Biocep-R

Open Science in the cloud, towards a universal platform for mathematical and statistical computing

Karim Chine

karim.chine@m4x.org

Croire possible le souhaitable est aussi dangereux que de croire souhaitable le possible. Utopies sentimentales et automatismes de la technique.

Nicolás Gómez Dávila

Il n'y a que le solitaire qui soit capable de penser plus que des vérités tactiques.

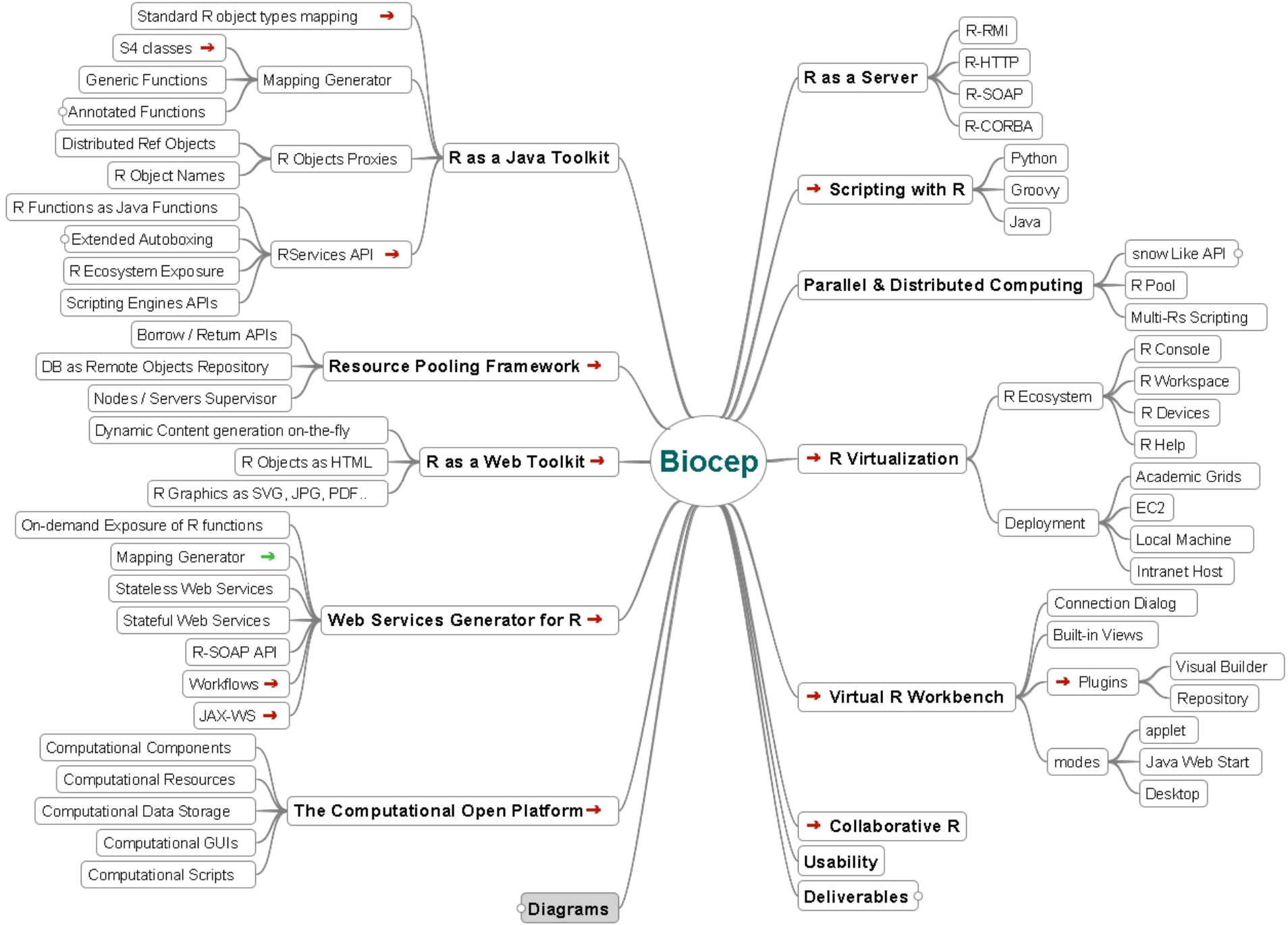
Nicolás Gómez Dávila

Extract from the GridSolve Description Document

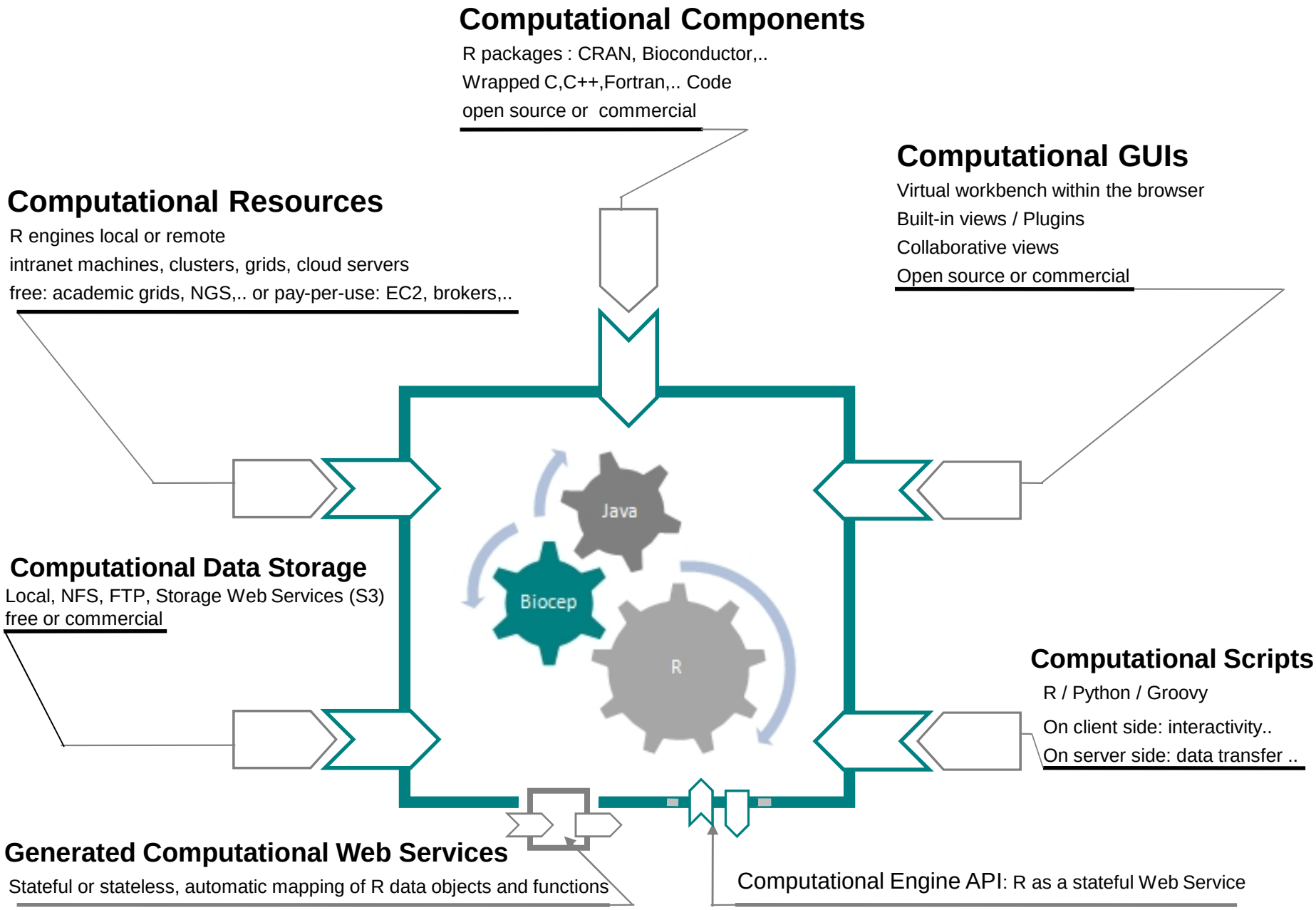
The emergence of Grid computing as the prototype of a next generation cyberinfrastructure for science has excited high expectations for its potential as an accelerator of discovery, but it has also raised questions about whether and how the broad population of research professionals, who must be the foundation of such productivity, can be motivated to adopt this new and more complex way of working.

The rise of the new era of scientific modeling and simulation has, after all, been precipitous, and many science and engineering professionals have only recently become comfortable with the relatively simple world of the uniprocessor workstations and desktop scientific computing tools. In that world, software packages such as Matlab and Mathematica represent general-purpose scientific computing environments (SCEs) that enable users — totaling more than a million worldwide — to solve a wide variety of problems through flexible user interfaces that can model in a natural way the mathematical aspects of many different problem domains.

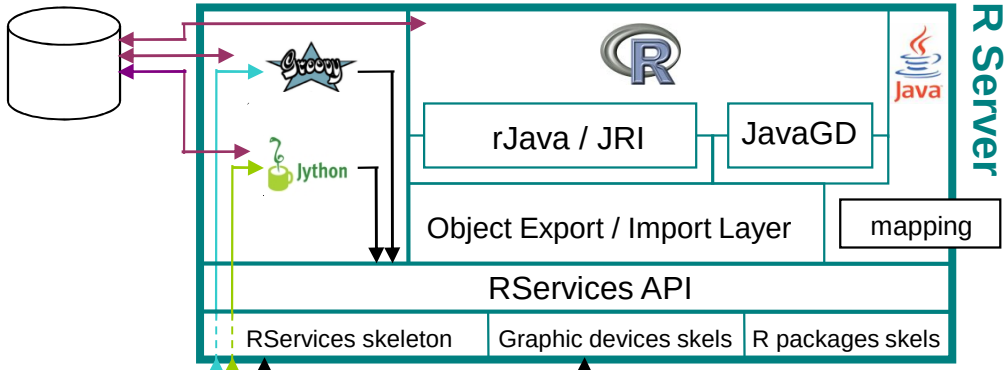
Moreover, the ongoing, exponential increase in the computing resources supplied by the typical workstation makes these SCEs more and more powerful, and thereby tends to reduce the need for the kind of resource sharing that represents a major strength of Grid computing [1]. Certainly there are various forces now urging collaboration across disciplines and distances, and the burgeoning Grid community, which aims to facilitate such collaboration, has made significant progress in mitigating the well-known complexities of building, operating, and using distributed computing environments. But it is unrealistic to expect the transition of research professionals to the Grid to be anything but halting and slow if it means abandoning the SCEs that they rightfully view as a major source of their productivity. We therefore believe that Grid computing’s prospects for success will tend to rise and fall according to its ability to interface smoothly with the general purpose SCEs that are likely to continue to dominate the toolbox of its targeted user base.


