SAKAI EVALUATION EXERCISE

Rob Crouchley and Adrian Fish
e-Science Centre of Excellence, University of Lancaster

Rob Allan and Dharmesh Chohan
CCLRC e-Science Centre, Daresbury Laboratory

e-Mail: `r.crouchley@lancs.ac.uk` or `r.j.allan@dl.ac.uk`

September 28, 2004

Abstract

This report summarises an evaluation of various aspects of the CHEF, Sakai and other related projects that may play a significant role in establishing a Virtual Research Environment (VRE) for the UK.

We evaluated the ease of administration, the feasibility of making existing tools available via Sakai/ CHEF and the feasibility of extending the functionality of the Sakai/ CHEF framework. The work carried out built on e-Science activities at Lancaster and Daresbury, which include software development and hardware deployment for collaborators in the physical sciences, substantive e-Social Science research and social science training and awareness-raising. It built upon the existing collaborations that the project team have with each other and with other university groups in the UK. It is now using existing mechanisms to disseminate the outcomes of the work. Throughout the period of this evaluation we have received clear expressions of interest in working with us from the developers of Sakai (http://www.sakaiproject.org). This included discussions and presentations at the Sakai Developers Workshop, Denver, USA, 23-27th June 2004.

The evaluation exercise was greatly facilitated by Lancaster deploying a CHEF implementation (`http://redress.lancs.ac.uk/cheflogon.html`) from day 1 to use as a collaboration tool for developers based at Lancaster and Daresbury.

This report summarises the outcomes and contains the conclusions of the evaluation which covered:

1. Comparing Sakai/ CHEF with Alternative Frameworks for VREs;
2. Establishing the ease of administration (EoA) of Sakai/ CHEF for a VRE;
3. Establishing the feasibility of making existing VRE (Grid) components available via Sakai/ CHEF;
4. Establishing the feasibility of extending the functionality of Sakai/ CHEF particularly to use Web Services for distributed development and deployment.

This report is available on line at `http://redress.lancs.ac.uk` and `http://www.grids.ac.uk/ReDRESS/sakai.pdf`.

Contents

1 Introduction

Our primary aim is to evaluate the appropriateness of the Sakai/ CHEF frameworks for the development of Virtual Research Environments (VREs) within the UK education, digital information and research communities. This report should help JISC to formulate the requirements of VREs. The relationship between, and capabilities of, the CHEF/ OGCE and Sakai frameworks is explained in Section 5. Another JSR-168 compliant framework, GridSphere, is also described for completeness as it has a certain popularity for Grid projects in Europe.

A broad definition of a VRE was provided following some debate by the JISC JCSR VRE Working Group [**?**]. It encompasses some, but not all functionality of:

e-Research: access to research tools, personal information, project-related issues;

e-Collaboration: contact with and working with researchers in the same or related fields;

Digital Information: access to relevant resources to provide background and supporting evidence for research and trainig;

e-Learning: components relevant to awareness and training. We assume that more research-related learning will be self-motivated and guided by example rather than managed course-work. Assessment tools are therefore igored;

e-Management: project and financial management in a research environment;

e-Authoring: authoring of all project and related research oriented material, often requiring input from partners and review by peers and chamions;

e-Publishing: publishing in appropriate format of research outcomes. Encompasses the peer-review process;

e-Leisure: BBC news, weather, current affairs, finance, local issues and events.

CHEF, the underlying technology which was the initial the focus of this review, is an open source, freely available framework designed to provide useful collaboration tools, such as a chat facility, discussion boards, shared calendars and file sharing in the context of a course or tutorial 'worksite'. It currently provides analogues of the most popular features of commercial groupware offerings such as IBM's Lotus Notes. Jetspeed and uPortal are open source, freely available portal servers that allows portal content to be imported, or linked to, by an administrator and then flexibly configured by users, thus empowering the users in the development of their own collaborative environment. The Sakai project aims to bring CHEF and uPortal together and then add more VLE facilities. It is in fact designed to encourage open collaboration and the sharing and contribution of new tools by a large and growing community of users, and this appears to be appropriate to the needs of the UK research community. Other generic portal engines are listed in Appendix H.

Virtual Research Environments, VREs, by their very nature will continue to evolve. It is important to make sure that our existing resources, services and applications can be made accessible in emerging standards-compliant frameworks and can be supported in the long term. A part of the funding requested for this evaluation was for a UK subscription as early adopters to the Sakai

Programme, the SEPP programme. There are obviously good financial reasons for moving to an open source platform for VRE delivery and collaboration, such as Sakai/ CHEF. The software is free and works on various platforms, being 100% Java code on the server side. There is thus no tie in to specific hardware, and the system will scale well financially with no licensing costs for extra server nodes etc. There are good logical and technical reasons also. Sakai will be completely open source and open architecture, so institutions will be able to customise existing, or add new, Java JSR-168 compliant portlet codes to connect to a legacy system running on their campus, e.g. to use existing project services such as databases, evaluations and timetables. Sakai will be configurable to use various different database management products, both commercial and open source, thus avoiding RDBMS lock in. At Daresbury an Oracle 9i RAC meta-data server and SRB are being used, but not all projects use Oracle. By exposing existing VRE services using the standard portlet API through a customisable portal framework, and re-using much existing code and sharing additional collaboration tools, we can achieve a large reduction in software development outlay and encourage closer community integration. Of course, this reduction has to be offset against the potentially increased need for software developers to throw away their bespoke solutions and adapt or interface to legacy systems so that they can co-exist with a VRE framework. If they do so however we can achieve a portable and maintainable solution.

This evaluation has addressed the following areas:

1. Comparing Sakai/ CHEF with Alternative Frameworks for a VRE. WP 1 is reported in Appendix A;
2. Assessing the Ease of Administration (EoA) of Sakai/ CHEF for a VRE. WP 2 is reported in Appendix B;
3. Establishing the feasibility of making existing VRE (Grid) components available in Sakai/ CHEF. WP 3 is reported in Appendix C;
4. Establishing the issues involved in extending the functionality of Sakai/ CHEF particularly to use Web services for distributed development and deployment. WP 4 is reported in Appendix D;
5. Developing a *Roadmap for a UK Virtual Research Environment*. There is now in a separate report from the JCSR VRE Working Group [29].

The deliverables of the project summarised in Section 5 and the appendices of this report are:

1. Evaluation Report part 1: Technology Survey (WPs 1,2);
2. A Review of the Issues for Building Standards Compliant Portlets (WPs 3,4);
3. An Assessment of the Potential of Sakai/ CHEF as a Platform for Customised Portals, e.g. ReDReSS, NCeSS, e-HTPX, e-Minerals and NGS (WPs 3,4);
4. Evaluation Report part 2: Developer and User Feedback (WPs 2-4);
5. Software Template for Sakai/ CHEF Institutional Adapters (WP 2);

6. Software Template for GRID tool wrappers for use in Sakai/ CHEF (WPs 3,4).
7. *Roadmap for a UK Virtual Research Environment* [29]

Our work has been written up in this report and also included in a couple of papers to the 2004 e-Science All Hands workshop, see References [30, 31]. An interim report was submitted to Nicole Harris of JISC, 3/9/04.

It is clear that, whilst we have attempted to be inclusive in our survey and review, such work is time-bound and never complete. We apologise if we have omitted to reference the full set of appropriate tools and beg their authors to contact us with more information.

1.1 Background to Portals and VREs

Early portals were very application-specific, stovepipe solutions with nothing re-usable. Examples include the work of Larry Smarr, who coined the term “Grid” in 1997 and started a portals project at NCSA. The Information Power Grid portal was started in 1998 at NASA. In 1999 Mary Thomas had a group developing portals at SDSC for the NPACI Grid, IPG was still making progress and GPDK, the Grid Portal Development Kit was started at NCSA. These were components of the US Grid Portals Project. There were also Geoffrey Fox’s DoD portal and the NCSA Bioportal.

At the *Portals and Portlets 2003* workshop [28], Dennis Gannon presented the “big picture” developed within the GGF Grid Computing Environment research group with particular reference to work at University of Indiana. Other material describing portal activities of the GCE-RG can be found from the GGF Web sites [32, 33]. The portal software stack was shown as four distinct layers:

1. Grid Portal – with the ability to launch, configure and control remote Grid application instances, possibly via a factory service;

2. OGSA layer – generic services such as:
   - registration and name binding
   - policy
   - security
   - data management
   - reservation and scheduling
   - event and messaging
   - administration and monitoring
   - logging
   - service orchestration
   - accounting

3. OGSI (or other) layer to link to infrastructure;
4. Resource layer with link to remote compute and data servers.

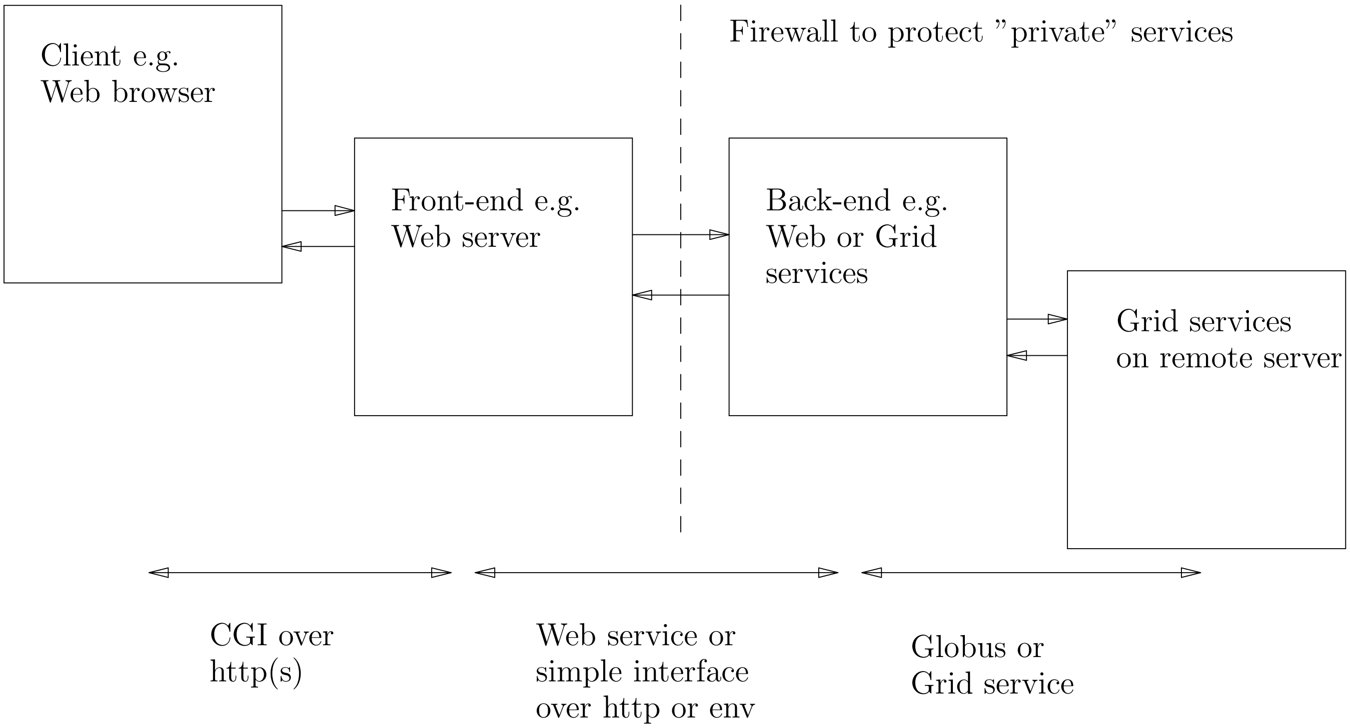

Figure 1: Four-stage Tool Architecure

Figure 1 shows a simplified picture of a fairly typical architecture for Grid-based tools as used in HPCPortal from CCLRC [8]. Whilst some of the stages may be merged (typically stages 2 and 3), logically the architecture comprises:

Client: client tools and user interface. The latter may be a Web browser, GUI, drag 'n' drop environment, script or programming library toolkit. Typically there will be a firewall between the client and the front-end server so CGI and/ or web services will be used through port 80 or port 8080 via a Web cache;

Front-end server: some services may be configured on other ports which could complicate the picture. It may also be desireable to have an internal firewall to protect "private" services, e.g. access to the real databases and computational resources. For this reason a front-end server acts as a gateway. If Web services are being used, it may validate the requests and it may also implement some form of primary access control;

Back-end server: the "real" services are implemented on the back-end server which is only sent requests following the primary access control and filtering. The back-end server may implement some simple functionality for the users and also carry out important operations such as session and state management. If no additional firewall is required front and back end may be merged using a simple and fast interface. Some development tools such as WebSphere facilitate this approach;

Remote resource: accessed from a back-end service by a remote procedure call using Globus or other Grid middleware such as LSF, Condor or PBS linked into the back-end services or a Grid service interface. They are not typically exposed directly to the end user. However, for anonymous services (e.g. registry lookup) a simple Web service will suffice.

This architecture has been found to be effective in practice and permits trade-offs between security, configurability and performance.

People are currently defining what types of services are needed in an Open Services Architecture [44, 54, 64]. The Grid could be defined as a collection of distributed services and a portal as a conduit to these. Solutions are built on components, where a component is a thing defined by (1) a public interface (2) semantics and a "standard" behaviour. Using a portlet framework's user interface, each component/ service can have its own interface, a portlet to uderlying services. These are "plugged" or "tiled" in a customisable fashion onto one or more views in the portal.

This picture is typical of a family of Grid portals now providing rapid development and extsibility features. A user interacts by logging on and implicitly creating a "context" or "session" which comprises his/ her recently used objects, files, jobs etc. These are represented by a set of tools for remote access and Grid services each associated with a unique portlet. Users can select the portlets they require for a particular job and to customise their portal workspace.

The "big picture" from the JISC Common Information Environment (IE) portal activities was presented by Chris Awre. For other related information see `http://www.jisc.ac.uk/index.cfm?name=programme_portals`, `http://www.jisc.ac.uk/ie/` and `http://www.jisc.ac.uk/index.cfm?name=strat_ieds0105_draft2`.

The IE programme wants a small number of sophisticated interfaces for increased ease of access and use of JISC resources. The IE technical architecture was developed by UKOLN. It encompasses: content providers (institutional or commercial); fusion layer (brokers, aggregators, catalogues, indices); authentication/ authorisation (currently Athens); service registry, preference services, metadata schema registries, resolvers, institutional preferences, terminology services.

A useful description of the JISC Information Environment architecture can be found on-line at `http://www.ukoln.ac.uk/distributed-systems/jisc-ie/arch/`. Portals are just a small part of this impressive programme of work.

There is a strategy for seamless and integrated access to digital collections. This includes connections to museums, archives and e-Science for learning and teaching. There are distinct building blocks with open standards.

The development programmes include:

- Content submission and disclosure
- Shared services
- Portals and fusion
- Presentation
- Service providers

For the JISC Information Environment portals provide the following functionality:

- Discovery tools, access search functions on (commercial) providers
- Lightweight, source remote content
- Common means of access via open standards, OAI, Z39.50
- Presentation of information
- User access via different routes: dedicated Web site, embedded functionality in known Web env
- Could be a PDA in future

We note that in this sense Google, which is probably now the most widely used "killer" search-engine application on the Web, is not a portal, but a "gateway", it only tells us where the information is, it does not aggregate it, although its specialised functionality, such as image searches, is growing rapidly. More information on these and related ideas is provided in an FAQ at `http://www.jisc.ac.uk/index.cfm?name=ie_portalsfaq`.

1.2 Portlet Standards: JSR-168 and WSRP

1.2.1 JSR-168

The Portlet Java Specification Request JSR-168 lays the foundation for a new open standard for Web portal development frameworks. Portlets define an API for building atomic, composable visual interfaces to Web content or service providers. A portlet provides a "mini-window" which can be placed within a portal page. Multiple portlets can be composed in a single page by the developer or user through the framework. Portlets extend servlets, the idea being to reuse common method signatures.

The Java portlet API JSR-168 emerged from the Java Community Process (JCP) principally from the Apache JetSpeed portal project in April 2001. JCP is an open process involving the organisation of Java developer institutions with the remit to develop and revise specifications and reference implementations for the Java platform. JSR-168 seeks to provide a portlet abstraction together with a portlet API thus enabling inter-operability between portals and portlets.

The Java portlet interface standard JSR-168 was ratified in August 2003, shortly after our workshop. Sun is pushing forward with a Java portlet API and has 18 application-server vendors supporting it (for more information go to the Web site `http://www.jcp.org/jsr/detail/168.jsp`). Some examples of "open" portlet frameworks are Jetspeed from Apache, uPortal, LifeRay, jPortlet (an open source project) and GridSphere from the EU GridLab project. See Table H in Appendix **??**.

1.2.2 WSRP

WSRP, the Web Services for Remote Portlets API defines a standard for interactive, user-facing Web servics that plug and play with portals.

WSRP emerged from the world of Web services which uses WSDL to publish service information after it was taken by an OASIS technical committee (which also reviewed the proposed JSR-168 standard). OASIS is the *Organization for the Advancement of Structured Information Standards*, a world-wide consortium that drives the development, convergence and adoption of e-Business standards. WSRP was combined with input from the proposed Web Service for Interactive Applications before a final specification was agreed in late 2002. Following a public review in May 2003, WSRP was also adopted as a full OASIS standard in the third week of July 2003.

WSRP seeks to establish a portlet abstraction with a WSDL description for how to publish, find and bind to remote WSRP-compliant services with metadata about related things such as security mechanisms, billing, etc. It is now a platform-independent bridge leveraging the language-independence of Web services and interfacing to the Java portlet API JSR-168, C# .NET API, and other WSRP implementations on J2EE or .NET. If a portlet is written to the portlet API it should be possible to publish it via a container as a WSRP service to a UDDI registry and import it into another portal using a portlet proxy.

The portlet JSR-168 specification handles the presentation end of information enabling re-use of portlets in different containers. In order for containers to present their contents as services IBM is taking the lead on WSRP, the Web Services for Remote Portlets standard (also ratified in August 2003), which is based on XML and Web services. WSRP will allow portals to retrieve content from other portals via their portlet containers and other data sources. The use of WSRP and JSR-168 in a typical portal architecture is shown in Figure 2. More information on WSRP can be found at `http://xml.coverpages.org/ni2002-01-21-b.html`.

1.2.3 JSR-168 vs. WSRP

JSR-168 and WSRP work at different levels. JSR-168 specifies the interfaces for local portlets into their container (e.g. Jetspeed, uPortal) whilst WSRP specifies the interfaces for accessing portlets across portal frameworks, i.e. from remote containers. These have to be aligned using the same notion of the objects and ability to instantiate portlets locally and remotely. Details of the portlet API have to be exposed via WSRP in order to do this. The use of WSRP and JSR-168 in a typical portal architecture is shown in Figure 2.

It is important to note that you don't need a portlet server to serve WSRP compliant content. It can be served as any Web service would be (e.g. using Apache with C gSOAP, Perl SOAP::Lite etc.). This avoids content providers having to tackle issues of installing additional software like Tomcat. This is also an important aspect of wrapping more traditional applications for presentation via a portal framework.

1.3 Content Aggregation: WSRP, RSS, P2P, URL etc.

A number of documents are available describing uses of aggregation protocols applicable to portal environments. These have been used in the evaluation exercise. In particular one describing the use of WSRP in Web service registries such as UDDI is of interest because it indicates how remote portlets can be published and discovered in a similar way to other services. See `WSRP and UDDI paper`.

`this section needs completing`

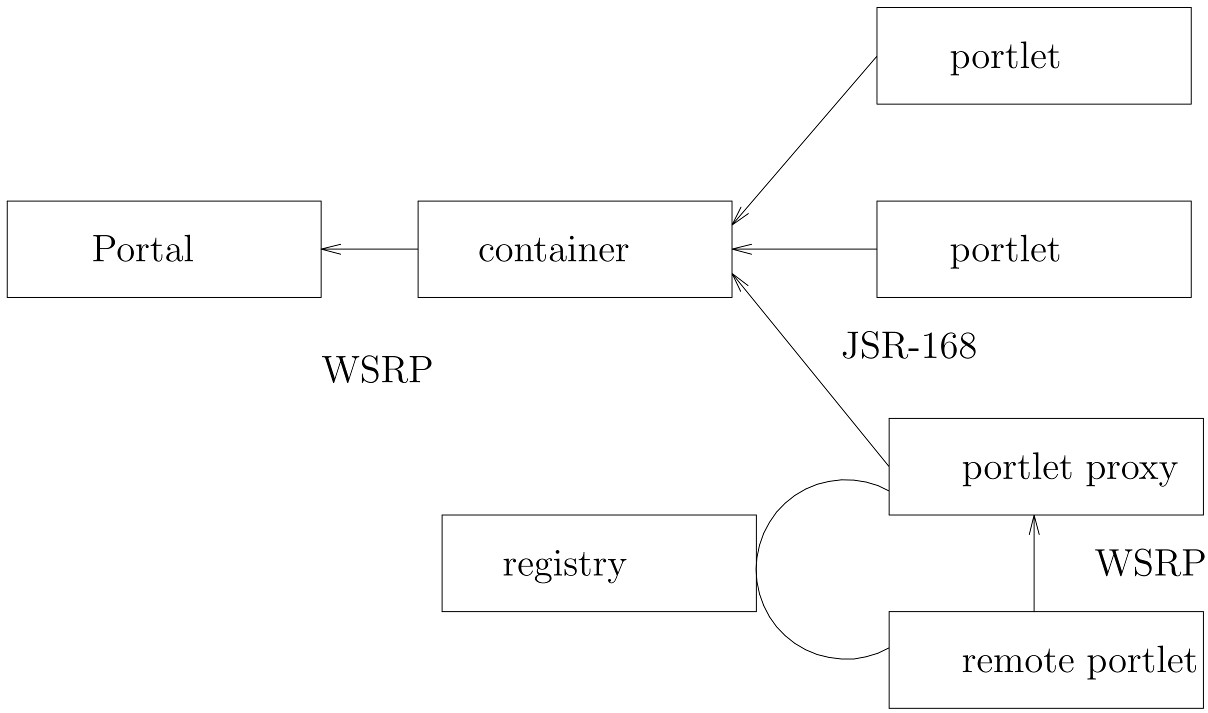

Figure 2: Relationship between WSRP and JSR-168

2 Portals and Portlets 2003

This section discusses the background to the international workshop held in summer 2003 to discuss these ideas and various issues concerning the deployment of portals, sharing of services and user perceptions. It focusses in particular on the impact of the JSR-168 portlet standard on portal development and the ability to share underlying tools and services. The workshop *Portals and Portlets 2003* was held at the National e-Science Centre, Edinburgh, 14-17th July 2003.

