A Component Framework for Building Web Science Gateways and Portals

Mehmet Nacar
Indiana University
Computer Science Department
mnacar@cs.indiana.edu

Advisor: Prof Geoffrey C. Fox

Outline

- Motivation
- Background
- Science gateways
- Case studies
- Research objectives
- Architecture
- Workflows and DAGs
- Contributions

Introduction

- Problem statement
  - "A novel approach to build a better component model for Grid portals that can enable Grid operations all in one application"
- A component model of Grid portals that can simplify portlet applications with Grid tags.
- Integrating workflow execution and monitoring in science gateways by using session management and persistency
- Case studies to research and generalize science gateway needs for different disciplines
- We introduce Grid tag libraries framework to support reusable Grid operations portable among Grid portals
- Our approach is to build Grid portals based on application developers standpoint

Motivation

- A new approach to simplify science gateway building
  - Fine-grained and modular Grid operations
- We want to build advanced portlets to apply science applications
- There are approaches to build science portals
  - Open Grid Computing Environments (OGCE) is collection of portlets and tools for building portals
  - GridSphere proposes additional services and portlets for Grid
- We need to find a way to collect separate Grid operations in one portlet (e.g. multi-staging)
  - VLab and QuakeSim portlets
- In addition to Grid operations, the framework should provide user session management
  - **Session**: user sessions on Web browsers
- We need an extensible component framework to add new capabilities

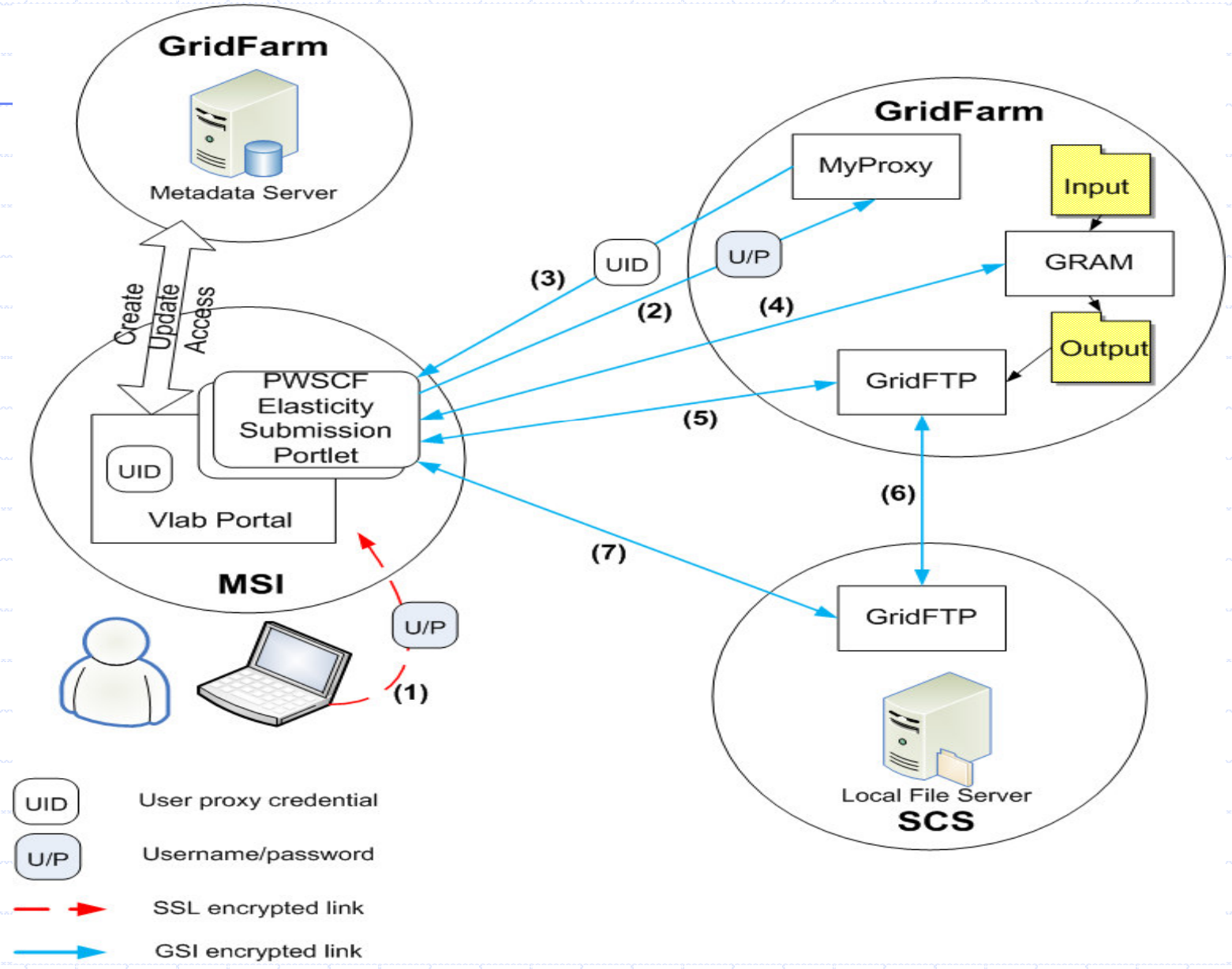
Definitions: Portal, portlet, portlet container

- **Portal** is a Web application provides
  - Personalization, single sign on and content aggregation
  - Web, Grid and educational portals
    - Web -> Yahoo, Google
    - Grid -> CIMA, VLAB and QuakeSim
    - Education -> OneStart and Oncourse
- **Portlets** are building blocks of portals
  - Generic portlets are calendar, chat, file transfer, weather and news.
  - Grid portlets are credential management, job submission and file management
- **Portlet container**
  - Provides lifecycle management of portlets
  - Defines window states and portlet modes
  - Equips packaging and deployment descriptors
  - JSR 168/286 specification defines
    - Portlet container
    - Portlet API

Science Portals and Gateways

- Grid portals are built on a common software/middleware stack .
  - TeraGrid Science Gateway
  - Open Science Grid (OSG)
- Many Java-based gateways are based on the portlet component model.
- What makes a portal Grid-enabled?
  - Grid account management
    - GAMA: Grid Account Management Architecture
    - PURSe: Portal-based User Registration System
  - Managing Grid credentials to access Grid services
    - MyProxy: remote authentication infrastructure
  - Accessing to Grid services (Grid portlets)
    - GRAM: resource access
    - GridFTP: file management
    - Condor: resource management
  - Information Services (GPIR, MDS, SRB)
  - OGSA-DAI: data service