Biocep
R as a Java Toolkit
Standard R object types mapping
Mapping Generator
S4 classes
Generic Functions
Annotated Functions
R Objects Proxies
Distributed Ref Objects
R Object Names
RServices API
R Functions as Java Functions
Extended Autoboxing
R Ecosystem Exposure
Scripting Engines APIs
Resource Pooling Framework
Borrow / Return APIs
DB as Remote Objects Repository
Nodes / Servers Supervisor
R as a Web Toolkit
Dynamic Content generation on-the-fly
R Objects as HTML
R Graphics as SVG, JPG, PDF..
Web Services Generator for R
On-demand Exposure of R functions
Mapping Generator
Stateless Web Services
Stateful Web Services
R-SOAP API
Workflows
JAX-WS
The Computational Open Platform
Computational Components
Computational Resources
Computational Data Storage
Computational GUIs
Computational Scripts
Diagrams
R as a Server
R-RMI
R-HTTP
R-SOAP
R-CORBA
Scripting with R
Python
Groovy
Java
Parallel & Distributed Computing
snow Like API
R Pool
Multi-Rs Scripting
R Virtualization
R Ecosystem
R Console
R Workspace
R Devices
R Help
Deployment
Academic Grids
EC2
Local Machine
Intranet Host
Virtual R Workbench
Connection Dialog
Built-in Views
Plugins
Visual Builder
Repository
modes
applet
Java Web Start
Desktop
Collaborative R
Usability
Deliverables

Biocep Computational Open Platform Ecosystem



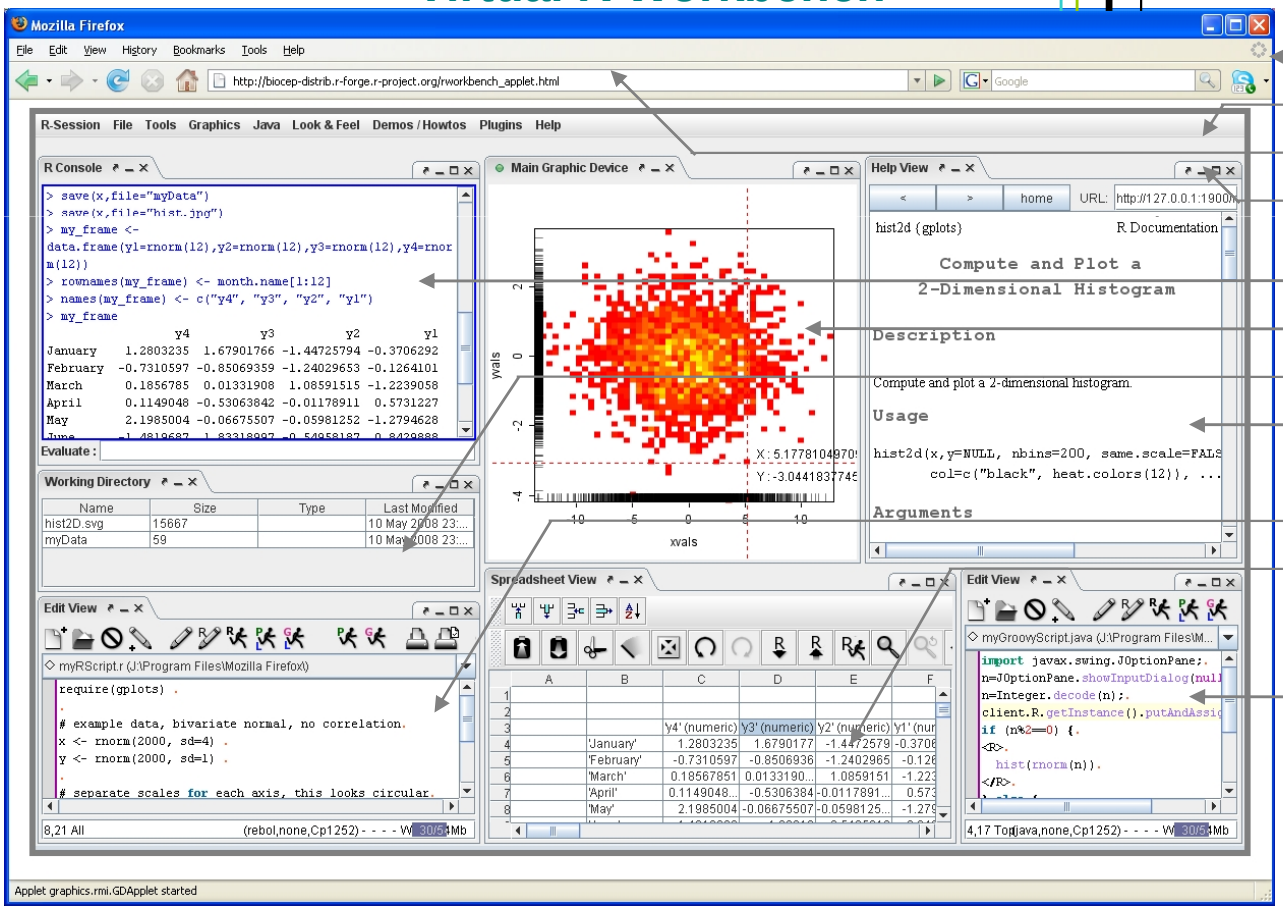
R Virtualization



Server Side - Personal Machine**,** Academic Grids, Clusters, Clouds

Client Side - Internet

Virtual R Workbench



Internet Browser

Java Applet

Virtual R Workbench URL

Docking Framework

R Console

R Graphic Device+Interactors

R Workspace

R Help Browser

R Script Editor

R Spreadsheet

Groovy / Jython Script Editor

Server-side, grid-enabled, collaborative spreadsheet

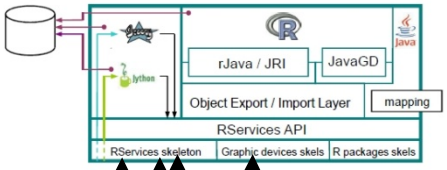
Server-side Data Model

1 - Data in server memory

2- viewing –editing on client machine

Cells access from R Console / R scripts

R functions: 1.*cells.get* 2. *cells.put* 3. *cells.select*

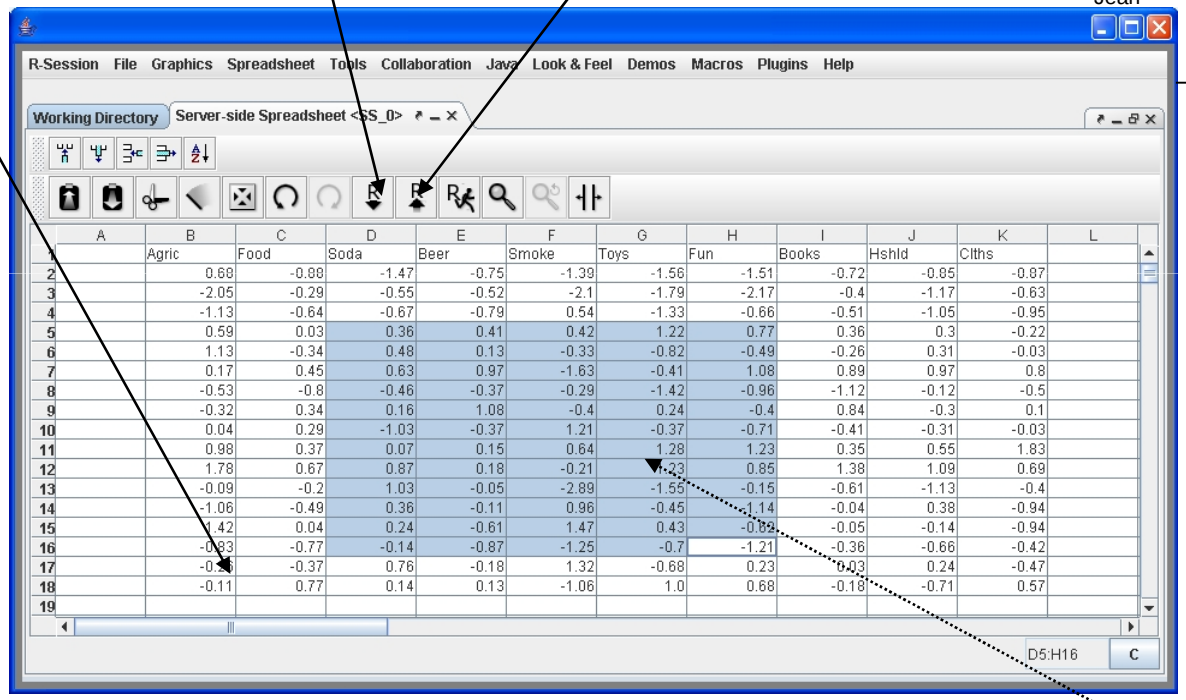


Dynamically evaluted cells using R functions

Paste R expressions into cells

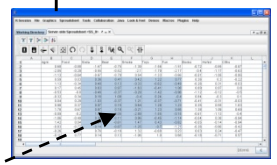
Export cells to R variables

Jean

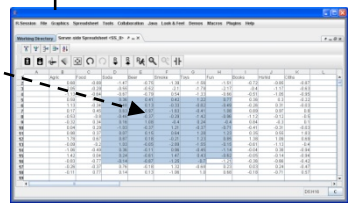


	A	B	C	D	E	F	G	H	I	J	K	L
1		Agric	Food	Soda	Beer	Smoke	Toys	Fun	Books	Hshld	Clths	
2		0.68	-0.88	-1.47	-0.75	-1.39	-1.56	-1.51	-0.72	-0.85	-0.87	
3		-2.05	-0.29	-0.55	-0.52	-2.1	-1.79	-2.17	-0.4	-1.17	-0.63	
4		-1.13	-0.64	-0.67	-0.79	0.54	-1.33	-0.66	-0.51	-1.05	-0.95	
5		0.59	0.03	0.36	0.41	0.42	1.22	0.77	0.36	0.3	-0.22	
6		1.13	-0.34	0.48	0.13	-0.33	-0.82	-0.49	-0.26	0.31	-0.03	
7		0.17	0.45	0.63	0.97	-1.63	-0.41	1.08	0.89	0.97	0.8	
8		-0.53	-0.8	-0.46	-0.37	-0.29	-1.42	-0.96	-1.12	-0.12	-0.5	
9		-0.32	0.34	0.16	1.08	-0.4	0.24	-0.4	0.84	-0.3	0.1	
10		0.04	0.29	-1.03	-0.37	1.21	-0.37	-0.71	-0.41	-0.31	-0.03	
11		0.98	0.37	0.07	0.15	0.64	1.28	1.23	0.35	0.55	1.83	
12		1.78	0.67	0.87	0.18	-0.21	1.23	0.85	1.38	1.09	0.69	
13		-0.09	-0.2	1.03	-0.05	-2.89	-1.55	-0.15	-0.61	-1.13	-0.4	
14		-1.06	-0.49	0.36	-0.11	0.96	-0.45	-1.14	-0.04	0.38	-0.94	
15		1.42	0.04	0.24	-0.61	1.47	0.43	-0.82	-0.05	-0.14	-0.94	
16		-0.83	-0.77	-0.14	-0.87	-1.25	-0.7	-1.21	-0.36	-0.66	-0.42	
17		-0.[illegible]	-0.37	0.76	-0.18	1.32	-0.68	0.23	0.03	0.24	-0.47	
18		-0.11	0.77	0.14	0.13	-1.06	1.0	0.68	-0.18	-0.71	0.57	
19												

