
IST-2001-32133
GridLab - A Grid Application Toolkit and Testbed

# GridLab Work Package 4 Requirements Analysis

| | |
|---|---|
| Author(s): | Ian Kelley, Jason Novotny, Michael Russell, Oliver Wehrens |
| Document Filename: | /wp-4/Documents/RequirementsDocument/Requirements.tex |
| Work package: | WP 4 - Portals |
| Partner(s): | GridLab Work Packages 1-3, 4-12 |
| Lead Partner: | - |
| Config ID: | - |
| Document classification: | Requirements Analysis, Deliverable 4.2, Working Draft |

**A**bstract: The primary goal of the Portals Work Package is design and build the Gridlab Portal. This document describes the high-level requirements we identified during the course of our requirements gathering and early design stages. This is not a complete nor accurate description of the specific components and related functionality we intend to provide with the GridLab Portal. Instead, we plan to accompany this document with design documents pertaining to the GridLab Portal framework as well as documents describing the specific components that we later develop to serve the needs of the scientists, administrators, service and toolkit providers that will use the GridLab Portal.

# Contents

# 1 Overview

The GridLab Project is structured as a collection of work packages where each work package is assigned a specific set of responsibilities. In most cases, each work package is responsible for developing one or more computing services and/or one or more application toolkits. The goal of the Portals Work Pacakge is somewhat unique in that we are tasked with building a Grid portal that will provide a single point of access to the services and resources provided by GridLab and the Grid community at large. Though this may entail the development of additional computing services to support this Grid portal, and though we are interested in developing a toolkit to enable other communities to more easily develop their own Grid portals, our primary aim is to build a reliable end-user environment for GridLab consumers, as defined below. This end-user environment should simplify the use of Grids on the one hand, and on the other make it easy to plug-in new services and functionality. The rest of this document, then, describes in further detail what is required of the GridLab Portal.

## 1.1 Products

As stated above, the primary product of the Portals Work Package is the GridLab Portal. The GridLab Portal will provide scientific researchers with a single point of entry from which they can gain easy access to the broad array of Grid services developed by the other work packages in the GridLab project. Users will be provided with a friendly and easy to use interface by allowing them to interact with these services through standard means such as a web-browser or PDA. Below, we present an hourglass model depicting the role of the GridLab portal as a gateway between end-users and GridLab service providers:

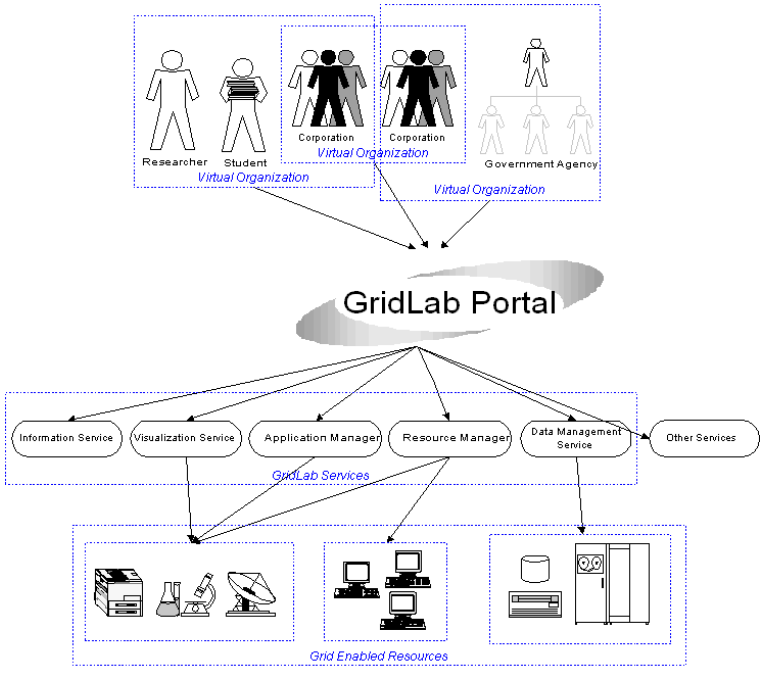


Figure 1: Shows how the portal acts as an intermediary between users of Grid Resources and the resources themselves.