The agenda and presentations from the workshop, including the majority of presentation materials, are available on line at the NeSC Web sit `http://www.nesc.ac.uk/action/esi/contribution.cfm?Title=261`. Links to portal projects are included in the rest of this report.

In spring of 2003 it was felt timely to organise an international workshop focussing on portals and portlets for e-Science. This had been discussed over the previous year among members of the GGF Grid Computing Environments Research Group. With strong interest from Rob Allan and Mark Baker in the UK, Jason Novotny and Michael Russell in Germany, Massimo Cafaro in Italy and Mary Thomas, Charles Severance and Dennis Gannon in the USA, it was eventually decided to hold a 4-day workshop in the summer of 2003 hosted at the National e-Science Institute in Edinburgh.

The workshop did indeed prove timely for several reasons. The GridLab group had recently finished the project's new GridSphere Java portlet framework and the 4th day of the workshop acted as its first major tutorial. Discussions had started between groups in the UK e-Science and JISC portals programmes with a recognition that the two groups had much to learn from each other. Indeed Chris Awre was able to bring in a number of speakers from the JISC Information Environment community. Finally, the Java JSR-168 and WSRP standards had been proposed to W3C for ratification, which actually happened in the week following the workshop.

There have been major changes in the UK e-Science programme since the workshop and even closer links have formed with JISC. The easy delivery of access to both Grid resources and information services to end users involved in multi-disciplinary research and training is more important than ever. There have recently been a number of discussions to consider how active UK groups can best collaborate to consolidate and extend best practice and functionality of existing portals. Some suggestions are provided in the workshop conclusions as follows.

It is clear that there is a lot of established expertise and momentum in the UK to develop Web-based portals for a variety of purposes. We have established strong links and potential collaborations bridging the UK, USA and other European developers and now also bridging the e-Science and JISC communities. It is important to continue this work and lead identified areas which will be taken via the Global Grid Forum research and working groups as input into the definition of standards leading to software sharing.

There are a number of UK groups already actively developing re-usable resource and Grid-based portals and portlet services. There are actually a large number of other groups developing informational, institutional, e-Learning and awareness and training portals. In many cases the frameworks being developed could be shared and the underlying portlets and services could be re-used if an appropriate architecture and standards were adopted. This implies a portlet framework plus a message-based service approach rather than a methods-based approach (advice from Geoffrey Fox, 26/2/04).

Based on our experiences and outcomes of the NeSC workshop we recommended the following:

1. Portal services development should be recognised as a strength of the UK middleware initiatives, e.g. as noted by Fox and Walker their *UK e-Science Gap Analysis* [34];
2. The active UK groups should collaborate. These include developers from the three JISC pillars: support for research, teaching and learning and the Common Information Environment plus the e-Science and Particle Physics Grid user communities;
3. UK developers should continue to work with the GGF Grid Computing Environments research group and the American Open GCE and Sakai projects plus other relevant international fora;
4. Developers should save effort by sharing services and methodologies and customising the existing Web-based presentation layers for delivery to all end-user projects;
5. The UK should be active in defining, classifying and developing portal services for input into the OGSA space between (possibly changing) infrastructure and application layers;
6. A range of toolkits (thin clients, portals, scripting languages, GUIs etc.) should be developed to extend and simplify access to Grid resources and information systems leading to the eventual emergence of one or more interfaces to a Virtual Research Environment.

It seems that there is a good possibility of linking future activities in the areas we have described more closely with developments in other sectors, including institutional and learning and teaching arenas. There does appear to be a convergence in technology and benefits from sharing some of the tools being developed. To this end we looked to JISC, the Joint Information Systems Committee, to fund a small number of evaluation studies to examine issues of re-usability and inter working of components in the emerging portlet frameworks. This a report of the Sakai Evaluation Exercise.

3 Sakai SEPP Developers' Conference

The authors of this report attended the first Sakai Developers' Conference in Denver, Colerado 23-35th June, 2004 which was also the inauguration of the Sakai Educational Partners' Programme. Other UK participants included Sarah Porter (JISC), John Norman (University of Cambridge) and Ian Dophin (University of Hull). The majority of the participants were SEPP members and the UK had subscribed to this programme for evaluation purposes.

Over 175 people from 45 institutions attended tutorials, presentations, meetings, and BOF sessions. The enthusiasm for the Sakai project was leavened with some hard hitting technical questions and concerns for the future. The Sakai Board of Directors, led by Joseph Hardin, emphasised that SEPP has a vital role to play as we move into Sakai 2.0 development and beyond. A new development process based on incubation in the SEPP Discussion Groups that spin out new projects was presented.

This first conference included a number of important presentations from the Sakai Core Team and others from SEPP members who have already begun to develop and port educational tools into the Sakai framework. We here focus on points raised in the core presentations. Most of the presentations shown are available on the SEPP Community Site. A Syllabus article covering the conference is also available at: `http://www.syllabus.com/article.asp?id=9635`.

The Sakai Tools Team has released three documents related to Sakai functionality. The first document covers Samigo (SAAM, Navigo) and its feature set:

`http://chefproject.org/access/content/group/1075771392979-922/SEPP_Conf_June_2004/Samigo\%20Marketing.doc`

A second document covers general Sakai functionality:

`http://chefproject.org/access/content/group/1075771392979-922/SEPP_Conf_June_2004/Sakai\%20Fall04\%20mktg.doc`

In addition, here is a link to the 15MB PowerPoint presentation that explains each gap in great detail as was promised at the SEPP Conference:

`http://chefproject.org/access/content/group/1075771392979-922/SEPP_Conf_June_2004/Gap_Prioritization_Revised2.ppt`

SEPP Requirements Group: The requirements group met at the SEPP conference with the BOF content and authoring group and later on its own. Some of the key areas that it identified include:

- Get more institutional input into the high-level build list (match institutions to tools) to contribute

to the 3.0 product;
• Matchmaking tool to encourage *ad hoc* alliances to form around tool development and common areas of interest;
• Promote a common specification and development process;
• Provide links to the core teams and ease communication efforts;
• Provide a common repository for discussion group requirements and specification documents ;
• Provide a gap analysis between what is currently in 1.0 and 2.0 and the partner requirements;
• Encourage partners to become more forthcoming with what they are doing via a "natural English" suggestion area;
• Provide a process for requirements feedback for projects to encourage a broader understanding of needs across institutions;
• Keep a running list of current projects, participants, and project information.

Some thoughts were shared during the conference regarding project management within *ad hoc* alliances:

• Mutual interest and timing (alignment) is critical;
• Create mini-projects using the SEPP process template;
• A project leader should be assigned from one of the participating departments who will be responsible for staff resources across institutions and management of project;
• Projects need to have visibility via Sakai forums.

We in the UK are now in the process of forming a Sakai Alliance which will work in this way.

Challenges identified:

• Maintaining openness and communication while staying on task;
• Ensuring visibility of projects;
• Maintaining a process where people understand where and how they can participate (especially if the project is underway).

The requirements group needs volunteers willing to put in some time and effort to creating the necessary matchmaking tools, compiling the build list, and performing a gap analysis.

SEPP Development Group. On Wednesday, Mark Norton gave an overview of the Sakai architecture and how to write tools using the Sakai framework and TPP document. This presentation can be downloaded from:

`http://chefproject.org/access/content/group/sakai-educ/Denver\%20Conference\%20Presentations/SEPP\%20Conf\%20-\%20Development2.ppt`

Since the conference, he has take the section on cloning a tools and written it up into a tutorial document that can be found at:

`http://chefproject.org/access/content/group/1085757908747-8684/Tutorials/writing-sakai-tools.pdf`

This tutorial contains step-by-step instructions on cloning an existing Sakai beta tool (sakai-module) and modifying it to be a new, standalone tool. While this is a very simple "Hello World" type application, it can serve as the basis for creating new Sakai tools.

The primary contributions to the Sakai tools project include the following.

- The first contribution will be the previously described Technology Portability Profile that will describe the integration of OKI's OSIDs, a user interface abstraction for localization and the new JSR-168 portlet specification. This Profile integrates the successes and lessons of Michigan's CHEF interoperability framework, and JA-SIG's uPortal, and describes a common path forward for their respective developer/user communities.
- The second contribution is that Michigan, Indiana, MIT, Stanford, and uPortal will all license their considerable intellectual property and/or experiences with large scale application software (e.g., Course Tools, Work Tools, Navigo Assessment, Oncourse, Stellar, uPortal, OneStart, Eden Workflow, CourseWorks, etc.) into a re-factoring of best features. This will include an enterprise-scale course management system, distributed research collaboration tools, and an enterprise services portal (described in more detail below), and others that have been conformed to the Technology Portability Profile. All Sakai tools will be both modular and also pre-integrated to work with each other. The software will be made available to the world at the same time via an open source license.
- The third contribution overcomes the barrier of institutional timing by synchronizing the development and implementation clocks of four complex institutions: Michigan, Indiana, MIT, and Stanford. All institutions are committing to an initial implementation of the Sakai tools, as a campus-wide CMS and/or campus-wide enterprise Portal by 3Q05 when the tools are fully released. Synchronized clocks will greatly facilitate further shared developments in the years beyond the Sakai Project.

Secondary contributions are no less valuable and in the longer run may be more valuable to the educational community. They are described here as secondary contributions as they are premised on the actions of others who are not within the direct control of the Sakai Project Core participants.

These large scale deployments of common applications based on the TPP at four complex institutions will clearly demonstrate the real viability of open source code mobility for higher education. The synchronization and TPP also provide a common means for pulling together the considerable extended user communities of uPortal, OKI, and Michigan's CHEF Project;

The Sakai Project is also developing an Educational Partners Program that includes resources for community development, training, shared best practice, and early access the TPP and Sakai applications. The goals of the Educational Partner's Program include

- To involve a wider community in the Sakai Project's open source vision, economics, and innovation
- To mobilize distributed resources for development and support of Sakai tools
- To initiate a "market of tools/components" that will run in the Sakai framework
- To manage interaction with the Sakai Core development team
- To coordinate activities with other organizations, such as IMS or country-level agencies

- To build on the experiences of the JASIG, CHEF and OKI training and conferences
- To generally engage in Sakai community development of shared best practice

The education community will benefit greatly from a Technology Portability Profile that provides an open, non-proprietary, and fully articulated specification for interoperable software. Any institution or commercial entity can build to this Profile, thus helping all institutions integrate software from multiple sources as their timing may require.

The economics of software for the education community are greatly served by a proven set of pre-integrated, modular, open source applications that any institution can adopt incrementally or as an integrated set of tools. Adopting institutions of any size or technical sophistication will be freed from annual licensing fees for Sakai Project software. Thus, money that would have been spent on licensing and integration can now be devoted to other academic priorities for an institution.

The second evening of the conferenced comprised an "All Hands" style demonstration session exhibiting the range of tools which are now being ported to Sakai by the SEPP members. This list was impressive:

Tufts	VUE
Virginia	Fedora
Rsmart	OSPI
Wisconsin	Virtual Dig
Wisconsin	Lesson Builder
Northwestern	Annotation tool
Indiana	Sakai Assessment Manager
University of Toronto	TILE and Atutor
Northwestern	Project Pad
University of Michigan	Sakai
Indiana	weBWorK
Knowledge Media Laboratory	KEEP
Tufts	TUSK
Harvard	iSites and Videotool
UC Davis/ Princeton	Image Management System
SUNY	Learning Environments
OKI OSIDs	Digital Repository

Our questions to the developers and SEPP members included:

- OKI
- Extending the API
- Contributing
- uPortal involvement
- the SEPP partners
- UK role: WSRP and tools

`need to finish this section`

3.1 Ongoing work of the SEPP.

`need to finish this section`

4 Role of Portals in a Virtual Research Environment

There are a number of fundamental research questions which must be answered before the large-scale deployment of a Virtual Research Environment in the UK. Our Sakai evaluation and additional work with the JISC JCSR VRE Working Group have lead us to identify and suggest responses to many of these issues. Portals are just one, but an important, component of a VRE.

4.1 The Need for a VRE

e-Science is a new paradigm of research, often characterised by a "deluge" of data analysed by massive distributed computing power. e-Science research collaborations are frequently large, distributed and multidisciplinary involving hundreds of institutions across the globe. Grid technology, emerging in response to these challenges, is enabling exciting possibilities for better research, even creating new disciplines like astro-informatics. In this context, a wide range of national and international initiatives are under way.

The concept of e-science is now broadening and evolving into e-research generally, to encompass the social sciences and the arts and humanities. At the same time it has to be recognised that different communities are at very different stages in their awareness of the new technologies: thus the current needs of a large international scientific collaboration are likely to be much more complex than those of the lone humanities researcher, wishing to collaborate more effectively with a handful of colleagues world-wide in the same field of interest. In our thinking we have tried to keep the whole range of requirements in view.

At the high end, the new developments are making the process of carrying out research more complex and demanding. The aim of a Virtual Research Environment (VRE) is to help researchers manage this complexity by providing an infrastructure specifically designed to support the activities carried out within research teams, on both small and large scales. JISC has recently been allocated £3.2 million as part of the Comprehensive Spending Review to develop a VRE.

The challenge is to create and sustain an infrastructure, ideally usable on a routine basis by researchers from all disciplines to enhance their productivity and effectiveness. Meeting this challenge is a task for those building the infrastructure, its potential user communities, the institutions to which users belong, the organisations which fund research and other stakeholders in the research process. These developments should not happen in isolation but will need to interwork with other components of the infrastructure being provided by JISC, Research Councils and HEIs themselves such as learning environments, digital libraries and national research facilities.

It is unclear, a priori, what type of framework a VRE should adopt, on which technologies it could be based, how it can be developed sustainably and how usability and take-up can be ensured. This document outlines a roadmap for developing a VRE. The target user community is all those engaged in research.

Disciplines, and communities within disciplines (especially in non-scientific subject areas), will have to identify the possibilities for them in the technology; may have to overcome cultural obstacles to collaboration; and may need training in relevant skills. Associated legal issues will need to be understood and clarified, and formal and informal codes of practice updated, to reflect understanding of novel forms of collaboration.

Locally, institutions will need to understand the business case for supporting research collaborations and how they can be reconciled with continuing institutional competition. Wider impacts will be felt through changes in scholarly communication and in the complexities of managing and sustaining long-term open access to data for reuse.

Additional background material and rationale for creating a VRE is contained in a report entitled *Building Collaborative eResearch Environments*, compiled for JISC by Andrew Cox, Department of Information Science, Loughborough University [41]. This summarises the proceedings and breakout group discussions from two workshops held in March/ April 2004 at Edinburgh and Warwick universities. The report also contains a SWOT analysis. A brief summary of some of the main recommendations from these workshops is included as Appendix A of the report.

Background material from the UK e-science Grid, compiled by the Architecture Task Force, is contained in the report [42] *UK Role in Open Grid Services Architecture* by Malcolm Atkinson et al. This vision is being realised through the work of the Engineering Task Force and is be introduced onto the production National Grid Service (including the JCSR-funded computing and data clusters) during 2005.

Related JISC work on e-learning frameworks and tools is described at `http://www.jisc.ac.uk/index.cfm?name=elearning\_framework` [43]. Work on developing a distributed architecture as part of the JISC Information Environment is described at `http://www.ukoln.ac.uk/distributed-systems/jisc-ie/arch/` and in reference [44].

4.2 Capabilities of a VRE

The nature of a VRE means that it is more realistic to describe it in terms of its intended capabilities rather than its component parts as the latter are likely to evolve over time, depending on contemporary standards.

In the views of the JCSR Working Group, a VRE should:

1. Support the processes of conducting research, including marshalling of resources, scholarly discourse and publication, and the creation and maintenance of collaborations, across disciplines, institutions and countries, including support for meetings and organisational processes.
2. Be based, as far as possible, on loosely-coupled, distributed, interoperable tools, rather than a monolithic piece of software.

3. Be designed to meet user requirements and address usability and accessibility, with appropriate evaluation mechanisms and benchmarks for new tool development.

4. Include modes of access which (almost) any user can download and install on their laptop/ desktop/ PDA/ home computer, with "servers" that can easily be installed by system administrators without specialist knowledge and national JISC-provided servers as appropriate, so that tools work "straight out of the box". Some tools will be integrated with domain-specific facilities (and vice versa).

5. Adopt and use appropriate open standards wherever possible.

6. Be secure and trustworthy. Hence the VRE components should interoperate with federated cross-institutional authentication and authorisation mechanisms.

7. Be accountable, by providing adequate logging and probity including supporting queries about provenance.

8. Be compatible with other widely used and deployed systems, including at least: web, email, instant messaging, SMS, Wikis and videoconferencing tools from lightweight desktop applications through to high-end videoconferencing via Access Grid. This means that the VRE should be accessible via web browsers and 3G mobile phones among other modes of access.

9. Support creation, sharing and curation of resources, through ease of authoring, publishing, discovery and access. This implies adoption of appropriate metadata schema and support for automatic generation of metadata. Resources to be described will include data, computation and potentially humans.

10. Be extensible with enhanced or new tools by any developer, through use of published standards and provided software development kits, software libraries etc. It should be as easy as possible to make existing software and services (e.g. e-print repositories, portals), including proprietary software, compatible with the VRE.

11. Be open source and standards-compliant wherever possible. The licensing of the tools should encourage and support improvements to the tools and development of new tools through open source development by the community.

12. Support tailoring of the environment by individuals or groups to reflect their interests and preferences.

13. Support the delegation of routine tasks to intelligent personal agents where the means to realise these exists, e.g. by incorporation into workflow processes.

4.3 Developing a VRE

Based on the background and examples summarised above, a VRE can be considered as a set of applications, services and resources integrated by a standards-based, service-oriented framework which will be populated by the research and IT communities working in partnership. The scope of the components needed to build this framework is further discussed with examples in Section 5.

Multiple domain-specific or community-specific gateways to the VRE will exist, in parallel serving the needs of different communities but achieving maximum synergy and cost-effectiveness by being based on a common framework which enables reuse of generic open-source components, referred to as services. The example of portals given above is just one example; lightweight programming libraries is another, permitting integration into “heritage” codes.

It is not the intention of JISC’s VRE programme to produce a complete VRE, but rather to define and help to develop the common framework and its associated standards and to encourage others to work within this framework to develop and populate VREs with applications, services and resources appropriate to their needs. The intention is to maximise the value and benefit of future investment in this area by both JISC and the Research Councils, to secure community contributions and to promote sustainability.

VREs must cater for a wide range of scale and complexity of research activities, from small research collaborations with a few partners to large teams with many partners in many institutions. Examples of the latter include projects within the current e-science programme. A balance must be achieved between meeting the needs of specific disciplines and developing capabilities of widespread utility. The wide variety of research activities means that a judgement must be made in each case on the appropriateness of including specific capabilities for the application in question.

In the long run, VREs will have to become self-sustaining within their user communities and service providers. They will therefore have to be seen to provide sufficient additional benefit to motivate this effort. Whilst VREs, gateways and resources will ”belong” to their user communities and will respond to and track these communities’ evolving requirements, it is expected that there will be mutual benefit in coordinating these separate VRE activities through a common framework with re-useable services and associated standards. JISC, in its closer relationship with the Research Councils, may continue to provide a suitable common reference point for this coordination. There is potentially a role for the Open Middleware Infrastructure Institute (OMII) in providing ongoing support for VRE middleware.

It is important that the VRE development activity has strong links with other related activities such as JISC’s VLE and MLE programmes, the research programmes and communities supported by the individual Research Councils, and international developments such as Sakai in the US. It will have many components and services in common with these.

4.4 A Service Oriented Architecture

A Service Oriented Architecture (SOA) is an approach to joining up services to provide integrated capabilities. It is a relatively new approach, but is rapidly gaining popularity because of the lower costs of integration coupled with flexibility and simplified configuration. This is becoming best practice for commercial distributed software development, see recent reviews e.g. [47, 48, 49, 50, 51]. An SOA builds upon the use of web services, the emerging industry standard for building and integrating distributed systems. The rationale for using an SOA in the JISC context for MLEs/VLEs is given in [43]. Other relevant projects worldwide are considering and indeed beginning to deploy similar approaches and architectures. One worth noting is Arda, the next generation framework for distributed analysis of Large Hadron Collider data [52].

Portal deployments have, until now, typically been monolithic with a rich set of tools, customisation possiblities and a database for content management. CHEF, OCE and Sakai fall into this category and are deployed as large Java jar files. At CCLRC the Integrated e-Science environment, IeSE, comprising services from HPCPortal, InfoPortal and DataPortal has already tried to break this mould [16]. Another more recent activity at University of Indiana is taking this a stage further and now using CHEF [53]. In Section 6 we propose an architecture for presentation of services throught portlets in an SOA which extends these ideas.

The following figures highlight some basic aspects of an SOA relevant to deploying a VRE with appropriate user interfaces such as portals, online commands, drag and drop desktops and programming libraries. A key aspect of the architecture is to maximise the re-use of common services and middleware including portlets.