Pierre



Paul



Macros

- User-defined actions : run user R / Groovy / Python scripts
- Events-driven macros **1-** on cells change **2**- on R variables change
- Data Links : dock R variables in the Spreadsheet : synchronized changes

Collaboration

- Simultaneous viewing of the same data by Jean, Pierre & Paul
- Collaborative cells editing
- Broadcasted cells Selection

Integrating R - State of the art

- **SJava and rJava/JRI**
  - Basic mapping via JNI of the R C API
- **TypeInfo**
  - Plug meta descriptions to R functions
- **RWebservices**
  - Generated Java Beans for basic R Types / S4 Classes
  - Axis Web Services based on SJava and ActiveMQ
- **JavaGD**
  - R devices connection to Java (JGR)
- **Rserve**
  - TCP/IP interface to R

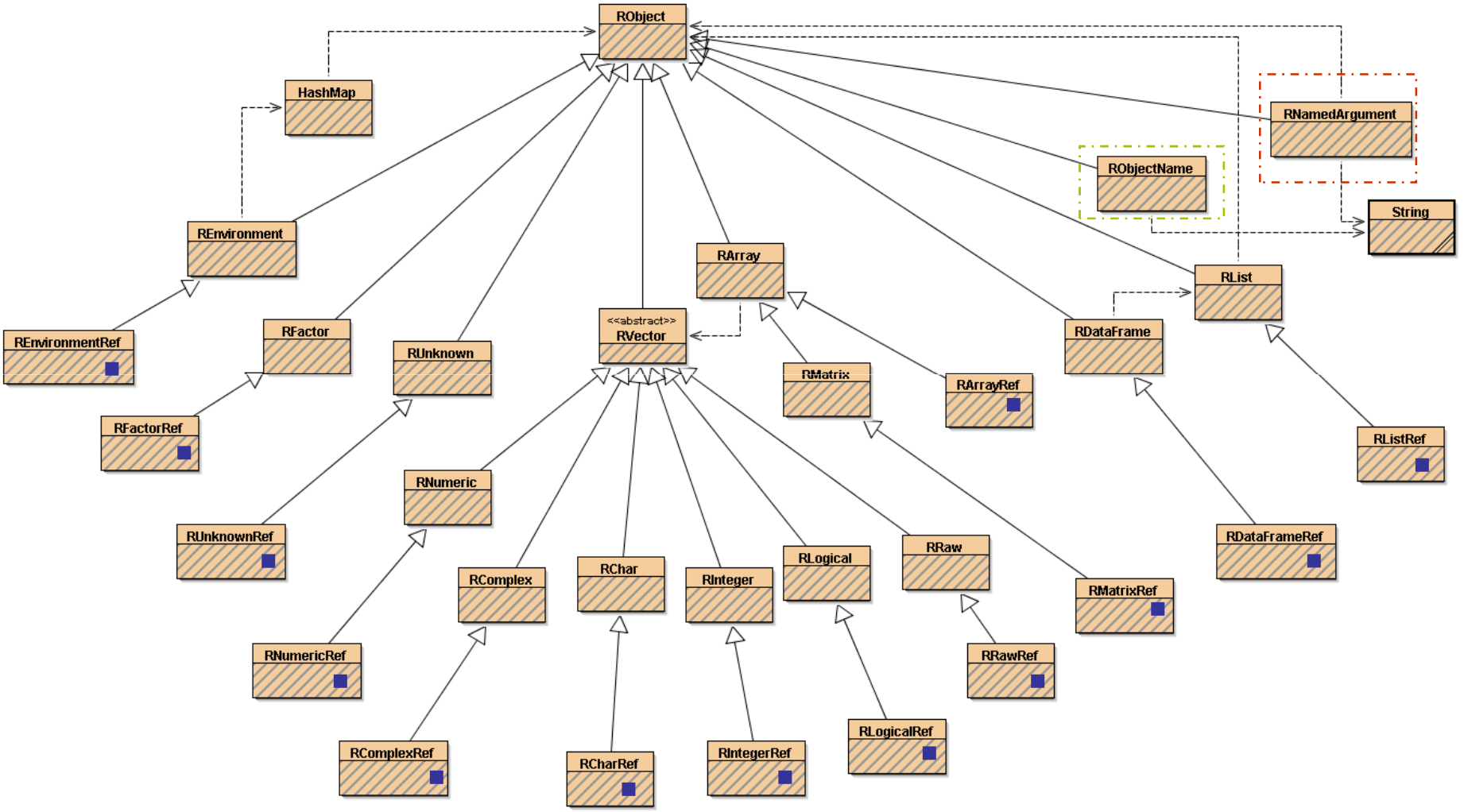
What was missing ?

- High Level Java API for Accessing R
- Stateful, Resuable, Remotable R Components
- Scalable, Distributed, R Based Infrastructure
- Safe multiple clients framework for components usage as a pool of indistinguishable Remote Resources
- User friendly Interface for the remote resources creation, tracking and debugging

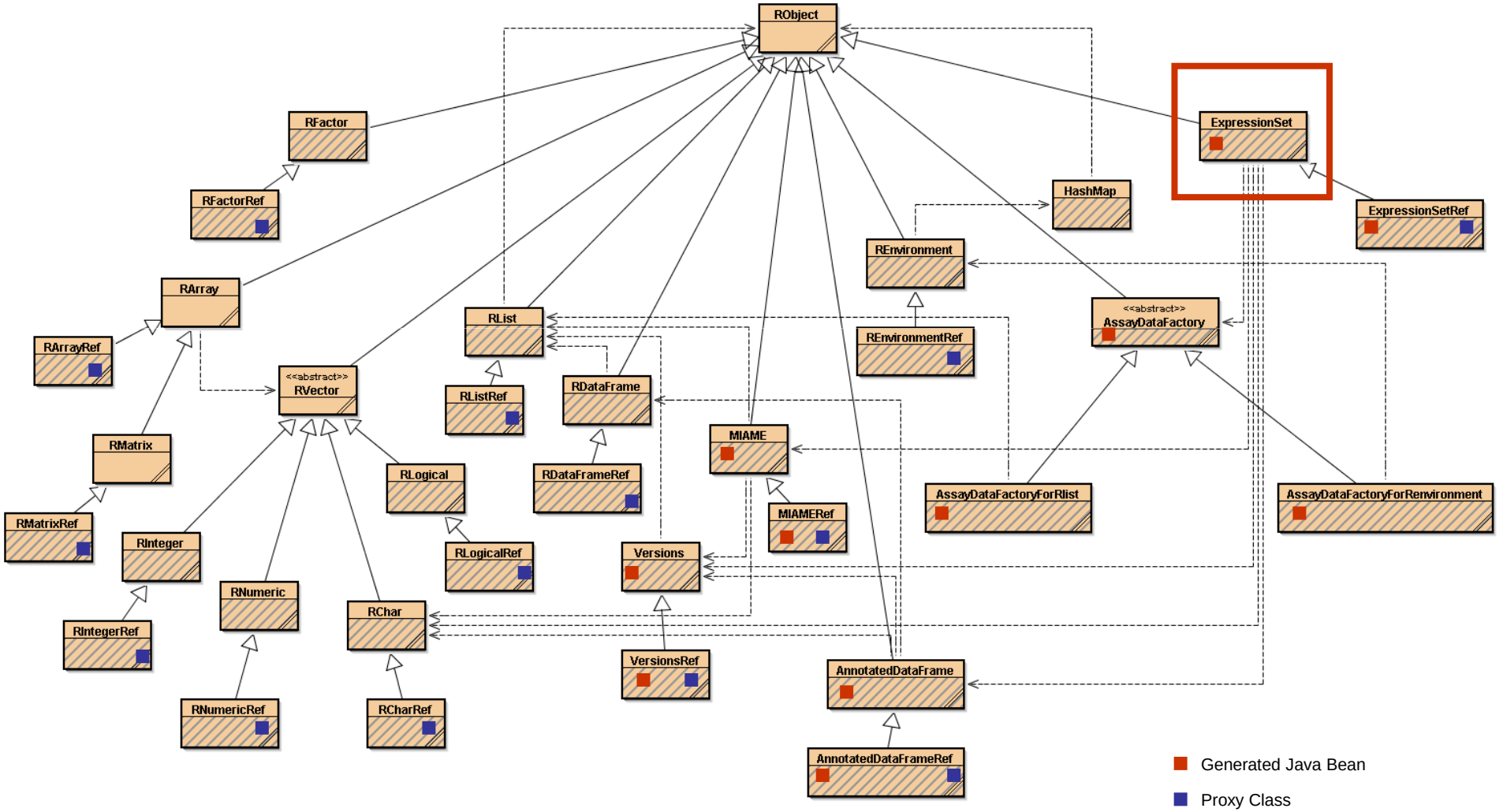
What was missing ?