Additional products, as we will be made clear in the GridLab Work Package 4 Design Document, include a Grid portal framework we call *GridSphere*, and additional services that are otherwise

not provided by GridLab.

## 1.2 Consumers

We have identified several groups of consumers, or end-users, of the GridLab Portal. These groups are described below.

### 1.2.1 Scientists

In particular, computational scientists represent the most important consumer group of the GridLab Portal. The ultimate usefulness of the GridLab Portal will depend on its ability to simplify to some extent the work done by computational scientists by providing an easy to use, feature-laden work environment. Supporting these end-users amounts to capturing the application requirements of specific research groups and providing the same capabilities from the portal. In practice, the use of a web browser or PDA is much more limiting than the workstation running specialized applications that a scientist is used to. This trade-off is acceptable as the value of the GridLab Portal is realized more in integrating the various applications and services required by the scientists accessible via a common platform as opposed to providing an equivalent level of functionality as the set of tools on a scientists workstation.

### 1.2.2 Portal Administrators

Portal administrators are responsible for the deployment and management of the GridLab Portal and are exposed to the technical issues of application server configuration and software installation. A requirement of portal administrators is that the GridLb Portal be easy to install and deploy with a minimum of site specific configuration changes. Ideally, the GridLab Portal will expose much of its administration tasks via the portal to allow administrators the ability to reconfigure the portal from a web browser or PDA. Portal administrators are also responsible for providing roles and permissions to users and general user management.

### 1.2.3 Portal Developers

Portal developers are responsible for the design and implementation of the GridLab Portal as will be outlined in our design document and satisfying requirements of the overall portal community. Portal developers also include providers of new portal functionality or portal enhancement. A key design consideration in the GridLab Portal architecture is to provide a framework for the easy development of new portal components, while requiring a minimum of knowledge about the underlying portal implementation.

### 1.2.4 Service Providers

The service provider consumer group is represented by GridLab work packages responsible for the development of additional Grid services such as the resource broker (WP 9), information services (WP 10), data management and visualization (WP 8), and monitoring (WP 11) . For any service a provider wishes to expose to the portal, a protocol specification or client API must be provided. The portal will provide generic support for OGSA[**?**] based Grid services making it easy for a portal developer or administrator to readily make the OGSA service portal accessible. Service providers that do not plan on interacting with end-users of the GridLab project can still benefit from providing portal access. The portal could be used for remote service administration and management by a service provider if so desired.

### 1.2.5 Testbed Administrators

Grid testbed administration is perhaps the most critical and difficult aspect of Grid computing. In general, testbed administrators do not have administrative control over the resources that makeup a Grid. However, testbed administrators can play the role of assisting our users to acquire access to resources, make and track requests to resource providers for upgrades to software and computing services, and to perform system tests on the various computing resources that are made available through the GridLab Portal.

### 1.2.6 Toolkit Providers

This group is represented by those Gridlab work packages responsible for providing application toolkits. As consumers of the Gridlab Portal, they may be interested in developing and testing those toolkits on the GridLab testbed.

# 2 General Requirements

There are a variety of requirements that any Grid portal project should satisfy or consider satisfying. In this section we describe a general set of requirements for the GridLab Portal, as gathered from the experience of the developers in this work package and lessons learned from the Grid portal development community at large.

## 2.1 High-level Design

The design of the GridLab portal should allow for maximum modularity and flexibility to support the many different needs and requirements of both the GridLab portal users, typically computational scientists and the GridLab service providers. The scientists require an easy to use, efficient and value-laden set of interfaces for interacting with the Grid to make their work more productive. The service providers need to provide a client API or protocol specification as well as information on how the service is to be used in practice. This includes hints on developing a useful visual portal interface to the service or information on how the service is intended to be used in conjunction with additional services made accessible from the portal.

