

Full Length Research Paper

Pluggable, interoperable and extensible (PIE): A grid portal for interoperability

Shashi Bhanwar* and Seema Bawa

Centre of Excellence in Grid Computing, Computer Science and Engineering Department, Thapar University, Patiala-147004, India.

Accepted 16 August, 2011

The main challenge grid computing environment faces today is to provide interoperability and enable collaboration between multitudes of grids. Standardization is a long term solution to enable interoperability. Grid portals and gateways have emerged to be important components in addressing interoperability challenge in grid environments to enable collaboration and expand the pool of resources available. In this paper, we proposed pluggable, interoperable and extensible (PIE) as a portal based solution for addressing interoperability challenge faced in grid environments. The architecture of PIE has been divided into three layer structure- resource layer, interoperability layer and portal layer. Portlets such as file management, credential management, job submission and resource registry has successfully demonstrated usage of heterogeneous resources irrespective of underlying platforms. The implementation of PIE as a portal based solution demonstrated interoperability between various heterogeneous resources hosted on varied platforms.

Key words: Grid portals, interoperability, portlets, grid computing environments.

INTRODUCTION

The emergence of grid computing has led to progressive growth in inventing the ways to conduct scientific research. Since its inception, there has been explosive growth in terms of multitude developments under its umbrella. These developments work reasonably well within their own boundaries, having little or no interoperability between them (Joseph and Fellenstein, 2003). The main reasons are the diversity of software platforms, the various policies governing the access to resources, use of distributed and heterogeneous re-sources and apparent lack of adherence to the emerging standards (Menon, 2007). However, grid computing paradigm is faced with security threats and challenges to connect resources from disparate developments having their own administrative policies and control. This becomes a paradoxical situation where grids, partly being developed to increase interoperability between different computing platforms; themselves have high extent interoperability

problems (Boardman et al., 2007).

Out of a number of solutions available to provide interoperability in grids, designing a portal and providing basic grid services is gaining lot of attention from scientific research groups to address interoperability challenge (Xang et al., 2006). Some of these portals are successful in their ability to provide interoperability at workflow level while the strength of others is aimed at discovering resources and making them available at a common platform (Network for Earthquake Engineering Simulation Grid, <https://www.nees.org/it/>). But, developing these portals, which includes integrating and coordinating multiple, distributed resources and differing grid technologies with various application programming interfaces (APIs), requires a significant amount of effort. Portal developers often re-implement functionality found in existing portals because there are no common discovery and access mechanisms in place to enable sharing of portal functions (Novotny et al., 2004).

This paper is an effort in this direction to address interoperability challenge and provide access to a large pool of grid resources to an end user using portal based solutions. It proposes a grid portal to address interoper-

*Corresponding author. E-mail: shashi@thapar.edu. Tel: + 91-0175-2393137, 9988529211. Fax: +91-0175-2393137

ability and provide heterogeneous grid resources to end users at a common interface. We have also designed a grid portal earlier using contemporary approach to integrate disparate resources (Bhanwar and Bawa, 2010). However, PIE: The proposed grid portal, is an effort in the direction to not only provide easy accessibility of heterogeneous grid resources to users but also to curtail the development effort involved by aiding developers in plugging new portlets into the portal to customize it to their requirements.

RELATED WORK

Grid portals and gateways have emerged to be important components in addressing interoperability challenge in grid environments to enable collaboration and expand the pool of resources available (Gannon et al., <http://www.extreme.indiana.edu/groc/ggf-portals-draft.pdf>). A portal framework can provide presentation abilities for grid services in addition to personalization and content management. Indeed, employing portal technology in grid environments offers the ability for users to work more effectively by allowing complex jobs to run with greater efficiency and at lower cost. Grid portals at different levels have already been constructed around the world to provide users an easy way of accessing or integrating distributed computing, data and other resources (Mock et al., 2002). Here, we discuss a few representative implementations of portals which aim to interoperate grid resources at various levels. Various grid portals have been developed till date, for specific applications. The NEESgrid Portal (Network for Earthquake Engineering Simulation Grid, <https://www.nees.org/it/>), the Alliance Portal (NCSA Alliance Scientific Portal Project, <http://www.extreme.indiana.edu/alliance/>), the GENIUS Portal for the European Data Grid (Barbera et al., 2007) and the Integrated e-Science Environment (IeSE) (Integrated e-Science Environment for CLRC, <http://www.e-science.clrc.ac.uk/>) are some of application based grid portals. There have been some efforts to build grid middleware also, such as gridport which is a middleware toolkit to facilitate the development of grid computing portals (Dahan et al., 2004). NSFNMI's OGCE project is considered as a more flexible portal. The Open Grid Computing Environments (OGCE) release 2 consists of a core set of grid portlets (Wang et al., 2006). There are other application based portals such as The Astrophysics Simulation Collaboratory Portal (ASC), an application portal developed to provide astrophysics researcher's web based access to the Cactus computational toolkit (Cactus Code, <http://www.cactuscode.org>). Cactus is a framework for solving various wave equations with an emphasis on astrophysical simulations of colliding neutron stars and black holes. The ASC portal allows users to submit various

Cactus simulations to a distributed set of HPC resources as well as remotely checkout and compile Cactus with specialized options on HPC resources (Laszewski et al., 2002; Russell et al., 2002). The NCSAportal development effort is focused on the development of several application portals for computational chemistry and particle physics simulations (NCSA Alliance Scientific Portal Project, <http://www.extreme.indiana.edu/alliance/>). Gaussian is a computational chemistry portal allows users to provide input files to the popular Gaussian chemistry package and easily submit simulations via the web (Gaussian, <http://www.gaussian.com/>). Similarly, the MIMD Lattice Computation (MILC) Collaboration portal is aimed at providing particle physicists access to a popular community code used to understand elementary particle interactions (MyProxy, <http://grid.ncsa.illinois.edu/myproxy>). Both portals rely on GSDK for job submission and file staging capabilities (Novotny, 2004).

P-GRADE is a web based, service rich environment for the development, execution and monitoring of workflows and workflow based parameter studies on various grid platforms. It hides low-level grid access mechanisms by high-level graphical interfaces, making even non grid expert users capable of defining and executing distributed applications on multi-institutional computing infrastructures. Workflows and workflow based parameter studies defined in the P-Grade are portable between grid platforms without learning new systems or re-engineering program code. Technology neutral interfaces and concepts of the P-GRADE portal help users cope with the large variety of grid solution. It has been a step further in allowing various scientists to submit workflows to grid and achieve interoperability (P-Grade, <http://portal.p-grade.hu/>). IBM's WebSphere portal is a framework that includes a runtime server, services, tools and many other features that can help integrate an enterprise into a single, customizable interface portal. It implements the JSR 168 Portlet Application Programming Interface (API) and WSRP. WebSphere Portal is a powerful tool and is widely used in many business companies and enterprises (IBM Websphere Portal, <http://www.ibm.com/developerworks/websphere/zones/portal/newto/>).