Case Study: VLAB Portal



VLAB: The Virtual Laboratory for Earth and Planetary Materials

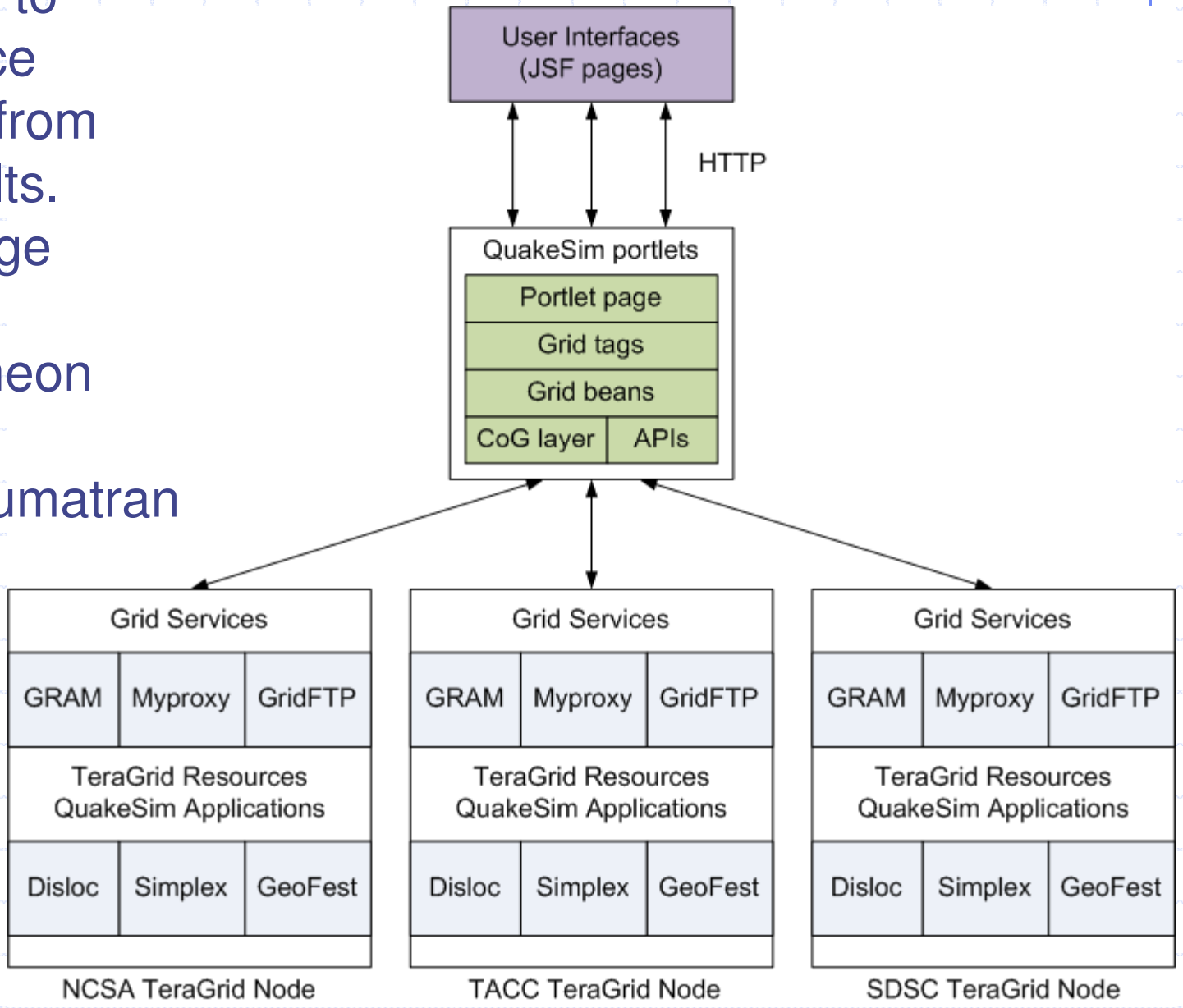
- Primarily a traditional job submission, monitoring, and management portal.
  - In first generation of OGCE, all capabilities are portlets
- Collaborative Grid services and portals support computational material science.
- Component based Grid portlet development makes application development easier.
- VLAB Challenges:
  - Grid Portlets must be easy to develop using component libraries.
  - HTML <form> actions in Grid portals typically have several steps:
    - Stage data files in and out of the desired remote host.
    - Run one or more executables.
    - Keep track of job progress
    - Store all of the information as "job archive" for reproducibility.

Case Study: Grid Portlets for QuakeSim

- **QuakeSim portlets** in production
  - Initial effort was to build the portal Web services
  - Portlets invoke the services and then execute simple workflows built in Ant scripts
- **Problem**: Portlets should also work with other TeraGrid resources.
  - TeraGrid restricts the services run on their resources.
- **Solution**:
  - We have prototyped Grid portlets to QuakeSim portal.
  - Described the process for creating Grid portlets using Grid Tag Libraries and Beans (GTLAB).
  - We gained rapid development by using reusable components
  - QuakeSim portlets enabled to utilize IU, SDSC, NCSA and other TeraGrid resources

QuakeSim portlets with TeraGrid allocation

- Disloc is used to calculate surface displacements from earthquake faults.
- 1994 Northridge earthquake
- 2003 San Simeon Earthquake
- 2004 Great Sumatran Earthquake.



Reuasability problem of portlets

- Portlets are entire web applications
  - JSP, Velocity and Struts
- Portlets are composed of two main parts View and Model
  - View is user interface (markup language)
  - Model is language codes (Java beans)
  - Efforts are focused on to provide modular Model
    - Java CoG Kit, Simple API for Grid Applications (SAGA)
  - There is no effort to markup Grid operations within Web frameworks
- Constructing portlet pages by using JSP introduces such problems for application programmers.
  - JSP pages mixes up markup and language codes
  - JSP approach reduces reusability of portlets
- We need to separate markups and language codes in order to use reusable components
  - Java Server Faces (JSF) is a good candidate

Simple portlet application (JSP)

```
<%
    PortletSession portletSession=renderRequest.getPortletSession();
    PortletURL url=renderResponse.createActionURL();
    String theActionString=url.toString();
%>
Fill out the form and then click submit to get your certificate.
<form method=post action="<%=theActionString%>">
    <input type=”text” name=”hostname” value=””>
    <input type=”text” name=”username” value=””>
    <input type=”password” name=”password” value=””>
    <input type=”submit” name=”getProxy” value=”Get Proxy Cert”>
</form>
```

OGCE
Open Grid Computing Environments

Logout
Welcome, Mehmet Nacar

Grid Information | Proxy Management | File Management | **Job Submission** | Condor | Sakai Portlets | SRB | IFrame | Welcome | Meme Job

Batch Job Submission Interactive Job Submission

? **GridPort Job Submission**

Job Submission

Host	grid-co.ncsa.teragrid.org
Port	2119
Executable	/bin/ls
Arguments	-l
Standard Input	
Standard Output	out
Standard Error	err
Directory	/manacar/jobs
CPU Count	
Wall Clock Time (min)	

Submit

Job Status

Refresh

Job Handle	Status

YOU CURRENTLY HAVE NO SUBMITTED JOBS.

