Integrative Biology VRE

*IB-VRE Infrastructure - Initial Design*  
*Report*

Geoff Williams

---

**Intended Audience**

Anyone.

This report details some of the decisions made in the implementation of the project management environment and portal hosting environment as part of the overall IB-VRE infrastructure.

---

**Issued: 31 Oct 2005**
1. Decision on Technologies

By the end of August 2005 the IB-VRE Project Plan [1] had identified some basic tools/applications required for the following two principal environments:

1.1. Project Management + Software Development Environment

   a) Internal project wiki e.g. trac
   b) Issue tracking e.g. trac
   c) Source code control e.g. CVS, subversion
   d) Document creation e.g. doxygen
   e) Build control e.g. luntbuild, cruisecontrol, ant
   f) Unit testing e.g. JUnit

1.2. Portal/VRE Hosting Environment

   a) uPortal portal
   b) WSRP4J WSRP engine
   c) OGCE portlets
   d) Pluto portlet container

2. Agreement on Design

2.1. Hardware :

Having reached decisions on the technologies the next stage was to consider the hardware and location of the computers. Although all of the above could have been accommodated on the same computer two computers were purchased, one for each of the environments listed above. This would allow the Hosting environment to remain unaffected by activities on the Project Management and vice versa as there exists the potential for the Project Management computer to frequently require excessive processing capabilities, particularly during the build and test processes, which would affect performance whilst IB-VRE users were using the system.

As the installation, configuration and systems administration of these computers was going to be performed by the systems developer in the Computing Laboratory the computers were placed on the comlab network with the following domain names:

1. ibvreprj.comlab.ox.ac.uk - Project Management computer
2. ibvredev.comlab.ox.ac.uk ibvretest.comlab.ox.ac.uk - Portal/VRE Hosting computer

The principal components of the two servers are:

- 64-bit Intel(R) Pentium (R) 4 CPU 3.20GHz
- 2 Gb DDR PC400/PC3200 RAM
2.2. Operating System:

As the systems developer had considerably more experience of systems administration in a Linux environment the SuSE operating system was used and this also would be the operating system in use on the live IB-VRE hosting computer being administered by the eDiamond project systems administrator.

2.3. Systems configuration:

Each of these computers each were set up and configured with only the software and hardware necessary for them to perform the tasks required of them. Moreover the software/applications to be used and developed are, where appropriate, configured and built specifically for the platform on which they are being hosted. For this reason the configuration and structure of both computers have been as consistent as possible so that software to be deployed on the host server can also be effectively be tested on the project management server. Although this results in an extended installation process due to performing non-standard installation procedures there are at least the following benefits:

1. Applications run faster as the binaries are not generic, but machine-specific.
2. Fewer applications to maintain/support when upgrading.
3. Configuration options can be documented.
4. Fewer potential sources of security compromises.
5. No unnecessary processes running.
6. Enforces an understanding of the software/applications being installed.

Where possible applications e.g. Tomcat, MySQL, were installed and configured to run as a non-root user (i.e. non-superuser) to both minimize the chance of exposure of the computer to security compromises with the potential to leave the computer open to root access, and also to avoid the need for anybody to log in as the root user.

A firewall and Intrusion Detection System were installed to minimize the risk of a compromise and to alert the systems administrator of any suspicious activities. There hasn't been, however, an appropriate action list designed for what needs to be done in the event of a compromise.

2.4. Choice of standards:

- JSR-168 (Java Specification Request 168) [6]

Created by the Java Community Process, "JSR-168 is designed to achieve interoperability between portlets, Java-based portal servers (such as uPortal), and other web applications" [9]. Portlets complying to this standard "... have a simple API (Application Programming Interface) to any portal client, support multiple types of clients (multidevice, multibrowser), support localization and internationalization, allow for hot deployment and redeployment of portal applications, and contain declarative security (identical to the mechanism found..."
in the servlet and enterprise JavaBean specification)" [9].

- **WSRP (Web Services for Remote Portlets) [7]**

  Created by the OASIS (Organisation for the Advancement of Structured Information Standards) technical committee "WSRP are presentation-oriented, interactive web services. WSRP provides a standard that enables all content and application providers to provide their services in a manner where they can easily be discovered and plugged into all compliant portals without programming effort on the portal's side" [10].

  By adopting these standards the project aims to provide portlets which can be used in any standards-compliant portal.