Having surveyed the major grid portal development tools, frameworks and components, it can be construed that, developing portlet-based grid portals is the latest for solution for solving short term interoperability challenge in grid environments (Thomas et al., 2002; UNICORE, <http://www.unicore.eu/>; uPortal by JA-S, <http://www.uportal.org>). The Grid Computing Environment (GCE) working group is one of the many working groups of the Open Grid Forum (OGF). It is a community-initiated forum of individual researchers and practitioners working on distributed computing, or "grid" technologies. This working group aims at contributing to the coherence and interoperability of frameworks, portals, problem solving

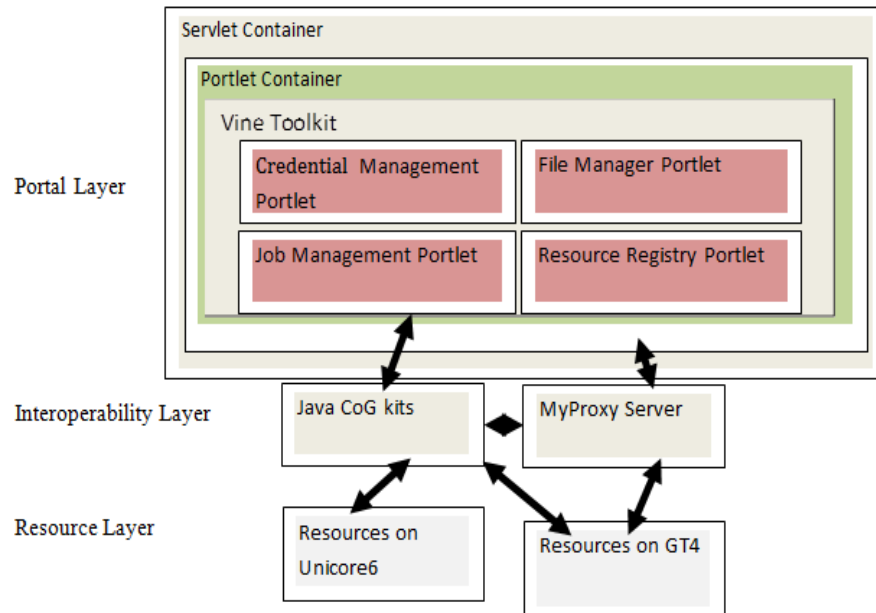


Figure 1. PIE architecture.

environments and other grid-based computing environments and Grid services by establishing standards that are required to integrate technology implementations and solutions (Condor High Throughput Computing, <http://www.cs.wisc.edu/condor/htc.html>). From the project's introduction, we know that the grid portlets can only access a typical set of grid sources, for example gt2, or gt3 only. It means that when the grid portlets are invoked to access information resources, the middleware in resource layer has to be of the same version. Obviously, this limits the usability of grid portlets. The end-users have to know the remote resources, for example which version Globus Toolkit installed, when deploying the grid portlets in portal framework. Usually end-users just care of their job implementation. They have not much more ideas about the kinds of services the grid systems can provide (Wang et al., 2006). It is also difficult for them to understand what resource middleware which is installed in the remote resource layer can be used, as not all end-users are expected to be grid experts.

ARCHITECTURE OF PIE

PIE is a pluggable, interoperable and extensible grid portal which can be used by scientists and end users to easily run their jobs on grid computing infrastructure. Scientists can easily develop their portlets and plug-in to PIE. It uses portal based solution to allow an end user access to heterogeneous resources available in Grid Computing Environments. End users can easily plug-in

heterogeneous resource without building the gateway. PIE system architecture is shown in Figure 1. PIE architecture is divided into three layers structure: Resource layer, interoperability layer and portal layer.

Resource layer

The lowest layer in PIE architecture is the resource layer. It hosts various heterogeneous resources as available in a grid. These sources may be running on different middleware in a form of cluster of systems which is plugged in to provide services to the Grid users. The resources can be computational, data storage, print etc. They are clusters of middleware such as GT4.2 and Uniform Interface to Computing Resources (UNICORE) (<http://www.unicore.eu/>) which are implemented using the middleware toolkits available through open source. MyProxy (<http://grid.ncsa.illinois.edu/myproxy>) server is installed to verify the credentials of the end users. Vine Toolkit used for grid portlets connects resources from various middleware using Java CoG kits at the backend (Vine Toolkit, <http://www.vine.org>). Other scientific resources can also be included.

Interoperability layer

The intermediate layer is the interoperability layer. It hosts the portlets which have Java CoG (Community Grid) kits at the backend to serve as gateway to the resources underlying. Commodity Grid (CoG) Kits allow

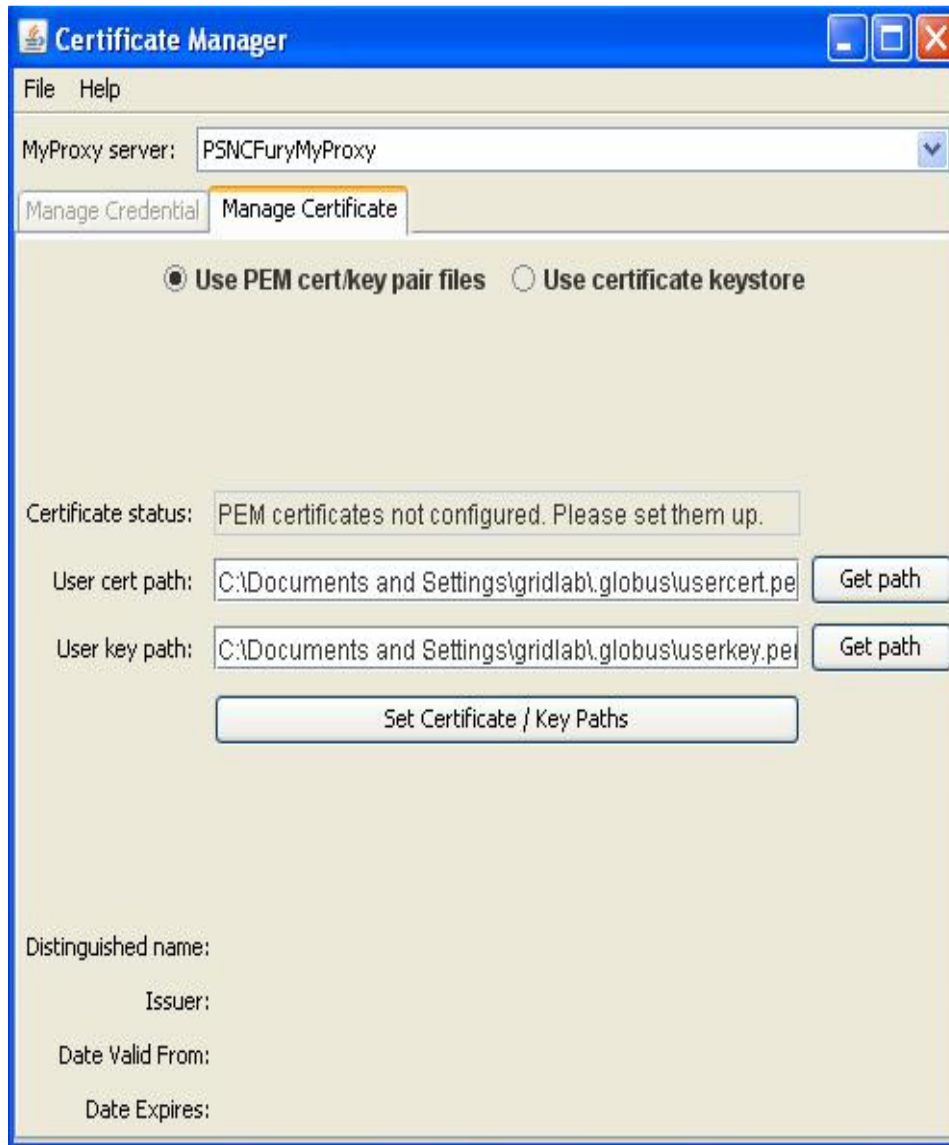


Figure 2. Certificate manager.