Problems with Grid portlets

- Grid portlets typically wrap each single Grid capability in a separate portlet
  - Myproxy, File management, Job submit portlets
- Portlets are coarse-grained components
- We need a component model for portlets: reusable portlet parts
  - In our approach fine-grained components are Grid tags that map to reusable Grid operations
- Grid portlets need to pipeline fine-grained Grid operations
  - How do we deal with many Grid operations in one portlet?
- Problem with session management on job monitoring
  - How to keep repositories for submitted jobs?
  - How to access metadata to resubmit or to search jobs

Research Issues

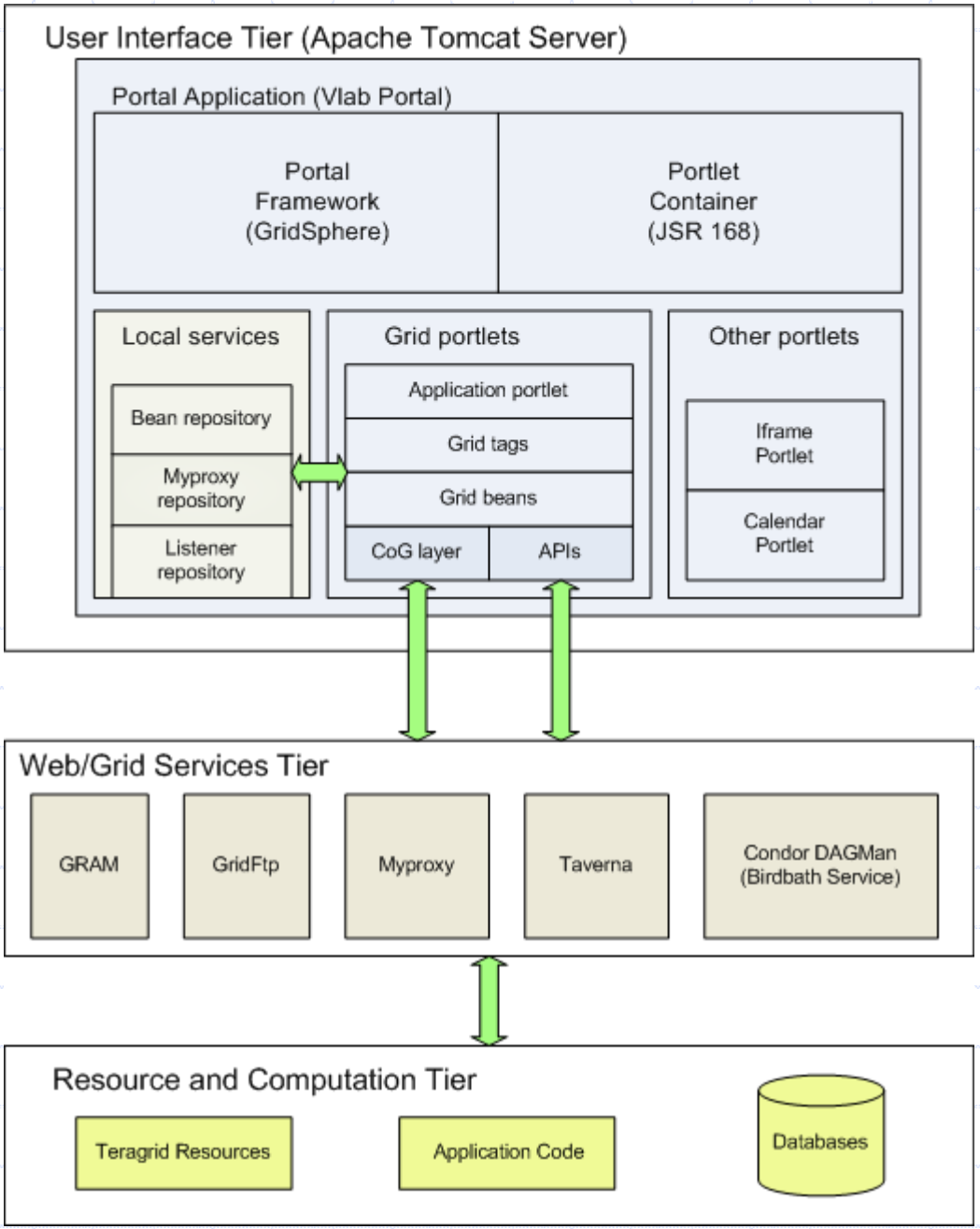
- Simplifying Grid portlet building for application developers
- Fine grained component model for Grid portlets
  - Issue with abstracting Grid operations as portlet applications
  - Portlet components are coarse-grained
- Metadata management
  - Issue with maintaining repositories for jobs
    - Providing persistency
  - Issue with archiving job information
  - Utilizing metadata for resubmitting or searching capabilities
- Pipelining and composing Grid operations
  - Issue is that a Grid portlet should be able to combine many operations
  - Issue with applying Directed Acyclic Graph (DAG) and workflows on Grid portlets
  - Abstracting Grid tags to support external DAG and workflow frameworks

Solution: Grid Tag Libraries and Beans (GTLAB)

- The goal of GTLAB is to simplify Grid portlet development
  - Enable rapid development
- GTLAB provides Grid components for building portlets using reusable tags.
- Grid beans provide methods to invoke Grid services
  - Simplify Grid service programming
  - Provides an abstract layer on top of low level APIs
- Grid tags attributes map to Grid beans methods
  - End users pass values to Grid beans by using tag attributes.
- GTLAB capabilities include Grid operations with XML based tags within Java Server Faces (JSF) framework.
- Grid operation codes provided by Grid beans
- GTLAB provide DAG capabilities
- Session manager deals with job handlers
  - Handlers and job metadata are stored on WS-Context server

Architecture Overview

- Grid portals are client to backend codes through Web/Grid services.
- Grid tags are part of user interface tier and embedded into portlet container.
- Grid tags uses local services in Apache Tomcat to manage sessions and handlers.
- Grid beans implement a layer on top of Grid client APIs such as Java CoG

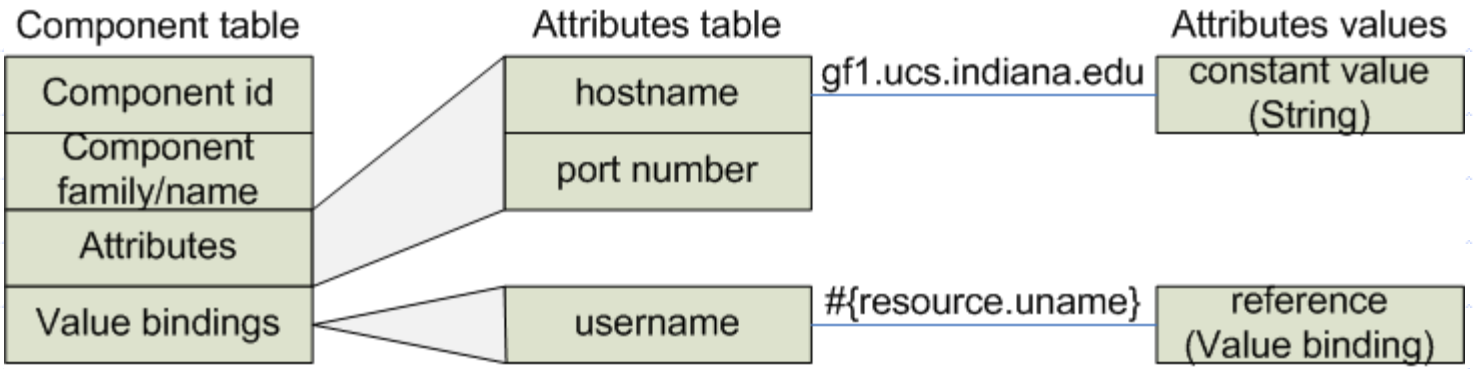


Java Server Faces (JSF)