- Generated light-weight Java proxies for R Types / S4 Classes
- On-demand mapping and deployment of R packages as RMI Components or as JAX-WS Web Services
- Remotable R Graphics / Swing Components for R
- Remote R components files exchange API
- Semi-thick client (applet) for web based tools using R

Standard R objects mapping to Java



Generated beans for ExpressionSet



RServices API - I

```
public interface RServices extends ManagedServant {
-------------------------------------------------------------------------
   public String  consoleSubmit(String expression) throws …
   public String  evaluate(String expression) throws …
-------------------------------------------------------------------------
   public RObject getObject(String expression) throws …
   public Object  getObjectConverted(String expression) throws …
   public RObject getReference(String expression) throws …
   public RObject getObjectName(String expression) throws …
-------------------------------------------------------------------------
   public void    putAndAssign(Object obj, String name) throws …
   public RObject putAndGetReference(Object obj) throws RemoteException;
-------------------------------------------------------------------------
   public RObject call(String methodName, Object... args) throws …
   public RObject callAndConvert(String methodName, Object... args) throws …
   public RObject callAndGetReference(String methodName, Object... args) throws …
   public RObject callAndGetObjectName(String methodName, Object... args) throws …
   public void    callAndAssign(String varName,String methodName,Object...args)throws .
-------------------------------------------------------------------------
   public RObject realizeObjectName(RObject objectName) throws …
   public Object  realizeObjectNameConverted(RObject objectName) throws …
   public RObject referenceToObject(RObject refObj) throws …
-------------------------------------------------------------------------
   public boolean isReference(RObject obj) throws …
   public void    assignReference(String name, RObject refObj) throws …
-------------------------------------------------------------------------
}
```

RServices API - II

```
public interface RServices extends ManagedServant {
-----------------------------------------------------------------------
  public String[] listPackages() throws …
  public RPackage getPackage(String packageName) throws …
-----------------------------------------------------------------------
  public GDDevice newDevice(int w, int h) throws …
  public GDDevice[] listDevices() throws …
-----------------------------------------------------------------------
  public String[] getWorkingDirectoryFileNames() throws …
  public FileDescription getWorkingDirectoryFileDescription(String fileName) throws…
  public void     createWorkingDirectoryFile(String fileName) throws …
  public void     removeWorkingDirectoryFile(String fileName) throws …
  public byte[]   readWorkingDirectoryFileBlock(String name,long off,int size)throws…
  public void     appendBlockToWorkingDirectoryFile(String name, byte[] block)throws…
-----------------------------------------------------------------------
  public String   getRHelpFileUri(String topic, String pack) throws …
  public byte[]   getRHelpFile(String uri) throws …
-----------------------------------------------------------------------
  public Vector<RAction> popRActions() throws …
-----------------------------------------------------------------------
}
```

```
public interface GDDevice extends Remote {
    public Vector<GDObject> popAllGraphicObjects() throws …
    public void fireSizeChangedEvent(int w, int h) throws …
    public void dispose() throws …
    …
}
```

RServices API - III

```
public interface RServices extends ManagedServant {
-------------------------------------------------------------------------
   public void    startHttpServer(int port) throws …
   public void    stopHttpServer() throws …
-------------------------------------------------------------------------
   public String  pythonExec(String pythonCommand) throws …
   public RObject pythonEval(String pythonCommand) throws …
   public void    pythonSet(String name, Object Value) throws …
-------------------------------------------------------------------------
   public String  groovyExec(String groovyCommand) throws …
   public Object  groovyEval(String expression) throws …
   public void    groovySet(String name, Object Value) throws …
-------------------------------------------------------------------------
   public void    setCallBack(RCallback callback) throws …
-------------------------------------------------------------------------
   public String  getStatus() throws …
   public void    stop() throws …
   public void    freeReference(RObject refObj) throws …
   public void    freeAllReferences() throws …
   public String  print(String expression) throws …
   public String  sourceFromResource(String resource) throws …
   public String  sourceFromBuffer(StringBuffer buffer) throws …
   public RNI     getRNI() throws …
-------------------------------------------------------------------------
   …
}
```

Remote Resources Pooling Framework

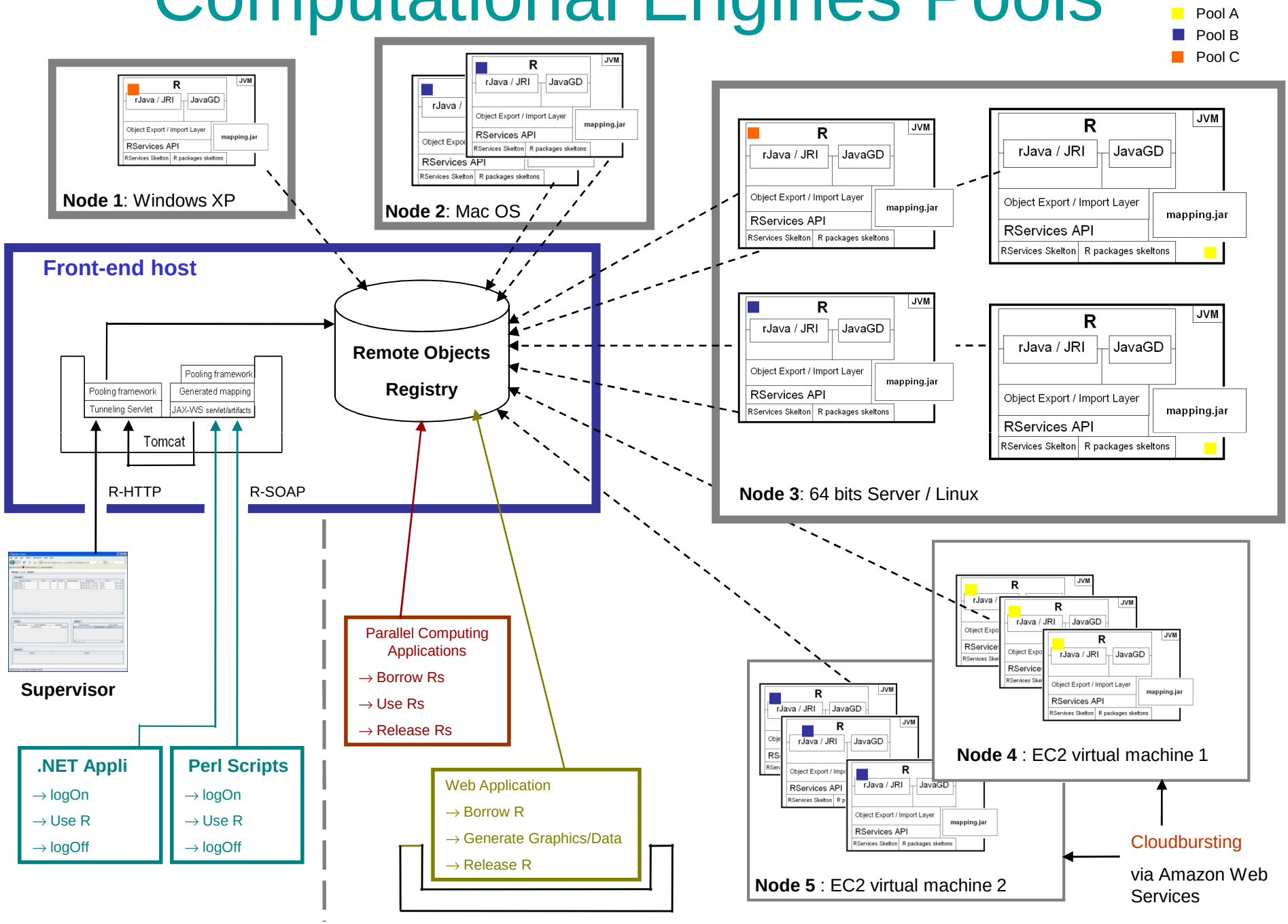
- Generic Standalone framework
- Pooling of any RMI components and if combined with JNI of any library / open architecture

- New Remote Object Registry based on Derby| Oracle| MySQL

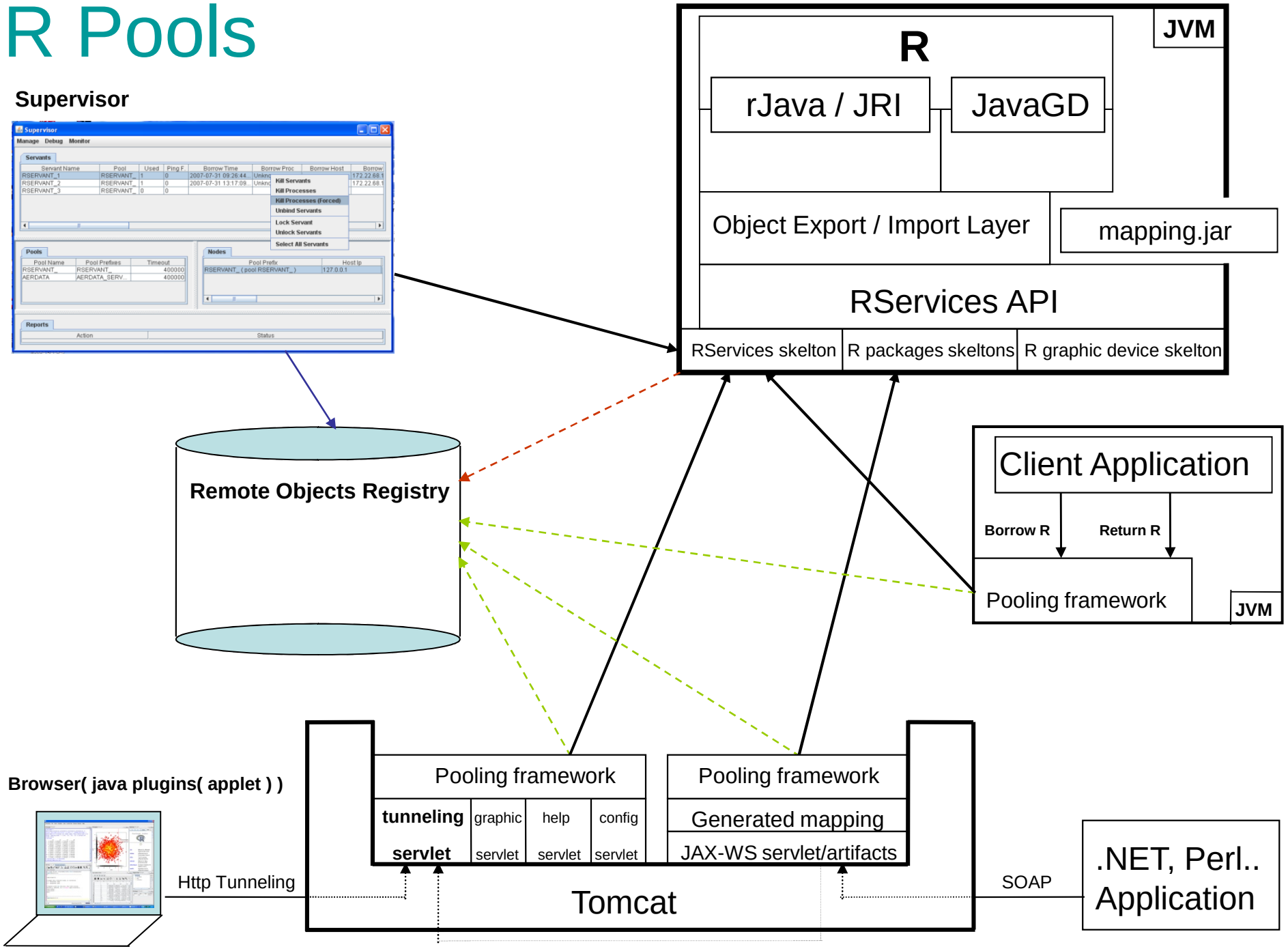
- Three implementations available
  - rmiregistry / mono-node / single client process
  - rmiregistry / multinodes / single client process
  - database ROR / multinodes / multiple client processes