grid users, grid application developers and grid administrators to use, program and administer grids from a higher-level framework (Laszewski et al., 2000). It helps the developers to bridge between even different version of same middleware such as GT2 and GT4 of Globus Toolkit. It also has Certificate Manager as shown in Figure 2, which stores credentials and maintains them. The Certificate Manager used in PIE is installed from Vine Toolkit which enables an administrator to maintain credentials for each user. The credentials maintained are from multiple middleware and are composed in multiple formats. A user can only submit job to the resource only if he has the required credential to access the resource. It becomes insecure to store credentials on a user's desktop. Therefore, a central repository of credentials

enables easier and efficient maintenance and access to resources. The access rights are available to the user on the basis of the credibility of the credentials available.

Portal layer

The topmost layer is the portal layer. It includes portal framework. The Grid Portlets are hosted in portal framework and managed by portlet container. When end-users plug the Grid Portlets in the user page, the portlets will visit the resource registry, retrieve resource information and display back on the portlets. So, the end-users can get the resource information and decide to run their jobs on the resources available. The Portal is further

hosted in servlet container. The portal layer consists of Portlets responsible for various primary functions as accomplished by portlets as shown in Figure 1. The portlets are credential management, file management, job management, security and resource management. The portlets are briefly discussed here.

Credential management

The credential management portlet enables user to manage and upload credentials required for running a particular job. New credentials can be stored and used. It works separately from the certificate manager.

File management

The file management portlet enables user to store and maintain files. Data forms the single most important asset in a grid computing system. This data may be input into the resource and the results from the resource on the execution of a specific task. If the infrastructure is not designed properly, the data movement in a geographically distributed system can quickly cause scalability problems. The data is expected to be near to the computation where it is used. This data movement in any grid computing environment requires absolutely secure data transfers, both to and from the respective resources. The file management portlet enables easier access to data files.

Job management

The job management portlet enables an end user to submit the job in a user friendly way by selecting the resources and configuring the parameters such as working directory, executable, arguments, environment variables, standard input and standard output etc. The job is converted into job service description language (JSDL) which is a standard format to submit job on grid. The job management service for users provides job allocation, status management, data distribution and start/stop jobs. The jobs are submitted to resources using the JSDL format which is the standard format for Grid jobs. JSDL specifies different information required for successful application execution. One of the sub elements is application element which allows the user to specify the application to be executed on the grid and the various application parameters and arguments, including input and output files. Requirements element allows the user to specify different requirements needed for the execution of the job, such as amount of memory, number of CPUs, etc. Data element allows the specification of input and output staging, in order for the user to optionally define which files should be copied to and / or

from the working directory on the host where the job will be executed before job submission and after job completion, respectively.

Security

The resource access and information is available to only authenticated users. The users are supposed to produce credentials to access resources which can be of multiple mechanisms such as MyProxy, VOMS etc. The heterogeneous nature of resources and their differing security policies are complicated and complex in the security schemes of a grid computing environment. These computing resources are hosted in differing security domains and heterogeneous platforms. The grid computing data exchange must be protected using secure communication channels.

Resource management

The resource registry portlet configures the resources available in the grid. The resources are configured usually using domain.xml file. The most interesting aspect of the resource management area is the selection of the correct resource from the grid resource pool based on the service-level requirements and then to efficiently provision them to facilitate user needs.

These portlets enable an end user to use grid infrastructure easily without having the need to bother about developing the grid infrastructure and managing separate credentials for different resources.

IMPLEMENTATION DETAILS

In a grid, managing resources, services and computations is a challenging due to the heterogeneous nature, large number, dynamic behavior and geographical distribution of resources. This presents the need to easy provision of the resources and enables interoperability between the resources. PIE has been deployed on various Pentium Core 2 Duo Machines running Windows Xp and Ubuntu as Operating System and GT4 and UNICORE 6 as middleware. Grid sphere 3.1 was considered for portlets container and Apache Tomcat was considered for Servlet container. Vine toolkit 1.2 (Wang et al., 2006) was deployed which run Java CoG kits at the backend to connect to the clusters of Globus Toolkit and UNICORE 6. The resources utilized were from various Computer Science and Engineering Labs across Thapar University. The resources heterogeneous in hardware and software configurations and connected through local area network (LAN) but located in various subnets were used to deploy and test PIE. Here, technologies chosen to achieve this goal have been outlined.

Grid middleware

At the resource layer clusters of Globus and UNICORE 6 were installed. The Globus Toolkit (GT) provides functionality to glue together distributed resources to form computational grids and to create grid-based applications. It provides a secure, uniform interface for resource access and provides file transfer, remote job submission and meta-data query services. It is also one of the most widely-used middleware packages for the grid. GT is increasingly being developed in Java, which is also well suited for building complex web applications. We prefer Java-based frameworks for structured development and the ease of use. GT4.0.2 version was chosen to be installed as it is compatible with Vine Toolkit Version 1.2.

UNICORE 6 has been chosen as another grid middleware where UNICORE 6 servers were installed. UNICORE offers a ready-to-run grid system including client and server software. UNICORE makes distributed computing and data resources available in a seamless and secure way in intranets and the internet. Unicore-servers-6.3.2 was installed to enhancing pool of resources.

Web technologies

Servlets are a viable choice for reusable web applications and a number of different frameworks built on the top of servlets were explored during the course of development. Portlets provide a standard technique for creating web applications for deployment in a *container portal*. Each application is self-contained and is displayed in sub-window of the portal. The portlet framework allows for communication between portlets without the need for tight inter-portlet coupling. Java Specification Request (JSR) 168 defines the standard for portlets and several systems implement it, including Apache Jetspeed2 (Apache JetSpeed, <http://portals.apache.org/JetSpeed-2/>) and Gridsphere (Novotny et al., 2004).

GridSphere

The GridSphere portal framework provides an open-source portlet based Web portal. GridSphere enables developers to quickly develop and package third-party portlet web applications that can be run and administered within the GridSphere portlet container (Gridsphere, <http://www.gridisphere.org/>). Therefore, it was chosen as a portlet container. It is deployed on popular Apache Tomcat5.5.26 web-application container. The Apache Ant1.8.2 build system, which is used by GridSphere, allowed us to semi-automate this process. GridSphere Version 3.1 was used for development purposes.