- Why is JSF suitable for Grid Portals?
  - OGCE community used templates to develop portlets
    - Velocity and JSP (Web development templates)
  - JSF eliminates intervention by proposing JSF tags that separate backing bean and server pages
  - JSF also provides an extensible framework (tag libraries)
- How to extend JSF?
  - There are two types of JSF standard components
    - Visual: CommandButton, InputText etc.
    - Non-visual: form, param etc.
  - Two components required for development of the tags:
    - UIComponent: Describes encoding/decoding HTML.
    - ComponentTag: Describes attributes
  - Tag components should describe attribute bindings.
    - Value binding -> #{bean.property}
    - Method binding -> #{bean.method}

Grid tags

- GTLAB framework is based on JSF framework
- Grid operations are modeled by using JSF components
- JSF components are represented by XML markup language.
- Tags corresponds to Grid operations
- Attributes corresponds to methods and properties
  - Attribute values could be: Constant value or Reference



How to prepare application pages

- Grid tags are embedded into JSF view pages and decorated with standard JSF form, input, output and button components
  - Grid components are non-visual and are not displayed in HTML.
- Resource bean provides bridging with form inputs and GTLAB framework.
  - *<h:outputText value="Taskname: "/>*
    <h:inputText value="#{resource.taskname}" />
    <o:multitask id="multi" persistent="true" task name="#{resource.taskname}" />
- Dynamic values to Grid tag attributes are provided by Resource bean.
- A visual component is *<o:submit/>* tag that is associated with action method of GTLAB.

Grid Tag Libraries (introductory example)

- Grid tags simplify association of composite Grid actions
  - increase reuse of code
- There are associated custom JSF tag extensions we've developed:
  - <o:submit/>, <o:myproxy/> and <o:jobsubmit/>
- Two Grid operations with no dependency

```
<html>
  <body>
    <f:form>
      <o:submit id="test" action="next_page" />
              <o:myproxy id="pr" hostname="gf1.ucs.indiana.edu" port="7512" lifetime="2"
                         username="mnacar" password="***" />
              <o:jobsubmit id="task" hostname="cobalt.ncsa.teragrid.org" provider="GT4"
                         executable="/bin/ls" stdout="tmp/result stderr="tmp/error" />
      </o:submit>
    </f:form>
  </body>
</html>
```

Grid Tags	Associated Grid Beans	Features
<submit/>	ComponentBuilderBean	**Creating components, job handlers, submitting jobs**
<handler/>	MonitorBean	**Handling monitoring page actions**
<multitask/>	MultitaskBean	**Constructing simple workflow**
<dependency/>	MultitaskBean	**Defining dependencies among sub jobs**
<myproxy/>	MyproxyBean	**Retrieving myproxy credential**
<fileoperation/>	FileOprationBean	**Providing Gridftp operations**
<jobsubmission/>	JobSubmitBean	**Providing GRAM job submissions**
<filetransfer/>	FileTransferBean	**Providing Gridftp file transfer**
Other JSF tags	ResourceBean	**Describes common properties among all tags and beans. Passing values given by standard visual JSF components.**