### 2.1.1 Portlets

*Portlets* have become an increasingly popular concept used to describe visual user interfaces to a content or service provider. From a technical perspective, portlets represent modular, reusable software components that may be developed independently of the general portal architecture and offers a specific set of operations. For instance, portlets may provide users with an updated list of stock quotes or content from a news feed.

We have determined that portlets, as described in the evolving Portlet API, represent the best possible model for the development of the GridLab portal framework. Portlets offer atomic functionality such as a job submission component or a remote file browser interface. In a Web browser or PDA, portlets can be aggregated together supplying the user with easy and efficient access to multiple sources of content, services and applications. A portal user may be able to add (subscribe) or remove portlets from their personalized portal page depending on their needs. The Portlet API is in the process of being reviewed within the Java Community Process (JCP) and a Java Specification Request (JSR)[**?**] is currently in the process of being standardized.

## 2.2 Functionality

Any application, whether its a desktop application or a complex Web application such as the GridLab Portal, is only as good as the functionality it provides to end-users. In this section, we describe the areas of functionality we have identified that the GridLab Portal should provide. This is not meant to be a complete list of functionality, but rather an overview that will lay the groundwork for the specific portlets and supporting services we must develop.

### 2.2.1 Authentication

- *Single Sign-on*: Users require single sign-on to the GridLab Portal and GridLab services. We can achieve this by requiring users to logon to the GridLab Portal with *Grid Security Infrastructure (GSI)* certificates. GSI certificates may be retrieved from a client's local filesystem or retrieved with a username and password from *MyProxy*, a secure certificate repository.

- *Multiple Certificates*: Our single sign-on solution should support the use of multiple certificates, as there are several certificate authorities upon which various resource providers within our testbed may rely upon.
- *Certificate Management*: This implies the need for client tools for managing multiple certificates locally or with MyProxy, in addition to enhancements to MyProxy. For instance, users should be able to receive notifications, by email say, when certificates need to renewed.

### 2.2.2 Authorization

- *Privacy/Sharing*: Users, like resource providers, should be able to protect their files from unauthorized access. At the same time, users should be able to easily share files and related information with other users as desired.
- *Security Policies*: On the other hand, we should not implement measures that usurp the security policies of resource providers. For instance, it would be conceivable to allow portal users to share their certificates in order to share access to resources. Instead, we should take precautions to prevent this kind of sharing, as it not only prevents resource providers from imposing restrictions on users, but may also affect accounting and related systems.

### 2.2.3 Computing Resources

- *Brokering*: In the ideal case, users would not care which resources they are using, only that their application needs are appropriately met. Indeed, this is one of the primary goals of the GridLab project.
- *Utilization*: However, in today's computing environments, users are often quite familiar with the resources they use and what is required to use them. Furthermore, it is not usually possible to predict application performance programmatically. Thus, while users would benefit from a brokering service that made decisions on behalf of users on where and how to run applications, they should be able to have full control over resource utilization.
- *Information*: On the other hand, users should be able to list and search for available resources as well as easily determine their access privileges. They should be able to view resource status online and have access to documentation about those resources. In addition, users should be also able to see their allocations and usage records online, and furthermore be notified when they have limited allocations.

### 2.2.4 Computing Services

In a similar manner to resources, users should be able to easily discover and utilize computing services online, though in today's computing environments users are generally familiar with the computing services they apply towards their research.

### 2.2.5 Job Management

- *Job Submission*: It's clear that users require to easy-to-use tools for submitting jobs. In particular, we require robust support for batch job submission, including the ability to generate and submit appropriate batch scripts.