Vine toolkit

Vine is a modular, extensible Java library that offers developers an easy-to-use, high-level Application Programmer Interface (API) for grid-enabling applications. Vine Toolkit Version 1.2 was installed for embedding portlets such credential management, file management, job management and security. Vine toolkit is the latest toolkit which provides ready to use grid portlets for credential management, file management, job management and resource management. Vine Toolkit is advancement over GridPortlets and OGCE, lately. The portlets can be designed in Adobe Flex and easily embedded into PIE. The developers can easily configure registry portlet to plug-in the resources for further usage purpose and extensibility of portal.

DISCUSSION

PIE portal is a set of vine grid portlets running in GridSphere portal framework. The interoperability is supported by several functions in PIE portal. A Grid is typically identified by its information system and hence when the PIE portal is installed, the portal administrator can connect the portal to several grid resources. The features of PIE are summarized in Table 1.

Configuring a resource

PIE provides a uniform interface to high performance computing infrastructures by mapping and configuring heterogeneous resources using Resource Registry portlet's domain.xml file as shown in Figure 3. The Resource information can be configured directly in Resource Registry Portlet or may be specified using domain.xml file. The domain.xml file has been configured for the resources available at Centre of Excellence in Grid Computing at Thapar University. The domain name depicts the domain related information of Thapar University. The <authenticationModule> in domain.xml file configures the type of authentication mechanism used to allow users to register to the Portal and make use of resources. The logic responsible for the user authentication is that the basic authentication module "PortletAuthModule" relies on the GridSphere authentication routines and checks whether user was successfully authenticated by GridSphere. The Portal File System tag is responsible for configuring the file system to be deployed. The <accountResource> is responsible for user's management. It manages user's roles and access rights to the underlying resources. The <hostResource> acts as a container or parent resource. It conforms to the physical host machine with some services installed on it. The <UnicoreGatewayResource> provides information related to the resource which has

Table 1. Features of PIE.

Feature	PIE
Programming effort	Low programming effort as the portlets can be easily designed in Adobe Flex
Resource registry	Pluggable portlets can be configured to use resources hosted on multiple middleware.
Authentication	Can embed any form of registration such as MyProxy, VOMS etc.
Authorization	Mapped to credentials and access control rights can be assigned to individual users by administrator
Scalability	More as simply resources can be configured using <hostResource> and hostname
Interoperability	Provided as jobs can be submitted to any resource irrespective of underlying platform
Extensible	Is extensible as other portlets can be easily embedded.
Security	Secure as access control is provided depending on credentials
Credential management	Credentials Manager available which stores and manages user credentials based on MyProxy etc.
Data management	File Management portlet available especially for managing data files
Information services	Information service can be used such as CIS of UNICORE and MDS of GT4
Collaboration	Available
Resource management	Resource registry maintains updated resource information

Unicore6 Server installed. The computational resources available were configured by updating domain.xml in the Resource Registry portlet of PIE. The hostnames were configured into domain.xml file which could update the set of resources available in job management portlet and allow end users to use the resources. The middleware installed on computational resources were GT4.2.1 and UNICORE6.

Managing certificates

After defining the grid environments, the user should upload his grid certificates belonging to the connected grids into the MyProxy server that works with strong interaction with the portal. Of course, if several grids accept the same certificate it is enough to upload their common certificate. The certificate manager takes care of certificates required for the user to run the jobs as shown in Figure 2. The certificates manager is installed and used separately to store different credentials of the user. MyProxy server was used to generate X.509 certificates for the end users which could also be maintained through certificate manager portlet.

Login into PIE

The login details are available to user through mailing the details to the PIE administrator which checks for credential authenticity. The users are authorized to access resources based on distinguished name (DN) mapping of their credentials provided as shown in Figure 4. The user accounts details are e-mailed to the user. The user can use the account details to login in PIE Login screen as shown in Figure 5. New credential can be managed using credentials manager portlet as shown in

Figure 6. An end user can simply login to PIE using the username and password allotted by the administrator and make use of resources only by providing authentic credentials such as X.509 certificates as provided for registering purposes.

Running a Job

Once the resources are configured as shown in Figure 7 of registry portlet they can be simply used, the next step is to upload the files to run the job. The file management portlet contains the files required by the user as shown in Figure 8. These files can be uploaded and downloaded at a specific location as and when required. The job management portlets in Figures 9 and 10 enables an end user to submit a job in a standardized format job service description language (JSDL) to the resource selected.

PIE also deals with security issues. It can solve different security issues while accessing different grid services. A significant interoperability problem results from the differences in the security models of systems such as UNICORE and Globus-based Grids (authentication based on GSI). The main difference between these systems is that a basic version of UNICORE and GLOBUS does not support dynamic delegation while GSI-compliant systems do. Currently, to overcome this problem, we allow a portal to authenticate on behalf of end-users. In case of UNICORE PIE supports GSI-enabled extension for UNICORE gateway what enables to authenticate users by using proxy certificates. Account creation allows automated, user-friendly user registration at the portal and to underlying Grid middleware and third-party services used through the portal (registration modules are used to register new user accounts in external services). User authentication allows the user to login into the portal and be authenticated.


```

- <domain name="vo.thapar.edu" label="THAPAR Grid" description="Thapar Grid">
- <!-- Portlet authentication -->
- <authenticationModule key="PortletAuthModule" order="1" />
- <!-- Credential repository authentication used with MyProxy server -->
- <!-- authenticationModule key="CredentialRepositoryAuthModule" order="2" / -->
- <!-- Proxy created from local user cert and key -->
- <!-- authenticationModule key="VomsDefaultCredentialAuthModule" order="3" / -->
- <vineRoleResource />
- <!-- Portal -->
- <hostResource name="portal" hostname="localhost" label="Portal" description="Portal">
- <!-- Portal file system -->
- <portalFileSystem label="Portal File System" description="Portal File System" />
- <!-- Account manager -->
- <accountResource name="GuestAccountManager" label="Guest Account" description="Guest Account Manager">
- <gridsphereRegistrationResource name="GridsphereRegistration" label="GridsphereRegistration" />
- </accountResource>
- </hostResource>
- <!-- MyProxy resource LMT Thapar -->
- <hostResource name="server" hostname="lmt.thapar.edu " label="server" description="Linux Host (2 CPUs)">
- <!-- MyProxy -->
- <myproxyResource label="knoppix (MyProxy)" useCredential="false" checkConnection="true"
timeoutMiliseconds="5000" />
- </hostResource>
- <!-- seagrass test unicore -->
- <hostResource name="gridlab" hostname="gridlab" label="gridlab" description="Linux Host (2 CPUs)">
- <!-- Unicore Gateway -->
- <unicoreGatewayResource name="unicore" label="gridlab (UNICORE6)" port="7777" protocol="https"
servicePath="/DEMO- SITE/services">
- <resourceAttribute name="description" value="UNICORE Gateway 6.0 version 1.0" />
- <resourceAttribute name="serviceTimeout" value="3" />
- <resourceAttribute name="userKeystore" value="true" />
- </unicoreGatewayResource>
- <!-- Unicore SMS -->
- <hostResource name="icmWMS" hostname="coegc.thapar.edu" label="COEGC" description="Resource broker in
ICM">
- </hostResource>
- </domain>

```

Figure 3. Domain.xml file of registry portlet.