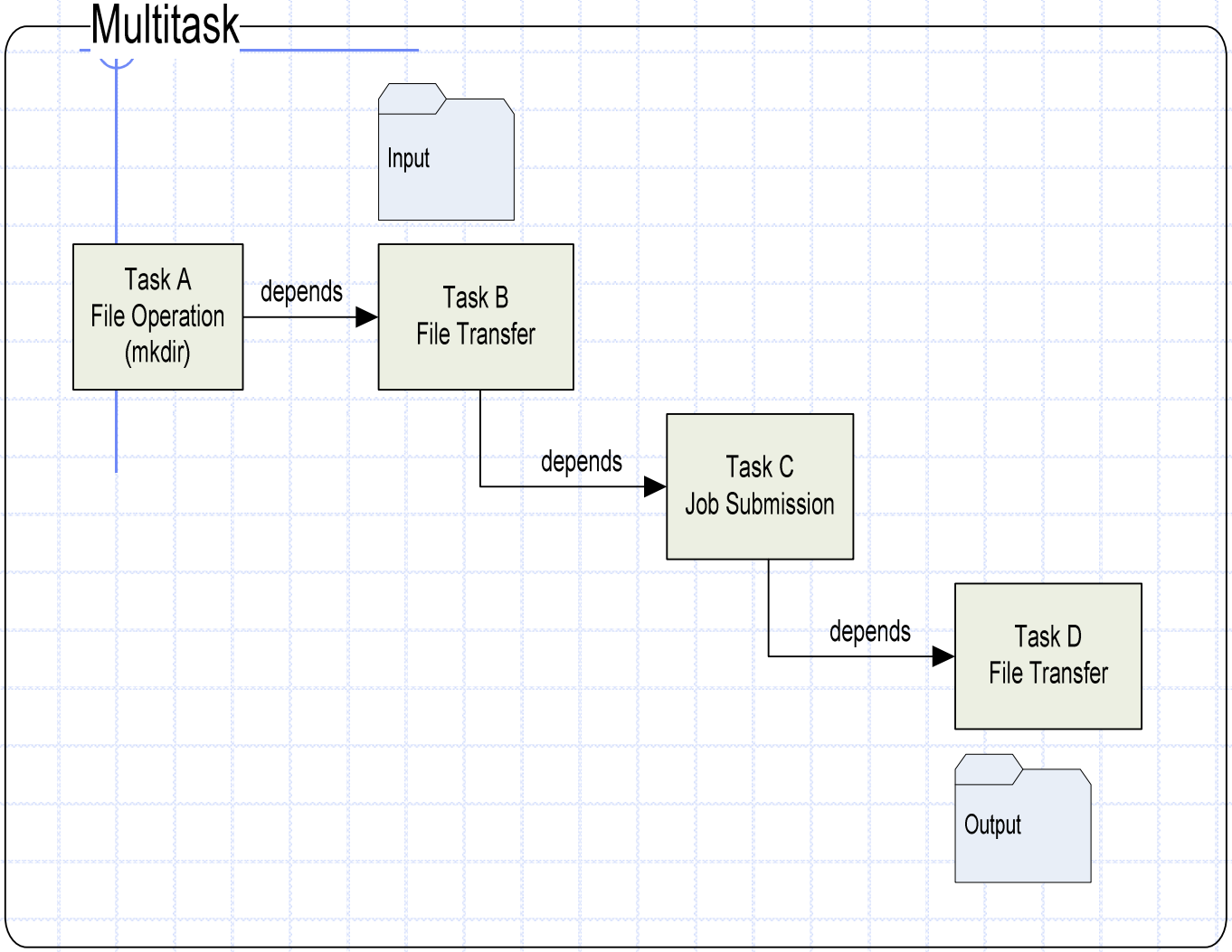
Managing Metadata Repositories

- Job metadata are stored in metadata repositories
  - Metadata includes: submission parameters and files, execution host and information, output parameters, files and their location
- Job data is stored on the specified servers. As a result data files are transferred by using GridFtp
- Metadata of the data files such as URL location or GridFtp locations are saved in the metadata repository
- Portal users see the job metadata that has links to the exact location of the data
- Metadata repository is built by using WS-Context repository
  - Users see job metadata with the time stamped hierarchical repository labels
  - /vlab/userA/sessionC/uuid/jobD

Handling of multiple operations

- We need to handle multiple jobs in logical order
  - Workflows are general approach to control execution flow
  - Grid pipelining problem specifically fit into DAG category
- We provide generic DAG component for Grid services
  - It's called "multitask"
- GTLAB handles lifecycle of DAG within JSF application
  - Start, stop, resume, cancel the operations
- <o:multitask/>, <o:dependency/> tags are representing DAG operations
- GTLAB can manages multiple DAGs in portlet application

Encoding DAGs to portlets



- This example demonstrates a composite Grid job using *multitask*

DAG Example JSF Page

```
<o:submit id="test" action="next_page" />
    <o:multitask id="mytask" taskname="test" persistent="true" >
      <o:myproxy id="pr" hostname="gf1.ucs.indiana.edu" port="7512" lifetime="2"
        username="manacar" password="***" />
      <o:fileoperation id="taskA" command="mkdir" hostname="cobalt.ncsa.teragrid.org"
        path="/home/manacar/tmp/" />
      <o:filetransfer id="taskB"
        from="gridftp://gf1.ucs.indiana.edu:2811/home/manacar/input_file"
        to="gridftp://cobalt.ncsa.teragrid.org:2811/home/manacar/tmp/input_file" />
      <o:jobsubmit id="taskC" hostname="cobalt.ncsa.teragrid.org" provider="GT4"
        executable="/bin/execute"
        stdin="tmp/input_file" stdout="tmp/result" stderr="tmp/error" />
      <o:filetransfer id="taskD"
        from="gridftp://cobalt.ncsa.teragrid.org:2811/home/manacar/tmp/result"
        to=" gridftp://gf1.ucs.indiana.edu:2811/home/manacar/result" />
      <o:dependency id="dep1" task="taskB" dependsOn="taskA" />
      <o:dependency id="dep2" task="taskC" dependsOn="taskB" />
      <o:dependency id="dep3" task="taskD" dependsOn="taskC" />
    </o:multitask>
</o:submit>
```

Note the specific values would typically come from the user's form inputs through bean methods.

DAG extensions: Condor DAGMan, Birdbath

- We want to show that GTLAB architecture is extensible
- Besides Globus Grid services support, Grid community also use other common Grid services such as Condor
- We extend GTLAB to support Condor capabilities
- Condor DAGMan is a tool for complex application workflows on Condor
- Birdbath is Web services provider of Condor capabilities
- Grid tags integrate DAGMan with the following tags:
  - <o:condorDagman/> and <o:condorSubmit/>
- Composing DAGMan workflow is out of scope.
- GTLAB executes and monitors DAGs by Condor
- GTLAB tags for DAGMan use Birdbath service to create client stubs.

Taverna workflows

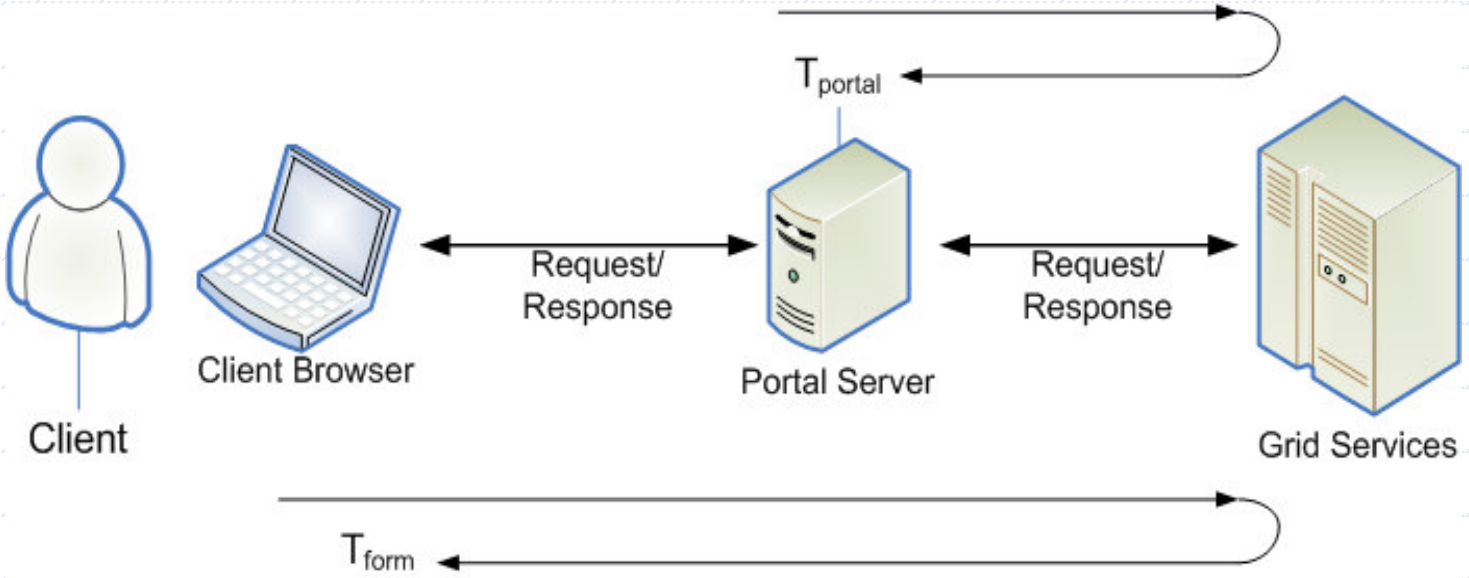
- DAGs provides numerous advantages to pipe Grid invocations
- But DAGs cannot support full-fledged workflow capabilities like conditional branches and loops.
  - We studied Taverna as use case.
  - We investigate Kepler and BPEL as future extensions
- Workflows can be handled in different categories:
  - Composing
  - Enacting
  - Monitoring
- GTLAB supports Taverna enactment and monitoring.
- GTLAB imports well studied built-in workflows collected by the community
  - Bioinformatics workflows and their metadata is available
- Workflow composition is out of scope of this dissertation
  - There are ongoing researches in this area