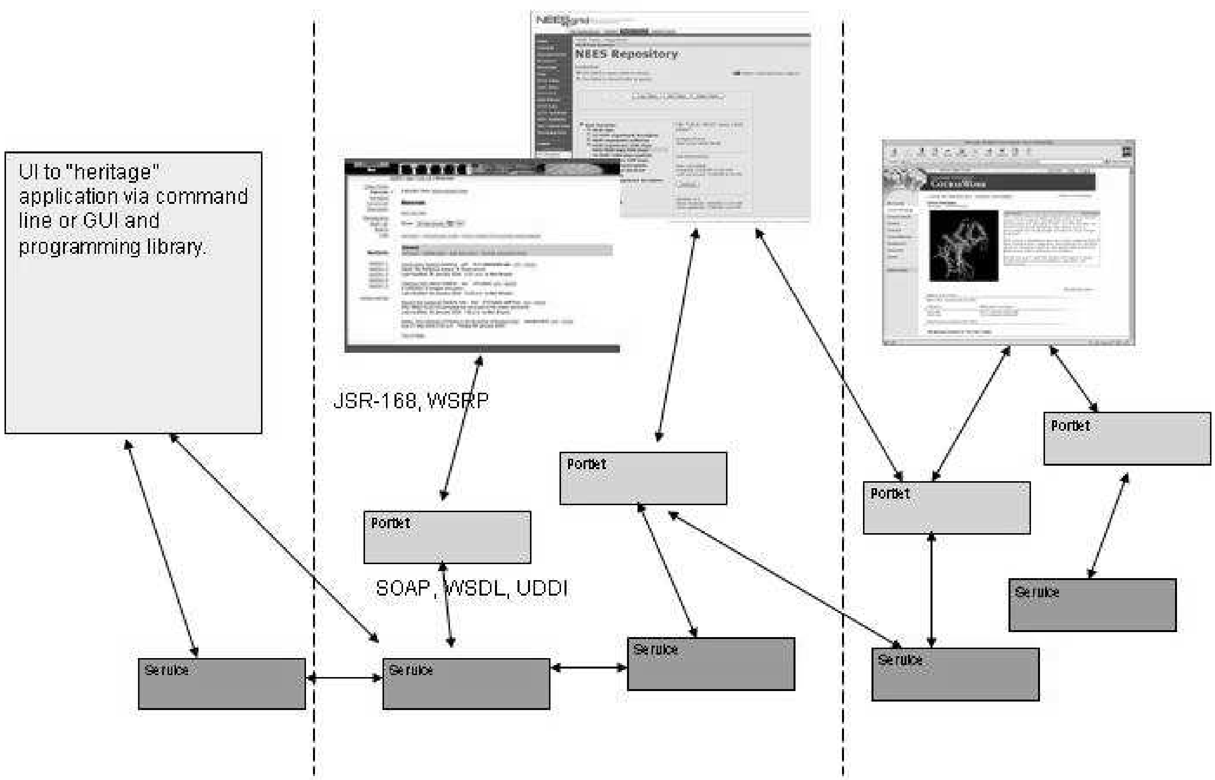

Figure 3: The GRID of Services

Figure 3 shows how an SOA approach would be of benefit in exposing a common set of services and middleware through a variety of user interfaces including Web portals employing the WSRP standard. It indicates how this architecture can be used to facilitate the horizontal aggregation that can occur for specific groups, e.g. the National Centre for e-Social Science (NCeSS) which is working alongside the Lancaster node for Quantitative e-Social Science (CQeSS) and the JISC/ESRC training and awareness programme ReDRESS, see `http://redress.lancs.ac.uk`.

An SOA clearly does not preclude also using portals or data warehouses, and is in fact agnostic about how the rest of the enterprise is configured, which is why it makes a good approach for a framework. In addition, because integration occurs in this fashion, it becomes a simpler task to replace the systems that provide services within the architecture or to look up new ones via a registry such as UDDI [55]. Because service consumers are configured to access a service without any knowledge of the system that provides the service, we can replace the underlying system without affecting systems dependent on its capabilities.

Figure 4 shows how services are used in a typical 4-layer architecture for portals and other client tools.

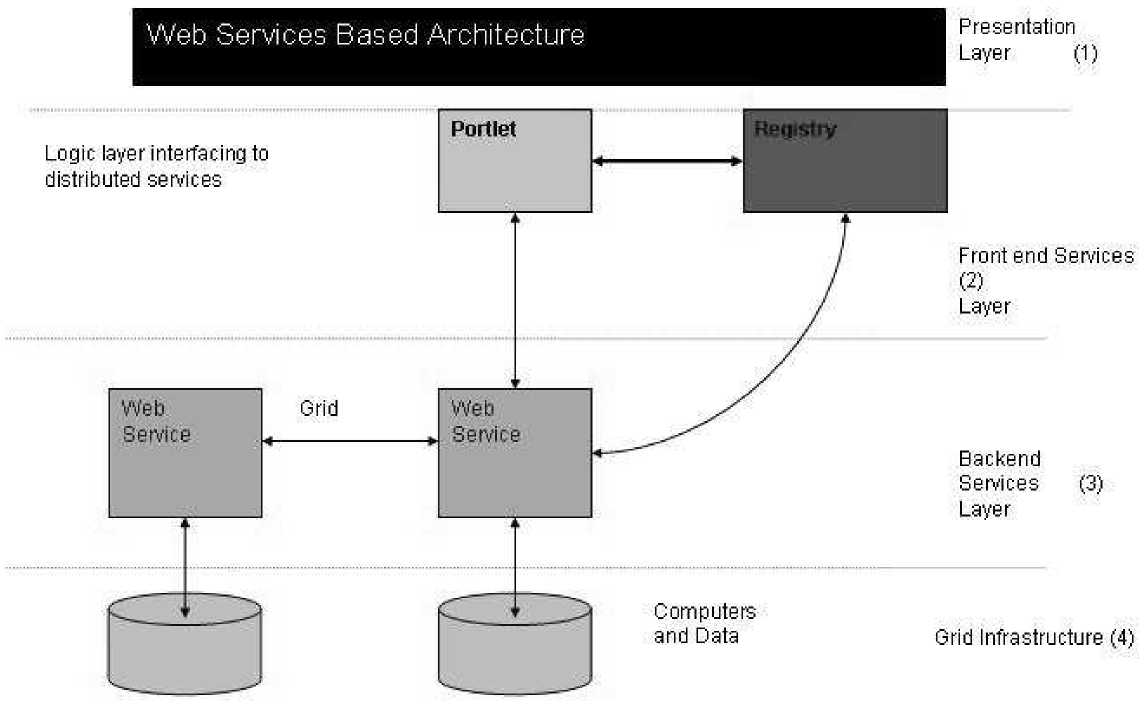

Figure 4: 4-layer Portal Architecture employing Web Services

Notes:

1. Presentation layer, e.g. uPortal, Jetspeed or similar integration and rendering framework. Tools such as Sakai/CHEF add value to this by providing a content management system and additional service interfaces.

2. Front end services, logic layer interfacing with (1) through the JSR-168 portlet API.

3. Back end services, accessed via web service or other well-known protocols: could be distributed, and/or could interact with other web services forming a Service Grid.

4. Remote resources on a Grid infrastructure, e.g. computers and data bases, such as the JCSR clusters.

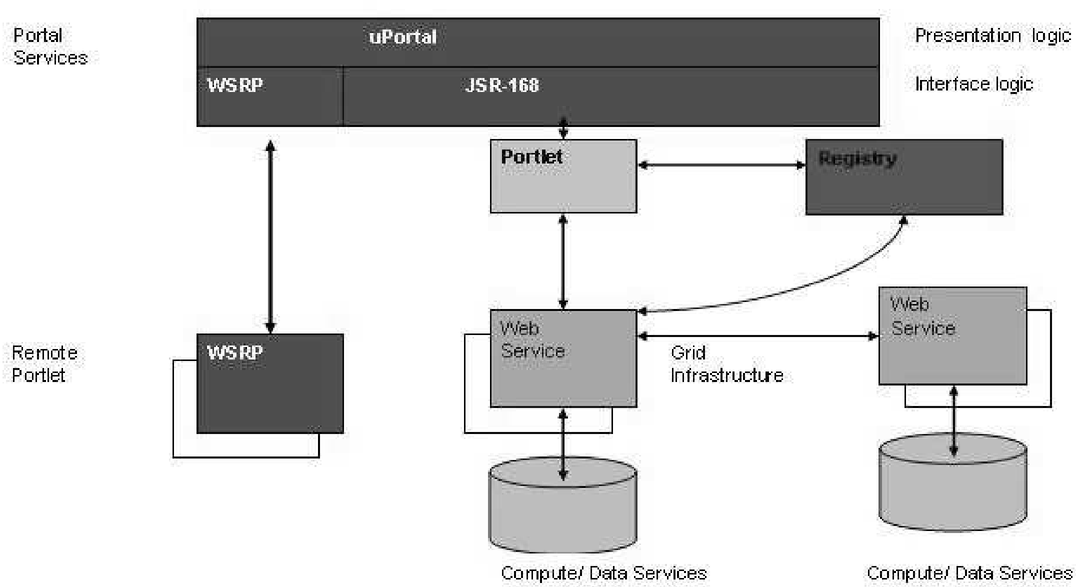

Figure 5: Portlets using Web Services

4.5 Technical Framework

This section briefly outlines technical aspects of a VRE which may indicate the scope of work required. This discussion arises directly from the identification of an SOA as the preferred architecture for the VRE and the community input which has led to the workshop recommendations. It is easy to be all-inclusive in a theoretical approach, but a more carefully chosen set must be suggested for pragmatic reasons. The following sections therefore attempt to suggest an initial preferred set.

Frameworks/ interfaces

Following discussions among interface providers, such as those writing 2nd Generation Portals and web services based middleware, we suggest adopting the following standards:

• JSR-168 (Java standard for portlet interface);
• Web Services (WSRP, WSDL, WS-I) which may extend to such things as WSRF in future (SOAP over HTTP is mainly implied as a request/response protocol but other delivery protocols may be needed, e.g. for large datasets, see below);

Frameworks should therefore be able to support these interfaces, e.g. in the case of portals Websphere, uPortal, Jetspeed, GridSphere etc. are possible candidates. For further discussion on JSR-168 and WSRP see [28]. There will be multiple solutions for access to the VRE and standards and inter-operability are key issues. In addition to frameworks such as portals, services must be able to be incorporated into heritage applications and GUI interfaces such as those developed and maintained by the long-standing Collaborative Computational Projects (CCPs), which are a unique contributor to UK research output.

Generic services

These generic services are an abstraction of actual services, some of which might be made available via the VRE. Appendix G provides a fine-grained, but still evolving, list of services from areas such as:

• AAA, Grid Services, Semantic Services, Resource Discovery, etc.
• E-Collaboration;
• Support and Management Services.

This is an attempt at an initial classification and description of these services based on work of other groups such as JCLT, ETF and GGF. Further work is required to identify how services can be broken into methods (functionality) and if appropriate where existing middleware provides such functionality. See `http://www.grids.ac.uk/ETF/public/WebServices/classes.html`.

Real Resources

Services must clearly be mapped onto real resources to be of value. In Appendix F we have listed some resources and services already available to the UK academic research community which could be accessible. This list is also not exhaustive, but we should identify the main resources and facilities which are likely to be available via the VRE. It may be the case that some of these resources need extra effort to provide interfaces or full availability. Gaps may also be identified as we analyse these resources and services.

Requirements and Use Cases

Development of requirements is an ongoing process and the awareness days and demonstrations which have taken place so far have informed this document. Following further demonstrations of VRE prototypes additional requirements will emerge as people change their ways of working, just as the invention of the elevator by Otis made the upper floors of skyscrapers into desirable real estate.

As an example of how requirements could be used to inform the list of services and identify workflows we firstly give a very simple use case:

```
User specifies required topic;
Uses search to discover location;
Uses authentication and authorisation to access source;
Downloads material using appropriate protocol;
Interacts with it using appropriate tool(s);
Does research;
Writes paper;
Publishes content;
Attends international conference.
```

Increasingly more complex and sophisticated use cases can be devised, and this should be done before large-scale deployment. Firstly we show a single-user case combining access to an HPC application on a remote resource with transfer of a large dataset from a data centre.

```
User logs in to VRE server
Login is based on e-mail and password
Login includes using X.509 certificates for Grid resources
Use Shibboleth or whatever emerges from JISC security middleware
      projects (future work)
User session established/ managed/ logged/ preferences selected
User selects a "Research Service Package" on the tools menu
Could involve a number of underlying tools/ services mediated by
      workflow to guide user through a complex procedure
User clicks "help" button or is provided with guidance in another form
Need to invest in "accessibility" configurations
User performs a query to list available services/ information sources
      with particular search criteria and semantic support
Service query is sent to Universal Description, Discovery &
      Integration (UDDI) registry
Enables dynamic invocation of Grid service
Service providers can edit, delete or modify published grid services
      through UDDI interface
UDDI returns a list of requested services and a URL describing on how
      to invoke the grid service
Information/ data query sent to cross-search broker and translated to
      use appropriate metadata via an information service
URL returned pointing to appropriate sources
Translation of format may be required through another service
Authorisation etc. with remote database  la OGSA-DAI
Protocol for download chosen, e.g. GridFTP
Clicking on the URL will generate a Web Services Description Language
      (WSDL) file or initiate download by pre-defined method
WSDL converted "on the fly" to a web form and presented to the user
User enters the appropriate additional details and submits the job
Data transmitted to host system
```

```
Job submission done using Web services and job logged
Monitoring of job done for future inspection
Notification of completion or other "event" by e-mail or message board
Collection of output, visualisation etc.
Repeat job
Send output to a different task in a pipeline
```

Finally an example of using a VRE for collaborative working between two or more users.

These steps could be part of a more complex workflow which could be modified with different data/ components and/ or repeated.

User has option of saving, deleting, printing or visualising the output and making decisions during the workflow session.

Registry of services and data needed and tools to query, find and consume them.

Security (authentication), confidentiality, privacy, accessibility and authorisation mechanisms essential.

Need appropriate presentation to user in ”familiar” environment, e.g. Web porta

```
User logs on as above
Private session managed as above
Determines if "peers" are present or if there is a hierarchical activity group
Asks to join group and is admitted
Peer session established and managed when more than one participant
Discourse with or share material with group
Use collaboration tools:
Whiteboards, visualisation, etc.
Chat, messaging, resource box
Controlled vocabulary
Decision support
Support for inter-disciplinary research
In research thi
```

Protocols/ standards

To make use of resources within the VRE additional protocols and standards are required over and above the access protocols identified above (e.g. Web services). Obvious requirements include the transfer of large quantities of data between data stores and computers, which requires optimal utilisation of bandwidth via a tuned FTP service. Search and content feeds also amplify the requirements. The following list may need to be extended, but should be contained to a manageable set implimented within the VRE and able to be consumed by deliery mechanisms such as portlets:

- Protocol examples: FTP, VIC, RAT, Z39.50, SRW/ SRU, OAI-PMH, RSS, OpenURL;

• Standards examples: IMS, Dublin Core, LOM, OAI, OKI, etc. including those as appropriate from W3C, IETF, OASIS, GGF.

Criteria for Success

In accordance with the discussion set out in the text, we suggest that a successful VRE would:

- Be applicable to all disciplines;
- Integrate e-research, e-learning and management of digital information to add value to all application areas;
- Be built to current specifications, standards and technology, in order to minimize development costs and time;
- Make UK services and resources available in familiar environments e.g. typically via a web browser;
- Have fast-track links into existing tools, services and resources ? some of which have been costly to produce and thus should be re-used;
- Enable an open community process for producing and consuming services and tools;
- Offer choice in presentation, delivery, and service and resource provision;
- Provide maximum ease of use, especially to new users, to optimise take-up;
- Demonstrate added value with respect to existing workbenches, toolsets, portals etc.

Call for Proposals

The discussion summarised above lead to a Call for Proposals by the JISC JCSR, circular number JCSR(04)/05. This aimed to spend £3.2M of money from HEFCE on demonstrators, tools and components for a UK VRE. The call deadline was 13th August and the proposal evaluation was completed on 20th September 2004. Some of the successful proposals using Sakai and related portal technology include:

Table 1: VREs using CHEF/ OGCE/ Sakai and related Portlet Frameworks

Project	Tool	Status
???	???	???

`need to complete this table when outcome is known`

5 Sakai Evaluation including CHEF, OGCE and GridSphere

In presenting the results of our evaluation it is important to distinguish and explain the relationship between the several separate but converging open source portal products which have recently attracted a lot of attention and a rapidly growing user base.

GridSphere: JSR-168 compiant portal development framework from the EU GridLab project;

CHEF: CompreHensive collaborativE Framework from University of Michigan, principally a collaboration portal hosting work tools and course tools;

OGCE: Open Grid Computing Environment portal, now adopted as the portal to be supported by the NSF Middleware Initiative (US equivalent of the UK Open Middleware Initiative Institute, OMII). Built on CHEF with Grid computing tools from NIEES Grid and NPACI GridPort;

Sakai: VLE portal built on CHEF and using OKI OSIDs with re-factored tools from Michigan, MIT, etc.

Charles Severance, chief architect of the CHEF and Sakai projects, was invited by JISC to participate as a plenary speaker at the two Awareness Raising Workshops on e-Collaborative Environments [41]. We were thus able to have early discussions with him about our proposed work. Previous discussions with others involved in portal developments, in particular Dennis Gannon and Geoffrey Fox of Indiana University, had taken place at the NeSC workshop in July 2003 [28]. The developers of GridSphere, Jason Novotny, Mike Russel and Oliver Wehrens, did however attend this workshop and gave a comprehensive tutorial of their software on the final day.

Sakai was not available to us at the beginning of our evaluation and required the UK to participate in the SEPP (Sakai Early Participants Programme). We were however already using CHEF in the JISC/ ESRC funded ReDRESS project `http://redress.lancs.ac.uk` and Sakai is based on a re-factoring of this system. We were able to gain access to the Sakai developers' CVS to obtain the source code in mid June 2004. SEPP membership also permitted us to represent JISC at the Sakai Developers' Workshop, Denver, USA, 23-27th June.

Sakai 1.0 beta was released to SEPP on June 23. You can download it from `http://cvs.sakaiproject.org/release/1.0.b1`. `the current release of Sakai is???`

According to Marlon Pierce, lead developer on the OGCE project, 21/5/04, "In a nutshell, Sakai is an collaboration between educational middleware and portal developers. It's basic component projects are a) implementing a JSR-168 container (uPortal group leads), building education/ courseware and collaboration tools (UMich leads that, I think), and building OKI-compatible middleware services. OKI is a previously unimplemented standard for education services and objects from MIT.

"OGCE concentrates more on Grid/ science portal applications. We are experimenting with early Sakai portal releases as they become available, but they still have a ways to go. We are planning to use them primarily as a) our base JSR-168 container, and b) a way to resynch with the latest version of the CHEF tools. I don't have an updated timeline from Sakai, but we hope to have this as part of our next major portal release at Supercomputing [2004]."

Evaluation of these related portal frameworks has involved deployment as follows:

Site	Framework	URL
Daresbury	CHEF	`http://thames.dl.ac.uk:8080:/chef/portal`
	OGCE	`http://thames.dl.ac.uk:1081/dnmi`
	Sakai	
Lancaster	GridSphere	Adrian's laptop
	CHEF	`http://redress.lancs.ac.uk/chef/portal`
	Sakai	`http://redress.lancs.ac.uk/sakai-uPortal`

These implementations are available for ongoing development, evaluation and demonstrations to interested parties. The following descriptions are taken mostly from the relevant Web sites.

5.1 GridSphere

The GridSphere project is building on experience in the Java-based Astrophysics Simulation Collaboratory and GPDK portal toolkits. It now has a reasonable uptake in Europe and to some extent Canada.

GridSphere provides a "white-box" framework (you edit the code) in which users can override base classes and "hook" in their own methods. It therefore requires users to become familiar with core framework interfaces which are however based on the community standard API JSR-168. Heavy use is made of design patterns which provide template solutions to commonly recurring software design problems. They also provides a common language that makes the code easier to read and understand. The Model View Control (MVC) pardigm is used to separate logic from presentation as in other portlet frameworks.

Features of GridSphere include:

- Portlet API implementation nearly fully compatible with IBM's WebSphere 4.2.
- Support for the easy development and integration of "third-party portlets"
- Higher-level model for building complex portlets using visual beans and the GridSphere User Interface (UI) tag library.
- Flexible XML based portal presentation description can be easily modified to create customized portal layouts.
- Built-in support for Role Based Access Control (RBAC). Enables managing of access for guests, users, admins and super users.
- Sophisticated portlet service model that allows for creation of "user services", where service methods can be limited according to user rights.
- Persistence of data provided using Castor JDO from ExoLab for RDMS database support, SQL and OQL
- Integrated Junit and Cactus unit tests for complete server side testing of portlet services including the generation of test reports.
- Documentation uses DocBook for HTML and PDF output of guides and tutorials
- GridSphere core portlets offer base functionality including login, logout, user and access control management.
- Localisation support in the Portlet API implementation and GridSphere core portlets support English, American, German, Czech, Polish, Hungarian and Greek.
- Open-source and 100% free!

Both the portlet definitions and the portal layout are coded in XML e.g. in Portlet.xml and Layout.xml and validated against schema. A portlet also has an associated class files and JSP pages. Authorised users can deploy portlets dynamically providing Tomcat >v4.1.18 is used.

The core and basic services provided in GridSphere are:

- Portlet Manager Service – Provides lifecycle methods to allow portlets to be installed, removed, initialized and destroyed by authorized users
- Login/ logout Service – Allows a User to be retrieved from a username and password;
- Locale – banners and support for 7 languages;
- User Manager Service – Add/ Remove User Accounts; Edit User Profiles;
- Access Control Service – Add/ Remove User Groups; Add/ Remove User Roles;
- Layout configuration – customise layout by creating new tabs which portlets can be added to;
- Credential Manager Service – Add/ Remove allowed User Credentials; Configure use of Credential Retrieval Service;
- Job Manager Service – For listing, starting, migrating, stopping jobs;
- Job Monitoring Service – Specify what to monitor for any given job and archive related information;
- File Transfer Service – For managing and scheduling file transfers;
- Data Manager Service – Access to data replica catalogues; Describe data with meta-data;
- Notification Service – Define events to be notified about; Specify how to be notified about those events;
- Text messager – communicate to IM users;
- Photo album – upload and display photos;
- Poll – create poll and display results;
- Chart – chart service using JFreeChart to display plots, timeseries graphs, etc.;
- Commander – secure portal file system to upload, download and transfer files.