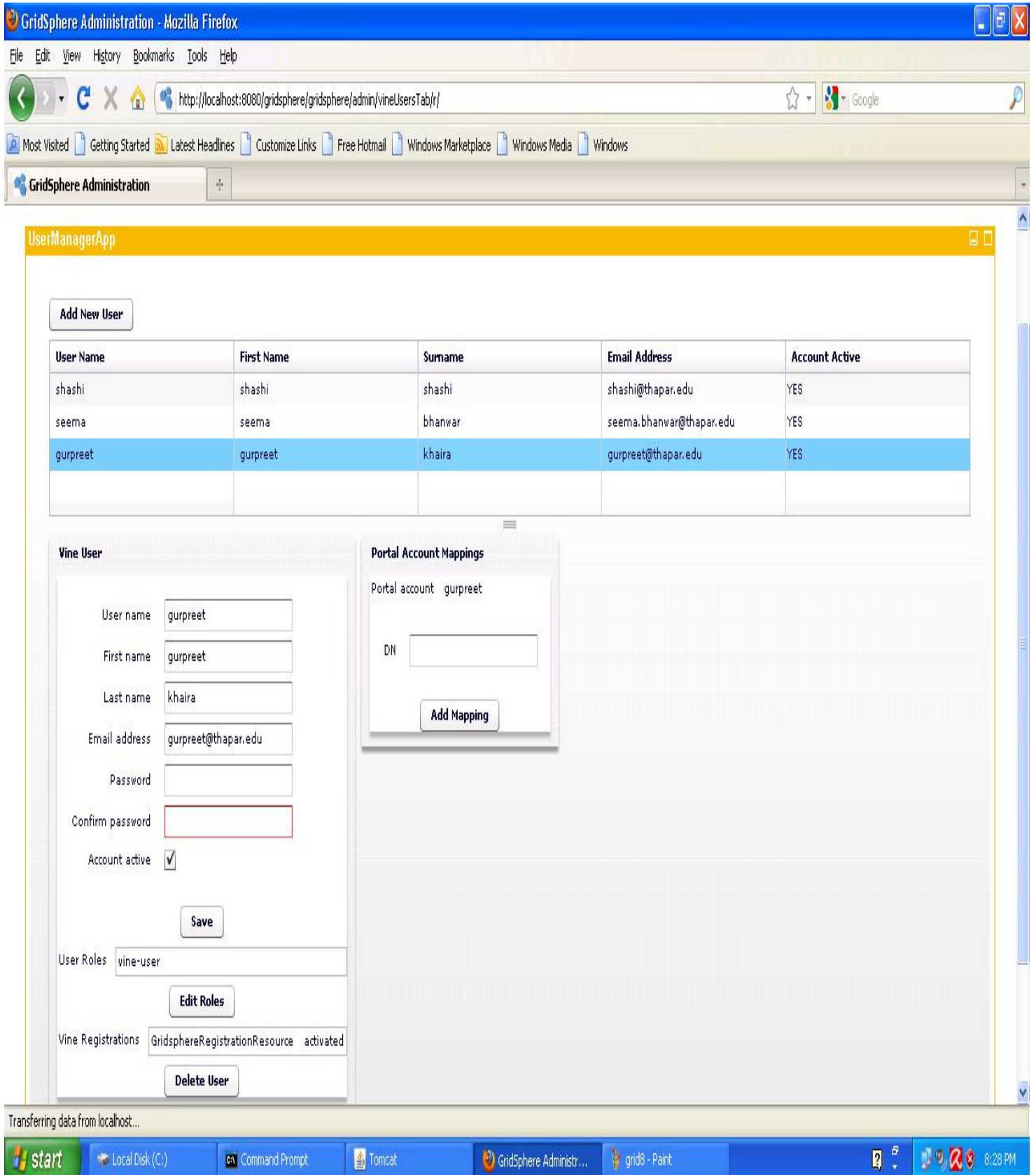


Figure 4. PIE new user registration.

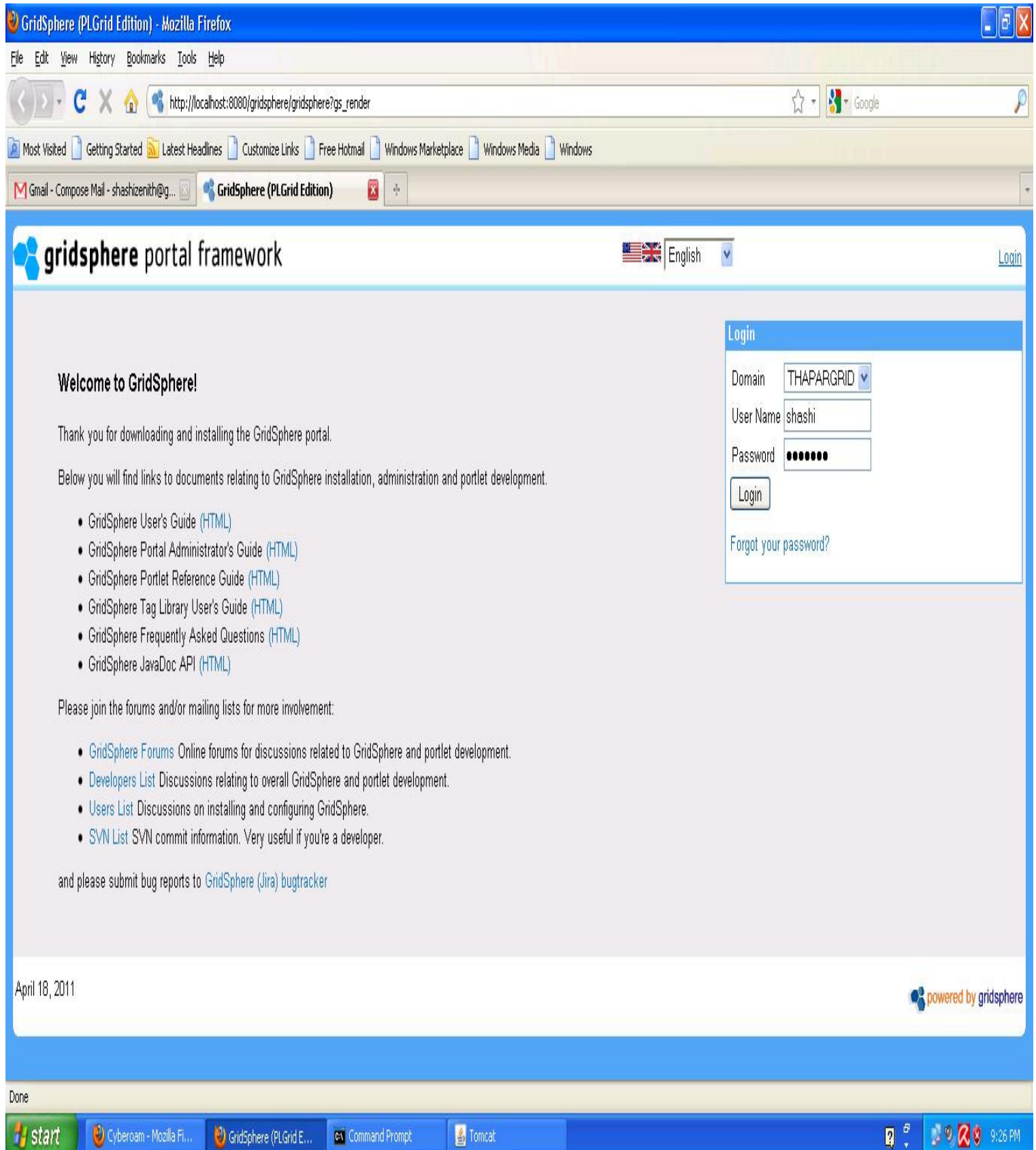


Figure 5. PIE login screen.