Advantages of GTLAB

- GTLAB provides simplicity to develop Science portals
  - Rapid development
  - Easy deployment
- Grid tags provide rich selection of attributes to initialize Grid beans.
- Composite tasks can contain an unlimited number of subtasks
- GTLAB gives flexibility to developers to use their own Grid beans library or add more Grid beans to the existing ones.
  - Following the method name convention of GTLAB
- Grid bean methods are bound to tags with attributes to simplify the building of new Grid portlets

Testing scenario

- We want to show baseline performance metrics
- We also measure GTLAB overhead in order to show it is tolerable
- There are two testing cases:
  - T_{form} : Request/response time between Client and Grid services via portal server
  - T_{portal} : Request/response time between portal server and Grid services



Testing setup

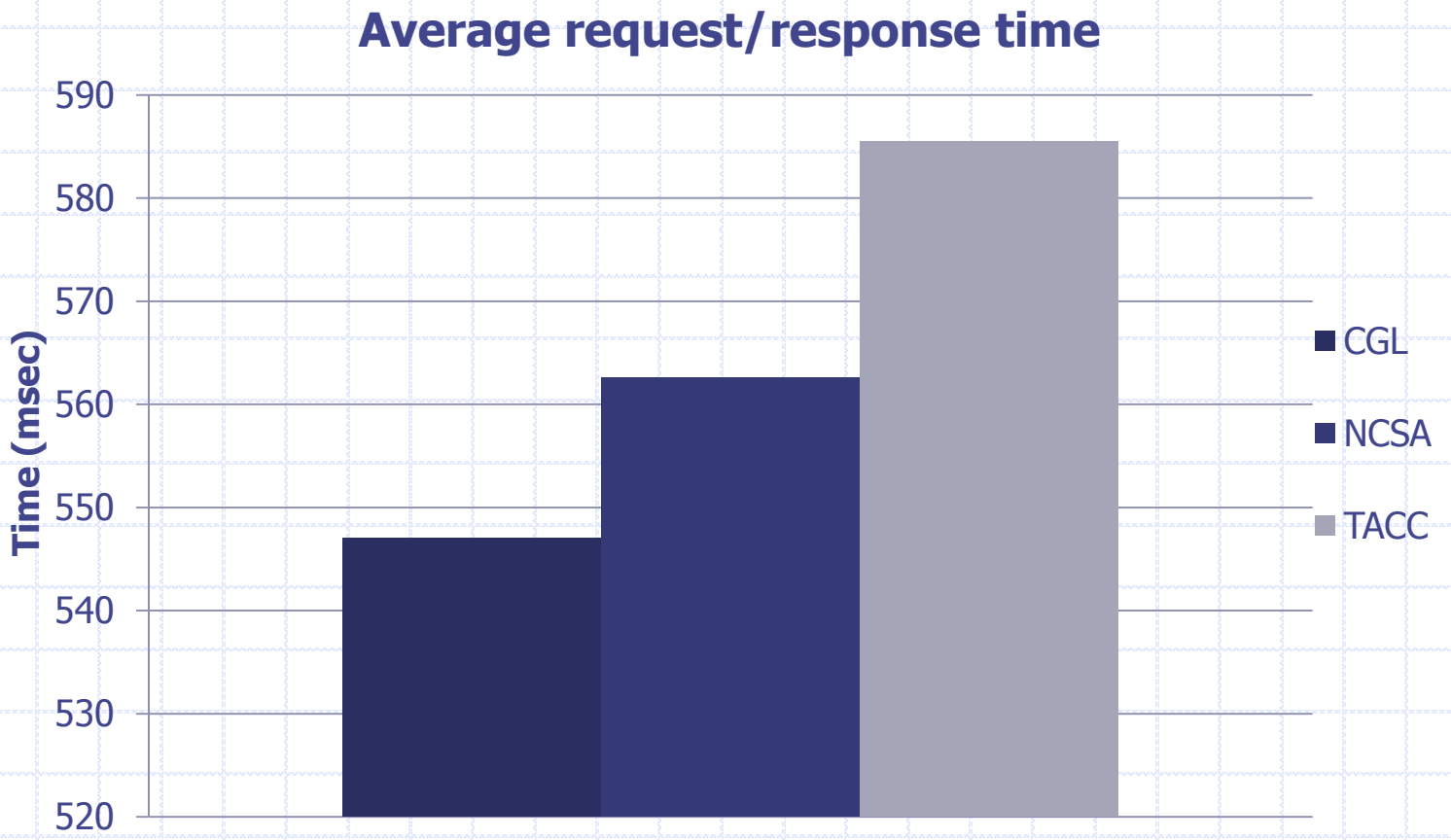
- The tests include multitasks with Grid operations. A simple multitask is shown below.

```
<o:multitask id="multi" persistent="true" taskname="#{resource.taskname}">
  <o:myproxy id="mypr" hostname="gf1" lifetime="2" password="manacar" port="7512"
        username="manacar"/>
  <o:jobsubmit id="js" executable="/bin/ls" hostname="gf1.ucs.indiana.edu"
        provider="#{resource.provider}" stdout="/home/manacar/tmp/out-test2"/>
  <o:jobsubmit id="js2" executable="/bin/ps" hostname="gf1.ucs.indiana.edu"
        provider="#{resource.provider}" stdout="/home/manacar/tmp/out-test3"/>
  <o:dependency id="dep" dependsOn="js" task="js2"/>
</o:multitask>
```

- We measure average request/response times (T_{form} , T_{portal})
- We have simulated 10 users from different locations and browser sessions.
  - Each user makes 100 requests
- Portal server runs on Grid Farm clusters at CGL
- End users invokes requests from CGL (local), NCSA and TACC (remote locations)