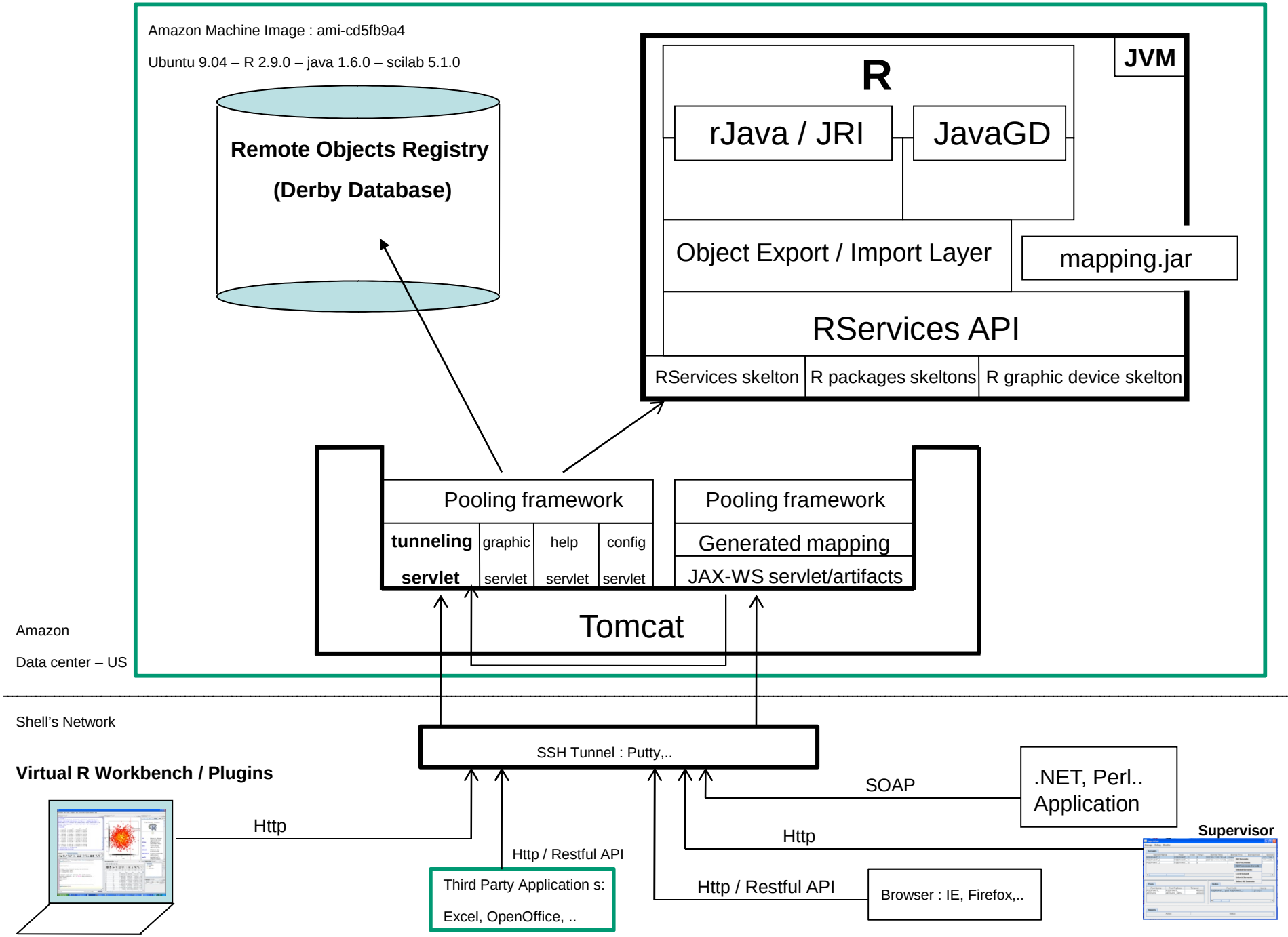
- User friendly interface for the remote resources creation, tracking and debugging, nodes and pools management

Computational Engines Pools

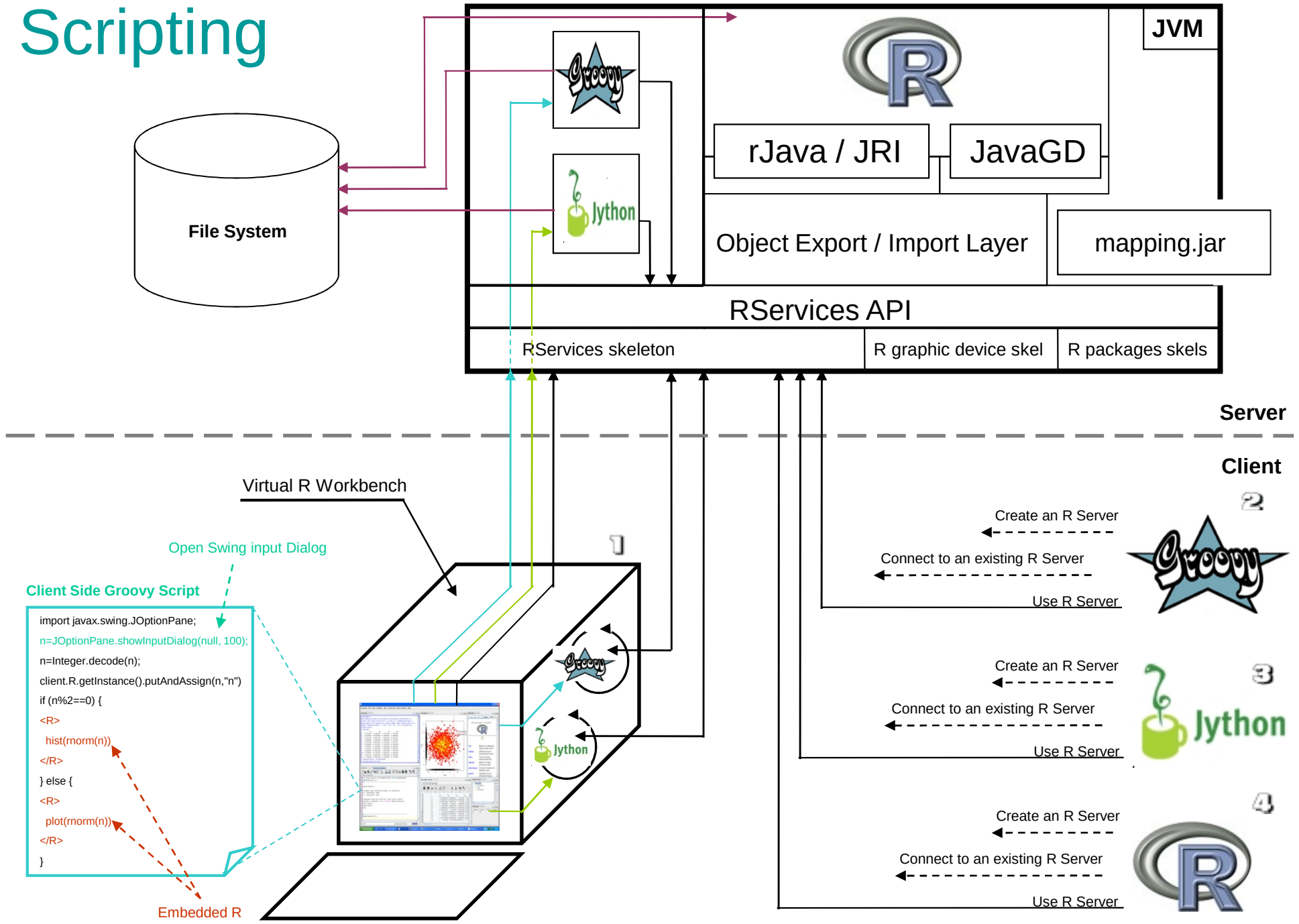


R Pools


Supervisor
JVM
R
rJava / JRI
JavaGD
Object Export / Import Layer
mapping.jar
RServices API
RServices skelton
R packages skeltons
R graphic device skelton
Remote Objects Registry
Client Application
Borrow R
Return R
Pooling framework
JVM
Browser(java plugins(applet))
Pooling framework
tunneling servlet
graphic servlet
help servlet
config servlet
Pooling framework
Generated mapping
JAX-WS servlet/artifacts
Tomcat
Http Tunneling
SOAP
.NET, Perl.. Application


Amazon Machine Image : ami-cd5fb9a4
Ubuntu 9.04 – R 2.9.0 – java 1.6.0 – scilab 5.1.0
Remote Objects Registry
(Derby Database)
JVM
R
rJava / JRI
JavaGD
Object Export / Import Layer
mapping.jar
RServices API
RServices skelton
R packages skeltons
R graphic device skelton
Pooling framework
tunneling servlet
graphic servlet
help servlet
config servlet
Pooling framework
Generated mapping
JAX-WS servlet/artifacts
Tomcat
Amazon
Data center – US
Shell's Network
SSH Tunnel : Putty,..
Virtual R Workbench / Plugins
Http
Http / Restful API
Third Party Application s:
Excel, OpenOffice, ..
SOAP
.NET, Perl..
Application
Http
Supervisor
Http / Restful API
Browser : IE, Firefox,..