Credentials | Files | Jobs | Resources

CredentialManagerApp

New credential | Edit credential | Deactivate credential | Delete credential

DN	Label	Status	Time remaining

New credential

Label: * Myproxy credential

User name: * research.gridlab.com

Name: research

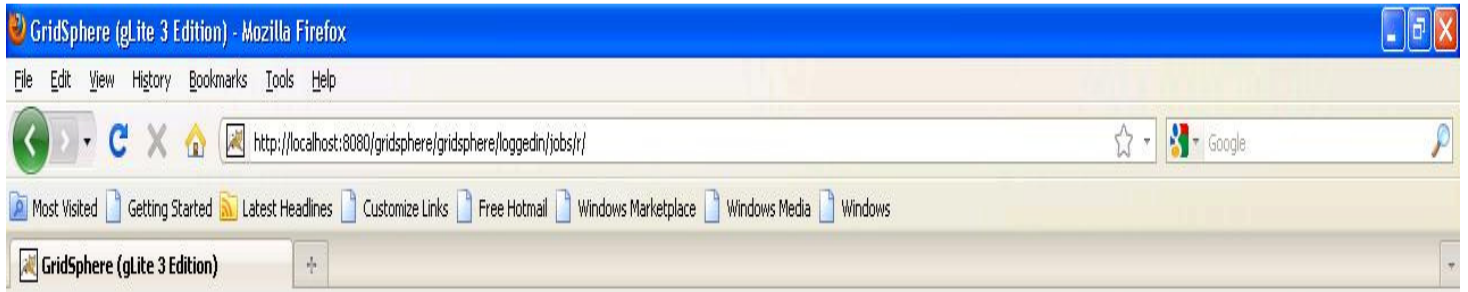
Lifetime: * 8100

Passphrase: * xxxxxx

Save credential | Cancel

Transferring data from localhost...

Figure 6. PIE credential management portlet.



gridsphere portal framework Welcome , shashi shashi [Administration](#) [Content](#) [Layout](#) [Profile](#) [Home](#) [Logout](#)

[Credentials](#) [Files](#) [Jobs](#) [Resources](#)

FileManagerApp

Portal File System

Name	Modified	Size	Kind
bridge.xml	Apr 17, 2011 08:37:29 PM	1536 bytes	File
comcli.xml	Apr 17, 2011 08:37:35 PM	631 bytes	File
depviewer.xml	Apr 17, 2011 08:37:40 PM	1595 bytes	File
matlab.exe	Apr 17, 2011 08:18:00 PM	229376 bytes	File
rtw_core.xml	Apr 17, 2011 08:40:00 PM	2093 bytes	File
sl_mi.xml	Apr 17, 2011 08:40:00 PM	1461 bytes	File

Portal File System

Name	Modified	Size	Kind
bridge.xml	Apr 17, 2011 08:37:29 PM	1536 bytes	File
comcli.xml	Apr 17, 2011 08:37:35 PM	631 bytes	File
depviewer.xml	Apr 17, 2011 08:37:40 PM	1595 bytes	File
matlab.exe	Apr 17, 2011 08:18:00 PM	229376 bytes	File
rtw_core.xml	Apr 17, 2011 08:40:00 PM	2093 bytes	File
sl_mi.xml	Apr 17, 2011 08:40:00 PM	1461 bytes	File

April 18, 2011

powered by gridsphere



Figure 7. PIE file management portlet.

ResourceManagerApp

Save Undo

Domain registry is in sync with server

Text view Tree view

```
<domain description="Globus Toolkit 4" label="Globus Toolkit 4" name="gt4">
<authenticationModule key="PortletAuthModule" order="1"/>
<authenticationModule key="CredentialRepositoryAuthModule" order="2"/>
<vineRoleResource/>
<hostResource description="Portal" hostname="localhost" label="Portal">
<portalFileSystem description="Portal File System" label="Portal File System"/>
<accountResource description="Account Manager" label="Account" name="AccountManager">
<gridsphereRegistrationResource label="GridsphereRegistration" name="GridsphereRegistration"/>
</accountResource>
<hostResource description="Portal" hostname="localhost" label="Portal" name="portal">
<accountResource description="Guest Account Manager" label="Guest Account" name="GuestAccountManager">
<gridsphereRegistrationResource label="GridsphereRegistration" name="GridsphereRegistration"/>
<gssCertificateRegistrationResource caCertificatePath="/home/user/ca/cacert.pem" caKeyFilePath="/home/user/ca/cakey.pem" caKeyPassword="secret" label="GssDemoCertRegistration" name="GssDemoCertRegistration"/>
<gt4RegistrationResource chownCmd="/home/adminuser/bin/chown" gridMapfileAddEntryCmd="sudo /usr/local/globus/gt404/sbin/grid-mapfile-add-entry" gridMapfileDeleteEntryCmd="sudo /usr/local/globus/gt404/sbin/grid-mapfile-delete-entry"
label="Gt4Registration" mkdirCmd="/home/adminuser/bin/mkdir" name="Gt4Registration" superUserName="adminuser" targetHost="unicore3.gridlab.com"/>
<uniconRegistrationResource caCertificatePath="C:\Program Files\UNICORE_Servers\certs\demo-ca-cert.pem" caKeyFilePath="C:\Program Files\UNICORE_Servers\certs\demo-ca-cert.pem" label="UniconRegistration" name="UniconRegistration"
superUserName="research" targetHost="research.gridlab.com"/>
</accountResource>
</hostResource>
</hostResource>
<hostResource description="Linux Host (2 CPUs)" hostname="fury.man.poznan.pl" label="Fury" name="fury">
```

transferring data from localhost...

[root@unicore1.gridlab.com] GridSphere (gLite 3 Edi... | Firefox |

Figure 8. PIE resource registry portlet.

