

Tutorial 6: Grid Portal

Jipu Jiang, National e-Science Centre



UNIVERSITY
of
GLASGOW



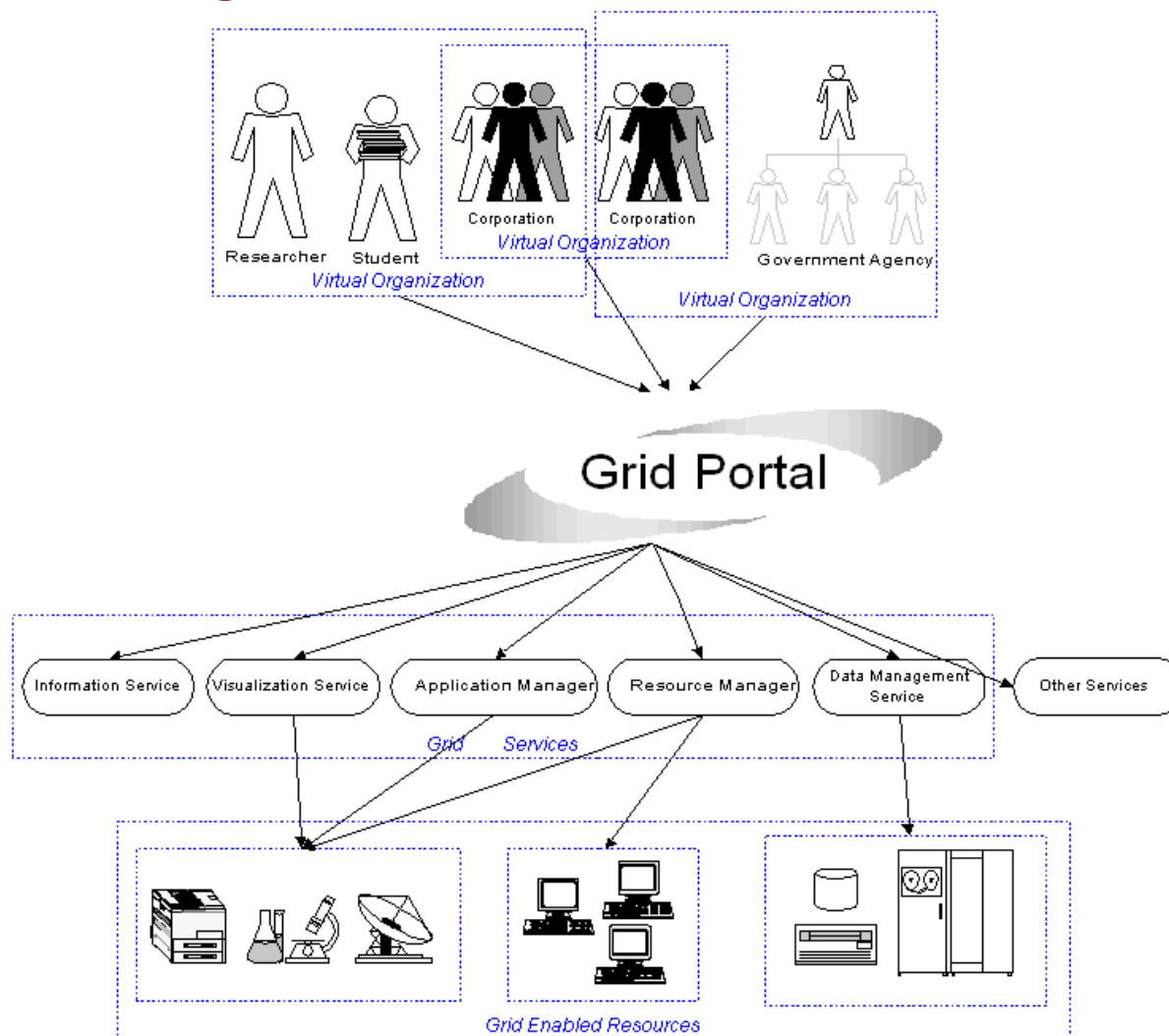
Portal

- A web portal is a base that acts as a starting point to help navigation through huge amounts of information on the World Wide Web.
 - Such as Yahoo!
- A web portal is different from a web search engine, such as Google.
 - Search engines tell us where the information is, it does not aggregate it.
 - Instead, a web portal aggregates information and provides index services.

