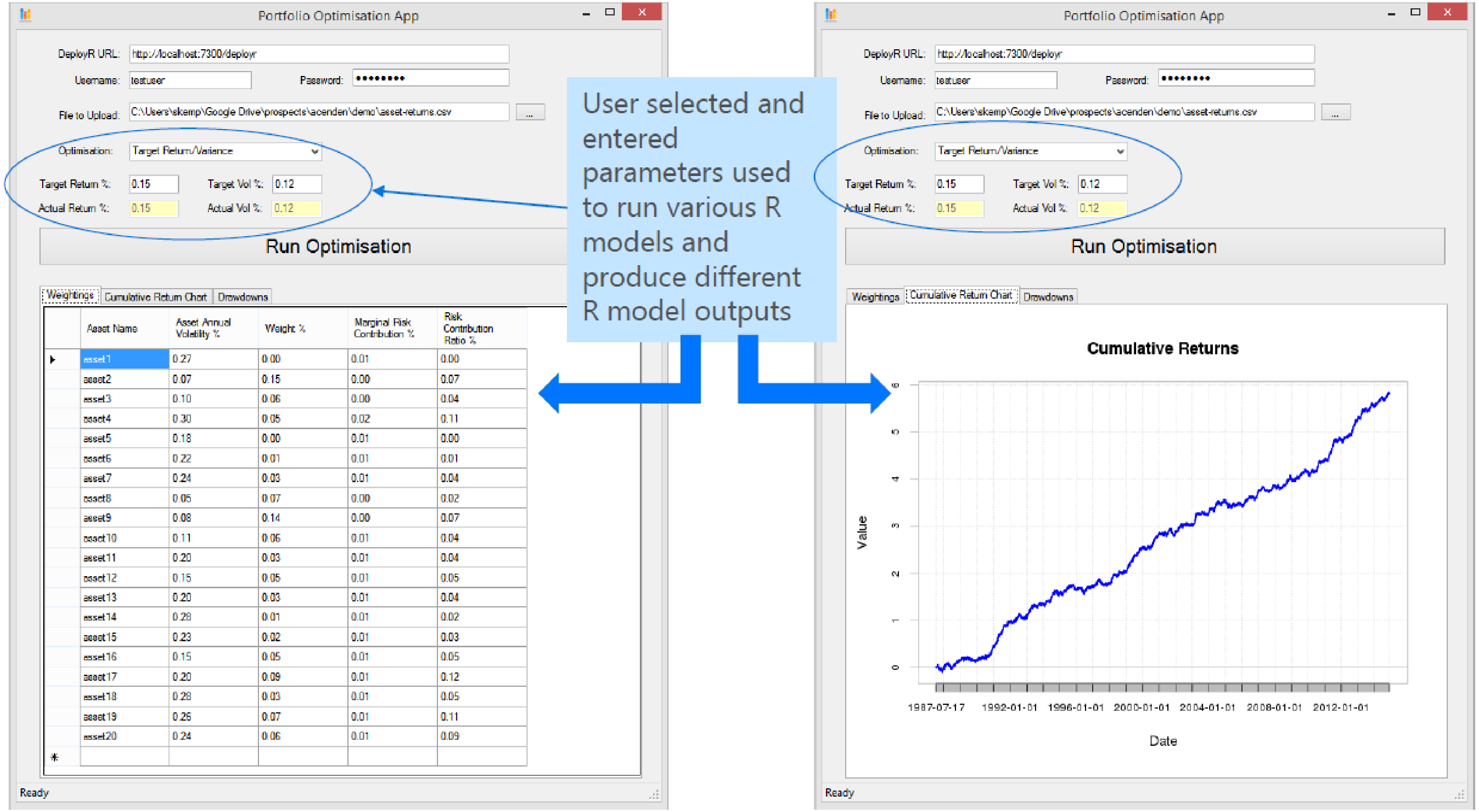


```
Call:
lm(formula=formula, na.action=na.omit)

Residuals:
    Min      1Q  Median      3Q     Max
-14.2608  -2.6150  0.0329  2.3850  7.7372

Coefficients:
         Estimate Std. Error t value Pr(|Intercept) 15.9671
0.6625  24.102  < 2e-16 ***
dataset$fat -1.3521   0.4654  -2.905  0.00482 **
---
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 4.086 on 75 degrees of freedom
Multiple R-squared: 0.1012,   Adjusted R-squared: 0.08917
F-statistic: 8.44 on 1 and 75 DF,  p-value: 0.004821
```

Example: integration with Excel



DeployR Example – C# web app business user front end for portfolio optimisation



Portfolio Optimisation App

DeployR URL: http://localhost:7300/deployr

Username: testuser Password: ••••••••

File to Upload: C:\Users\skemp\Google Drive\prospects\acenden\demo\asset-returns.csv

Optimisation: Target Return/Variance

Target Return %: 0.15 Target Vol %: 0.12

Actual Return %: 0.15 Actual Vol %: 0.12

	Asset Name	Asset Annual Volatility %	Weight %	Marginal Risk Contribution %	Risk Contribution Ratio %
▸	asset1	0.27	0.00	0.01	0.00
	asset2	0.07	0.15	0.00	0.07
	asset3	0.10	0.06	0.00	0.04
	asset4	0.30	0.05	0.02	0.11
	asset5	0.18	0.00	0.01	0.00
	asset6	0.22	0.01	0.01	0.01
	asset7	0.24	0.03	0.01	0.04
	asset8	0.05	0.07	0.00	0.02
	asset9	0.08	0.14	0.00	0.07
	asset10	0.11	0.06	0.01	0.04
	asset11	0.20	0.03	0.01	0.04
	asset12	0.15	0.05	0.01	0.05
	asset13	0.20	0.03	0.01	0.04
	asset14	0.28	0.01	0.01	0.02
	asset15	0.23	0.02	0.01	0.03
	asset16	0.15	0.05	0.01	0.05
	asset17	0.20	0.09	0.01	0.12
	asset18	0.28	0.03	0.01	0.05
	asset19	0.26	0.07	0.01	0.11
	asset20	0.24	0.06	0.01	0.09
*					

Ready

User selected and entered parameters used to run various R models and produce different R model outputs

Portfolio Optimisation App

DeployR URL: http://localhost:7300/deployr

Username: testuser Password: ••••••••

File to Upload: C:\Users\skemp\Google Drive\prospects\acenden\demo\asset-returns.csv

Optimisation: Target Return/Variance

Target Return %: 0.15 Target Vol %: 0.12

Actual Return %: 0.15 Actual Vol %: 0.12

Ready

Microsoft

The information herein is for informational purposes only and represents the current view of Microsoft Corporation as of the date of this presentation. Because Microsoft must respond to changing market conditions, it should not be interpreted to be a commitment on the part of Microsoft, and Microsoft cannot guarantee the accuracy of any information provided after the date of this presentation. MICROSOFT MAKES NO WARRANTIES, EXPRESS, IMPLIED OR STATUTORY, AS TO THE INFORMATION IN THIS PRESENTATION.