Scripting


JVM
File System
rJava / JRI
JavaGD
Object Export / Import Layer
mapping.jar
RServices API
RServices skeleton
R graphic device skel
R packages skels
Server
Client
Virtual R Workbench
Open Swing input Dialog
Client Side Groovy Script
import javax.swing.JOptionPane;
n=JOptionPane.showInputDialog(null, 100);
n=Integer.decode(n);
client.R.getInstance().putAndAssign(n,"n")
if (n%2==0) {
<R>
hist(rnorm(n))
</R>
} else {
<R>
plot(rnorm(n))
</R>
}
Embedded R
1
2
Create an R Server
Connect to an existing R Server
Use R Server
3
Create an R Server
Connect to an existing R Server
Use R Server
4
Create an R Server
Connect to an existing R Server
Use R Server

Parallel Computing

```
final double[][] m=..;
Future<Double>[] result=new Future[m.length];
ExecutorService exec = Executors.newFixedThreadPool(50);
for (int i=0; i<result.length; ++i) {
final double[] v=m[i];
result[i]= exec.submit(
            new Callable<Double>() {
      public Double call() throws Exception {
        RServices r=null;
        try {
          r=(RServices)ServantProviderFactory.getFactory().getServantProvider().borrowServantProxy();
          Rnumeric mean=(RNumeric)r.call("mean", new RNumeric(v));
          return mean.getValue()[0];
        } finally {  ServantProviderFactory.getFactory().getServantProvider().returnServantProxy(r);  }
   }
          });
}
while(true) {
int count=0; for (int i=0; i<result.length; ++i) if (result[i].isDone()) ++count; if (count==result.length) break;
Thread.sleep(100);
}
for (int i=0; i<result.length; ++i) System.out.println(result[i].get());
```

Snow with Biocep

From the R Console :

- makeCluster(n,...) stopCluster(cl)
  - Starting and Stopping clusters
- clusterEvalQ(cl, expr)
  - The expression is evaluated on the slave nodes.
- clusterApply(cl, seq, fun, ...)
  - Calls the function with the first element of the list on the first node, with the second element of the list on the second node, and so on.
- clusterExport(cl, list)
  - Assigns the global values on the master of the variables named in 'list' to variables of the same names in the global environments of each node.

…

Web Services Generation

Script / globals.r

```
square  function(x) {return(x^2) }
typeInfo(square)  SimultaneousTypeSpecification(
TypedSignature(x = "numeric"), returnType = "numeric")
```

Script / rjmap.xml

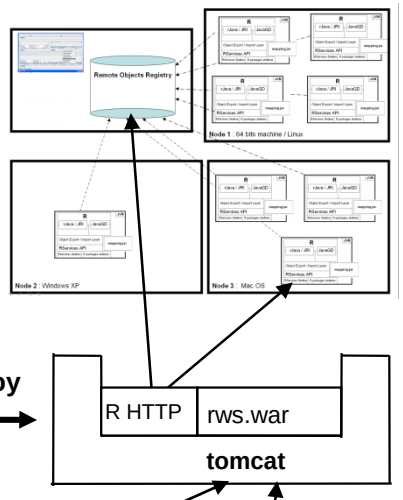
```
<rj>
  <publish>
    <functions> <function name="square" forWeb="true"/> </functions>
  </publish>
  <scripts>  <initScript name="globals.r" embed="true"/> </scripts>
</rj>
```

WS generator

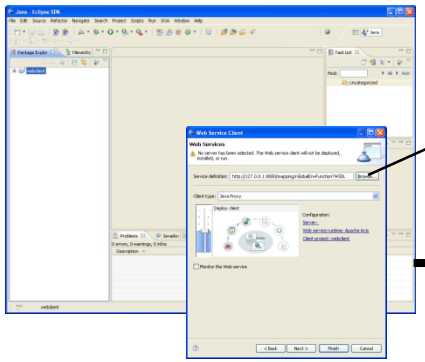
rws.war

+ mapping.jar
+ pooling framework
+ R Java Bridge
+ **JAX-WS**
  - Servlets
  - Generated artifacts

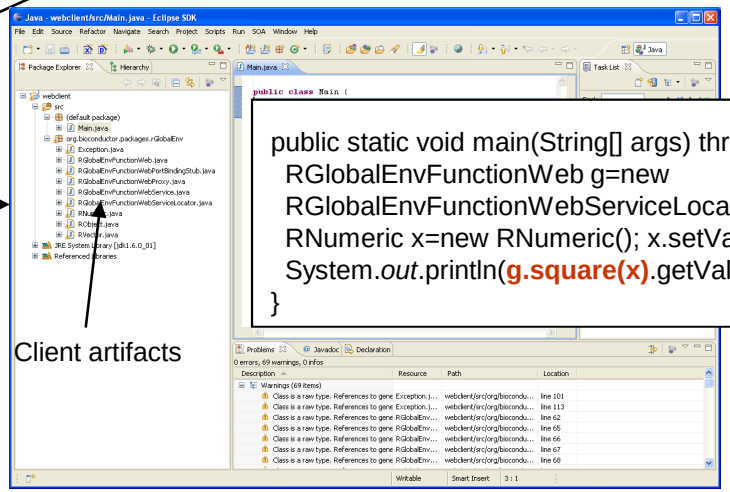
Deploy



http://127.0.0.1:8080/rws/rGlobalEnvFunction?WSDL



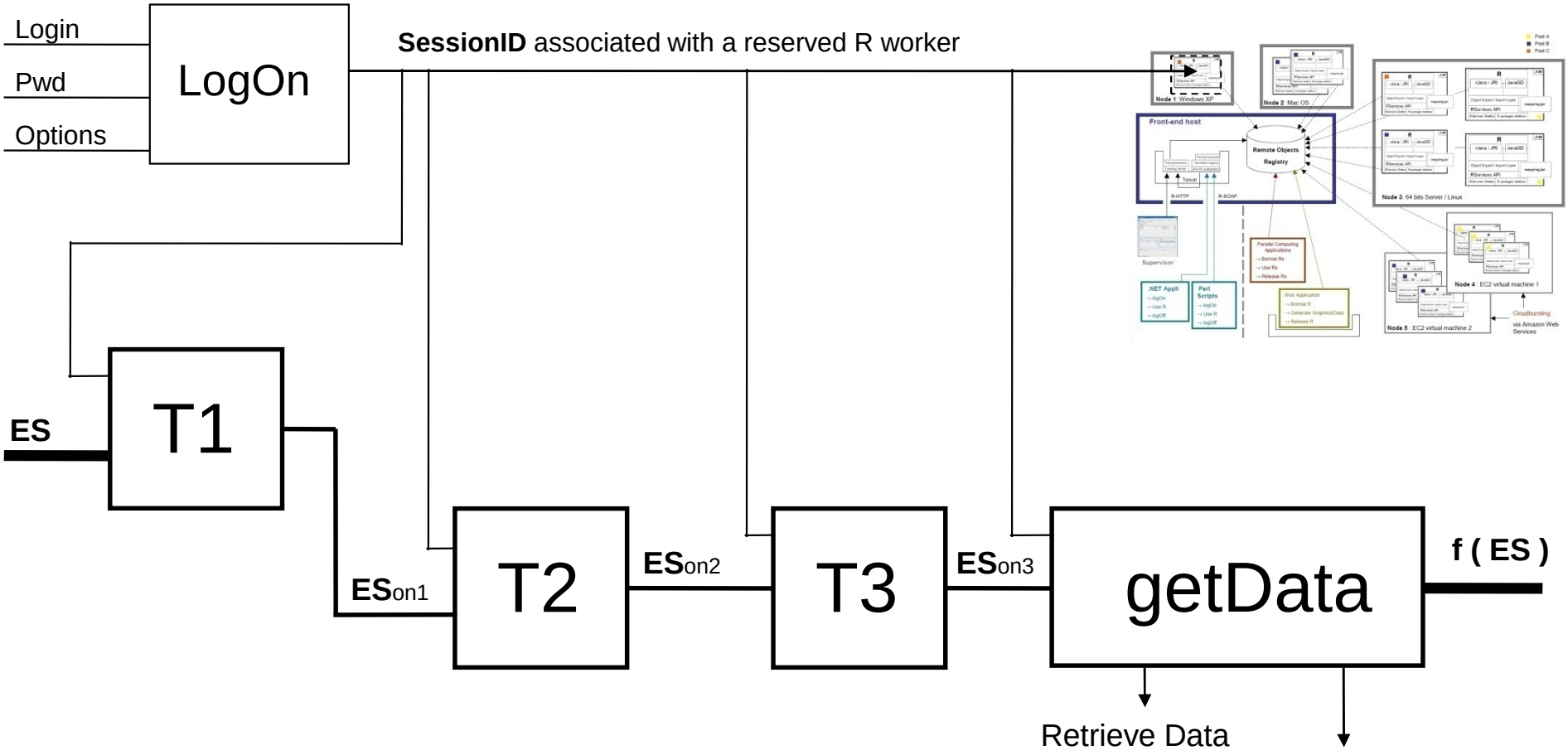
Eclipse Web Service Client Generator



Client artifacts

```
public static void main(String[] args) throws Exception {
  RGlobalEnvFunctionWeb g=new
  RGlobalEnvFunctionWebServiceLocator().getrGlobalEnvFunctionWebPort();
  RNumeric x=new RNumeric(); x.setValue(new Double[]{6.0});
  System.out.println(g.square(x).getValue()[0]);
}
```