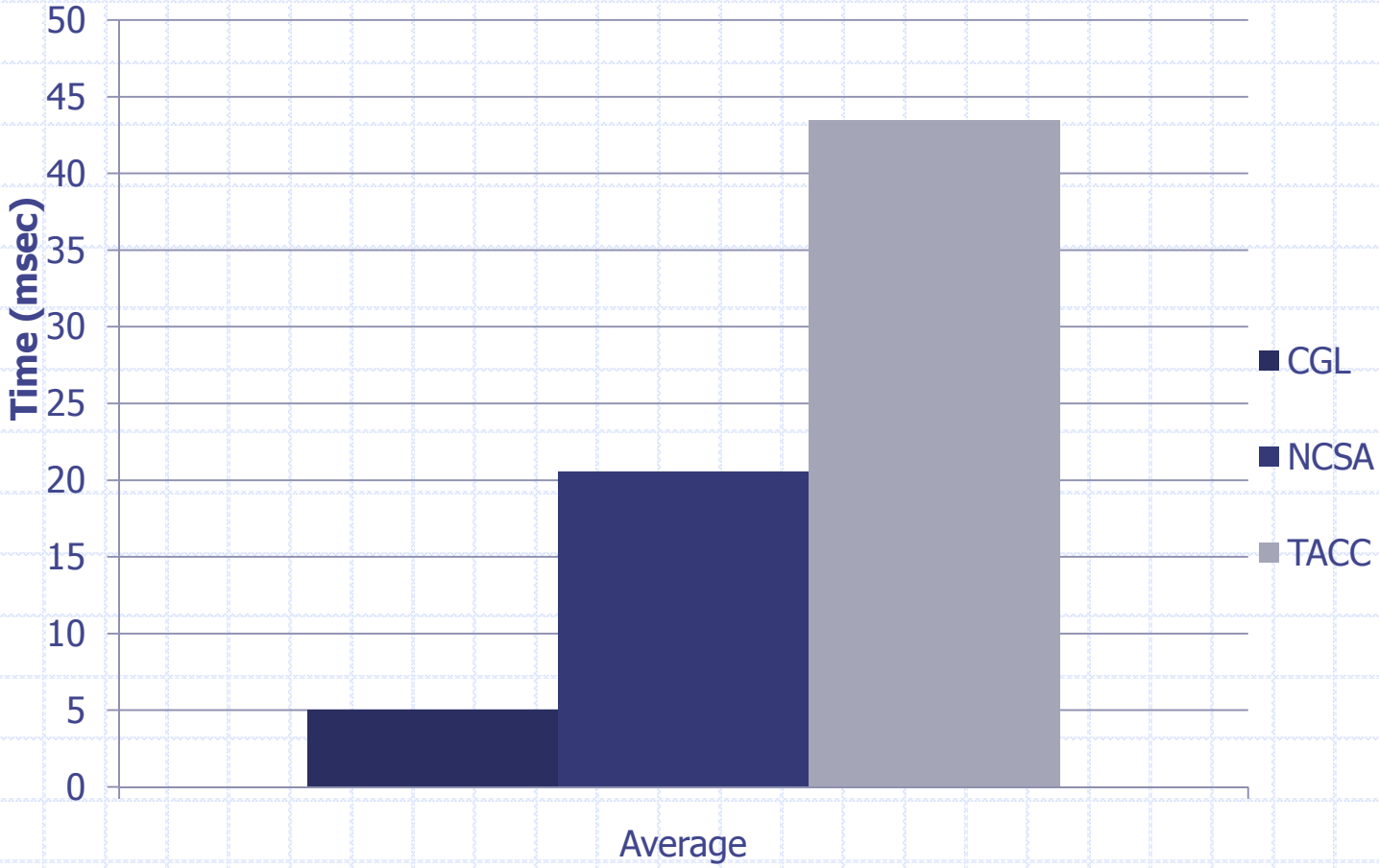
Benchmarks I

- Average request/response time (T_{form})

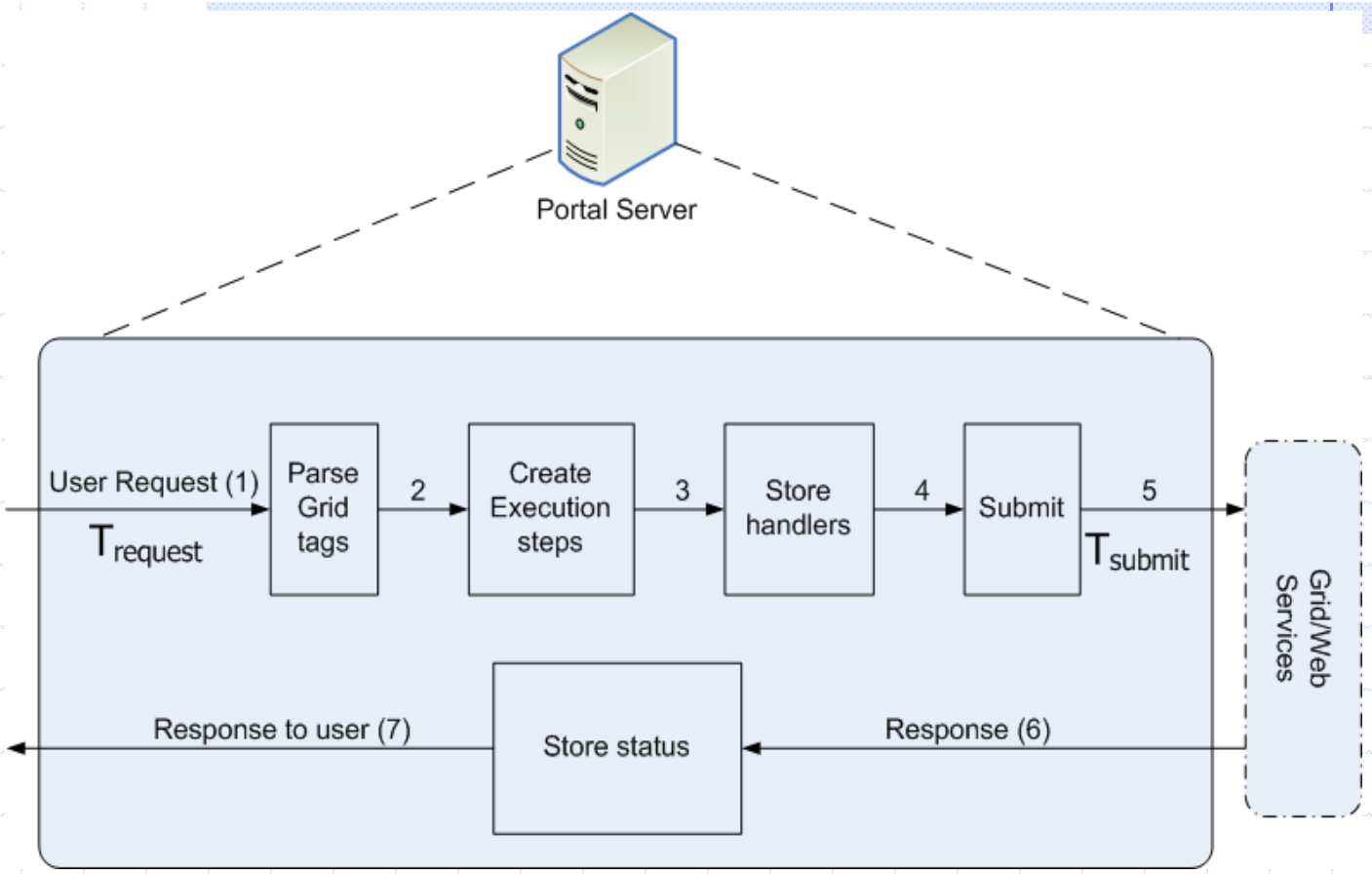


Benchmarks II

- Average network latency between end users and portal server T_{form}-T_{portal}
  - Users from IU, NCSA and TACC



GTLAB Overhead



	GTLAB Processing	JSF Processing	Handler storing	Submitting
Time (msec)	2	153	1	410

- Total overhead = T_{submit}-$T_{request}$ = 156 msec (that includes JSF overhead)

Evaluation of tests

- Benchmark results summarize performance metrics about GTLAB framework
- Average overhead of GTLAB is about few milliseconds
- GTLAB does not add up significant delay on processing the requests.
- The results also indicate that geographical location of the end users causes minor delays on response time
  - Delays vary about 10-50 msec depending on distance
- Application developers save cost of development time when building or reusing GTLAB, but they do not loose performance.