- *Job Monitoring*: Users should be able to monitor the status of their jobs online and recieve notification when jobs complete or fail. In addition, they should be able to monitor the *performance* of their jobs.
- *Job Migration*: Given poor performance or the choice of better available resources, it should be simple for users to *migrate* their jobs. Furthermore, they should be able to programmatically specify conditions for migrations at time of job submission.
- *Job History*: Users should be able to examine their job history, along with status information, output to stdout and stderr, and performance information. This capability could be enhanced with the ability to archive associated executables, input files, and output files.

### 2.2.6 Data Management

- *Data Transfers*: Our users require the ability to transfer large datasets between resources. However, often times users are interested in analyzing some particular subset of their data, and thus would benefit from transfer services that allow fine-grained access to data, such as the ability to browse or serach for a subset of data within a file and to transfer just that portion of the file for local analysis.
- *Data Storage*: Users required the ability to store or archive their large datasets. And as mentioned in the resource section above, users should be able to determine their disk or archive space allocation.
- *Data Location*: At time of analysis, users should be able to describe their datasets with meta-information and be able to use this meta-information to quickly locate their datasets at a later time.
- *Data Replication*: Given that researchers may be distributed around the globe, researchers should be able to replicate relevant datasets to local storage facilities as required for anlaysis.

### 2.2.7 Data Visualization

- *Visualization Applications*: Users should be able to quickly launch the appropriate visualization application for their datasets. Already, the Cactus Project has provided mechanisms to support the launching of visualization applications based on *mime-types*, and we should be able to provide the same kind of support in the GridLab Portal.
- *Visualization Services*: In some cases, users may wish to pre-process datasets before final analysis, especially if this pre-processing takes a considerable amount of time to complete. In other cases, users require low-resolution images of their datasets. There are many kinds of services we have yet to categorize for our users. One of the goals of the GridLab project is to determine what kinds of services would most benefit our users.

### 2.2.8 Application Management

- *Application Deployment*: Users should be able to quickly and easily deploy applications and related input files to resources. It should be possible to automate this on a per resource basis or accross an entire testbed, if necessary.
- *Resource Brokering*: As it is a goal of the GridLab Project to provide automated resource brokering, it should be possible to tailor resource brokering for on a per application basis.

- *Application Monitoring*: Users should be able to monitor application-domain information and receive notification about events occuring within applications. This implies the need for toolkits to make such functionality available to application developers, and indeed this is a goal of the GridLab Project as well.

#### 2.2.9 Collaboration

- *Online Chat*: From the experience of the developers in this work package and the observed popularity of online chat services in the general community, we feel that our Grid portal should provide or enable provide online chat services to our users.
- *Bulletin Boards*: Bulletion boards are useful for posting information to an entire community of users. We have identified several such use-case scenarios, including the ability to post information about resources and announcements about upgrades to software or computing services.
- *Videoconferencing*: Videoconferencing may be another useful service to explore. For instance, it would be a benefit to the developers with the GridLab project to be able to hold video-conferences to discuss interaction between work packages.

#### 2.2.10 Administration

- *Portal Administration*: A Grid portal should be available at all times. Therefore, it's a must that a Grid portal be can be administered online. Users with administrative rights should be able to add other users to the GridLab Portal, set access to portlets within the GridLab Portal, and otherwise be able administer portlet services.
- *Service Administration*: It should also be possible to administer GridLab services through the GridLab Portal, where administrative control is made available through network mechanisms.
- *Testbed Administtration*: It should be possible to administer a testbed as described earlier in this document.

#### 2.2.11 Other Features

- *Events and Notification*: Users should be able to receive notification of various events, to specify for which events they want to receive notifcation, and how notification should occur.
- *Fault Tolerance*: For certain critical operations, users may require fault tolerance mechanisms to insure that either operations complete successfully or that users are notified of their failure
- *Work-flow*: In some cases, a user may be able to determine a work-flow to perform a set of operations, and we should be able to support the creation of work-flows as some point in the future.
- *Shell Access*: Many of our users have requested the ability to

## 2.3 User Interfaces

User interface is difficult and time-consuming. Therefore, we should develop useful guidelines to insure that the user interfaces we develop provide the right look-and-feel and make the functionality most relevant to the end-user easily accessible. Below we provide brief descriptions of the issues we are considering in the design of the GridLab Portal.