Workflows with Stateful Web Services



T1,**T2**,**T3** : Generated Stateful Web Services for R functions T1,T2 & T3

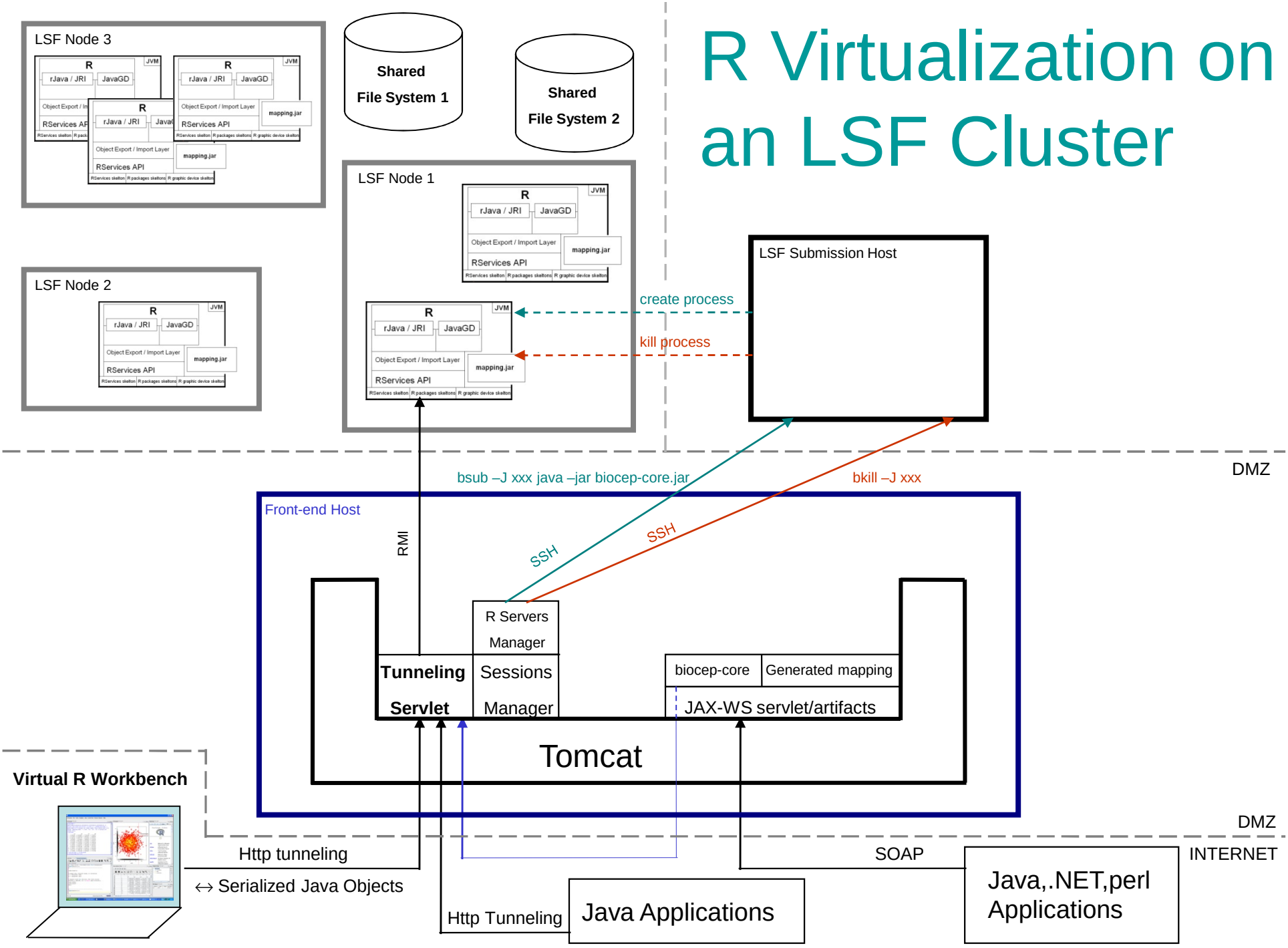
LogOn, **getData** : R-SOAP methods

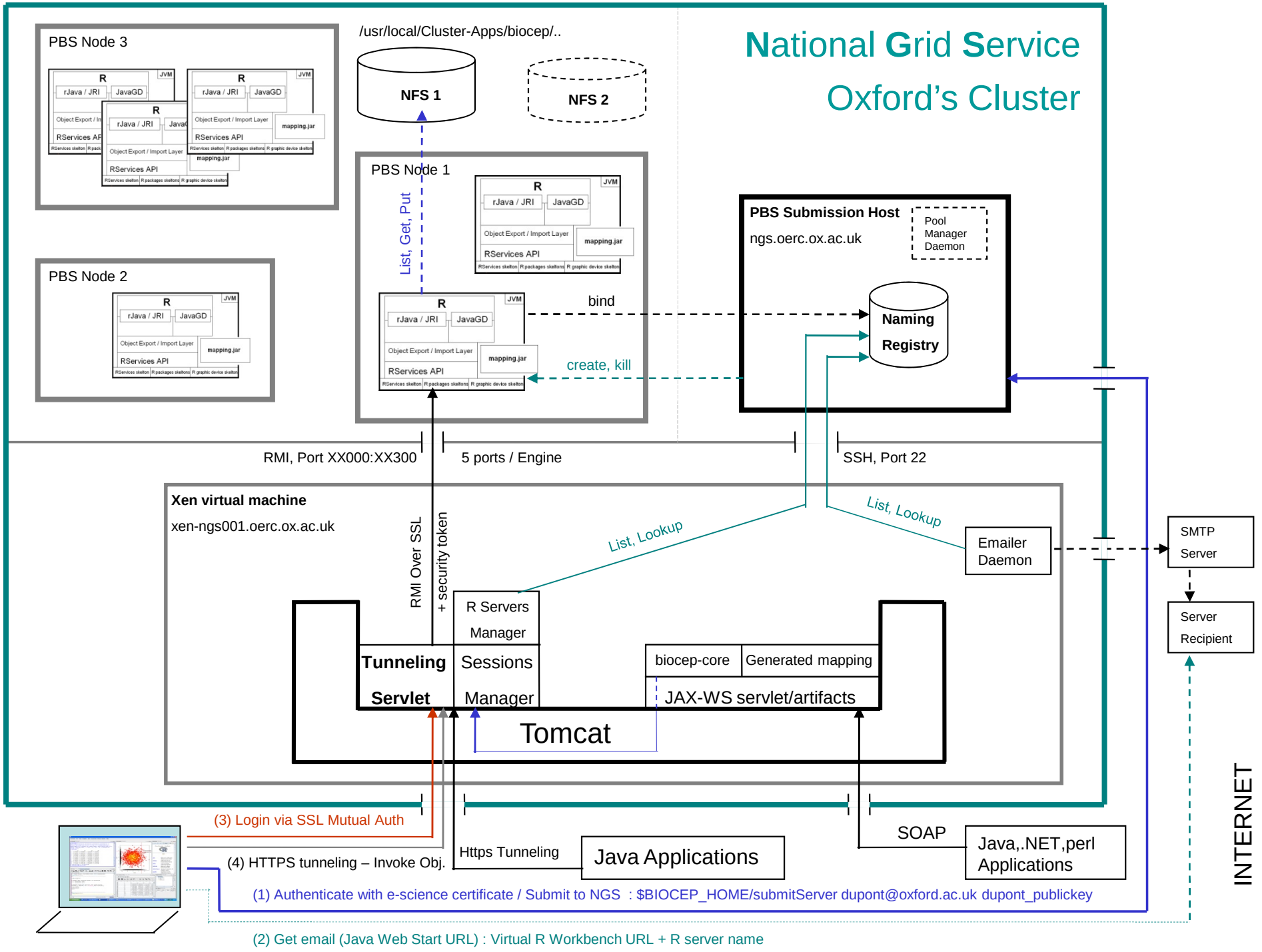
ES : ExpressionSet

ESon1, **ES**on2, **ES**on3 : ExpressionSet Object Names

f = T3 o T2 o T1

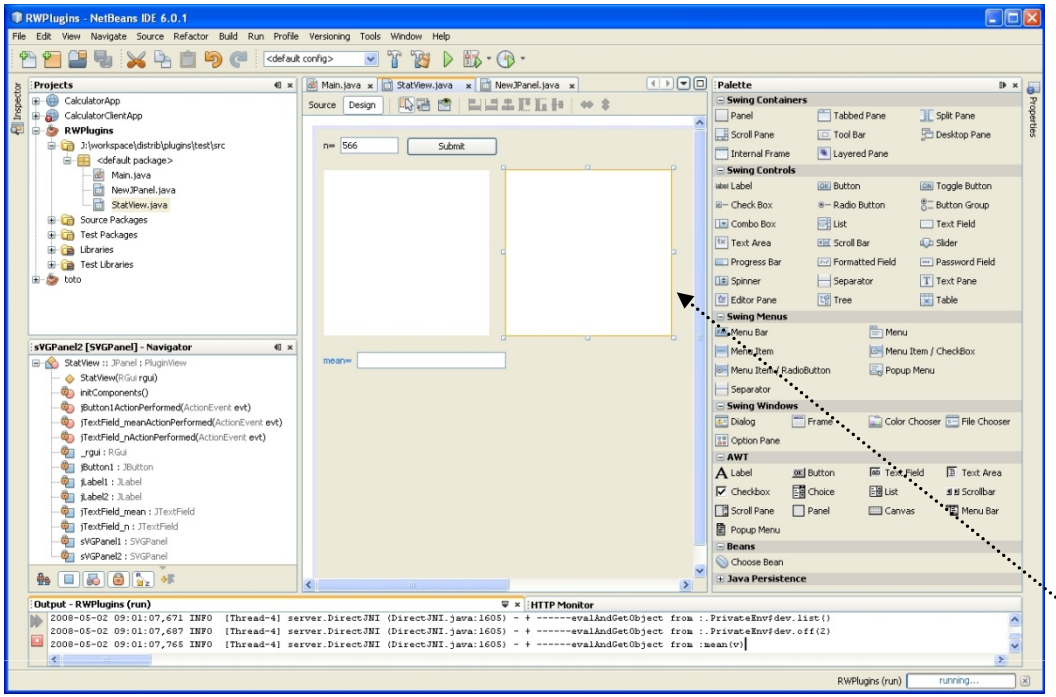
R Virtualization on an LSF Cluster




National Grid Service
Oxford's Cluster
PBS Node 3
PBS Node 2
PBS Node 1
/usr/local/Cluster-Apps/biocep/..
NFS 1
NFS 2
List, Get, Put
bind
create, kill
PBS Submission Host
ngs.oerc.ox.ac.uk
Pool Manager Daemon
Naming Registry
RMI, Port XX000:XX300
5 ports / Engine
SSH, Port 22
Xen virtual machine
xen-ngs001.oerc.ox.ac.uk
RMI Over SSL
+ security token
List, Lookup
List, Lookup
Emailer Daemon
SMTP Server
Server Recipient
R Servers Manager
Tunneling Servlet
Sessions Manager
biocep-core
Generated mapping
JAX-WS servlet/artifacts
Tomcat
(3) Login via SSL Mutual Auth
(4) HTTPS tunneling – Invoke Obj.
Https Tunneling
Java Applications
SOAP
Java,.NET,perl Applications
(1) Authenticate with e-science certificate / Submit to NGS : $BIOCEP_HOME/submitServer dupont@oxford.ac.uk dupont_publickey
(2) Get email (Java Web Start URL) : Virtual R Workbench URL + R server name
INTERNET