Grid Portal

- **Grid Portals build upon the familiar Web portal model.**
 - To deliver the benefits of Grid computing to virtual communities.
 - Providing a single access point to Grid services and resources.
- **A Grid portal is a web based application that will typically provide personalization, single sign on, authorization, content aggregation, and seamless access to Grid heterogeneous resources and services supporting end user communities in one or more tasks through a Web-based user interface.**
 - Often given as Virtual Research Environments
- **Web Portal vs Grid Portal**
 - The major difference is that a Grid Portal is for Grid applications and resources rather than Categorized web pages/sites.

Portals Bridge Users and Services



Benefits of Grid Portal

- **Grid Portals hide the complexity from end users**
 - The very heterogeneity that is the strength of computational and data Grids can also make application for such an environment extremely difficult to use, especially for non-computer scientists.
 - For example, to invoke the BRIDGES GridBLAST service
 - ▶ `%> java -classpath ./build/classes/./build/lib/./lib/:\$CATALINA_HOME/shared/lib/.$CATALINA_HOME/common/lib/.\$CATALINA_HOME/common/endorsed/.$CLASSPATH \bridges.jobsub.portlets. TestBridgesClient \http://127.0.0.1:8080/ogsa/services/bridges/JobServiceFactoryService \jiangj, Job1, >seq1\nATCTAGTACTAGTACTGTACTGATCA\n\>seq2\nACCATTTGATACACGATTAGCAATGA\n\>seq3\nTCGTAGATAGATGATTGATGATGTGA\n, \-p blastn, ecoli.nt, 10, 7, yes, no, jiangj@dcs.gla.ac.uk, html`

To invoke the GridBLAST through a portal

GridBlastPortlet

GridBLAST -- Job Submission Page

Enter your input data in FASTA format here or upload a file (below):

```
>seq1
ATCTAGTACTAGTACTGTACTGATCA
>seq2
ACCATTTGATACACGATTAGCAATGA
>seq3
CGATGACGATTGAGGATTAGATTAGTA
```

Please enter a name for this job:

Please enter an email address where notification of the job completion is to be sent (if this is left blank no notification can be sent but your job will still work):

BLAST database:

BLAST program:

e-value (default 10):

word size:
(default= 3 for protein, 7 for nucleotide)

Output file format:

Further Benefits of Grid Portals

- **Grid Portals allow scientists or engineers to focus on their problem area by making the Grid a transparent extension of their desktop computing environment.**
 - The Web-based interfaces are more familiar to non-IT savvy users
 - Ideally require minimum learning time to use the services
- **Support thin clients**
 - No client side Grid installations needed.
 - A browser is all they need.
- **Visualization**
 - Graphical user interface shows information in a more sophisticated way.

Further Benefits of Grid Portals

- Enhanced interaction among services.
- Workflow management is straightforward.
- Grid Portals facilitate the establishment of Virtual Organizations (VOs) providing
 - a single point of access to heterogeneous Grid-based resources and services.
 - a single access point through which authorization can be checked/enforced.
 - a customized view of software and hardware resources specific to their particular problem domain.
 - easy to add commodity IT tools to the Grid portal.
 - ▶ e.g. wiki, etc., to share more information to globally distributed researchers in the VO.

Portal Frameworks

- **Previous Grid portal developments show up several problems.**
 - **Most portals have similar core functions.**
 - ▶ E.g. login, access control, customization, etc.
 - **Difficult to reuse code or components.**
 - ▶ Previous portals were generally hard coded to the underlying Grid infrastructure details and the associated code base
 - ▶ Weren't explicitly designed to be reusable.
 - **Poor user interface design.**
 - ▶ Developing teams had no Human Computer Interaction specialist.
 - ▶ Focused more on functionality rather than user interface thereby making the portal harder to use.
 - **Required programmers to repeat coding across projects.**
 - ▶ Web application development often a tedious task with little in the way of reusable components, forcing developers to constantly "re-invent" the wheel.
- **To solve those problems, Grid Portal Frameworks were developed to support rapid establishment of portals and improve component reuse.**

Portal Frameworks today

- **Commonly used portal server packages**
 - **GridSphere Portal Framework**
 - ▶ An open source research project
 - ▶ Widely used as Grid portal framework.
 - **WebSphere Portal**
 - ▶ Commercial software solution from IBM
 - ▶ Heavy-weighted portal solution
 - **Pluto**
 - ▶ One of the Apache Portal projects
 - ▶ Standard JSR 168 reference implementation (more about JSR168 in next slide)
 - **uPortal**
 - ▶ Originally designed as a Java-based framework for creating educational web portals. Then integrated with Apache Pluto to support JSR 168
 - ▶ Lack of architecture makes it hard to use.
 - **Liferay**
 - ▶ Focusing on enterprise side of the story
 - ▶ Not the Grid Community's favorite.
 - **Etc.....**