### 2.3.1 Portal Clients

- *Web clients*: By definition, Grid portals provide access to Grids through Web applications. Web applications typically deliver clients as Web pages or Java applets. In our case, it is a must that the majority of the functionality described here be accessible through simple and easy to use Web pages. However, Java applets have applicability wherever *interactive* functionality is required, such as will be the case with SSH clients to remote hosts.

- *Mobile devices*: There are numerous scenarios we have identified where mobile devices provide can value-added services to the GridLab Portal. For example, when an application completes or when an event of interest occurs in a long-running application, scientists might benefit from receiving an SMS message on their mobile-phone. Furthermore, many mobile devices, including mobile-phones and PDAs, provide the ability to browse the Internet. The GridLab Portal should support those types of clients wherever possible.

- *Offilne tools*: Many of our users own laptop and work offline on a regular basis. We are study scenarios in which users may want to schedule tasks to run once they are back online. For example, many users would like to be able to prepare an application for use before using the GridLab Portal to run that application on the GridLab testbed.

### 2.3.2 Design Issues

There are several design issues when it comes to good user interface design, all of which are made more relevant by the inherit limit in developing Portal clients with HTML. Below, we provided a brief list of those issues.

- Lightweight interfaces that download quickly.
- Fast response time between actions.
- Effective error handling and reporting.
- Intuitive and easy-to-use interfaces.
- Consistent design and organization.
- Consistent terminology.
- Limited but meaningful use of colors.
- Interfaces should be customizable.
- Effective menus and navigation systems.
- Online help and documentation.

# 3 Application Requirements

Earlier we stated that computational scientists represent the most important consumer group. From our perspective, the best way to support computational scientists is to start by providing support for the software and applications they develop and use in the course of their research. The general goal is to make it easy to build, run, and visualize the results of applications on Grids. This represents a successful division of labor if scientists are also provided with toolkits for developing applications that effectively utilize Grids to meet their resource requirements. Indeed, this is one of the key strategies of GridLab, in addition to building better Grid technologies.

There are two target application toolkits identified in the Gridlab Proposal, *the Cactus Computational Toolkit* and *Triana*. In each case, our goal is to provide support for building, running, and visualizing the results applications constructed with these toolkits. Additionally, we are interested in developing portlets that provide *value-added* services, or functionality that simplifies even the use of these toolkits.

## 3.1 Cactus

We held extensive meetings with the Cactus Project and users of the Cactus Computational Toolkit at Max-Planck-Institute for Gravitational Physics to identify Cactus user requirements for the GridLab Portal. In particular, we identified the following use-case scenario as the most critical Cactus use-case for the GridLab Portal.

*A numerical relativist at Max-Planck-Institute for Gravitational Physics wants to run a Cactus physics simulation represented by a list of Cactus a parameter file and, optionally, a list of thorns. (Note that the required thorns can be determined from the parameter file.) This physicist would like to be able to instruct the GridLab Portal to locate and reserve the best available computing resource for their simulation, construct a Cactus simulation executable containing the given list of thorns and submit that simulation executable along with the given parameter file to the resource job scheduling system. This physicist would like to be notified, say via email, when the Cactus simulation begins and ends, as well as to be notified at certain time-steps during the course of the simulation. After the simulation ends, they would like the GridLab Portal to archive the data produced by the simulation as well as to make this data available for analysis upon request. Generally, a physicist is interested in visualizing 1-dimensional data while the Cactus simulation is running and immediately after the Cactus simulation ends. It is generally preferable to archive 2- and 3-dimensional datasets, due to their large size, until they are needed for analysis.*

We identified several other requirements that either support this use-case scenario, such as the ability to test new code before committing changes to CVS, or extend this use-case scenario, for example physicists at MPI regularly perform *parameter studies*. Because we require more interaction with the Cactus user base, in the rest of this section we describe only those portlets that we determined are necessary to support the above use-case scenario and the requirements most essential to the users of Cactus software.