A number of core portlets are provided by which these services can be accessed. Future plans include the provision of a complete general JSR-168 compliant framework (done in February 2004), integration of the GAT toolkit and OGSA services, an IDE, inclusion of Flash presentations and some forms of collaboration tools.

5.2 CHEF

The **C**ompre**H**ensive collaborativ**E** **F**ramework (CHEF) project is developing a flexible environment for supporting distributed learning and collaborative work, and doing research on distributed learning and collaborative work.

This project is staffed by University of Michigan School of Information, Media Union, and Medical School staff. The developers are working closely with and are contributing to the OKI reference architecture from the Mellon Foundation, and are collaborating with other groups interested in open source collaboration standards.

Communities targeted for CHEF use include those involved in the scholarly activities of teaching, learning and research at the University of Michigan, and their students and colleagues involved in teaching, learning and research that are outside of the Michigan community.

The CHEF framework includes various tools that are used in supporting collaborative work and distributed learning, and tools necessary for the study of collaborative work and distributed learning. The framework provides organization for the disparate functionality used to support research, collaborative and learning activities and combines locally developed, commercial off the shelf and free off the shelf components.

CHEF is aimed at making available a set of functional elements that can be easily configured by users to accomplish a wide variety of activities. This framework supports existing and emerging capabilities, and seeks to make the integration of new functionality as easy as possible.

In our past and current work we have seen much commonality in the needs and tools used by people in these communities, and we are seeing the emergence of frameworks for user configurable toolset delivery. CHEF mobilizes that experience in the effort to develop a comprehensive framework to support these activities, and to make this framework available for wide use.

5.3 OGCE

In the USA several groups previously in the so-called Portal Collaboration are now collaborating in the Open Grid Computing Environment (OGCE) Portal Project funded by the National Science Foundation (NSF) as part of its middleware initiative, indeed it is also referred to as the NMI portal. The Portal Collaboration principally involved the San Diego Supercomputer Centre with Perl-based GridPort and HotPage toolkits developed by Mary Thomas and Steve Mock supporting the NPACI Grid and NCSA at Urbana-Champaign with the Java-based GPDK toolkit developed by Jason Novotny. OGCE now includes researchers and end users from Indiana, NCSA, Michigan, Texas and Argonne National Lab. OGCE now embraces the new technologies of portlets and Web/ Grid services to serve the needs of NCSA, NPACI, DOE and NASA Grid users. Jetspeed, CHEF and OGSA/ OGSI are currently being used with the Argonne Globus Java CoG kit. Jason Novotny has gone on to develop GridSphere as a parallel activity with Mike Russell directly supporting the EU GridLab project. GridSphere is also portlet based and there is regular discussion between the various groups.

The functionality of OGCE includes a number of the collaboration and Grid tools mentioned below.

A lot of this is explained in the paper by Gannon et al. [53].

In general OGCE (which is also sometimes called the NMI portal project), represents a union of many of the American Grid related portal projects. Dennis Gannon notes that it has a long way to go over the next twelve months, including its integration into Sakai and the port of all the portlets to JSR-168 and futher consideration of WSRP. Currently much of the effort of the OGCE project is focused on building Grid/ Web services and their access by client interfaces. Some of the work at Indiana has the goal to deploy Grid services for things like workflow tools and make it possible for portal users to discover and load the client interfaces into their portal environment or compos it into applications as components.

Planned portal deployments that are based on OGCE include:

1. The NSF TeraGrid portal (prototype at `http://www.extreme.indiana.edu:18081/teragrid/portal`);
2. The Linked Environments for Atmospheric Discovery NSF ITR project (prototype at `http://lead.extreme.indiana.edu:10081/lead`);
3. The DOE Fusion Portal. see: `www-fp.mcs.anl.gov/middleware-review/ProjectReports/PortalWebservicesfinal.pdf`;
4. The Southern California Eathquake Center see: `http://epicenter.usc.edu/cmeportal/proposal.html` for the project which is just starting its portal effort;
5. The NEES Grid portal is based on CHEF and not OGCE but it was the portal effort that first used many components that are now going into OGCE;
6. the NCSA Alliance Portal, see: `https://portal.extreme.indiana.edu:8443/alliance/index.jsp`

The current version of OGCE is based on CHEF v1.0 and will be ported to Sakai by autumn 2004 (see above note from Marlon Pierce) so that it will be JSR-168 compliant. The generic OGCE release is available from `http://www.ogce.org`.

5.4 Sakai

Sakai is a $6.8M project which implements a model for the purposeful coordinating of work in a community of teachers and learners. It is based on many of the principles of open source development efforts, but community source efforts rely more explicitly on defined roles, responsibilities, and funded commitments by community members than some open source development models. The project is founded by the University of Michigan, Indiana University, MIT, Stanford, the uPortal Consortium, and the Open Knowledge Initiative (OKI) with the support of the Andrew W. Mellon Foundation and William Hewlett Foundation. The Sakai Educational Partners' Program (SEPP) extends this community source project to other academic institutions around the world.

The consortium members are joining forces to integrate and synchronize their considerable quantity of educational software into a pre-integrated collection of open source tools. This will yield three big wins for sustainable economics and innovation in higher education:

- A framework that builds on the recently ratified JSR-168 portlet standard and the OKI open service interface definitions to create a services-based, enterprise portal for tool delivery;
- A re-factored set of educational software tools that blends the best of features from the participants, disparate software (e.g. course management systems, assessment tools, workflow, etc.);
- A synchronization of the progress of these schools in developing, and sharing a common set of open source software.

The products of this project will include an Enterprise Services-based Portal, a complete Course Management System with sophisticated assessment tools, a Research Support Collaboration System, a Workflow Engine, and a Technology Portability Profile as a clear standard for writing future tools that can extend this core set of educational applications.

Modular, pre-integrated tools will in princuple greatly reduce the implementation costs of one or more of these tools at any institution. The core partners are committing over $2 million per year to launch and support this two year project. The core universities are also committed to implementing these tools at their own institutions starting in Fall 2004 through the duration of the project. The commitment of resources and adoption is purposefully set on an aggressive timeline to swiftly integrate and synchronize the educational software at the core institutions. This effort will demonstrate the compelling economics of "software code mobility" for higher education, and it will provide a clear roadmap for others to become part of an open source community.

Overcoming Barriers to Application Sharing

The technical barriers can be overcome by distilling the accumulated architectural knowledge and programming experience gained in building systems into a Technology Portability Profile (TPP) that provides four essential elements for code mobility.

- Service Interface Definitions – the Open Knowledge Initiative Open Services Interface Definitions (OSIDs) have provided an essential first contribution to solve this technical challenge. Local implementations of the OSIDs at an institution integrate heterogeneous local architectures (e.g. an authentication system or directory service) by using common connectors that enable code mobility for OKI-based application software (e.g. a CMS, library system, collaboration tool, etc.);
- Standard Portal – advanced CMS's are based on portals that aggregate class information and services and allow the user to personalize and customize their views of these classes, services and information. At the same time, university-wide services are migrating from independent web-based interfaces that accessed siloed systems (e.g. Bursar, Library, Registrar, CMS, etc.) to enterprise-wide portals that integrate a personalized view of the full range of the university's services and information. The uPortal effort has brought forth a powerful portal environment that has commanded broad adoption, but it currently lacks the recently ratified JSR-168 portlet specification needed for tool interoperability. A standards-based portal that can be used as both the academic portal for the CMS as well as for delivering other university services via the JSR-168 portlet standard is a core building block of the TPP;

- Tool Interaction Framework – tools need a framework to provide a consistent way for invoking other tools and passing information among them. For example, a homework 'drop box' tool developed at one university may need to invoke and pass a grade to a 'grade book' tool developed at another. This framework provides a common place for these tools to interact with each other in a standard way. It also provides services like notification that cross tool boundaries. The U. Michigan CHEF project has developed such a portal-based framework which provides the environment for a large set of course management tools;
- Localized User Interface – institutional adoption by faculty and students often relies on user interfaces that match other familiar systems. A user interface includes colors, fonts, logos, and navigation aids that can be localized as needed without disturbing the underlying functionality of the software. The TPP will provide standard methods and description of best practices for Sakai-based tool interfaces to be customized and modified by user institutions

The maturing of the OKI OSIDs, recent demonstration of a working tool interoperability framework at U. Michigan, and industry ratification of the JSR-168 portlet specification make the timing perfect for developing a full Technology Portability Profile for higher education. But, while specifications, standards, and profiles are numerous in higher education, it is large adoptions that give a specification momentum to become a universal and wide spread standard.

Sakai depends upon a number of technologies which are now described. The risks associated with these technologies is noted.

5.4.1 uPortal

uPortal 3.0 is required by Sakai, see `http://www.uportal.org`. Indeed the uPortal Consortium is a member of the Sakai Collaboration. uPortal is an open-standard effort using Java, XML, JSP and J2EE. It is a collaborative development project with the effort shared among several of the JA-SIG member institutions. You may download uPortal and use it on your site at no cost. The most significant change to in uPortal 3.0 as compared to 2.3 is the native portlet support will be provided. Currently it is done using an adaptor into uPortal channels. Pluto will be used in both cases. This is further explained on the Web site. uPortal will support WSRP (when?).

5.4.2 Java Server Faces GUI Layer

JavaServer Faces technology includes:

- A set of APIs for representing UI components and managing their state, handling events and input validation, defining page navigation, and supporting internationalization and accessibility.
- A JavaServer Pages (JSP) custom tag library for expressing a JavaServer Faces interface within a JSP page.

Designed to be flexible, JavaServer Faces technology leverages existing, standard UI and web-tier concepts without limiting developers to a particular mark-up language, protocol, or client device. The UI component classes included with JavaServer Faces technology encapsulate the component functionality, not the client-specific presentation, thus enabling JavaServer Faces UI components to be rendered to various client devices. By combining the UI component functionality with custom renderers, which define rendering attributes for a specific UI component, developers can construct custom tags to a particular client device. As a convenience, JavaServer Faces technology provides a custom renderer and a JSP custom tag library for rendering to an HTML client, allowing developers of Java 2 Platform, Enterprise Edition (J2EE) applications to use JavaServer Faces technology in their applications.

Ease-of-use being the primary goal, the JavaServer Faces architecture clearly defines a separation between application logic and presentation while making it easy to connect the presentation layer to the application code. This design enables each member of a web application development team to focus on his or her piece of the development process, and it also provides a simple programming model to link the pieces together. For example, web page developers with no programming expertise can use JavaServer Faces UI component tags to link to application code from within a web page without writing any scripts.

Developed through the Java Community Process under JSR-127, JavaServer Faces technology establishes the standard for building server-side user interfaces. With the contributions of the expert group, the JavaServer Faces APIs are being designed so that they can be leveraged by tools that will make web application development even easier. Several respected tools vendors are contributing members of the JSR-127 expert group and are committed to supporting the JavaServer Faces technology in their tools, thus promoting the adoption of the JavaServer Faces technology standard.

The expert group is actively developing the specification for JavaServer Faces. To check its current status, please see the JSR-127.

5.4.3 JSR-168

JSR-168 is the Portlet standard ratified by JA-SIG in August 2003 following community comment. See Section 1.2. Open source implementation is available, Pluto from the Jetspeed developers.

5.4.4 JSR Servlets

???

5.4.5 WSRP

WSRP: Web Services for Remote Portlets is a Web services standard ratified by OASIS in August 2003. See Section 1.2.

5.4.6 OKI Open Service Interface Definitions

O.K.I., the Open Knowledge Initiative, develops specifications that describe how the components of an educational software environment communicate with each other and with other enterprise systems. O.K.I. specifications address broad interoperability agreements that allow for adaptation and further specification by communities of practice. In this way, O.K.I. seeks to open new markets for educational tools and content.

The O.K.I. team at MIT continues to provide architectural and technical leadership for software initiatives in higher education, among industry affiliates, and through national and international organizations such as the IMS Global Learning Consortium. This work helps ensure that the next generation of e-learning software will integrate seamlessly with the educational enterprise, and it establishes O.K.I. as a global leader in behavioural specifications for educational technology interoperability.

The Open Knowledge Initiative was initially funded through a grant from the Andrew W. Mellon foundation. This work is now being taken forward in the Sakai project through the design of a pragmatic portlet API to the OSIDs, Open Standard Interface Definitions. The Java specification for the OSIDs is available from SourceForge at `http://sourceforge.net/projects/okiproject/`.

The Open Knowledge Initiative (OKI) is a multi-institutional, multi-year project led by MIT and Stanford and funded by the Andrew W. Mellon Foundation. It is a highly collaborative effort to enhance learning by creating an open programming environment that supports sharing and pedagogical experimentation. The initiative is in its second year of work, and has spawned much interest and high expectations in the international educational community. It is important that expectations for the products of the OKI are realistic, especially in light of rapid growth in enterprise-wide academic systems such as Blackboard and WebCT, the academic counterparts to our administrative systems.

The following summary of the initiative is couched in non-technical terms to aid small colleges and universities as they prepare to use and to contribute to the OKI.

The OKI has short- and long-term goals. The short term goals are to set standards for software development, to develop a set of basic software tools which are needed by most educational software, and finally to create some exemplary educational software. As of this writing (October, 2002), many of these short-term goals have been met although it is likely to be late in 2003 before our institutions will be able to use OKI products. The long-term goals are to create a community of software developers who can adopt, extend and improve on the OKI standards and software, and to ensure that the Initiative can be sustained in the face of constantly changing technologies.

At the core of the OKI are fully public software standards that define an open programming environment to foster pedagogical experimentation and sharing. This open programming environment is based on delivery of learning materials through standard web browsers. This means that software written to the OKI standards will be accessible to users of nearly all modern computers, and the content will be accessible wherever there is an Internet connection.

The OKI can be most easily understood when viewed as a set of four layers. Each activity of the Initiative is based upon the layer below it. The bottom-most layer is the infrastructure of the campus: the campus network and associated hardware and software that is the responsibility of each campus. Built upon this foundational layer are OKI "common services" and "educational services." Common services are modules of software that perform relatively limited functions. Educational services comprise the next layer and perform more complex functions. Many educational services are created by the integration of several common services. The OKI defines the function of these services and how they interact with one another. Finally, at the topmost layer are "educational applications" which are complex systems that integrate common services and educational services into more comprehensive learning systems.

The availability of publicly defined common services and educational services makes it possible for programmers to focus on their pedagogical goals by using and reusing software modules as building blocks for more complex educational software. For instance, the OKI defines common and educational services such as how students and faculty log into an OKI-compliant system, how workgroups of learners are managed, or how files are shared among students and faculty. Programmers will need many such services to integrate into more complex educational software that meet pedagogical needs.

An educational application is software that has one or more specific pedagogical goals and is built upon the architecture defined by the OKI. For instance, student electronic portfolios could be an educational application. An e-portfolio application might integrate many OKI-defined services such as authenticating the identity of a student, managing the files in a portfolio, and allowing portfolio files to be annotated and assessed.

OKI staff members have not only defined the standards for these services, they are implementing many of them so that developers of educational software can take early advantage of the OKI effort. The common services are being written in the Java programming language but because the definitions of these programs are public, others can create the same functionality by writing their own services in programming languages of their choice. Thus, the OKI is not dependent upon a single programming language or a single computing platform for shared development or access to educational content.

It is important to note that the OKI is not a comprehensive course management system, although comprehensive educational applications are being developed on the OKI standards. These included `http://aboutcoursework.stanford.edu/` Stanford's CourseWork, `http://stellar.mit.edu/` MIT's Stellar and `http://www.chefproject.org` The University of Michigan's CHEF among others. For instance, the latest release of CourseWork complies with many of the OKI standards and is being assessed in courses at several institutions this academic year.

It is also important to mention that the Initiative will not produce 'free' software. The OKI is a non-commercial project based on the 'open-source' model of software licensing. This gives us the freedom to adapt the products of OKI to our needs, but supporting our use of OKI products will not be free. Our institutions will not avoid the costs of course management systems simply by adopting the products of the OKI.

Open source software is distributed freely and includes all the source materials that will allow a programmer to improve the software, adapt it to their needs and to fix bugs. Open source software is thus maintained and extended by a community of programmers. However, the usual avenues for software support don't exist for open source software that is distributed freely. There is not likely to be a 24-hour phone number to call for help. More likely, our institutions' knowledgeable support staff will be able to post inquiries to an e-mail list or browse web resources for answers to support questions. Thus technical support for open source software is a cost center.

Our institutions will be able to adopt the products of the OKI if they have the staff time and expertise to install and maintain the software they select. The costs of using OKI products will come from meeting hardware needs for the products (servers, backup, etc), installing the applications, connecting them to institutional databases as needed, and supporting the local user community. The benefits of the OKI will come as we gain access to diverse educational applications contributed to the OKI community by colleges, universities and other developers. For example, Stanford University is developing an assignment and assessment manager to deliver online assessments that are discipline-specific and teaching method-specific, and to allow faculty to embed assessments into online content for better formative feedback.

Commercial developers are able to contribute educational applications to the OKI (and Blackboard and WebCT have announced they will), but we will have to license those commercial products even if they do fit into the freely distributed OKI framework.

Our institutions will be able to contribute to the OKI if they have the staff time and expertise to develop new modules that comply with the OKI standards. Developing OKI applications implies a higher level of resource commitment and expertise than just using the products of the OKI. Several major institutions in the US and UK are core collaborators or application developers for the OKI, so the outlook for a rich suite of educational applications is excellent. There is still plenty of room for the smaller colleges and universities to contribute. For many of our institutions, a strong focus on undergraduate learning outcomes puts us in a unique position to design effective educational applications in the OKI framework. Our ability to implement such applications will depend on our commitment of technical resources and the challenges of integrating basic OKI services into effective pedagogical software. Those who wish to contribute to the OKI effort should participate in the OKI Developer's Network the first meeting of which will take place at MIT in March, 2003.

For much more official information about the Open Knowledge Initiative, visit `http://web.mit.edu/oki/`.

The Open Knowledge Initiative has produced a series of Open Service Interface Definitions (OSIDs) informed by a broad architectural view of the educational technology landscape. As learning management systems have become a core component of the campus information technology infrastructure, O.K.I. seeks to simplify and enhance the creation of educational applications.

The OSIDs are an abstraction layer between the programmer and the enterprise infrastructure systems of his or her campus. Each OSID is characterized by a tightly defined set of methods and strict boundaries. This approach offers a number of important benefits to applications designed to the OSIDs:

- Simple integration with existing infrastructure;
- Local innovations can be shared across campuses or universities;
- Adaptation to new technology without destabilizing the overall;
- environment.

The OSIDs are divided into the following classes. See also `http://www.grids.ac.uk/ETF/public/WebServices/classes.html`. We note that many of these are not mature and the descriptions are sketchy.

Common services:

Agent:

Authentication: The Authentication OSID gathers required credentials from an agent, vouches for their authenticity and introduces the agent to the system.

Authorization: The Authorization OSID allows an application to establish and query a user's privileges to view, create, or modify application data, or use application functionality.

Dictionary: The Dictionary OSID provides a means to support multiple languages, domain-specific nomenclature and culture-specific conventions through interchangeable property files.

Filing: The Filing OSID provides platform-independent means to handle files arranged in simple hierarchical containers.

Hierarchy: The Hierarchy OSID manages parent-child relationships among elements. In addition to simple tree structures, the OSID supports hierarchy that are recursive and have nodes with multiple parents.

ID:

Logging: The Logging OSID records and retrieves a variety of application activity history

Scheduling: The Scheduling OSID manages events in shared calendars.

Shared: The Shared OSID contains fundamental objects used in the other OSIDs to provide their functionality.

SQL: The SQL OSID provides relational database access functionality at a higher level of abstraction than the DBC OSID. Unlike DBC, it is not dependent on JDBC.

User Messaging: The Usermessaging OSID supports communication and notification among users.

Workflow: The Workflow OSID provides a way to manage an interdependent succession of activities each of which has completion constraints.

Educational services:

Assessment:

Cource Management:

Grading:

Repository:

Java and PHP specifications are available from SourceForge.

5.4.7 Spring

???

5.4.8 Avalon

???

5.4.9 Turbine

???

5.4.10 Pico

???

5.4.11 J2EE/ EJB/ JBoss

???

5.4.12 Database

Oracle, MySQL, PostgreSQL.

5.4.13 Sun's Evaluation of Sakai

Sour grapes?

5.5 Available Portlet Tools

We here follow the classification scheme of Appendix G and identify what portlets are available from various sources which are or could be incorporated in the Sakai or other JSR-168 compliant framework. The evaluation tasks reported in the other appendices have showed in some cases how this can be done.

5.5.1 Collaboration

Some collaboration portlets available include:

• Login/ Logout - as it says on the tin
• Membership - join/ unjoin from workspaces
• Schedule - calendar for workspace
• Resources - resource list and links
• News - news service
• Discussion - threaded discussion service
• Chat - online chat room
• Anabas - join and participate in Anabas Impromptu collaborative session
• Newsgroup (Read/ post) - topic-based threaded newsgroup tool
• Newsgroup (Request) - request a new topic
• Newsgroup (Admin) - for admins only!
• Bibtex (Read/ edit) - add/ edit entries in Bibtex topics for shared authorship of reference lists
• Bibtex (Request) - request new Bibtex topic
• Bibtex (Admin) - for admins only!
• Bibtex (Super Admin) - for super admins only!
• Customize - chose which tools you want in the workspace, edit portlet layouts etc.