### 2.5. Choices made for specific technologies on the IB-VRE computers:

a) **uPortal / WSRP4J / Pluto - Portal framework [2]**.

  uPortal is being used as per the project requirements as it has integrated WSRP and JSR-168 functionality.

  Early experimentation with uPortal 3 in its infancy resulted in the decision to use version 2.5.x for the IB-VRE, and to move to version 3 at a later date. This was because in the version 3 available at the time there was only a minimal functionality as it appeared to be a proof-of-concept version. Version 3 however is being designed to be more standards-compliant e.g. JSR-168, WRSR compatible, than its earlier incarnation.

b) **Trac - Integrated SCM & Project Management [3]**

  Trac has been adopted as the project management tool as the benefits of its use have already been experienced in local software development projects and thus has provided locally accessible knowledge of its implementation and use.

  The Trac system provides very useful functionality such as:

  - A wiki.
  - Easy access to a Subversion version control system.
  - Tickets - a mechanism for linking project requirements/decisions with outcomes.

b) **Subversion - Source code control [4]**

  Claiming to be a superior replacement to CVS, it benefits from:

  - Improved concurrent access to source code.
  - Faster operations.
  - Transaction-based commits.

  It is also the version control system used by the Trac project management tool.
d) Ant - Build control

As most of the applications being used in the portal environment (e.g. uPortal, WSRP4J, Pluto, Portlets) are Java-based they also come with their specific build scripts which need only minor alterations to enable them to be used locally.

The Ant scripts are also being assisted by Bash scripts which are automating (and in effect providing minimal documentation to) the build and deploy process.

e) MySQL - Database

Early experimental work using uPortal was done using the Hypersonic SQL database [5] but this was not used for the production environments as the uPortal documentation recommended the against its use (see http://www.uportal.org/administrators/hsq1.html).

As the systems developer was familiar with MySQL this was used instead although there were some issues with its use, particularly uPortal's need for primary key fields on some tables to exceed MySQL's 1k limit when the database was configured to use UTF-8 character encoding.

2.6. Source code base:

As this project is going to be using some of the latest "cutting-edge" technologies it was considered appropriate to keep some of the source code base being used under our own version control system so as to prevent as much as possible the direction and progress of our project being dictated by third parties. The ability to build our own portal and components from source code will also enable us to identify and locate bugs more effectively by firstly being able to potentially place debugging statements in code, and secondly to visualise lines in source code where error/debug messages give them. It will also assist in determining performance issues effectively e.g. communication issues, memory requirements, database accessing, etc.

Another benefit to retaining the source code in the Subversion repository is that some of the users/developers on the project are not familiar with linux environment and so the project would benefit from code being "checked out" of the repository and then compiled and tested locally on individual user's computers before being "committed" back into the repository. This ensured that there was some cross-platform development and testing being performed which would assist the project in developing platform-independent software. As the same users/developers had their own preferences for specific technologies e.g. databases, IDEs (Integrated Development Environments), etc., then this would also assist in developing software which was not only platform-independent, but also software which would be operational in a variety of common environments.

2.7. Current / Ongoing operations:

- Backup configuration.

Both server computers are currently being configured to subscribe to the HFS (Hierarchical File Server) backup facilities available at OUCS (Oxford University Computing Services) [8]. This service is to be used as a means of ensuring that in the unfortunate event of a serious hardware problem on either server the project will not be jeopardised. For less critical problems local
copies of important files will be held on the respective computers.

- Automated code building, testing and deployment.

Where possible the systems developer will continue to aim to reduce duplication of effort by creating scripts and automated processes that will enable the developers to automate frequently required tasks. The intention is to develop an automated build and test process that will create a "clean" version of the IB-VRE portal with only IB-VRE-specific user structure, security, functionality and appearance. For this to be possible a good working knowledge of uPortal administration will be required with the aim of integrating it into the build process.

- Security patches/upgrades.

Various websites and security alert feeds will be monitored by the systems developer to ensure that the applications running on either of the servers do not pose a security risk.

3. Initial IB-VRE Infrastructure

Early infrastructure developments

3.1. The first IB-specific developments have been the inclusion of existing IB portlets into the portal. These portlets were modified slightly from the existing portlets in that they have been separated as much as possible into distinct units, rather than as a complete package. The intention is to begin testing these portlets once the backup facilities have been implemented.

3.2. An additional requirement for the Hosting server will be the creation of the Web Service layer corresponding to the IB project's technical requirement for such a layer to act as an interface to the Grid.

4. References

1. ?
8. http://www.oucs.ox.ac.uk/hfs/