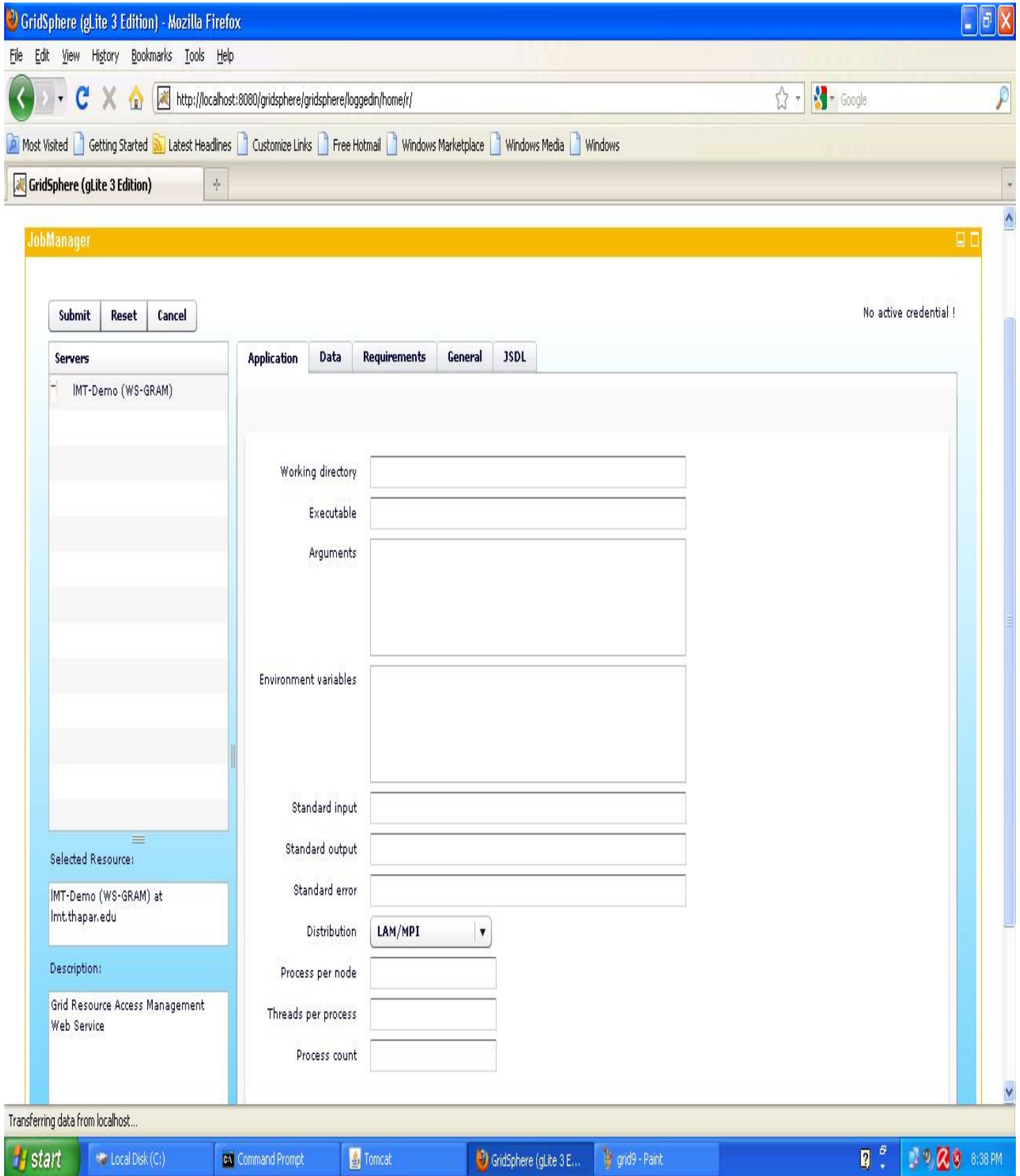


Figure 9. PIE job submission portlet.

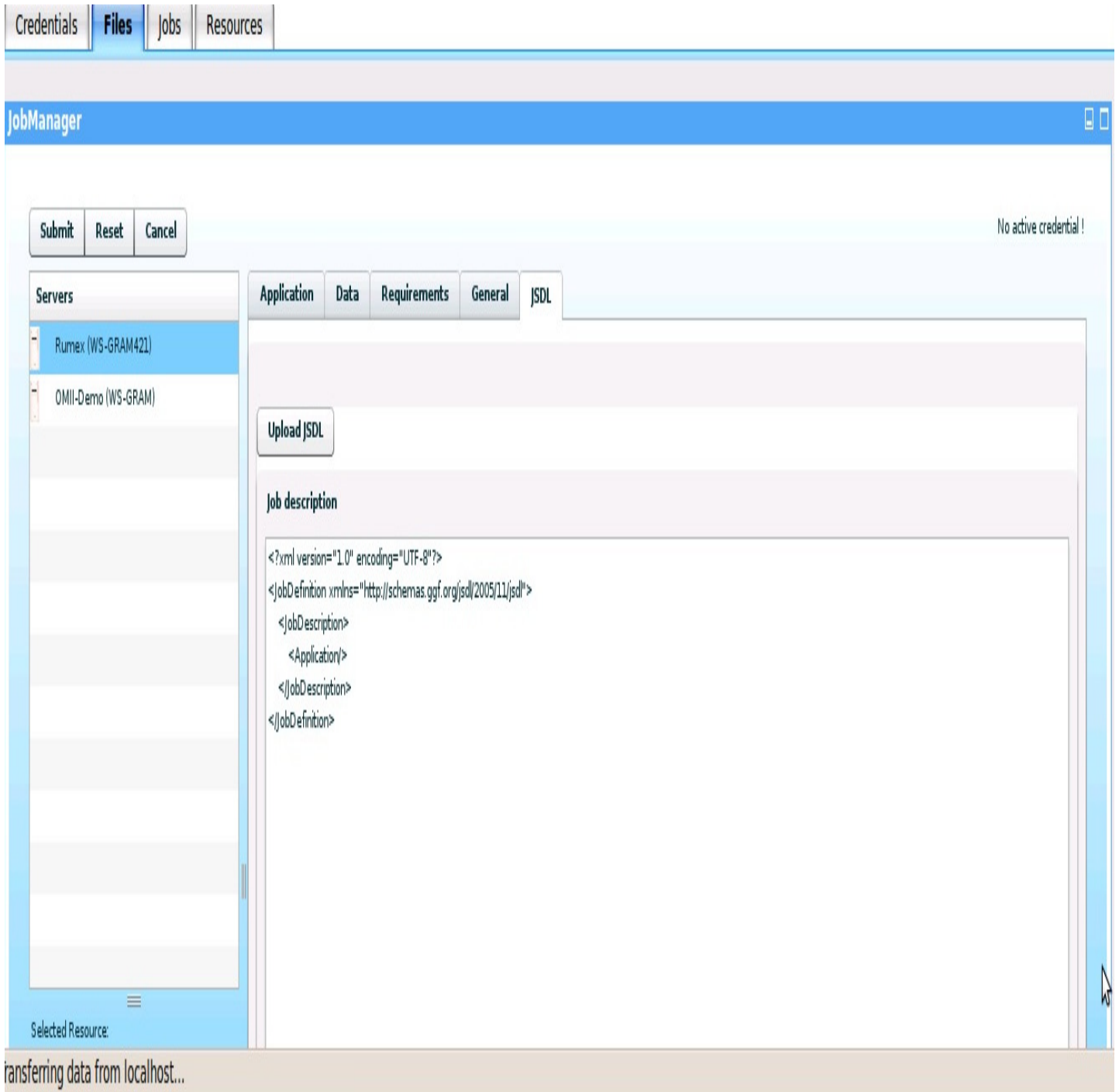


Figure 10. PIE job management.