### 3.1.1 Cactus Software

- CactusInstallationPortlet

  The CactusInstallationPortlet enables users to install and update Cactus software on target

hosts with the *getcactus* script. This script is developed and maintained by the Cactus Project and uses cvs checkout and update tools to obtain the Cactus flesh and thorns. It is included in the Cactus CVS standard repository at cvs.cactuscode.org. The *getcactus* script enables users to select from which Cactus repository to obtain the Cactus flesh. It also enables users to provide a CactusThornList. By default, the *getcactus* obtains the latest revision of the flesh and thorns, but it users can also specify which revisions to obtain. The CactusInstallationPortlet will support the options offered by the *getcactus* script.

- CactusThornListPortlet

  The CactusThornListPortlet enables users to maintain one or more CactusThornLists for use with CactusInstallationPortlet.

- CactusConfigurationPortlet

  The CactusConfigurationPortlet enables users to create and update Cactus configurations with the *configure* script that is provided when one installs Cactus software with the *getcactus* script. The *configure* script is based on *autoconf*. It produces a directory within a particular Cactus installation and generates a *makefile* and the necessary files to produce a Cactus executable with *gmake*. Thus, *CactusConfigurationPortlet* also enables users to build executables for each Cactus configuation with *gmake*, allowing users to specify appropriate tags and command-line options.

- CactusParameterFilePortlet

  The CactusParameterFilePortlet enables users to upload Cactus parameter file to a target resource. When the parameter file is uploaded, users can specify whether they would like to have that parameter file modified to contain parameters that are known to be required for the target resource in order for Cactus simulations using that parameter file to behave correctly. Users may also edit Cactus parameter files online or download Cactus parameter files with this portlet.

- CactusSimulationPortlet

  The CactusSimulationPortlet enables users to run Cactus simulations with the *qs2* script. The *qs2* script is a batch-script written and maintained by the Cactus Project for submitting a Cactus executable and parameter file to batch queues on target resources. The CactusSimulationPortlet requires the *minimum information* for running a Cactus simulation:

  - *Resource requirements:* Currently, users are very familiar with the resources they employ to run Cactus simulation. The information they are most interested in obtaining from the GridLab Portal is queue status information.
  - *Parameter file:* In this case, the CactusSimulationPortlet will check to see whether Cactus is installed on the target resource and whether an appropriate configuration exists. If not, it will produce a Cactus configuration given the Cactus parameter file, using the *makethornlist*, *getcactus*, and *configure* scripts provided with Cactus software. And if required, the CactusSimulationPortlet will subsequently use *gmake* to build an executable before running the simulation as described above.

  The CactusSimulationPortlet also enables users to specify whether to produce an archive of a simulation after it has completed. By archiving, we mean that one or more tarballs

is created containing a *Cactus executable*, *Cactus parameter file* and *Cactus datasets*. In some cases, users may wish to archive 2- and 3-dimensional data in separate tarballs, due to their large size. This archive information stored on behalf of users is accessible from both the CactusSimulationPortlet and the CactusSimulationArchivePortlet.

- CactusSimluationArchivePortlet

  The CactusSimulationArchivePortlet enables users to tarball and untarball Cactus simulation archives. It also enables users to transfer archives to other locations with GridFTP.

- CactusFileAdvertiserPortlet

  The CactusSimulationArchivePortlet enables users to view the files produce during a Cactus simulation and after it completes. It uses the *CactusFileAdvertiser* to make this information available.

### 3.1.2 Cactus Applications

## 3.2 Triana

We are still in the early requirements analysis stages for Triana.