5.5.2 e-Research

Some e-Research portlets, mainly for managing remote jobs on a computational Grid include:

• Proxy Manager - store and retrieve credentials from MyProxy repository
• GRAM Job Launcher - launch a Globus job via interactive GRAM protocol
• GridPing - test a remote Grid service
• Grid Job Submission - submit a remote batch job to a Globus resource
• LDAP Browser - browse Grid Information Service for remote resources
• GridFTP - invoke 3rd party file transfer
• GridContext - brows and manage DNs in Xdirectory server
• GPIR Browser - browse Grid resources via Grid Port Information Repository
• Historical GPIR - use GPIR to view stored information
• CSF Job Submission - submit job to a Community Scheduling Framework service
• Sequencer - ditto via GridPort sequencing service
• OGRELauncher - Open Grid Computing Runtime Engine: launch simulation event via OGRE from NCSA, workflow based on ANT
• OGREEventView - monitor OGRE events
• COG-Workflow - use workflow enactment via the Java CoG kit
• App Manager - register, de-register and edit information about a Grid application
• Condor - submit jobs via Condor or DAGMan and monitor them
• Customize - chose which tools you want in the workspace, edit portlet layouts etc.

5.5.3 e-Learning

???

5.5.4 Digital Information

???

5.6 Outcome of Evaluation Work Packages

???

5.6.1 Evaluation Report Part 1: Technology Survey

The Technology evaluation Report draws upon two sources. Firstly the proceedings of the *Portals and Portlets 2003* workshop [28] which was written up in early 2004. Secondly the technolody survey undertaken by Adrian Fish for the ReDRESS project [37]. The body of this report is provided in Appendix **??**.

Here we provide some conclusions.

5.6.2 A Review of the Issues for building standards-compliant Portlets

5.6.3 An Assessment of the Potential of Sakai/ CHEF as a Platform for Customised Portals

5.6.4 Evaluation Report Part 2: Developer and User Feedback

This report focuses on issues of ease of administration, user management, customisation, etc. The body of this report is provided in Appendix **??**.

Here we provide some conclusions.

During the time frame of this evaluation exercise, we have had dialogues with a large number of people engaged in state of the art research in e-Science projects funded via the Research Councils, Core Programme or DTI. Many have been supportive to the idea of using portal interfaces and of developing a VRE encapsulating wide-ranging functionality. In fact there is no doubt that a VRE building on, and extending the capabilities of the current Grid and using Web services is a high-priority goal for many areas of applied research. Collaboration tools are high on the list of desireables. Portals are one means of delivering the fuctionality of this VRE, but additional requirements for lightweight toolkits have also been expressed by several groups [59, 57, 58, 61]. Expressions of interest have been forthcoming as shown in the Table 5.6.4.

5.6.5 Software Template for Sakai/ CHEF Institutional Adapters

???

Table 2: Projects using CHEF/ OGCE/ Sakai and related Portlet Frameworks

Project	Tool	Status
	in the UK	
CCLRC InfoPortal v3.0	CHEF	migrating
e-HTPX	CHEF	evaluating
MyGrid	CHEF	developing portal
ReDRESS	CHEF	using
NCeSS	CHEF	evaluating
Portsmouth		
Oxford	uPortal	input to JSR-168 and WSRP standards
RealityGrid	GridSphere	developing portal
GeneGrid	GridSphere	developing portal
P-GRADE	GridSphere	portal for GEMLCA workflow
Cambridge		
DAME	Jakarta Struts	
White Rose Grid	Jakarta Struts	
SPP	Jetspeed	developing basic portlet functionality
Hull	uPortal	
Nottingham		
Bristol		
	in the USA	
NMI portal	OGCE	using for US Grid projects
CMCS	CHEF	using wih Ecce meta-data tools
BioGrid	CHEF	using
Alliance Grid	CHEF/ OGCE	using with Xportlets
NPACI Grid	CHEF/ OGCE	using with HotPoage GPIR
U. Michigan	CHEF	WTNG and CTNG tools migrating to Sakai
MIT		tools migrating to Sakai
Stanford		tools migrating to Sakai
???		tools migrating to Sakai

5.6.6 Software Template for Grid Tool Wrappers for use in Sakai/CHEF

???

5.6.7 Roadmap for a UK Virtual Research Environment

A *Roadmap for a UK Virtual Research Environment* was prepared for the JISC Joint Committee for the Support of Research with the VRE Working Group chaired by Dr. David Boyd in April-May 2004 [29]. Along with this report a comprehensive service classification was produced with input from the Grid Engineering Task Force, JISC Information Environment and JCLT architecture groups. See `http://www.grids.ac.uk/ETF/public/WebServices/classes.html`.

6 Conclusions and Way Forward

This report has focussed on CHEF/ Sakai as a user-facing delivery mechanism for services in a Virtual Research Environment. It is however only one component of the environment. During the course of our work we have needed to address other aspects of VRE architectures which dictate how Sakai is to be used alongside other components and delivery mechanisms in a Grid such that the many and diverse services supported by JISC and research institutions can be made available.

6.1 VRE Architecture

Figure 6 shows how major components in a federated VRE architecture could be linked. In developing this architecture we coined the acronym HIVE: *Highly Integrated Virtual Environment*. This inherits many aspects from CCLRC's prototype IeSE: Integrated e-Science Environment [16, 26].

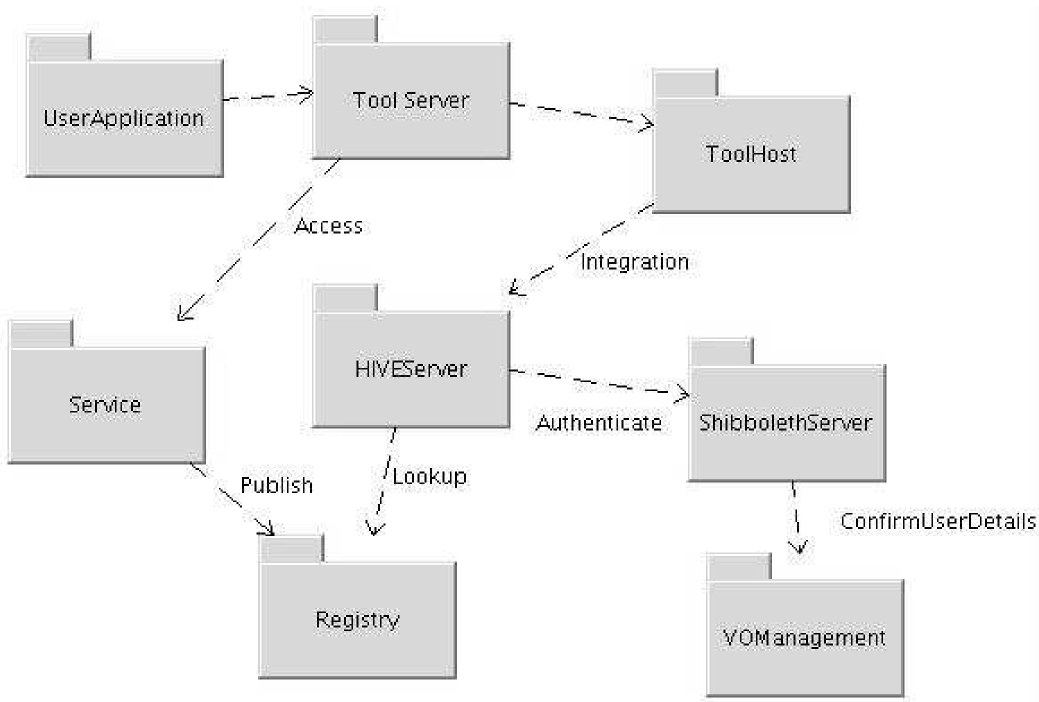

Figure 6: Federated Components in a VRE

In this architecture there can be multiple instances of each component serving slightly different functions.

Notes:

User/ Application: Consumer of delivered services via tools;

Tool Server: User facing part of the system. Browser, programmig library, desktop icons etc.

Tool Host: The tools server can be Web or desktop based. Will delegate authentication to HIVE server and thus permit single sign-on across remote toolsets;

HIVE Server: HIVE server provides access to integration services such as authentication, workflow, registries. Can handle federated services;

Shibboleth Server: Will provide the authentication services to the HIVE server. Could be part of a federation and thus provide trust-based access to all the tools hosted for all researchers in the federation's institutions;

VO Management: Provides information about users, their roles and project affiliation. Can extend to resources and services;

Registry: Registry holds details of services and provides template to access them along with relevant semantic information. May be a number of registries handling different types of services. ETF UDDI and JISC IESR are examples;

Services: Multiple services provide access to end resources and applications. Language agnostic so can wrap heritage applications and facilities.

6.2 Integration Services

Within a portal a number of internal services are needed to address of issues of coordination of tools (portlets) within an overall framework. Methods can be provided as an "internal" class library which sits alongside the portlet API and service APIs (the model part of the MVC paradigm). Each portal framework could have the same or a different set of tools, but the way they are integrated may differ between user groups – similar services are required to allow different frameworks to inter-operate. These services could be federated and available via Web service calls to specialised servers elsewhere in a Virtual Research Environment.

Research issues are implied with most of these services. Some simple ones, such as managing the look and feel of the portal, personalisation and accessibility are provided directly by the portlet container. This is a specialisation of portals and not required in other frameworks. Services which are not specialised, e.g. single sign-on, should not be limited to the portlet API. Some example portal integration services are now listed:

Session Management: management of a Session Key and issues related to single sign-on and session activities. Involves database access for storing and retrieving other items relevant to the session. User can authenticate and start a new session or revert to a previous one. Service can open and close sessions and log state of a session from state handle. Rollback and replay including peronal workflow can be available.

Authentication using MyProxy: MyProxy is a repository of valid proxy certificates for authenticated users. The portal can download these for delegation to trusted external services. Service can also check that certificates etc. are still valid and refresh them if not. Part of the integration API would enable storing and retrieval of the proxy in the portal database for later use. This will be done using the Session Key and uid (e.g. DN or unique e-mail address). Having the cert associated with the session key enables authorisation issues to be tackled, e.g. using subsidiary cert or other method. Same user but different session => working in different role?

VO Management: a virtual organisation could be based around a project (as described by UDDI [18]) which would typically have its own portal and mini-Grid. VO users are real people who have been authenticated and have received a digital identity (certificate). They are then given rights based on the roles they are taking in this VO and thus can be authorised to access services. A prototype schema is given in a separate paper [23].

Integrated State: manage database storage and retrieval of state information by portlet id. Research needs to develop concepts of integrated state. State can be used as an event trigger. State needs to be logged for session manager/ workflow. What states can portlets and services have which are meaningful for rollback and replay?

Service and Portlet Location: registry input, query and lookup of remote services and portlets. This requires semantic support, i.e. what does the service do and why?. It also supports identification and location of internal portlets with unique keys (portlet id) for use in IPC etc.

Portal Preferences: build up a "preferred set" of services, portlets etc. based on usage, e.g. from registries. This service can also log semantic information and build a related ontology. It extends the idea of a workspace toolset enabling dynamic and semantic/ function-driven choice.

Semantic/ Ontology Support: semantic and knowledge-based information about services and portlets in the framework. These are used for decision support and choice augmenting stored preferences. This does not cover generic semantic issues which would need separate tools.

Workflow: directed links between components (typically graph based). Event mechanism used to trigger actions within portals and attached services. Graphs in the portal will be mostly predefined, but with constrained facilities to swap in and out components and provide additional inputs at decision points. Again, not completely generic workflow, but based on instances in the preferences list and their states.

Trails and Personalisation: logging of usage for off-line mining and analysis, e.g. for developers to improve presentation, ease of use, and optimisation, e.g. by aggregation of low-level tools and services. This tracks state and component changes for each session. Basically same information as session management service.

IPC - Inter Portlet Communication and Event Management: a message-based communication mechanism between portlets, possibly with event triggers and asynchronous handler. Could be interrupt driven? Use a message queue in the database with associated triggers. Example: Completion of a computational job or background query triggers portlet to send SMS or SMTP message or view results. Could therefore give the user a flag, a chat entry, mobile phone message or an e-mail.

Some key research issues in implementing the above services include:

Identification of user/ session/ portlet/ services: name value pairs => sessionKey, uid, portletId, serviceId. Same stuff as typically put in a cookie in 1st generation portlets. Can however also be used by non-portal based tools by building into the method calls.

State definition: a pre-defined set of states needs to be identified. Can this be done? This could be the key to using the event mechanisms and session logging. Do portlets and/ or services make clear state changes?

6.3 Portal Architecture

This evaluation and consideration of the wider implications and how to implement a Virtual Research Environment for the UK have raised some interesting architectural questions. Similar questions have been raised within the JISC VLE programme, e.g. see the paper by Bill Olivier [60]. The key to both VLE and VRE deployment is to ensure the maximal use of existing resources via a re-usable set of distributed services delivered through a variety of mechanisms such as portals and desktop applications.

`Conclusions`

`Way forward`

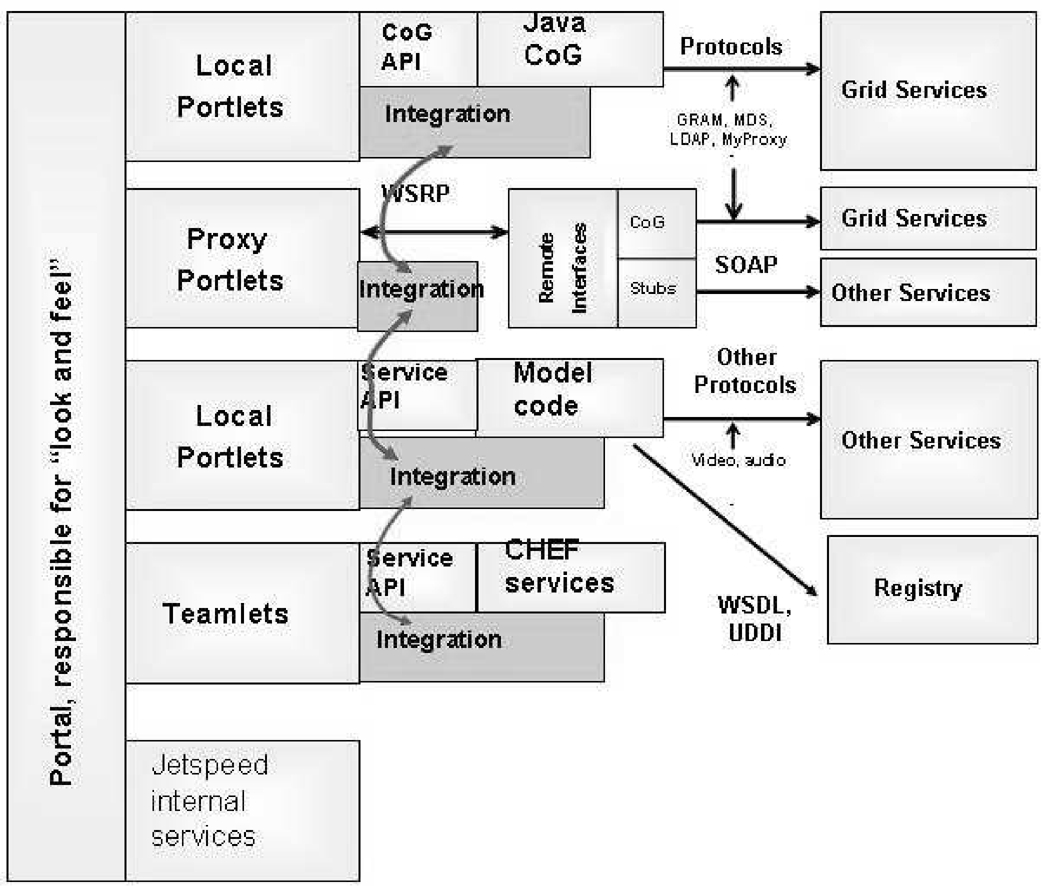

Figure 7: Pluggable portal components

`why Sakai?`

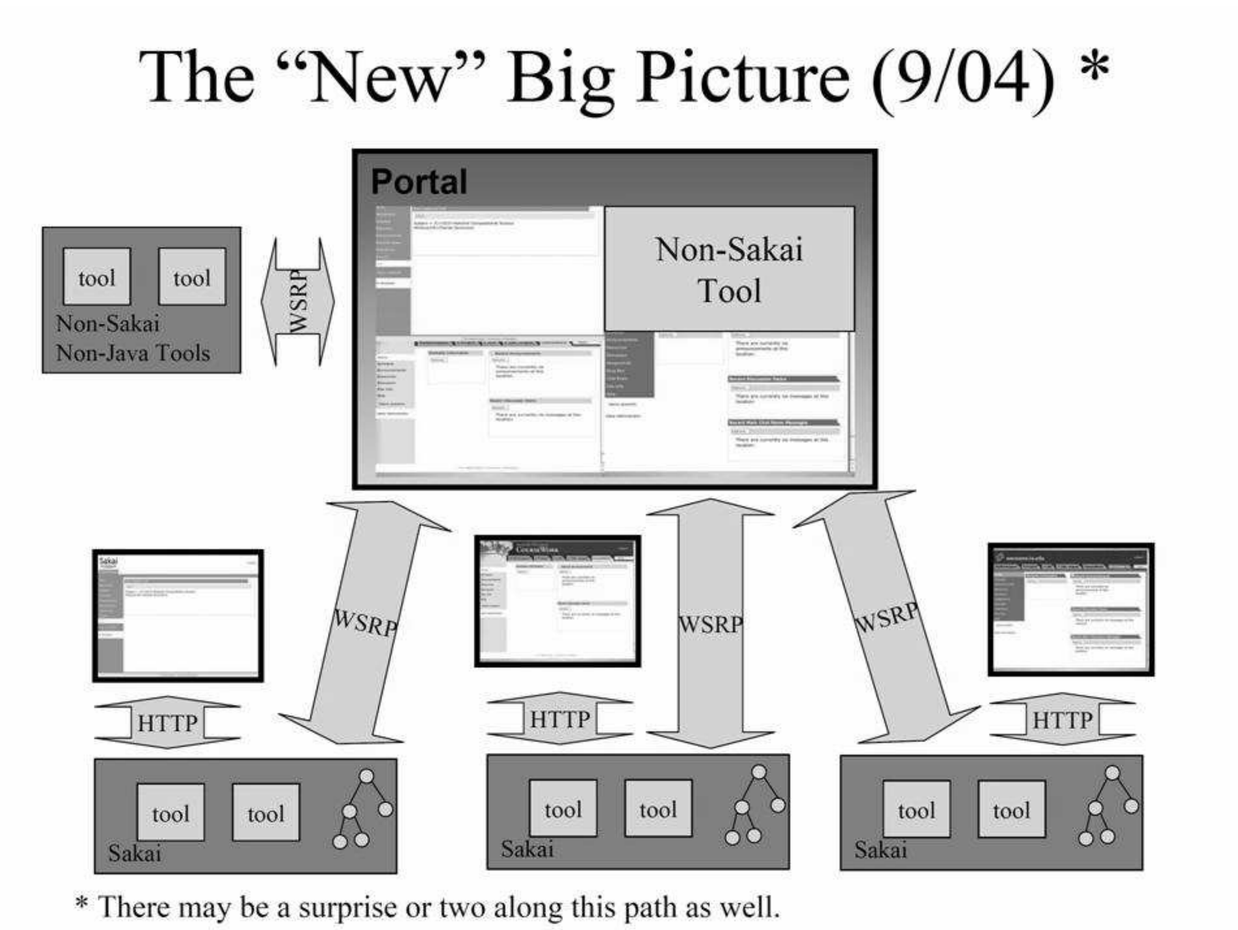

Figure 8: New Big Picture

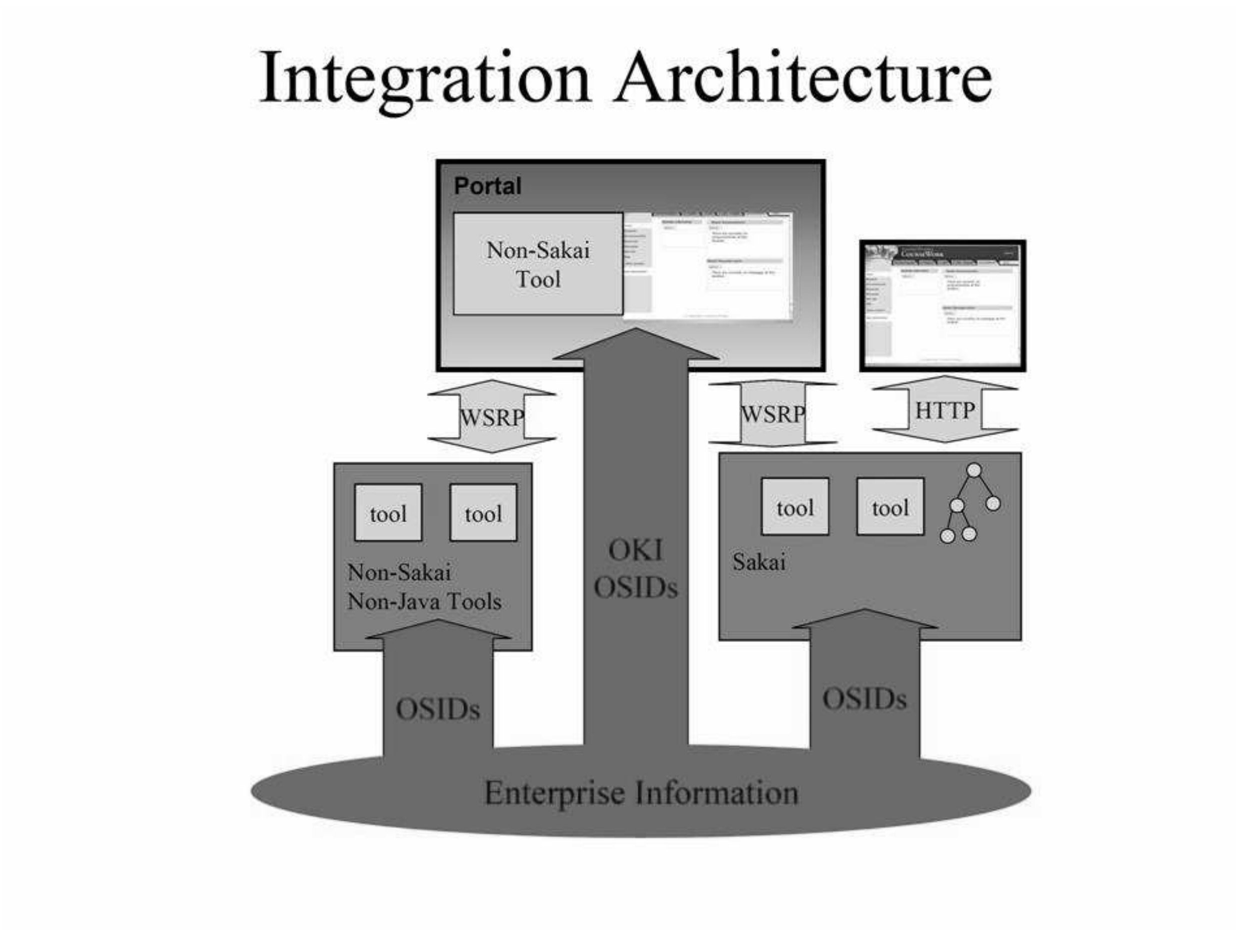

Figure 9: Integration Architecture

6.4 Appropriateness of Sakai in a VRE

`cite some of the commercial portal comments...`

`VLE focus on portals`

The HIVE approach to e-Research presented in this report can be applied in many other contexts. In e-learning we could for example use it to construct a GRID of distributed content that could be dynamically aggregated in the ways required by the user for each learning situation they face. The content of each HIVE instance could be watermarked to identify its origin. This use of the HIVE would require the development new tools, e.g. cross searching tools. Eventually we could use the HIVE to coalesce the appropriate combinations of information, e-Learning, e-Research, e-Collaboration, e-Management, e-Authoring and e-Publishing, e-Leisure tools as required by our current activity.

6.5 Other Frameworks

Clearly a VRE is more than just a portal! We believe that portal and portlet technology has a major role to play in delivering VRE services and tools, particularly the collaboration tools, to end users, but existing applications, e.g. GUIs must also access VRE functionality. In order to do this a lightweight toolkit is required to be downloaded onto workstations and PCs which will avoid the "client problem" of bloated and difficult software installations and associated firewall issues.

A need has been identified within the e-Science community for a client toolkit which can provide very light-weight but extensible access to Grid resources GROWL: *Grid Resources on Workstation Library* `http://www.grids.ac.uk/GROWL` is a prototype library which could be used for this evaluation work. We are initially creating libraries in C/ C++ and R, interfacing to a set of existing services derived from HPCPortal, DataPortal and InfoPortal which are part of IeSE. It should be possible to install this library on a variety of client workstations with a minimum of additional software. The library is targetted at existing applications in physics, chemistry and statistics.

Presentation of services through both portlets into the portal framework and as a programming library can be achieved using language-agnostic Web service client interfaces to the VRE exposed in a Service Oriented Architecture. Early work with GROWL at Daresbury and Lancaster has shown that this is feasible in C and R languages in addition to Java. "Heritage" applications can in this way be Grid enabled or themselves exposed (wrapped) as remote services. This is a response to the recent discussion of lightweight toolkits by existing and potential Grid users, some of which is referenced on the Web site. It is also a crucial step in bringing such applications into play for a wider research community and is a step to achieving inter-disciplinary research agendas.

A number of evaluation tasks listed in the table below were proposed to JCSR, these were however deemed to require too much effort and involved too much development for this initial evaluation. We still believe strongly that they must be caried out to yield valuable prototypes of alternatice VRE integration methodology.

Task	Title	Responsibility	Description
6.1	C-GROWL	Daresbury	Link the prototype GROWL C library with SEE services
6.2	R-GROWL	Lancaster	Link the prototype GROWL R library with SEE services
6.3	R heritage	Lancaster	Expose an R statistical analysis package, e.g. SABRE, as a SEE service
6.4	C/ Fortran heritage	Daresbury	Expose a Fortran program as a SEE service

`Bees...`

Ref. Olivier paper again.

`e.g. GROWL`

6.6 Back-end Resources and Services required in a VRE

A lot of effort in this project has gone into considering the integration of existing resources and services throught the architecture proposed here. This proposal has emerged alongside very similar ones for the JISC VLE and IE programmes. All teams are in contact with the same American and European developers. A list of UK resources which *should* be integrated into the VRE were included in paper [29] and are repeated in Appendix F. Similarly the services which we believe *should* be available via a VRE are listed in Appedix G and a fuller version is being kept up to date on the ETF Web site `http://www.grids.ac.uk/ETF/public/WebServices/classes.html`.

It is important for the tools in the VRE to be able to access all the underlying resources that a particular group of collaborating researchers would require. These include computer systems, databases, data/ information collections, application codes and instruments for on-line observation and data recording and annotation. Some of the resources to which the above tools will provide immediate access include:

- HPCPortal and InfoPortal functionality applied to NGS, JCSR Grid clusters and researchers' own facilities;
- Network Monitoring tools;
- DataPortal cross search tools for scientific data;
- CCLRC's Atlas Data Store;
- HPCx capability compute facility;
- Sample experimental facilities on Daresbury Synchrotron Radiation Source;
- NGS Nodes;
- ReDReSS training and awareness content and services.

Indirect access would be provided to:

- RDN data via SPP cross search tools;
- MIMAS, Manchester Information and Associated Services;
- CSAR computing and data facilities.

We would also seek to bring in other resources and services if a Sakai demonstrator project were funded. Further information is provided in Appendixresources-sec.

6.7 UK Sakai Alliance

The idea of a "Sakai Alliance" has arisen in discussion with other UK members of the e-Science and Virtual Learning communities who are interested in developing and porting tools into a portlet framework. Sakai is the leading example of such a Java framework built upon the open source uPortal hosting container. It also builds upon the JSR-168 and WSRP portlet standards and the OKI OSIDs which could be extended for a variety of purposes as described above.

Participants in JISC VRE funded projects who are interested in portal delivery could be invited to join the Alliance as a means of informing each other. The aim of the Sakai Alliance is to ultimately allow the UK developers to play a recognisable part in the wider Sakai development community in a similar way to that in which the UK participates in the Globus Alliance.

The Terms of Reference of the Sakai Alliance will be formulated in discussion with early participants and subjected for agreement at an open meeting.

The Sakai Alliance will host an open research group platform for UK HE based on the Sakai platform. Anyone will be able to form a group to support their research group in a similar fashion to Yahoo Groups. Initially there will be light authentication of users as the demand for the service is tested, but the Alliance will develop an implementation of a distributed trust authentication architecture such as Shibboleth and integrate it into the platform so that individual Universities accredit the researchers.

If the tool proves popular, the Alliance will explore sustainability models but the partners offer to keep the basic service running for a minimum of 5 years. Basic service means deployment and maintenance of Sakai production release 2.0. Again if the tool proves popular, it is likely that a number of enhancements will be requested. These may be developed by the Alliance and added to the service as part of other proposals not covered here. The Alliance will seek to QA and deploy open source extensions of Sakai on request, subject to a compatible license and available resource. This process will be in line with the development and test cycle proposed for the SEPP contributors.

The vision is an open resource for all of UK HE inter-institutional research groups which does not depend on limited time project finance and institutional resources for sustainability and so which enhances inter-organisational research collaboration and better preserves and disseminates the research outputs.

7 Acknowledgements

We acknowledge the following people who have freely provided their comments and input to the work reported here.

Alison Allden, Mark Baker, David Boyd, David de Roure, Geoffrey Fox, Dennis Gannon, Liz Lyon, Mark Norton, Maron Pierce, Andy Powell, Alan Robiette, Charles Severance, Mary Thomas, Scott Wilson,

We especially thank Mary Thomas, Dennis Gannon and Charles Severance for their continual guidance and help.

References

[1] R.J. Allan *Survey of Computational Grid, Meta-computing and Network-based Tools* Edition 2 Technical Report DL-TR-99-02 (Daresbury Laboratory, February 2000) `http://esc.dl.ac.uk/StarterKit`

[2] R.J. Allan *Resource Selection and Billing via Agents, Brokering and a Grid Hub* (CLRC e-Science Centre, 2000) draft

[3] R.J. Allan, D. Hanlon, R.F. Fowler and C. Greenough *A Globus Developers' Guide with Installation and Maintenance Hints* Draft Technical Report, UKHEC Collaboration and CLRC e-Science Centre (2001) `http://www.dl.ac.uk/TCSC/UKHEC/GridWG`

[4] F. Costen, J.M. Brooke, R.J. Allan and M. Westhead *Grid-based High Performance Computing* (UKHEC, 2000) `http://www.ukhec.ac.uk`

[5] R.J. Allan, S. Pickles, J.M. Brooke, F. Costen and M. Westhead *UK HEC Grid: background and first steps* UKHEC Report to the Research Councils (UKHEC 2000) `http://www.ukhec.ac.uk`

[6] R.J. Allan, S.J. Andrews, M.F. Guest, P.M. Oliver, D. Henty, L. Smith, S. Telford and S. Booth *Designing and Building Beowulf-class Cluster Computers* (UKHEC 2000) `http://www.ukhec.ac.uk`

[7] R.J. Allan *Developing a Web Portal for the Computational Grid* Draft Technical Report, UKHEC Collaboration and CLRC e-Science Centre (2001) `http://esc.dl.ac.uk/GridWG`

[8] R.J. Allan *Developing Web Portals for the UK e-Science Grid.* technical report (UKHEC Collaboration and CCLRC e-Science Centre, 2002) `http://esc.dl.ac.uk/GridWG` `http://www.ukhec.ac.uk/publications/reports/portal_guide.pdf`.

[9] R.J. Allan, D.R.S. Boyd, T. Folkes, C. Greenough, D. Hanlon, R.P. Middleton and R.A. Sansum *Evaluation of Globus and Associated Grid Middle-ware.* Report to the DTI (CLRC e-Science Centre, May 2001) `http://esc.dl.ac.uk/StarterKit`

[10] R.J. Allan and D.R.S. Boyd *Some Architectural Issues to be considered in building a UK Grid* Report for the UK Grid Engineering Task Force (CLRC 2001)

[11] R.J. Allan, D.R.S. Boyd, T. Folkes, C. Greenough, D. Hanlon, R.P. Middleton, R.A. Sansum, Elson Mourão, Rob Baxter, David Henty, Mark Parsons, John Brooke, W.T. Hewitt, Mike Daw, Jon MacLaren, Jon Gibson, Graham Riley and Stephen Pickles *Globus and Associated Grid Middleware. Consolidated Evaluation Report from UKHEC Sites.* (UKHEC, December 2001) `http://esc.dl.ac.uk/StarterKit`

[12] R.J. Allan, S.J. Andrews and M.F. Guest *HPCGrid* the national magazine for HPC applications (Daresbury Laboratory, twice yearly) ISSN 1473-2076 `http://www.cse.clrc.ac.uk/Activity/HPCProfile`

[13] M. Baker, H. Ong and G. Smith *A Report on Experiences operating the Globus Toolkit through a Firewall* (University of Portsmouth, 2001) `http://www.dl.ac.uk/TCSC/UKHEC/GridWG`

[14] M. Baker, H. Ong and G. Smith *A Status Report on developing a Grid-enabled Web Portal using Commodity Toolkits* (University of Portsmouth, 2001) http://www.dl.ac.uk/TCSC/UKHEC/GridWG

M. Baker, H. Ong and G. Smith *A Portal for Liquid Crystal Modelling Applications* (University of Portsmouth, 2002) http://www.dl.ac.uk/TCSC/UKHEC/GridWG

[15] M. Baker, H. Ong and G. Smith *Grid Monitoring Architecture* draft report (University of Portsmouth, 2002)

[16] R.J. Allan, D.J. Hanlon, K. Kleese van Dam, S. Sufi, L. Sastry, D.R.S. Boyd and B.M. Matthews *An Integrated e-Science Environment for CLRC: Grid Portals, HPC Computation, Data Management and Advanced Visualisation* Project Overview (CLRC, 2002), http://www.dl.ac.uk/TCSC/UKHEC/GridWG/

R.J. Allan, K. Kleese van Dam, L. Sastry, and B.M. Matthews *An Integrated e-Science Environment for CLRC: Grid Portals, HPC Computation, Data Management and Advanced Visualisation* Project Definition, Use Cases and Component Specifications (CLRC, 2002),

[17] O.F. Rana, V. Getov, E. Sharakan, S.J. Newhouse and R.J. Allan *Developing Grid Services with Jini and JXTA* White paper for Jini Working Group presented at GGF4, Toronto (February 2002) http://esc.dl.ac.uk/WebServices

[18] R.J. Allan, D. Chohan, X.D. Wang and M. McKeown *UDDI and WSIL for e-Science* Grid Support Centre (2002) http://esc.dl.ac.uk/WebServices

[19] Rob Allan, Dharmesh Chohan, Xiao Dong Wang and Daniel Hanlon *An Introduction to Web Services and related Technology for building an e-Science Grid* Grid Technology Group, Daresbury Laboratory (2002) http://esc.dl.ac.uk/WebServices

[20] Rob Allan and Xiao Dong Wang *A Review of some Web Services Tools.* Grid Technology Group, Daresbury Laboratory (2002) http://esc.dl.ac.uk/WebServices

[21] R.J. Allan and D.R.S. Boyd *The UK e-Science Grid.* Grid Support Centre (2002)

[22] R.J. Allan et al. *Level 2 Grid – Final Report* (ETF, April 2003)

[23] Rob Allan, Xiao Dong Wang and Andy Richards *XML Schema for HPC Applications and Resources* Grid Technology Group, Daresbury Laboratory (2002) http://esc.dl.ac.uk/WebServices

[24] R.J. Allan et al. *Middleware, Applications and Tools deployed at Level 2* Proc. AHM'2003 (Nottingham, September 2003)

[25] R.P. Tyer, L.J. Blanshard, K.K van Dam, R.J. Allan, A.J. Richards and M.T. Dove *Building the e-Minerals Mini-Grid* Proc. AHM'2003 (Nottingham, September 2003)

[26] A.J. Richards, R.J. Allan, G. Drinkwater and K. Kleese *Advanced Grid Computing* Proc. AHM'2003 (Nottingham, September 2003)

[27] R.J. Allan, D. Chohan, X.D. Wang, A.J. Richards, M. McKeown, J. Colgrave, M. Dovey, M. Baker and S.M. Fisher *Grid Information Services* Proc. AHM'2003 (Nottingham, September 2003)

[28] Rob Allan, Chris Awre, Mark Baker and Adrian Fish *Portals and Portlets 2003* Proc. NeSC Workshop 14-17/7/2003 http://www.grids.ac.uk/Papers/Portals/portals.pdf (CCLRC and NeSC, 2004)

[29] Rob Allan and Rob Crouchley *Roadmap for a UK Virtual Research Environment* JCSR(04)14. Report of the JISC JCSR VRE Working Group (JISC 2004) http://www.jisc.ac.uk/uploaded_documents/VREroadmapv4.pdf

[30] Mark Baker, Hong Ong, Rob Allan and Xiao Dong Wang *Advanced Portal Services* Proc. AHM'2004 (Nottingham, September 2004) http://www.allhands.org.uk/proceedings/papers/223.pdf

[31] Rob Allan, Rob Crouchley, Adrian Fish and Dharmesh Chohan *Portal Services for Awareness and Training* Proc. AHM'2004 (Nottingham, September 2004) http://www.allhands.org.uk/proceedings/papers/234.pdf

[32] A report on grid computing environments which was an early GGF GCE document: *Overview of Grid Computing Environments* http:/grids.ucs.indiana.edu/ptliupages/publications/ggf-gce-overview2.doc which has lots of survey stuff about Grid portals from the "early days" of 2001. This is also GGF document so you can find the final pdf on the GGF documents web site at: http://forge.gridforum.org/docman2/ViewCategory.php?group_id=90&category_id=28

[33] D. Gannon, G. Fox, M. Pierce, B. Plale, G. von Laszewski, C. Severance, J. Hardin, J. Alameda, M. Thomas, J. Boisseau *Grid Portals: A Scientist's Access Point for Grid Services* GGF Community Practice Document (GCE-RG, 19/9/03) http://forge.gridforum.org/projects/ggf-editor/document/GCE-Portal-working-draft/en/1/GCE-Portal-working-draft.pdf http://www.extreme.indiana.edu/~gannon/ggf-portals-draft.pdf

[34] G. Fox and D.W. Walker *UK e-Science Gap Analysis* (2003)

[35] M. Baker, H. Ong and G. Smith *A Status Report on developing a Grid-enabled Web Portal using Commodity Toolkits* (University of Portsmouth, 2001) http://esc.dl.ac.uk/GridWG

[36] M. Baker, H. Ong and G. Smith *A Portal for Liquid Crystal Modelling Applications* (University of Portsmouth, 2002) http://esc.dl.ac.uk/GridWG

[37] Adrian Fish *ReDRESS Project Review: Survey of Portals and Related Software* (University of Lancaster, February 2004)

[38] D. Gannon, M. Christie, O. Chipara, L. Fang, M. Farrellee, G. Kandaswamy, W. Lie, B. Plale, A. Slominski, A. Saarangi, Y. Simmhan, *Building Grid Services for User Portals*, GGF Workshop on Grid Services, Chicago, (Oct. 8th 2003)

[39] S. Krishnan, R. Bramley, M. Govindaraju, R. Indurkar, A. Slominski, D. Gannon, J. Alameda, D. Alkaire *The XCAT Science Portal* Proceedings SC2001, Nov. 2001, Denver. Extended version in Scientific Programming 10(4) (2002) 303-317

[40] J. Novotny, V. Welch *MyProxy* (NCSA) www.grid.ncsa.uiuc.ed./myproxy/

[41] *Building Collaborative eResearch Environments* Final report compiled for JISC by Andrew Cox (University of Loughborough, March 2004)

[42] *UK Role in Open Grid Services Architecture* draft report v0.7 from the Grid Architecture Task Force (12/3/02)

[43] Liz Lyon and Andy Powell *A Distributed National Electronic Resource Technical Architecture: Scoping the Information Environment* (UKOLN, University of Bath). See also http://www.ukoln.ac.uk/distributed-systems/jisc-ie/arch/

[44] S. Wilson, B. Olivier, S. Jeyes, A. Powell, T. Franklin *A Technical Framework to Support e-Learning* report to JCLT (2003)

[45] R. Crouchley and R.J. Allan *A Service-Oriented Architecture for VRE/ VLE building on the Sakai Framework* Report to JCLT (2004)

[46] Mark J. Norton *A Comparison between the JISC and Sakai Frameworks* (24/3/04)

[47] Michael Stevens *The Benefits of a Service-Oriented Architecture* Developer.com http://www.developer.com/services/article.php/1041191

[48] Michael Stevens *Service-Oriented Architecture Introduction* (2 parts) Developer.com http://www.developer.com/services/article.php/1010451

[49] *Web Services and Service-Oriented Architectures* http://www.service-architecture.com/

[50] Sayed Hashimi *Service-Oriented Architecture Explained* (O'Reilly) http://www.ondotnet.com/pub/a/dotnet/2003/08/18/soa_explained.html

[51] Bill Ruh *Succeeding at Service Oriented Architecture* (ZDNet) http://www.zdnet.com.au/builder/architect/work/story/0,2000034884,20276810,00.htm

[52] *The Arda Project – a Realisation of Distributed Analysis for LHC.* http://lcg.web.cern.ch/LCG/peb/arda/Default.htm

[53] D. Gannon et al. *Xportlets Architecture* (Xtreme Computing Laboratory, University of Indiana, 1/4/2004)

another paper DC found

[54] Web Services interest area of UK Grid Engineering Task Force htmllinkhttp://www.grids.ac.uk/ETF/public/WebServices/index.shtml

[55] registry interest area of UK Grid Engineering Task Force htmllinkhttp://www.grids.ac.uk/ETF/public/registry/index.shtml

[56] Portals and Grid Clients interest area of UK Grid Engineering Task Force http://www.grids.ac.uk/ETF/public/Portals/index.shtml

[57] Savas Parasitidis *GAF: Grid Application Framework* See http://www.grids.ac.uk/ETF/public/WebServices/index.shtml

[58] Mark McKeown *OGSI::Lite* README file with the Perl source from http://www.mvc.man.ac.uk/???/ILCT. See http://www.grids.ac.uk/ETF/public/WebServices/index.shtml

[59] P.V. Coveney and J. Chin *Towards tractable toolkits for the Grid: a plea for light-weight, usable middleware.* http://www.realitygrid.org/lgpaper.html

[60] Bill Olivier *Information Paper for JISC VLE Programme* (JISC, June 2004)

[61] R.J. Allan *GROWL: Grid Resources on Workstation Library* http://www.grids.ac.uk/GROWL

[62] Mary E. Jackson *The Current State of Portal Applications in ARL Libraries* (Association of Research Libraries, May 2004) http://www.arl.org/access/portal/PAWGfinalrpt.pdf

[63] WSRP and UDDI

[64] S. Wilson, K. Blinco and D. Rehak *Service Oriented Frameworks* DEST (Australit, JISC-CETIS (UK and Industry Canada http://www.jisc.ac.uk/uploaded_documents/AltilabServiceOrientedFrameworks.pdf

A WP 1 – Comparing Sakai/ CHEF with Alternative Frameworks for a VRE

This work package compared Sakai/ CHEF with Alternative Frameworks for a VRE. By "framework" we really imply a means to deliver the VRE services in an appropriate way to end users and developers. This was an ongoing task and lasted for the duration of the project. Even though Sakai/ CHEF appears to be the leader as a VRE portal delivery framework at the moment, this may change. This work package resulted in the Technology Report (Evaluation Report 1) below.