Portlets

- Portal frameworks don't automatically mean components are reusable though.
 - Needs standards - Portlet
- Portlets:
 - are building blocks of a portal.
 - look like little windows inside a typical Web page.
 - act as pluggable user interface components.
- Two perspectives:
 - From a users' perspective:
 - ▶ a window in a portal page that provides a specific service
 - From an application development perspective:
 - ▶ a portlet is a software component commonly written in Java, managed by a portlet container, which handles user requests and generates dynamic content.

A Portlet Example

gridisphere portal framework
open-source / portlet jsr168 compliant

Logout
Welcome, Root User

GLASS Welcome Administration Grid gridblast Introduction Portlet

Project Introduction Portlet

National e-Science Centre UNIVERSITY of GLASGOW GLASS DyVOSE Shibboleth JISC

Welcome to the GLASS project Portal

The GLASS (GLASgow early adoption of Shibboleth) project is proposed to investigate how the integrated directory infrastructure for unified user account management currently being rolled out across the University of Glasgow can be utilised in a Shibboleth environment. In particular we will explore the usage and adoption of Shibboleth through a variety of case studies in the education and medical domains including those requiring Grid-based authentication and authorisation. In doing this we will draw on a variety of security focused e-Science projects within the National e-Science Centre at the University of Glasgow including the JISC funded DyVOSE project and a collection of e-Health projects.

More information is on the GLASS project Website: <http://www.nesc.ac.uk/hub/projects/glass>

Contacts

For more information on GLASS project please contact:

- ◆ Prof. Richard Sinnott (r.sinnott@nesc.gla.ac.uk)
- ◆ Dr. John Watt (j.watt@nesc.gla.ac.uk)
- ◆ Mr. Jipu Jiang (j.jiang@nesc.gla.ac.uk)

Add portlets: A Introduction Portlet Add

Resource Browser Portlet

Resources Services Job Queues Jobs Accounts

Refresh List General Change Profile

Resource	Hostname	Description
Portal	localhost	Hosts the GridSphere Portlet Container
NeSC		
MyProxy	myproxy.grid-support.ac.uk	Hosts the GridLab MyProxy Credential Repository

One Portlet

Add portlets: A Introduction Portlet Add

19 February 2007

Portlet Standards

- **Portlet standards:**
 - JSR-168 (Java Portlet Specification) defines a standard Portlet API that enables interoperability for portlets between different web portals . Next generation portlet specification is JSR-286
 - ▶ This standard attempts to ensure the compatibility of portlets developed under different portal frameworks.
 - WSRP (Web Service for Remote Portlets) defines a universal API that allows portals of any type to consume portlets of any type.
 - ▶ This standard attempts to ensure that remote portlets can be invoked
- **WSRP and JSR-168 are complementary specifications**
 - JSR-168 builds the foundation for WSRP and provides the possibility of inter-operability of portlets

Portlets vs Servlets

- Portlets are very similar to Java Servlets. They both process HTTP requests and produce HTML output
- But Portlets produce fragments of markup code that are aggregated into a portal page
 - Portlets API extends the Servlet API.
 - Portlets run in a portlet container inside of a Servlet container (e.g. Tomcat, Axis etc.), which is a layer that runs on top of an application server.
 - Portlets' HTML output is only a part of a Web page. The portal server fills in the rest of the page with headers, footers, menus, and other portlets.
- Most mainstream portal frameworks are built on portlet specification.

Writing a JSR 168 portlet: HelloWorld

```
import javax.portlet.*;

public class ActionHelloWorld extends GenericPortlet {

    public void processAction(ActionRequest req,
        ActionResponse res) throws PortletException, {
        .....
    }

    public void doView(RenderRequest request,
        RenderResponse response) throws PortletException, {
        ...
    }
}
```

- Red words shows the differences from a Servlet.