# 4 GridLab Requirements

By definition, our project requires us to provide interoperability and support for services and toolkits developed by our GridLab collaborators. As was illustrated in the hour-glass model earlier, the entire range of GridLab services will be coordinated through use of the GridLab Portal. Furthermore, as a "window to the Grid", the GridLab Portal depends entirely on these services to deliver functionality to end-users. This implies the need to provide support for developing, testing, and administering services through the GridLab Portal. Thus, in addition to portlets that deliver related functionality to end-users, we would like to provide portlets for administrators and developers of Grid services, portlets, and the GridLab Portal. Below, we describe how we depend upon each work package and portlets that should pertain to that work package.

## 4.1 GAT Work Packages

The GAT work packages will provide us with application toolkits that offer high-level operations for performing operations on Grids. When these toolkits are available, we can apply them towards the development of portlets.

### 4.1.1 Portlets

As stated above, we will presumably support the development of portlets with the application toolkits provided by these work packages. However, as represents of the toolkit provider group, we are still determining the kinds of portlets that might serve the development and testing of application toolkits.

## 4.2 Testbed Work Package

The Testbed work package aims to organize a reliable testbed for use with the GridLab Portal and Grid services. From the GridLab Portal perspective, this group represents the group responsible insuring an infrastructure exists for meeting the needs of our scientific end-users.

### 4.2.1 Portlets

Testbed portlets will provide Web-interfaces to libraries and services developed by the Testbed Work Package. The primary consumers of these portlets will be Grid Testbed administrators, though some portlets are likely to be useful to general users of the GridLab Portal for discovering and testing specific elements of the GridLab Testbed or other Grid testbeds.

## 4.3 Security Work Package

The primary product of the Security work package, as indicated in their design document, is a Grid authorization service. All GridLab services will presumably rely on the GridLab Authorization Service to authorize client requests. In principle, we could use the Gridlab Authorization Service to assigning access privileges to elements of the GridLab Portal.

### 4.3.1 Portlets

Security Portlets will provide Web-interfaces to libraries and services developed by the Security Work Package. The primary consumers of these portlets will be administrators of the GridLab

Authorization Service for assigning privileges to consumers of GridLab resources and services, including elements of the GridLab Portal.

## 4.4 Adaptive Components Work Package

The Adaptive Components work package will provide an application toolkit for enabling applications to adapt to applications for migration scenarios. We are still investigating the kinds of portlets that would pertain to this work package.

## 4.5 Data Management and Visualization Work Package

The Data Management and Visualization work package will provide libraries and services for manipulating and visualizing datasets on Grids.

### 4.5.1 Portlets

Data Management Portlets will provide Web-interfaces to data management libraries and services developed by the Data Handling and Visualization Work Package. The primary consumers of these portlets will be scientists that require Web-interfaces for locating, transferring, replicating, and archiving datasets.

Data Visualization Portlets will provide Web-interfaces to data visualization libraries and services developed by the Data Handling and Visualization Work Package. The primary consumers of these portlets will be scientists that require Web-interfaces for visualizing and/or launching applications or services to visualize datasets

## 4.6 Resource Management Work Package

The Resource Management work package will provide services for brokering resources and managing jobs.

### 4.6.1 Portlets

Resource Management Portlets will provide Web-interfaces to resource management libraries and services developed by the Resource Management Work Package. The primary consumers of these portlets will include scientists that require Web-interfaces for launching applications on Grid resources and administrators of the GridLab Resource Management Services.

## 4.7 Information Services Work Package

The Information Services work package will maintain the GridLab Information Index Service. This work package is also responsible for assisting the other GridLab work packages to design and implement schemas for obtaining the specific meta-information they require about Grid resources and services.

### 4.7.1 Portlets

Grid Information Portlets will provide Web-interfaces to libraries and services developed by the Information Services Work Package. The primary consumers of these portlets will include GridLab administrators and developers interested in viewing information provided by GridLab Information Services.