Work carried out includes:

Task	Title	Responsibility	Description
1.1	Portals and Portlets 2003	Daresbury	Review projects presented and surveyed in the Portals and Portlets 2003 NeSC workshop and update material
1.2	ReDRESS TR	Lancaster	Review projects identified in the ReDRESS Technology Survey and update material
1.3	Evaluate new work	Both	Ongoing technology watch for related projects, e.g. by liaison with GGF Grid Computing Environments research group, JCLT VLE groups, FPVI EGEE project developers, etc.
1.4	Compile Technology Evaluation Report	Both	Use material to compile report, keep up to date over time of project on Web sites and provide to directors for submission to JCSR and other appropriate groups, e.g. GGF GCE research group

A.1 Portals and Portlets 2003

Review projects presented and surveyed in the Portals and Portlets 2003 NeSC workshop and update material.

A list of public-domain and propriatory portal offerings is given in Appendix H.

A.2 ReDRESS Technology Evaluation Report

This report compares Sakai/ CHEF with Alternative Frameworks for a Virtual Research Environment.

`Adrian's report...`

`update of Portals2003...`

A.3 Evaluate new Work

Ongoing technology watch for related projects, e.g. by liaison with GGF Grid Computing Environments research group, JCLT VLE groups, FPVI EGEE project developers, etc.

A.4 Technology Evaluation Report

Use material to compile report, keep up to date over time of project on Web sites and provide to directors for submission to JCSR and other appropriate groups, e.g. GGF GCE research group.

B WP 2 – Ease of Administration (EoA) of Sakai/ CHEF for a VRE.

This work package assessed the Ease of Administration (EoA) of Sakai/ CHEF for a VRE. Administration facilities in the chosen VRE framework are of paramount importance to its maintenance and extension. There are many questions that we will attempt to answer, e.g. "How easy is it to add new users to Sakai/ CHEF?", "Can we customise the portal 'skin' in appropriate ways for existing e-Science projects?", "Can we easily allow projects to create work sites for subsets of users in the portal and to include the tools they require from the given set?", "Can we isolate users from specific work sites and create a secure access mechanism using certificates (e.g. with GridSite)?" This work package resulted in the production of a report (Evaluation Report 2) together with any software adapters developed during the assessment process. These adapters may be required to plug Sakai/ CHEF into Lancaster/ Daresbury authentication mechanisms and will serve as useful templates for other institutions.

Administration facilities in the chosen VRE framework are of paramount importance to its maintenance and extension. There are questions that we will attempt to answer:

Task	Title	Responsibility	Description
2.1	Add users	Lancaster	"How easy is it to add new users to Sakai/ CHEF?"
2.2	Customisation	Lancaster	"we customise the portal 'skin' appropriate ways for existing e-Science projects?" demonstrators for ReDRESS, e-HTPX, e-Minerals and NGS
2.3	Workspace creation	Lancaster	" we easily allow project to create workspaces for subsets of users in the portal and to include the tools they require from the given set?"
2.4	Security	Both	"Can we isolate users from specific workspaces and create a secure access mechanism using certificates (e.g. with GridSite)?"
2.5	Performance and scalability	Both	Investigate what server is required to host the VRE front end for a large project, e.g. using BladeCenter at Daresbury

B.1 Add Users

In addition to using the built-in Sakai configuration we carried out two tests to show that Sakai can be integrated with existing user management systems.

Lancaster AuthN Integration

CHEF's login mechanism has been modified to use Lancaster University's LDAP directory interface for the primary means of authentication with the CHEF internal database-backed system being used as a secondary method. This means that any user, be they student or staff, can log straight into CHEF using their standard network username and password. The piggy backing of an LDAP directory lookup on CHEF's default mechanism like this means that we can also authenticate external ReDReSS users by adding them to the CHEF user database. The CHEF documentation gives some guidance on how to achieve this. This is what we did:

1. Create a java class that implements the interface *PiggybackUdp* You need to implement all of the methods, including the lookup methods, so that they query your institution's LDAP directory implementation. The Udp doesn't just authenticate users, it is used during worksite creation to check that the participants are known to the CHEF software. This gives a degree of referential integrity to the worksite creation process, so this is why the lookup methods are also important. The CHEF documentation recommends that this go in a "plugin" sub-package.

2. Modify the "compile" target in CHEF's `build.xml` file so that the new class is compiled into the CHEF Web application's classtree.

3. Modify the `chef_dev_nc_resources.properties` file in `src/conf` by changing the plugin component class *SampleUserDirectoryProvider* to the fully qualified name of your newly implemented class.

Lancaster Student Records System Integration

Lancaster uses a students records system called LUSI. This is based on a Microsoft SQL database with a Windows front end for report presentation. It was not possible to gain direct access to this system for the evaluation, so a similar test database has been created. This is linked to a CHEF service called *StudentRecordsService* which has three methods, *getCourseList()*,*getStudentList()* and *getTeacherList()*, the latter two of which take a course id as their sole parameter. This was wired into the Turbine framework using the CHEF configuration files. To permit the adding of a course worksite the *UmiacClient* functionality was replaced with calls to the custom *StudentRecordsService*. This is done, within the *SiteAction* class, by modifying the *finishCourseSite()* and *addNewSite()* methods.

B.2 Customisation

"we customise the portal 'skin' appropriate ways for existing e-Science projects?" demonstrators for ReDRESS, e-HTPX, e-Minerals and NGS

CHEF has been re-skined to more closely resemble a Lancaster University Web site. This reskinning has been lightweight in that none of the layouts have been changed, only colours and graphics.

`Adrian ???`

B.3 Workspace creation

" we easily allow project to create workspaces for subsets of users in the portal and to include the tools they require from the given set?"

B.4 Security

"Can we isolate users from specific workspaces and create a secure access mechanism using certificates (e.g. with GridSite)?".

B.5 Performance and scalability

Investigate what server is required to host the VRE front end for a large project, e.g. using BladeCenter at Daresbury

C WP 3 – Feasibility of making existing VRE (Grid) Components available in Sakai/ CHEF.

This work package was aimed at establishing the feasibility of making existing VRE (Grid) components available in Sakai/ CHEF. The project partners at Lancaster and Daresbury had access to a range of research tools that could be integrated into a VRE framework. We investigated the feasibility of creating an integrated suite of such tools, in order to determine whether there were any generic problems we needed to forewarned about. This work package is closely coupled with WP 4. This work package contributed to a report (Evaluation Report 2) together with any software wrappers employed in making Daresbury's GRID tools available in the Sakai/ CHEF framework. These wrappers will serve as useful templates for other GRID tools.

The project partners currently have access to a suite of separate tools that may well have their usefulness enhanced by integration into a VRE framework. Work will be done on the construction of a demonstrator, with a suite of such tools integrated. It is anticipated that the construction of such a demonstrator will be the optimal way to establish feasibility and get feedback from users in sample e-Science projects. This work package is closely coupled with WP 4, but we list the components here.

Work carried out includes:

Task	Title	Responsibility	Description
3.1	InfoPortal	Daresbury	Grid Information Services and XML metadata about projects, users, applications and resources
3.2	HPCPortal	Daresbury	"Active" services for authentication, file management, data management, workspace, job submission and applications
3.3	CCF Whiteboard	Daresbury	Collaborative Computing Framework being deployed in the e-Minerals project from U. Reading (UK) and U. Emory (USA). Collaboration with Vassil Alexandrov (Reading)
3.4	SPP Cross search	Lancaster	Subject Portals Project portlet for RND resource cross searching. Collaboration with SPP project developers (Bristol, Bath, Oxford)

Sakai is an ambitious project which already comprises a rich set of tools and templates for extending the set. Pre-built collaboration tools include: Worksite membership management, Schedule (calendar), Resources, News, Discussion, Chat, Anabas Impromptu interface, Newsgroup, Bibtex, eMail, DropBox, Personal Customisation. Another set of tools already exists from the OGCE portal: Proxy Manager, GRAM Job Launcher, GridPing, Grid Job Submission, LDAP Browser, GridFTP, GridContext, GPIR, CSF Job, OGE Jobs, Workflow, Application Manager, Condor Job. Sakai also has a comprehensive and growing set of VLE tools developed in the Sakai Educational Partners' Programme, see `http://www.sakaiproject.org`.

Discussions have indicated a number of additional tools which could (should) be developed and integrated into a Sakai-based VRE framework. We are aware that work is ongoing both in the UK, USA and elsewhere to develop appropriate Java-based tools in many of these areas which could be adapted quickly. Further information about this is provided in Appendix E.

Search Tools: Context-based Information Retrieval, Collection Cross Search, Google Web service interface, Application Discovery, Scientific Data Query, UDDI, other information management tools as appropriate;

Collaboration/ Discussion: IRC (Internet Relay Chat), Blog, Address Book, Wiki, Research Scheduler, Video Conferencing, Visualisation, Distributed Whiteboard, Flowchart Editor, Distributed Display, Graphing/ Charting;

Anaysis Tools: Scientific Calculator, Active Spreadsheet, Statistical Computing, Data Management, Simple Visualisation, Grid Information, Network Information, Grid computing;

Publication/ Documentation: Text Editor, Word Processor, Distributed Presentation Tool, Bibliography Builder, XHTML Editor, LaTeX Processor, Simple Interactive Plotting, Data and Metadata Upload and Harvesting, Project Publication, Application Publishing, UDDI;

Others: Portal Statistics, Personal Information Manager (Chandler?), e-Notebook, Shopping Cart, Bugzilla, CVS, Portal Workspace, Authorisation Policy Management;

Training and Awareness: Personal Information Guide (PIG), CopperCore Content Player, Tool Templates, Documentation Manager, Helpdesk interface, other e-Learning tools as appropriate.

```
Geoffrey Fox has a project to develop OpenOffice tool...

Open GroupWare (John Norman)

Digital Information tools via Oxford?  - Z39.50 and SRW, SPP?
```

C.1 InfoPortal

InfoPortal has Grid Information Services and XML metadata about projects, users, applications and resources. InfoPortal and HPCPortal outcomes are discussed together below as this has become a separate project to implement a single portal for NGS users.

C.2 HPCPortal and NGS Portal

HPCPortal has “Active” services for authentication, file management, data management, workspace, job submission and applications.

One aim of the ReDReSS project is to access Grid resources to enable live demonstrations of the sort of application social scientists might want. We are also developing a prototype portal for the National Grid Service (NGS) to provide easy access to resources at CCLRC an the Universities of Leeds, Oxford and Manchester. To enable this we have ported a couple of tools from HPCPortal [26] into the CHEF framework. This was made easier by the fact that the US NMI portal, OGCE [**?**], is already using CHEF.

URL feeds from InfoPortal resource status display panels were easily added to CHEF and retain all original functionality. Active services previously found in HPCPortal were included by porting the relevant portlets from OGCE.

A National Grid Service portal is being developed using the CHEF/ Sakai framework for use in a production environment. The NGS portal will consists of the standard CHEF collaboration tools such as Welcome page; Announcements for posting current, time-critical information; Schedule for posting and viewing deadlines, meetings, events, etc; Resources for posting documents, URLs, etc; Discussion for conversation in written form; and Work Site to allow NGS members to setup their own project workspace either as a private Web page or allowing other members to participate in their workspace forming a VO. In addition, Grid tools are being provided to allow users to perform Grid related activities such as Grid Proxy Delegation for creating the x.509 proxy certificate; Grid Resource Broker for job submission; LDAP Browser for resource discovery and GridFTP for file/ data tranfer between the NGS compute and data nodes. examples of these Grid tools had been developed as xportlets by the OGCE team [**?**, 53]. we have shown that a new Grid tool can be easily integrated into the CHEF/ Sakai portal by adding the tool definition into the XML site description table, adding the tool portlet in the CHEF/ Sakai portlet registry and finally deploying the tool portlet on Apache Tomcat.

`Xportlets...`

`Dharmesh ???`

Review of the Issues for Building Standards Compliant Portlets

`Dharmesh ???`

Assessment of the Potential of Sakai/CHEF as a Platform for Customised Portals, e.g. ReDReSS, NCeSS, e-HTPX, e-Minerals and NGS

`Me ???`

Template for Grid tool wrappers for use in Sakai/ CHEF

`Dharmesh ???`

C.3 CCF Whiteboard

Collaborative Computing Framework being deployed in the e-Minerals project from U. Reading (UK) and U. Emory (USA). Collaboration with Vassil Alexandrov (Reading)

The port of CCF Whiteboard software into a portlet could not be completed as development of the software has not been continued and the e-Minerals project is no longer associated with it. Instead a portlet tool has been developed to interface to CooperCore (see below).

C.4 SPP Cross search

Subject Portals Project portlet for RND resource cross searching. Collaboration with SPP project developers (Bristol, Bath, Oxford).

C.5 CopperCore Integration

CHEF has been loosely coupled with the CopperCore IMS Learning Design player from the Open University of the Netherlands. CopperCore is implemented as a set of enterprise Java beans, and as such requires a bean container to run. JBoss is the bean container of choice for the CopperCore team and this is what we installed. We followed the instructions on the CopperCore Web site [**?**]. Our aim was to enable the attachment of a number of units of learning to a CHEF course worksite for ReDReSS, via a Web-based tool. The mechanisms employed to do this involved JNDI lookups on the JBoss component registry, to retrieve the appropriate CopperCore stub objects; coupled with extensions to the CHEF worksite modification tool *SiteAction.java* that used these stubs to access the CopperCore engine. We modified the CHEF worksite tool by adding a link pointing to a template containing the upload form. We then added a command to the *SiteAction* class to take the uploaded file, save it to disk, and then pass the file to the CopperCore *LDCourseManager* stub for publishing. To enable the transparent addition of all the "access" level CHEF worksite participants to the "learner" role on each unit of learning we had to modify the *LDCourseManagerBean* and its corresponding interface *LDCourseManager* to enable the retrieval of the learner role ids for a supplied Uol (Unit of learning) id. The final stage of the exercise involved the implementation of a worksite tool enumerating the available units of learning and allowing the subsequent playback of the learning material.

D WP 4 – Issues involved in extending the Functionality of Sakai/ CHEF

This work package was aimed at establishing the issues involved in extending the functionality of Sakai/ CHEF particularly to use Web services for distributed development and deployment. There are some key VRE questions that need to be clearly formulated and which, to our knowledge, are outside the scope of the current Sakai project. One involves the skill set required to extend Sakai/ CHEF functionality and whether this is available in the UK. The degree to which Sakai/ CHEF is standards compliant also needs to be investigated, with a particular attention to standards relating to integration of external tools and content. Sakai is focussing on the OKI OSIDs, but will need to also work with other interfaces. It is anticipated that there will be an inverse relationship between such compliance and the need for substantial extensions to the framework software. A final and key question is how Sakai can be used as a front end to distributed services as envisaged in a service-oriented VRE. This work package contributed to a report (Evaluation Report 2) and developed and architecture and prototype software wrappers to extend the functionality of the Sakai/ CHEF framework. These wrappers could serve as useful templates for other extensions.

An architecture has been proposed following discussions with Sakai technical director Charles Severance and others and will be instantiated for some of the test services listed in WP 3. The outcomes of these discussions are described in the concluding section.

Work carried out includes:

Task	Title	Responsibility	Description
4.1	WS Architecture	Both	Develop and agree WS-based Open Service Architecture linked to Sakai. Collaboration with Charles Severance (U. Michigan, USA). Input from Geoffrey Fox (U. Indiana) and other members of the GGF Grid Computing Environments research group.
4.2	Implement	Both	Instantiate this architecture to access some of the Grid tools in WP 3.
4.3	Technical Spec	Both	Write up technical specification of this work package so that other services can be incorporated via portlets
5.1	Narada Messaging	Lancaster	Scalable message broker and interfaces from U. Indiana (USA). Collaboration with Geoffrey Fox
5.2	Video Conferencing	Lancaster	Video conference, H323, Access Grid, and XSGP interfaces from U. Indiana (USA). Collaboration with Geoffrey Fox

Items 5.1 and 5.2 were deemed by JISC to require too much effort and involved too much development for this initial evaluation.

An initial technical specification following a number of discussions with relevant developers was presented to the JCSR Committee in a paper on 16th June 2004 [29]. The outcomes of additional discussions with the Sakai developers and the Indiana Xportlets group are described in the concluding section.

Work on the ReDReSS and other e-Science projects has indicated the need for extensions to the existing Sakai framework and tools additional to those already provided in CHEF/ Sakai. Our bids to the JISC VRE call will focus on achieving these goals as noted below.

After this we report on progress with evaluation and development of a couple of the tools for Sakai. The outcome of these initial evaluations is promising as we are gaining increasing knoweldge of working with the CHEF framework code base. The release of Sakai 1.0.b2 on 23rd July will enable us to provide a fuller report on this work at the AllHands Conference in September.

D.1 Web Services and VRE Extensions to Sakai Framework

In discussion with the Sakai development team, we have identified four generic areas for a demonstrator VRE project to enable a wide selection of tools to be integrated as illustrated above. Framework extensions would be made to accommodate emerging authentication and authorisation systems and SOAP-based interaction with remote services such as Web services WS-I and WSRF Grid services as well as peer-to-peer services. These will be included as an "Integration API" which could eventually extend the OKI OSIDs. These extensions are focussed on enabling a portal, such as Sakai, to work in a distributed Service Oriented Architecture. This could be by accessing remote Grid or other services, aggregating content, or enabling communication between portals in a way which differs from the usual client-server model, e.g. peer-to-peer. This breaks the potential bottleneck associated with the usual monolithic portal interface (although in practice that has not been seen to be a problem with thousands of users of institutional portals based on CHEF).

In the evaluation we have attempted to investigate the design of a generic JSF interface to Web services via WSDL and a UDDI registry.

Identification, or specification, of an XML grammar for describing of collaborative research. This work is concerned with the problem of describing a collaborative research session in a standard, easily machine parsable fashion. Firstly we need to be able to describe the time of the proposed session. We then need to be able to describe its subject matter in as rich a set of terms as possible. This contextual description will be utilised in other tools, e.g. for information retrieval. Finally, we need to be able to describe the participants. The *eduPerson* initiative `http://www.educause.edu/eduperson/` will be assessed as a suitable source for a descriptive grammar. The *eduPerson* initiative is aimed at providing a standard way for institutions to list individuals in their LDAP directories. One of the outputs of the *eduPerson* project will be a controlled vocabulary that could be used in an XML grammar such as the one proposed here. Some background work has been done by the project partners to extend the UDDI service registry schema to describe projects, people, resources and applications `http://www.grids.ac.uk/Papers/Schema/schema.pdf`. Other input will be taken from the CCF, Collaborative Computing Frameworks joint project between University of Reading and Emory, USA.

A Service Authentication and Identity Verification System. A concern with establishing virtual collaborations, is one of identity. How can you be reasonably sure that the colleague you are working with 500 miles away is the person they say they are? Any Grid tools accessed via Web services are likely to require Grid Security Infrastructure (GSI) type or other appropriate authentication. This work package will implement a Shibboleth Federation (see `http://shibboleth.internet2.edu/`) consisting of the collaborators' institutions. Access to the VRE will be protected by a resource manager, which will delegate authentication to the callee's institution. The keys exchanged during the Shibboleth authentication phase will then be used for generation of the appropriate Grid credentials required for use of the Grid tools contained within the VRE. This will build on the JISC evaluation in which Sakai has already been linked to an institutional LDAP people system. Other technology could be harvested from HPCPortal or OGCE which use x.509 extension certificates and a MyProxy certificate repository hosted at the UK Grid Support Centre `http://myproxy.grid-support.ac.uk`.

Another concern is with establishing appropriate authorisation mechanisms respecting local policies. PERMIS provides a system using a role-based set of authorisation policies. It is the job of the tool provider to outline the privileges of the different roles and to the remote site to assign roles to its users. We will seek input from JISC-funded Security Middleware projects on this and related matters. A separate bid is being prepared to enhance the management interface for PERMIS in these areas for use with Sakai by the Salford group.

A JSF-based Web service interface generator. If we wish to provide access to Web services as tools within the Sakai framework, we need to provide a user interface for parameter input. Within Sakai such interfaces are rendered using a pipeline consisting of an abstract XML layout description and a final Java Server Faces user interface. This work package will produce code that generates such an interface from the WSDL file of a desired Web service. All that will be required is the URL of the service WSDL. We will seek input from the Indiana Xportlets group who have done related work `xportlets`.

Service registries are a key technology enabling shared development, code distribution and re-use and version management. This functionality will be capable of being coupled with UDDI registry lookups so that a VRE user will be able to search for appropriate Web services to integrate as tools, select the most desirable ones and have user interfaces transparently created there and then. Prototype UDDI servers are being hosted at Daresbury and Oxford. Background work is reported in `http://www.grids.ac.uk/Papers/UDDI/uddi.pdf`.

This work will have to be closely coupled with the identity work as the credentials gathered at Sakai logon will have to be passed onto any Web services requiring authentication. This is done in HPCPortal and GROWL using agents, a session key and MyProxy and similarly in the GT3.9.1 implementation of WSRF.

Interface to Peer to Peer Tools This task is to provide generic interfaces to integrate tools using peer-to-peer technology such as JXTA in a straightforward way. Its scope is largely still to be defined, but it is a response to the emergence of peer-to-peer tools such as LionShare which would ideally be accessible in a VRE through a single interface. The interface will also support the peer-to-peer communication between portlets in multiple portal instances (another form of aggregation). A complementary proposal to use P2P services to establish a VRE is being submitted by the Reading and Westminster groups. Background work is reported in `http://www.grids.ac.uk/Papers/Rana/rana.pdf`.

D.2 Implementation

Instantiate this architecture to access some of the Grid tools in WP 3.

D.3 Technical Specification

Write up technical specification of this work package so that other services can be incorporated via portlets.

E Integration of UK VRE Tools into Sakai

In this section we describe a number of tools which have been discussed during the evaluation and which we believe would form useful components of a Sakai-based VRE.