Contributions: System Research

- Simplifying invocation of Grid applications through reusable libraries.
  - Tag libraries for portlets are “fine-grained” components.
- Providing two types of components
  - Grid tags (UI)
  - Grid beans (API)
- Abstracting DAG capabilities to manage multiple Grid operations
- Extending workflow capabilities to support external workflow engines
- Persistently storing job metadata
- Metadata services enable accesses to archives.
- Defining an access control policy of portlet contents

Contributions: System Software

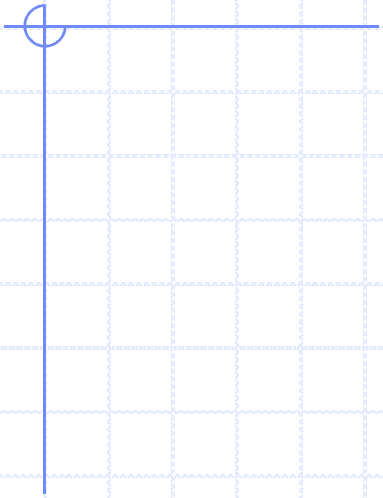
- Design, development and application of GTLAB architecture
  - Extends the current field
    - Limited by coarse-grained portlet components
  - Our approach provides a natural way to implement Grid portlets
    - Case studies are VLab and QuakeSim portals
- VLab portal
- QuakeSim portlets
- CIMA portal
- OGCE portlets
- GridShell project

Related Work

- GridSphere's Grid Portlets 1.3
  - Grid Portlets 1.3 provide API and User Interface (UI) tags to build Grid portlets
  - An effort called *Vine (Portlet Vine)* refactors Grid portlets and decouples the portlets from GridSphere
- Karajan
  - XML based workflow language and engine for Grid computing
  - Requires additional effort to aggregate it into Grid portals
- Simple API for Grid Applications (SAGA)
  - Low level programming interface (API)
  - It provides tools to build Model in Web programming (Beans)
  - It does not provide any User Interface widgets to embed it into Web applications

Software

- GTLAB v1.0 release available at
  - http://grids.ucs.indiana.edu/users/manacar/GTLAB-website
- See link from main OGCE web site
  - http://www.collab-ogce.org
- Vlab portal
  - http://pedro.msi.umn.edu:6080/gridsphere
- CIMA portal
  - http://156.56.94.164:8080/gridsphere
- QuakeSim portal
  - http://gf7.ucs.indiana.edu:8080/gridsphere/gridsphere



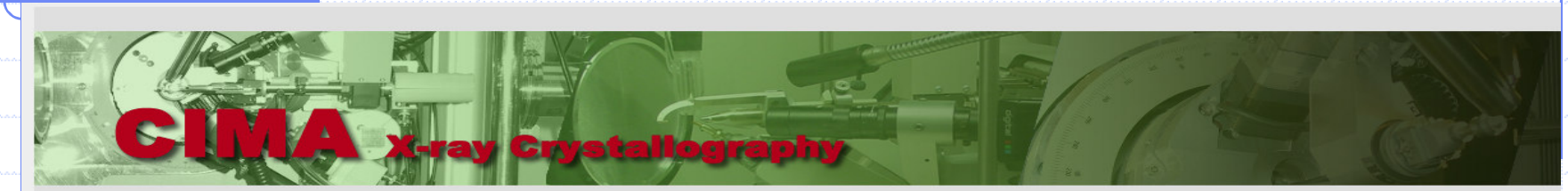
Thanks !

Selected Publications

1. **Nacar, Mehmet.A**., M. Pierce, and G. Fox. *GTLAB:Grid Tag Libraries Supporting Workflows within Science Gateways*. in *SKG 2007*. 2007. Xian, China: IEEE Proceedings.
2. **Mehmet A. Nacar**, M.S.A., Marlon Pierce, Zhenyu Lu and Gordon Erlebacher, Dan Kigelman, Evan F. Bollig, Cesar De Silva, Benny Sowell, and David A. Yuen, *VLab: Collaborative Grid Services and Portals to Support Computational Material Science* Concurrency and Computation: Practice and Experience, 2007. **19**(12): p. 1717-1728.
3. **Mehmet Nacar**, MarlonPierce, Gordon Erlebacher, Geoffrey Fox, *Designing Grid Tag Libraries and Grid Beans*, in *Second International Workshop on Grid Computing Environments GCE06 at SC06*. 2006: Tampa, FL.
4. Jay Alameda, Marcus Christie Geoffrey Fox Joe Futrelle Dennis Gannon Mihael Hategan Gopi Kandaswamy Gregor von Laszewski **Mehmet A. Nacar** Marlon Pierce Eric Roberts Charles Severance Mary Thomas. *The Open Grid Computing Environments collaboration: portlets and services for science gateways.* Concurrency and Computation: Practice and Experience, 2007. **19**(6): p. 22.
5. Bollig, Evan F. Jensen, Paul A. Lyness, Martin D. **Nacar, Mehmet A**. da Silveira, Pedro R. C. Kigelman, Dan Erlebacher, Gordon Pierce, Marlon Yuen, David A. da Silva, Cesar R. S. *VLAB: Web Services, Portlets, and Workflows for Enabling Cyber-infrastructure in Computational Mineral Physics.* Physics of The Earth and Planetary Interiors. 2007. **163**(1-4): p. 333-346.
6. Hao Yin, Donald F.Mcmullen, **Mehmet A. Nacar**, Marlon Pierce, Kianosh Huffman, Geoffrey Fox and Yu Ma, *Providing Portlet-Based Client Access to CIMA-Enabled Crystallographic Instruments, Sensors, and Data*, in *7th IEEE/ACM International Conference on Grid Computing (GRID 2006)*. 2006: Barcelona, Spain.
7. **Mehmet A. Nacar**, J.Y.C., Marlon E. Pierce, and Geoffrey C. Fox. *Building a Grid Portal for Teragrid's Big Red*. in *TeraGrid 2007*. 2007. Madison, WI.
8. **Nacar, Mehmet A**., M. