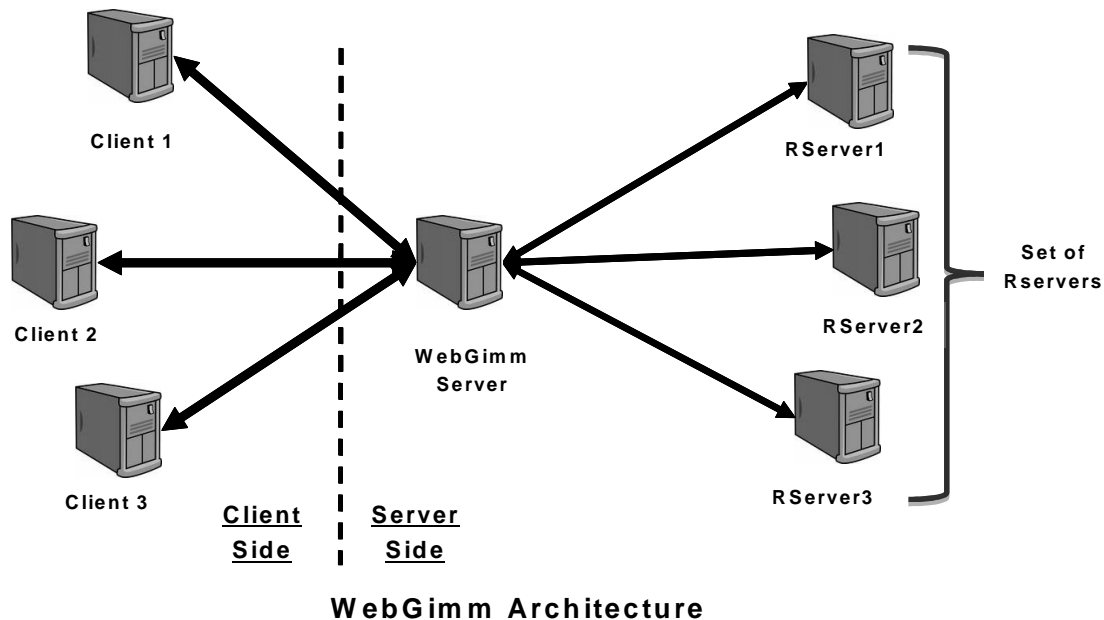


Deployment of WebGimm System

The WebGimm software system consists of the following three components:

1. Client
2. WebGimm server
3. Rserver(s)

This document will outline the details about deployment of these components.



Hardware Requirements:

Hardware requirements for the WebGimm server infrastructure depend on the expected computation load. We do not have strict requirements about the hardware. Though it is recommended to run RServer and WebGimm software on separate machines, the system does not make this a strict requirement. For example, our deployment consists of a WebGimm server running on AMD Quad-Core Opteron processor running at 2.4 GHz and 16GB RAM. RServer(s) run on Intel i7 processor running at 3.3 GHz and 12 GB RAM.

Software Requirements:

We recommend use of Linux for WebGimm and RServer(s). It might be possible to use other operating systems, perhaps after some modifications to the software and deployment procedure, though we have not tested it. We assume availability of Bourne (Again) shell (*sh* or *bash*) and UNIX-style pathnames in our programs.

Deployment Instructions:

Source code for WebGimm system can be downloaded from the URL mentioned on the web page for WebGimm (<http://eh3.uc.edu/gimm/webgimm>). Please follow the following steps to deploy this:

1. Untar and unzip the source. Typically the command using GNU tar is:
`tar -xzf WebGimm.tar.gz`
2. Build the server using command *ant* and then execute *package.sh*. This will create the jar for deploying the client as a Java Web Start application in the directory called *dist*. The server is built in the directory called *target*.
3. Sign this jar for Java Web Start application created in step 2 using *jarsigner*.
4. Edit the configuration file called *config.txt* in the directory called *target* with the parameters of the server deployment.
5. The directory created in step 2 includes a subdirectory called *lib* that contains a few jar files. Include these jar files in the Java *classpath* because they will be needed when the server starts.
6. Install the following software on RServer(s):
 - a. R (<http://www.r-project.org>),
 - b. RServe (<http://www.rforge.net/Rserve/>),
 - c. BioConductor (<http://www.bioconductor.org>),
 - d. gimmR (<http://eh3.uc.edu/gimm/gimm/index.html>) and
 - e. CLEAN (<http://eh3.uc.edu/clean/>).
7. The directory created in step 2 includes a subdirectory called *r* that contains a file called *call_rserve.R*. Place this file on the *RServer* machines. Insert the path on *RServer* to this file in the *config.txt*.
8. Start the *RServer* with the command *R CMD Rserve*. See the website for *RServe* for more details about this.
9. Setup and start an *http server* on WebGimm server machine.
10. Create a *jnlp* file to serve the signed jar created in step 3 and place this *jnlp* and signed jar files on the http server. Make the *jnlp* file available through an http url.
11. Open port 1371 on the WebGimm server for the external network, and port 8790 for the internal network.
12. Start *WebGimm* server with the command:
`java webGimmServer.Server --cfg config.txt`