GridSphere vs WebSphere

Feature	<i>Gridsphere</i>	<i>Websphere</i>
License	Open Source	Commercial Product
Complexity	Low	High
Re-use of Publicly Available Components	Good - deploys into Apache Tomcat and uses log4j	Poor - proprietary implementations of application server, logging package and http server
Documentation	Good - about the right amount	Almost too much - things are hard to find
Installation/Configuration	Easy & Quick	Cumbersome - has to be downloaded as lots of separate files
Special Hardware Requirements	No	At least 1 GB memory per processor and At least 4GB disk space
Debugging Your Own Portlets	Easy - log4j can be used	Hard - in-code debugging statements need to be changed to support IBM logging mechanism
Stable	Not very stable	Better than GridSphere
JSR168 Compliance	Full	Full from version 5.1 onwards
WSRP Compliance	No	Yes
Security	Not many security features	Very strong, even URLs are encrypted
Overall Impression	Lightweight, easy-to-use, basic	Complex, monolithic, feature-rich

GridSphere: pros and cons

- GridSphere was chosen as the Grid Portal Framework at NeSC because
 - Lightweight framework
 - White-box framework, open source, free to modify the code
 - Well support for Grid/Globus, by installing GridPortlets plug-in
 - Based on most open source technology that have been widely used by the academia, such as Log4j, Apache, Tomcat, etc.
 - Support rapid portlets development
 - Visual UI tags and beans makes presentation development much easier
 - Tools to create, build, and deploy portlets.
- But GridSphere still has some insufficiencies as a Grid Portal to fill the Grid Community needs.
 - no support for portlet/service searching.
 - Issues of GT3 and GT4 portlets cannot co-exist in the same Grid Portal
 - ▶ Challenge of evolving Grid technologies and evolving portal environments
 - Closed authorization environment.
 - ▶ Though GridSphere supports Role Based Access Control (RBAC)
 - ▶ Hard to plug in other authorization frameworks, e.g. PERMIS.

GridSphere in Action

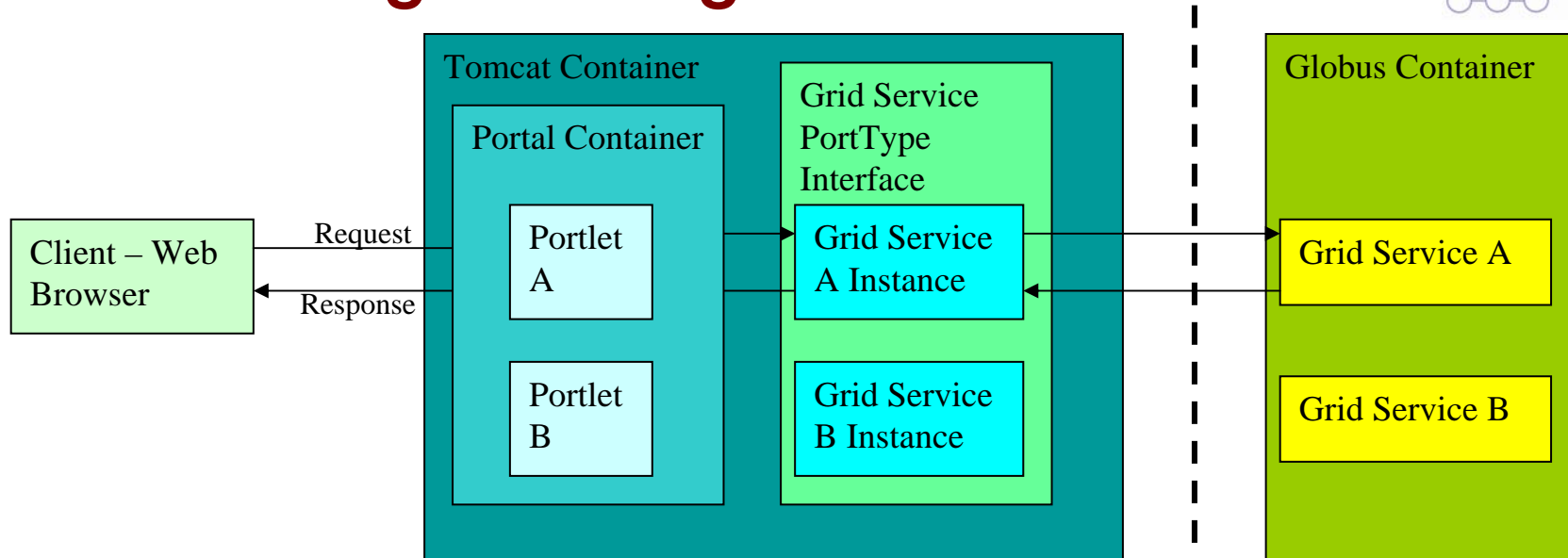
- Installation manual can be found on GridSphere website. But there are few points must be noticed
 - Before you install GridSphere, add a line to \$CATALINA_HOME/conf/tomcat-user.xml

```
<user name="gridsphere" password="gridsphere" roles="manager,admin" />
```
 - If using Apache Tomcat 5.5.20, the [apache-tomcat-5.5.20-compat.zip](#) must be patched.
- **Create a new portlet**
 - GridSphere provides tool to create new portlet
 - ▶ In <gridsphere_root>, run 'ant new-project'
 - ▶ A complete directory structure will be constructed under <gridsphere_root>/projects/ with the name you give.
- **Walkthrough**