GUI Plugins

Netbeans 6 – Visual GUI builder



Compile

myPlugin.jar

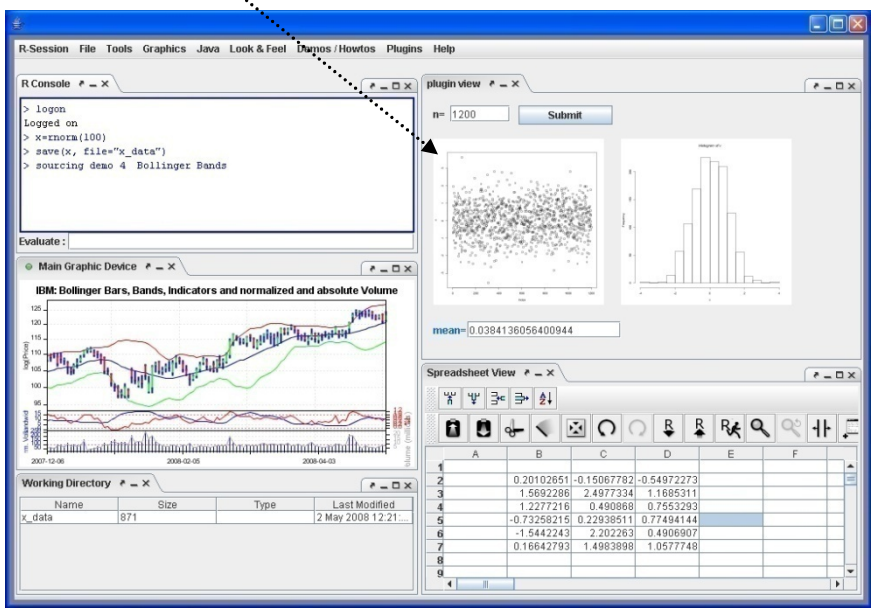
+ myView1

+ myView2

+ descriptor.xml

Import Plugin

Virtual R Workbench



Upload plugin

Browse Repository

Download Plugin

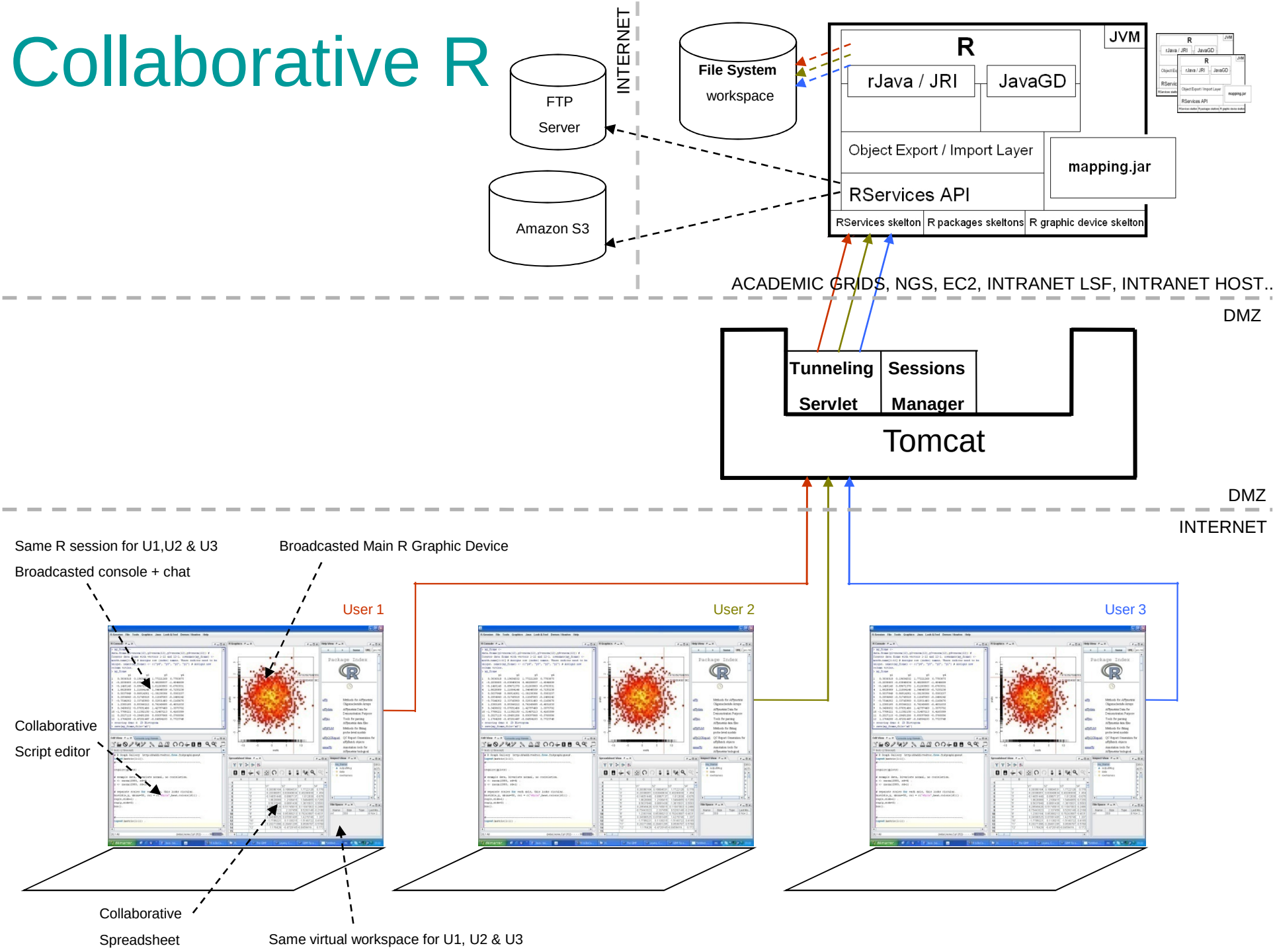
Plugins Repository

* myPlugin *myDashboard

* Klimt * iPlots

* Mondrian *E. Profiler

Collaborative R



Ease Of Use - I

- **Reasonable Pre-requirements**

Java 5 and R>=2.5 accessibles from the command line : to run R servers, generate mappings & Web Services, run the miniature virtualisation and the R-SOAP Web Apps..

- **All-in-one Highly Productive Workbench**

Docking framework, spreadsheets, syntax highlighting enabled editors, objects viewer, help browser, storage views, zooming system on R graphics, settings persistence..

- **Easy Computational Resource Acquisition**

Provide nothing to run R servers on local machine

Provide HOST / PORT / LOGIN / PWD to run R Servers on remote hosts (SSH)

Provide URL & (LOGIN/PWD or X.509 Certificate) to Connect to Grid Rs or Cluster Rs

- **Easy Scripting**

Simple API for running/connecting to R servers

Embeddable R code (<R> </R>) within scripts

Automatic conversion from/to R Objects for common data types(standard,arrays,collections)

Ease Of Use -II

- **Easy Plugins Integration**

  Import local file / Browse Plugins repository and choose a plugin

- **« Push button » Web Services Generation/Web Services Deployment**

  Add TypeInfo to your function / add your function name to an XML / run biocep-tools

  Deploy: *java –port=80 –cp biocep-core.jar HttpServer rvirtual.war MyWebServices.war*

- **Self-contained jar & war files distribution :**

  biocep.jar biocep-core.jar biocep-tools.jar rvirtual.war rws.war

- **Configurationless Parallel Computing from R console :**

  makeCluster(n,..), stopCluster(cl), clusterEvalQ(cl, expr), clusterApply(cl, seq, fun, ..) ...

Acknowledgements

ACS: Madi Nassiri **Amazon**: Simone Brunozzi, Deepak Singh **AT&T Research Labs**: Simon Urbanek **ATUGE**: Imen Essafi, Béchir Tourki, Ilyes Gouja, HatemHachicha, Amine Elleuch **Banca d'Italia**: Giuseppe Bruno **Bio-IT World**: Kevin Davies **Cambridge Healthtech Institute**: Cindy Crowninshield **City University of New York**: Mario Morales, Makram Talih **Columbia University**: Omar Besbes **Dataspora**: Michael E. Driscoll **EBI**: Alvis Brazma, Wolfgang Huber, Kimmo Kallio, Misha Kapushesky, Michael Kleen, Alberto Labarga, Philippe Rocca-Serra, Ugis Sarkans, Kirsten Williams, Eamonn Maguire **EPFL**: Darlene Goldstein **ETH Zürich**: Yohan Chalabi, Diethelm Würtz, Martin Mächler **EVRI.com**: Seth Falcon **FHCRC**: Martin Morgan, Nianhua Li **Google**: Olivier Bosquet **FVG LLC**: Lisa Wood **Harvard Business School**: Ousseynou Nakoulima **Harvard University**: Tim Clark, Sudeshna Das, Douglas Burke,Paolo Ciccarese **IBM**: Jean-Louis Bernaudin, Pascal Sempe, Loic Simon, Lea A Deleris, Alex Fleischer, Alain Chabrier **Imperial College London**: Asif Akram, Vasa Curcin, John Darlington, Brian Fuchs **Indiana University**:Michael Grobe **INRIA**: David Monteau **JISC**: David Flanders **Johnson & Johnson - Janssen Pharmaceutica**: Patrick Marichal **Lancaster University**: Robert Crouchley, Daniel Grose **Leibniz Universität Hannover**: Kornelius Rohmeier **Limagrain**: Zivan Karaman **Mekentosj**: Alexander Griekspoor **Microsoft**: Eric Le Marois, Tony Hey **Mubadala**: Ghazi Ben Amor **Nature Publishing Group**: Ian Mulvany, Steve Scott **NCeSS**: Peter Halfpenny, Rob Procter, Marzieh Asgari-Targhi, Alex Voss, YuWei Lin, Mercedes Argüello Casteleiro, Wei Jie, Meik Poschen, Katy Middlebrough, Pascal Ekin, June Finch, Farzana Latif, Elisa Pieri, Frank O'Donnell **New York Java User Group**: Frank D Greco **OeRC**: Dimitrina Spencer, Matteo Turilli, David Wallom, Steven Young **OMII-UK**: Neil Chue Hong, Steve Brewer **OpenAnalytics**: Tobias Verbeke **Oracle**: Dominique van Deth, Andrew Bond **OSS Watch**: Ross Gardler **Platform Computing**: Christopher Smith **San Diego Supercomputer Center**: Nancy R. Wilkins-Diehr **Sanger Institute**: Daniel Jeffares, Matt Wood, , Phil Butcher **Shell**: Wayne.W.Jones, Nigel Smith **Stanford University**: John Chambers, Balasubramanian Narasimhan, Gunter Walther **SYSTEM@TIC**: Karim Azoum **Technische Universität Dortmund**: Uwe Ligges, Bernd Bischl **The Generations Network**: Jim Porzak **Tunisian Ministry of Communication Technologies**: Lamia Chaffai-Sghaier, Mohamed Saïd Ouerghi **Tunisian Ecole Polytechnique**: Riadh Robbana **UC Berkeley**: Noureddine El Karoui, Terry Speed **UC Davis**: Rudy Beran, Debashis Paul, Duncan Temple Lang **UCLA**: Ivo Dinov **UCSF**: Tena Sakai **Université Catholique de Louvain**: Christian Ritter **University of Cambridge**: Ian Roberts, Robert MacInnis Peter Murray-Rust, Jim Downing **University of Manchester**: Carole Goble, Len Gill, Simon Peters, Richard D Pearson, Iain Buchan, John Ainsworth **University of Plymouth**: Paul Hewson **University of Split**: Ivica Puljak **UTK**: Ajay Ohri **World Bank Group-IFC**: Oualid Ammar **Yahoo**: Laurent Mirguet, Rob Weltman / Charles Dallas, Romain François, Manfred Duchrow, Joerg Mueller, Slava Pestov, ..

www.biocep.net