Authorization allows the administrator to access third-party authorization systems to change user permissions.

Common problem to solve while using different middleware infrastructures is user management and account provision. PIE is able to register users in different middleware infrastructures like UNICORE 6 or Globus

GT4. It contains registration modules which allow registering users on different resources. Currently, the following are provided: GridsphereRegistrationResource, allows adding new users to Gridsphere portal, Gt4RegistrationResource, creates user accounts on Globus Toolkit 4 host machine, Unicore6RegistrationResource,

creates user accounts on Unicore 6 XUADB, DmsRegistrationResource, registers users on Data Management Service. TUX-INTERO (Bhanwar and Bawa, 2010) aimed at designing a portlet based portal for secure interoperation of Grids where plugging of resources was available by making use of gateway one per middleware.

PIE is an effort to design and develop pluggable, interoperable and extensible portals which entail less programming effort from scientists and developers.

Conclusion

This paper discusses PIE which is a pluggable, interoperable and extensible portal. PIE has been designed with the aim of allowing end users to easily run jobs on grid and allow pluggable portlets to be embedded into portal as per use. The paper demonstrated that portal based solutions help users to achieve interoperability and collaborate disparate grids. It achieves the original vision of Grid which is aimed at single massive infrastructure. Here, any user is allowed to submit job instead of getting a user trained in special type of software or technique. Resource based model allows straightforward service support and configuration by uniform way regardless of the target technology. PIE addressed security issues which always occur while accessing different middleware stacks and solved them by using Vine registration, authorization modules and proxy certificates support. Role management mechanism allows administrators and end users to easily manage permissions.

REFERENCES

- Apache JetSpeed [Online]. Available: <http://portals.apache.org/JetSpeed-2/>.
- Barbera R, Falzone A, Ardizzone V, Scardaci D (2007). The GENIUS Grid Portal: Its Architecture, Improvements of Features, and New Implementations about Authentication and Authorization. Proceedings of 16th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises., pp. 279-283.
- Bhanwar S, Bawa S (2010). Interoperability: Bridging the Gap. Int. J. Comput. Appl., 7: 15-21.
- Bhanwar S, Bawa S (2010). TUX-INTERO: A Portal for secure interoperation of Grids. Int. J. Eng. Sci. Technol., 2: 3335-3343.
- Bhanwar S, Bawa S (2010). A Reputation based enhancement of Enhancing Reputation in a trust based system, Proceedings of CNSA-2010, pp. 440-451.
- Boardman R, Crouch S, Mills H, Newhouse S, Papayet J (2007). Towards Grid Interoperability. In: *All Hands Meeting 2007, OMII-UK Workshop*, 10 September - 13 September, 2007, Nottingham, UK.
- Cactus Code [Online]. Available: <http://www.cactuscode.org>.
- Condor High Throughput Computing [Online]. Available: <http://www.cs.wisc.edu/condor/htc.html>.
- Dahan M, Thomas M, Roberts E, Seth A, Tomislav U, (2004) GridPort: Using Globus for Grid- Enabled Web Portals in Proceedings of the 13th IEEE International Symposium on High Performance Distributed Computing (HPDC'04), pp. 272-273.
- Gannon D, Fox G, Pierce M, Plale B. Grid Portals: A Scientist's Access Point for Grid Services (DRAFT 1). Available: <http://www.extreme.indiana.edu/groc/ggf-portals-draft.pdf>.
- Gaussian [Online]. Available: <http://www.gaussian.com/>.
- Gridsphere [Online]. Available <http://www.gridsphere.org/>.
- IBM Websphere Portal [Online]. Available:<http://www.ibm.com/developerworks/websphere/zones/portal/newto/>.
- Integrated e-Science Environment for CLRC [Online]. Available: <http://www.e-science.clrc.ac.uk/>
- Joseph J, Fellenstein C (2003). Grid Computing. IBM Press.
- Laszewski G, Russell M, Foster I, Shalf J (2002). Community software development with the Astrophysics Simulation Collaboratory. Concurrency and Computation: Practice and Experience, 14: 1289-1301.
- Laszewski G, Foster I, Gawor J (2000). CoG Kits: A Bridge between Commodity Distributed Computing and High-Performance Grids. Proceedings of the ACM Java Grande Conference, pp. 97-106.
- NCSA Alliance Scientific Portal Project [Online]. Available: <http://www.extreme.indiana.edu/alliance/>
- Novotny J (2004). The Grid Portal Development Kit. Concurrency and Computation: Practice and Experience, Special Issue: Grid Computing Environments, 14: 13-15.
- Network for Earthquake Engineering Simulation Grid [Online]. Available: <https://www.nees.org/it/>.
- Novotny J, Russell M, Wehrens O (2004). GridSphere: An advanced Portal Framework. Proceedings of Euromicro Conference - 2004, pp. 412-419.
- Menon S (2007). Interoperability: A new Paradigm, Novell Corporate Presentation, Putrajaya.
- MIMD Lattice Computation (MILC) Collaboration [Online]. Available: <http://www.physics.utah.edu/~detar/milc/>
- Mock S, Mueller, K, Pierce M, Youn C, Fox G, Thomas M (2002). A BatchScript Generator Web Service for Computational Portals. Proceedings of 11th IEEE International Symposium on High Performance Distributed Computing, pp. 421.
- MyProxy [Online]. Available: <http://grid.ncsa.illinois.edu/myproxy>.
- P-Grade [Online]. Available: <http://portal.p-grade.hu/>.
- Russell M, Allen G, Foster I, Seidel E, Novotny J, Shalf J, Laszewski G, Daues G (2002). The Astrophysics Simulation Collaboratory: A Science Portal Enabling Community Software Development. Future Generation Computer Systems, Special issue: Advanced grid technol., 21(2): 259 - 270.
- Thomas M, Dahan M, Mueller K, Mock S, Mills C (2002). Application Portals: Practice and Experience. Grid Computing Environments, Special Issue of Concurrency and Computation: Practice and Experience, 14: 1427-1444.
- UNICORE [Online]. Available: <http://www.unicore.eu/>
- uPortal by JA-S [Online]. Available: <http://www.uportal.org>.
- Wang XD, Yang X, Allan R (2006). Flexible Grid Portlets to Access Multi Globus Toolkits. Proceedings of GCC Workshops 2006, pp. 565-570.
- Xang X, Dove MT, Hayes M, Calleja M, He L, Rust PM (2006). Survey of Major Tools and Technologies for Grid-enabled Portal Development. Proceedings of the UK e-Science All Hands Meeting, pp. 353-356.
- Vine Toolkit [Online]. Available: <http://www.vine.org>.