## 4.8 Application Monitoring Work Package

The Application Monitoring work package will provide libraries and services for monitoring the performance of applications.

### 4.8.1 Portlets

Application Monitoring Portlets will provide Web-interfaces to libraries and services developed by the Application Monitoring Work Package. The primary consumers of these portlets will include GridLab scientists interested in monitoring closely the performance of their applications and administrators of GridLab Application Monitoring Services.

## 4.9 Mobile Services Work Package

The Mobile Services work package is investigating use-case scenarios and solutions that may be applied towards the use of the mobile devices. There are numerous scenarios we have identified where mobile devices provide can value-added services to the GridLab Portal. For example, when an application completes or when an event of interest occurs in a long-running application, scientists might benefit from receiving an SMS message on their mobile-phone. Furthermore, many mobile devices, including mobile-phones and PDAs, provide the ability to browse the Internet. The GridLab Portal should support those types of clients wherever possible.

# 5 Areas of Research

We are beginning to identify new areas of computing not explicitly addressed by the GridLab Project for which we are interested in developing portlets and supporting services. This section serves as a place holder for requirements pertaining to these areas of research.

## 5.1 Support for Virtual Organizations

Taken as a whole, the GridLab Project represents a *virtual organization*. This term was coined in Anatomy of the Grid and generally refers to a collection of distributed researchers collaborating on complex problems and sharing resources across institional, geographical, and political boundaries. If we are able to effectively support our own virtual organization, the results of our work may one day be applied to other viritual organizations. However, from the perspective of a Grid Portal, a virtual organization may be "just another layer". Thus, while there may not yet be a clear definition for a virtual organization, we are aware of the potential for supporting virtual organizations with one Grid portal.

## 5.2 General Support for Applications

While each application might have a unique set of requirements to effectively support its use on Grids, we can often categorize these requirements across application domains, such as computational resource requirements and data storage requirements. Similarly, while there might be a unique set of portlets for interacting with each application, we can categorize portlets across application domains, such as portlets for building application executables and portlets for visualizing the output of applications. Thus, as we develop solutions for the target applications described above, we are actively searching for new application consumer groups so that we might start factoring solutions into a more common, general design for supporting the use of applications from the GridLab Portal.

## 5.3 Support for Application Development

Many of our Cactus users have expressed the desire to be able to checkout applications from CVS and compile applications on remote hosts from the GridLab Portal. In addition, users would like to be able to edit parameter files online, which in Cactus defines an application's behavior. We feel that a natural extension of these capabilities would be to enable users to develop their applications with the aid of the GridLab Portal. This would include the ability to edit and commit source code to CVS, or other shared document repositories, as well as to debug and test code online. The end-goal might be one or more *integrated-development environments (IDE)* that work with commonly available development tools, such as autoconf and automake, or ant for Java-based applications.

# 6 Community Involvement

It is a requirement of a project such as ours that we stay actively involved in the Grid community. This is because the world of Grid computing is quickly evolving, as are related areas in high-performance, Web, and enterprise computing. By community involvement, then, we mean to visit conferences, publish papers, give presentations, and participate in community working groups.

## 6.1 Global Grid Forum

The Global Grid Forum (GGF) represents perhaps the most important Grid community from our perspective. It contains several working groups that aim to produce documents for describing best practices in Grid computing, and in some cases develop common protocols and technologies. Of those working groups, the Grid Computing Environments Working Group (GCE-WG) is the working group most closely associated with our project.

# 7 Summary

In this document, we described the high-level requirements we identified during the course of our requirements gathering and early design stages. This is not a complete nor accurate description of the specific components and related functionality we intend to provide with the GridLab Portal. Instead, we plan to accompany this document with design documents pertaining to the GridLab Portal framework and more documents describing the specific components that we later develop to serve the needs of the scientists, administrators, service and toolkit providers that will use the GridLab Portal.