`Geoffrey Fox has a project to develop OpenOffice tool...`

`Open Groupware`

`Digital Information tools via Oxford? - Z39.50 and SRW, SPP?`

CopperCore ???

e-Learning Tools as appropriate

IRC Chat Tool. (Lancaster) IRC is a widely used protocol for on-line chat sessions including group or 1 on 1. Users can choose between several threads. A Chat portlet is already present in the Sakai framework but work is needed to integrate chat sessions from project worksites into a user's personal worksite and to link to external IRC channels. `Jabber is an alternative.`

Blog. (Daresbury) A candidate exists, Blojsom at `http://www.blojsom.com`.

Programmable Scientific Calculator. (Daresbury)

Active Spreadsheet. (Daresbury)

Personal Address Book. (?)

Personal Information Guide (PIG). (?) Office Assistant and help tool in the form of a flying pig.

Helpdesk Interface

Distributed Word Processor. (?) A distributed word processing tool to facilitate collaborative document editing.

Shopping Cart. (Daresbury) Adaptation of the personal resource tool designed to package data and executable programs for processing, e.g. via the Grid.

Distributed Presentation Tool. (?) A distributed presentation tool for multicasting slide shows to co-researchers.

Wiki. (Lancaster) Java-based portlet interface to popular collaborative Web editing tool. A prototype exists from Michael Fischer's computational anthropology group, University of Kent.

Research Scheduling Tool. (Lancaster/ Daresbury) This package will involve the development of software adapters intended to query potential co-researcher's Exchange (or any iCalendar compliant mail server) calendars. A software module will be produced designed to query potential co-researcher's calendars and list potential research time slots.

Cross Search Tool. (Oxford) An appropriate tool for cross searching digital resource collections is available as a portlet from the Subject Portal Project. This currently searches the RDN, Resource Discovery Network, collections.

Relevant Information Retrieval Tool. (Lancaster/ Daresbury) This work package will focus on the delivery of a WSRP portlet based tool enabling lookup of resources relevant to the research being undertaken in the VRE at the current time. This focusses on context based content aggregation using the output of the grammar extension deliverable. It will be used in that the subject matter portion of the VRE session description to help generate the information search criteria. Each researcher involved in a project will be able to manage content and publish it for other researchers who can retrieve it from other portal instances. Recent work has shown how WSRP services can be published via UDDI `FPB subgroup of WSRP OASIS technical group`. There are relevant Web-based UDDI clients which could form the basis of this tool.

Personal and Project Notebooks. (Daresbury) This will be an adaptation of the e-Notebook tool from ??? Database backed 'blog' style tools will be produced to enable the making of research notes. `An american project has one...`

Personal Information Manager. e.g. for sharing schedules with other software. Example, Chandler.

Project Bibliography Construction. (Daresbury) Tool to enable collaborators to upload bibliographies from their work machine into a shared project bibliography. A BibTeX tool exists in OGCE which can be adapted to produce output in different formats.

A Statistical Computing Web Service (SABRE and R). (Lancaster) Work package 2 is concerned with creating a resource producer, in the form of a web service version of the R and Sabre statistical tools. The Sabre routines will be made available via R calls, and these in turn will be made available as a multiple message web service. A prototype of this has been demonstrated in the GROWL project at Daresbury, see `http://esc.dl.ac.uk/GROWL`. This service will be able to run on the National Grid Service (JCSR nodes at Manchester, Leeds, Oxford, RAL) and will require grid authentication via Sakai's Grid logon facilities, as enhanced in the grammar. Initially, a dataset will be uploaded from a collaborator's computer to the service for processing, although the possibility of remote dataset specification will be investigated. GridFTP or SRB client calls would be used for this.

Video Conferencing and AGN Participation. (All) This work package will build on the work done by Geoffrey Fox from Indiana on the Narada Brokering based GlobalMMCS video conferencing system. It is anticipated that the bulk of the work will involve configuring GlobalMMCS, devising an elegant deployment solution and coding the conference instantiation facilities into CHEF. GlobalMMCS uses a message exchange protocol based on SOAP for setting up conferences. Communications are then carried out using message packets, which can be shunted over a self-organising network overlay of Narada Brokering nodes. A browser borne video conferencing tool tool, for use by VRE research project members will be delivered.

Scientific Data Management. (Daresbury) We will port CCLRC's DataPortal JSPs to portlets to provide cross-search access to a range of scientific data via a common metadata model. The "shopping cart" portlet will connect a list of identified datasets to a GridFTP facility or other functions.

Visualisation. (Daresbury/ RAL) Visualisation tools will be imported from other projects, for instance the e-Science Visualisation Group at CCLRC Rutherford-Appleton Laboratory and Ken Brodlie, University of Leeds, which provides tools for the GODIVA, GViz and Integrative Biology projects and University of Manchester's SVE Group.

Distributed Whiteboard. (Lancaster/ Reading) Input will be taken from the CCF, Collaborative Computing Frameworks joint project between University of Reading and Emory, USA which includes a multicast-enabled "clearboard" and other collaboration tools. A distributed "whiteboard" tool is required, to facilitate collaborative discussion and annotation of research materials such as images and documents.

Distributed Flowchart Editor. (Lancaster) A distributed flowcharting tool to assist in the planning of research activities.

Text Editor Emacs style editor

XHTML Editor

Distributed Display. (Lancaster) A distributed display tool for multicasting live data from HPC.

Bugzilla. Collect bug reports and feedback to developers.

CVS. A Web front end for CVS exists.

TeX/ LaTeX Processor Back-end text processing and rendering linking to the BibTeX tool.

Simple Interactive Plotting. e.g. GNUplot for uploading and graphing 2D datasets.

Portal Statistics. Extension of portal group collaboration tools to support gathering of usage statistics. Similar tools exist in PHP-Nuke and are very popular.

Graphing/ charting Web service Takes a dataset and graphs it. Could use the graphing functionality of R behind the scenes.

Extended presence tool Shows co-researcher's location on a map. Hover over the point and you get metadata about that individual (like contact details).

Grid Information

Network Information

Grid Computing

Data and Metadata Upload and Harvesting

Project Publishing

Application Publishing

UDDI

Portal Workspace Manager

Authorisation Policy Manager

Documentation Policy Manager

F Resources

Some resources and facilities available to the academic research community were identified during the VRE work with JCSR:

- Access Grid Nodes (e-Science Centres);
- Course Content (University and Training Institutions);
- Condor pools of workstations (University and Teaching institutions);
- Resource Discovery Network resources (JCIE) http://www.rdn.ac.uk/. See the RDN Internet Resource Catalogue http://www.jisc.ac.uk/index.cfm?name=rdnipage
- AHDS (AHRB) and e-SS (ESRC) and related training and awareness material, e.g. REDRESS;
- Directories: Z-Directory (UKOLN), Z39-50 target directory (Index Data), RSS-express (UKOLN), OAI Data providers (OAI), IESR (JISC)
- Text mining service (BBSRC), Data Curation Centre and any other specific research resources funded in partnership with Research Councils;
- Resources referenced in the JISC subject resources guides http://www.jisc.ac.uk/index.cfm?name=resguides. These cover the seven subject areas: Arts and Humanities; Engineering, Mathematics and Computing; Geography and the Environment; Health and Life Sciences; Hospitality, Leisure, Sport and Tourism; Physical Sciences; Social Sciences. They include resources such as: Bibliographic, reference and research information; Publications online; Subject gateways; Data services; Learning and teaching; Support services.
- Tools referenced in JISC Collections publications list: collections of high quality online research tools, learning materials and digital archives for UK HE and FE institutions http://www.jisc.ac.uk/index.cfm?name=coll
- National Grid Service nodes (JCSR) http://www.ngs.ac.uk;
- Supercomputing facilities such as HPCx, CSAR (managed by EPSRC): http://www.hpcx.ac.uk and http://www.csar.ac.uk;
- Data Archive and MIMAS (ESRC);
- Protein Data Bank (Hosted by Wellcome Foundation at EBI);
- Large-scale facilities such as SRS, ISIS, Diamond (hosted at CCLRC) and associated scientific data collections;
- LHC Data Grid (PPARC);
- NERC Data Centres and CEH;
- Telescopes, e.g. via eSTAR services (PPARC);
- British Library, National Museums, etc. e.g. http://www.nmsi.ac.uk/
- Others such as British Geological Survey, UK Met. Office, Hadley Centre.

G Service capability set

This appendix summarises ongoing e-Service classification work which will be extended online at `http://www.grids.ac.uk/ETF/public/WebServices/classes.html`.

We list services which could form the basis for virtual environments for a variety of purposes. We specifically draw upon considerations of an information environment; virtual learning; and e-research. Reports and papers from which ideas have been taken are listed in the references. We thank the authors of these and also groups such as JCLT, JCIE and ETF whose members have been debating e-services for at least a year. Additional input has been taken from the various working and research groups of the Global Grid Forum which is currently identifying services, specifications and standards leading to an Open Grid Services Architecture (OGSA). In the UK, the recommendations of the Grid Architecture Task Force and the e-Science Gap Analysis carried out by Geoffrey Fox and David Walker have been taken on board.

We have attempted a rather broad and arbitrary classification of the services identified into the following areas: collaboration; e-research; e-learning; digital information; common infrastructure.

We do not consider these to be definitive lists of the services that can be provided, only examples, and we hope that additional services will be identified and developed in ensuing programmes, or identified services refined in the light of future requirements analyses ? such lists are organic and will grow and shrink. The aim is to kick-start a programme whereby a framework can be deployed to enable community input and contribution of more specialised services and resources. There is a tendency at the start to list many small-sized atomic services for every function imaginable, later the need to optimise the large-scale distributed system may indicate that services have to be aggregated (federated) in different ways to improve performance. We expect any e-Environment to support only a range of the services listed here.

e-Collaboration application services

Collaboration is about people working together, either as peers or in some more formally-defined relationship, such as researcher-supervisor. Collaboration extends to high-end technologies such as Access Grid.

- Calendar
- Collaboration Management
- Content Management
- Content Sharing
- Group or VO Management
- Peer Group Join
- Peer Service Location
- User Registration

e-Research application services

Research services are specific to supporting research processes and tools, including the Grid. They should include collaboration with experts and peers, encapsulation of complex procedures for non experts to facilitate growth of inter-disciplinary sciences and aids for results publication and proposal writing. We aim for a holistic system-based approach!

- Application Management
- Deployment
- Distribution
- Fabric Management
- Grid Information
- Information Access
- Information Aggregation
- Information Content Registration
- Information Query
- Information Metadata
- Information Presentation
- Information Notification
- Information Update
- Job Management
- Knowledge Extraction
- Knowledge Syndication (Join)
- Process Building
- Proposal Writing
- Resource Discovery
- Resource Management
- Scheduling
- Security
- Validation and Verification
- Visualisation

e-Learning application services

Learning services are about supporting a Managed Learning Environment with particular relevance to teachers and students supporting both peer groups and training hierarchies. A variety of approaches to teaching can be included with also self learning and assessment. A training and awareness environment might be distinguished from a more formal teaching environment because the former may not require assessment but could have more interactive demonstration material (e.g. via the Grid). There should be the ability to walk through material in various ways, log activities and attention to accessibility issues. Again the aim is to provide a holistic approach, but there must be access to humans if a student runs into difficulties, be they ones of understanding or personal. An electronic system cannot completely replace tutors!

- Activity Authoring
- Activity Management
- Assessment
- Competency
- Course Management
- e-Portfolio
- Grading
- Help
- Learner Profile Management

- Learning Flow
- Rating/Annotation
- Resource List
- Resource Management
- Scheduling
- Sequencing
- Trails and Personalisation
- View

Digital information services

These are services for digital information and data management, which will be informed by issues identified by the Digital Curation entre. Special attention to database servers and large collections, some of which are of qualitative nature, will be required. Conversion to electronic format and data mining may be targeted too.d

- Archiving
- Cataloguing/Curation
- Data Access and Integration
- Data Virtualisation
- Data Replication
- Data Management
- Deposition
- Dictionaries and Ontologies
- Digitial Rights Management (DRM)
- Resolver Services
- Resource Discovery
- Terminology

Common services

We have identified the following common services which may underpin many of the other services. They could in many cases be provided as part of the "infrastructure", and not directly visible to the users, maybe acting as "agents" to other high-level services. Session management is an example of this. Common services also give scope for system optimisation, e.g. through aggregation or federation in special cases and could be encapsulated in workflow for various scenarios. Some use cases are required here to extend and refine the list.

- Accounting
- Alert/Notification
- Authentication
- Authorization
- Billing
- Component Communication
- Filing
- File/Dataset/BLOB Management
- Identifier
- Logging

- Messaging
- Metadata Registry
- Monitoring
- Network Management
- Packaging
- Personalisation
- Portal Services
- Proxy Management
- Search
- Service Registry
- Transaction
- User Preferences
- Workflow

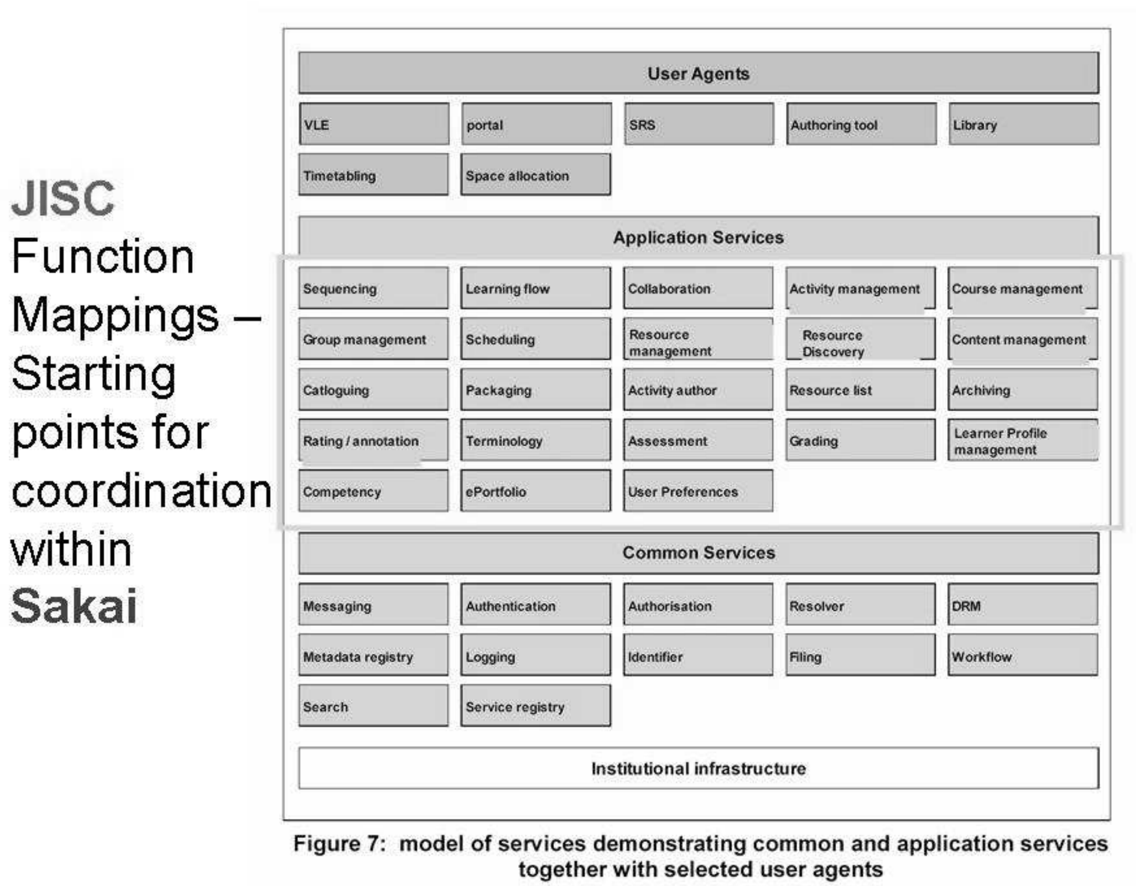

Figure 10: JISC Framework 1

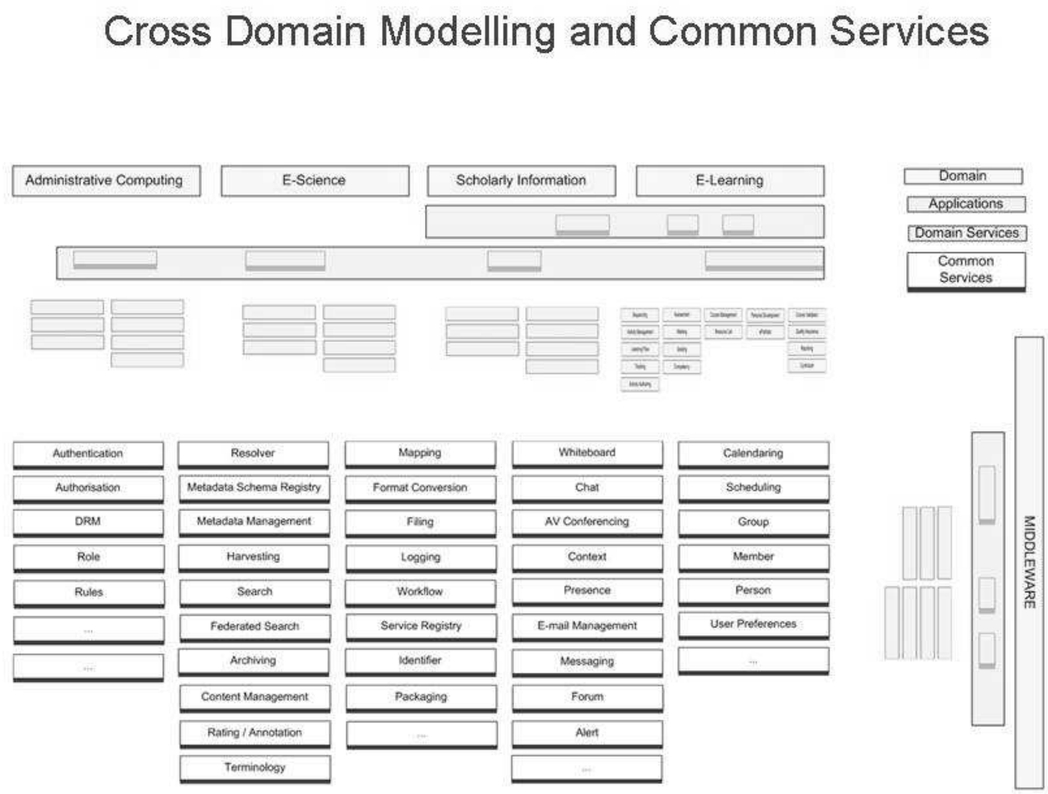

Figure 11: JISC Framework 2

H Generic Portal Engines

Tables H and H lists a number of generic portal engines and some related software from commercial vendors and public-domain developers. These were identified in the *Portals and Portlets 2003* workshop [28] and subsequent technical survey by Adrian Fish for the ReDRESS project [37].

Table 3: Generic Portal Engines 1

Commercial	
ASP.NET	`http://ibuyspy.com`
Blackboard Community portal	`http://products.blackboard.com/cp/bb5/index.cgi`
Campus Pipeline Luminis	`http://www.campuspipeline.com/`
CA CleverPath Portal	`http://www3.ca.com/Solutions/Product.asp?ID=262`
CMIS ePortal	`www.ccmsoftware.com/cmis.htm`
Elipva	`http://www.elipva.com`
EnginFrame	`http://www.enginframe.com`
Epicentre	`http://www.epicentric.com`
IBM WebSphere Portal	`http://www-4.ibm.com/software/webservers/portal/`
WebSphere Portal API	`http://www7b.software.ibm.com/wsdd/zones/portal/portlet/4.1api/`
Jahia	`http://www.xo3.com`
Lotus Notes Domino	`http://www.lotus.com/`
Lychee	`http://www.netcentriceurope.com/content/product_overview.htm`
Macromedia Cold Fusion	`http://www.macromedia.com/software/coldfusion/`
MediaApps NetPortal	`http:/www.mediaapps.com/`
Merant Collage	`http://www.merant.com/Products/WCM/collage/home.asp`
Microsoft Exchange 2000	`http://www.microsoft.com/exchange/default.asp`
Microsoft Sharepoint	`http://www.microsoft.com/sharepoint/`
Mongoose Portal Studio	`http://www.mongoosetech.com/products/portalstudio.html`
Novell Silverstream ePortal	`http://www.silverstream.com`
Novell Portal Services	`http://www.novell.com/products/portal/quicklook.html`
Oracle 9iAS Portal	`http://otn.oracle.com/products/iportal/`
Oracle Portal	`http://portalcenter.oracle.com`
RedHat Portal Server	`http://www.redhat.comn/software/rha/portalserver/`
SITE e:Vision	`http://www.sits.co.uk/`
Sun iPlanet Portal Server	`http://www.iplanet.com/products/iplanet_portal`
Unicon Academus	`http://www.uicon.net/academus`
WebCT Vista	`http://www.webct.com/products/viewpage?name=products_vista`

Table 4: Generic Portal Engines 2

Public Domain	
CHEF	http://www.chefproject.org
Enhydra	http://www.enhydra.org/
ExoPlatform	http://tuan.dyndns.org/exo/faces/public/portal.jsp
FreshMeat PHP Portal	http://freshmeat.net/projects/phportal/
Gluecode Portal Foundation Server	http://www.gluecode.com/website/html/PFS.html
Grid Portal Development Kit	http://dast.nlanr.net/Projects/GridPortal/
GridPort	http://gridport.net/index.cgi
GridSphere	http://www.gridsphere.org
JetSpeed	http://jakarta.apache.org/jetspeed/site/
jPortlet	http://jportlet.sourceforge.net/
LifeRay	http://www.liferay.com/products/index.jsp
MyLibrary	http://www.lib.ncsu.edu/eresources/mylibrary/
PHP-Nuke	http://phpnuke.org
PostNuke	http://www.postnuke.com
Sakai	http://www.sakaiproject.org
Thatware	http://thatware.org
uPortal	http://mis105.mis.udel.edu/ja-sig/uportal
Zope	http://www.zope.org/