Portlet development process

- Typically, the basic programming process is made up by 7 steps:
 - Create a new Gridsphere project: create the empty portlet by running the Ant script "ant new-project". Then a new folder with the name that you entered will be created under the `<gridsphere_root>/projects/`
 - Write the Java code and place the portlet java source code in the `<portlet_root>/src/`
 - Place your JSP (Java Server Page) files in `<portlet_root>/webapp/jsp`
 - Edit the 3 XML configuration files located under `<portlet_root>/webapp/WEB-INF/`
 - ▶ Portlet.xml, layout.xml, and group.xml
 - Run "ant install" under `<portlet_root>` to compile the source
 - Run "ant deploy" under `<gridsphere_root>` to deploy the project into tomcat
 - Restart the Tomcat container

Portlet Programming for Grid Services



- Method *processAction(ActionRequest req, ActionResponse res)* should contain the code to invoke Grid Services.
- Trickier than write a plain Grid client. Must be very careful with the JAR files in your \$CATALINA_HOME/share/lib.
 - Be patient if getting error messages. Read them carefully to see where the errors are.

Portlet with Globus Toolkit 4

- **Pre-requisition**

- Must have Globus Toolkit 4 and GridSphere portal installed
- Deploy Globus into Tomcat
 - ▶ `%> ant -f share/globus_wsrf_common/tomcat/tomcat.xml -Dtomcat.dir=$CATALINA_HOME deployTomcat`
- Copy Globus library to GridSphere
 - ▶ `%> cp $GLOBUS_LOCATION/lib/*.jar <portlet_root>/lib`
 - ▶ `%> cp $CATALINA_HOME/webapps/wsrf/*.jar <portlet_root>/lib`
- Copy Client-config.wsdd file to portlet directory in Tomcat
 - ▶ `%> cp Client-config.wsdd $CATALINA_HOME/webapps/<portlet_name>/WEB-INF/classes/`

- **Then we can start writing the Java source in the portlet**

- In the *processAction(ActionRequest req, ActionResponse res)* method, initialize CreatEPR and Client classes and execute them in order.
- Now you have your Globus service results in your portlet, display them.

Insufficiency of JSR 168

- **JSR-168 is not perfect**
 - Tweaking is often needed to drop a JSR-168 portlet to a different portal containers
 - Interportlet communication to let portlets coordinate some tasks and provide richer user experience is not supported.
 - Doing all the controller work in the processAction method of a portlet means cluttering your code with complex if/else blocks.
- **Portal frameworks typically provide their own “improved” APIs based on JSR-168.**
 - GridSphere team provides the ActionPortlet model
 - ▶ library of UI tags
 - ▶ ActionPortlet interface instead of GenericPortlet.
 - ▶ Every button on the JSP page connects to a single method.
 - But usually, portlets build on these APIs would not work in other portal frameworks.

Outlook for the future

- Although elementary Grid services exist that enable scientific application developers to authenticate, access, manage, and discover sophisticated remote resources, these frameworks are not compatible with the commodity technologies and frameworks used by application scientists today.
- Portal technology can be the bridge connecting Grid with other researchers
 - (i.e. the majority of people, not just IT experts).
- With Grid Portal technology, accessing to the Grid is no longer restricted to computers.
 - PDAs, mobile phones, and any other portable devices will also be able to enjoy the power of Grid technology through the web based access to portals.
- The future Grid should be as easy to use as the Internet!

Useful resources

- GridSphere Programming
 - <http://www.gridsphere.org/gridsphere/html/gridsphereguide.pdf>
- GridSphere API, Tag Library, and GridPortlet Guide
 - <http://www.gridsphere.org/gridsphere/gridsphere?cid=docs>
- JSR-000168 Portlet Specification
 - <http://www.jcp.org/aboutJava/communityprocess/review/jsr168/>
- Pluto - Pure JSR 000168 implementation
 - <http://portals.apache.org/pluto/>
- Email: j.jiang@nesc.gla.ac.uk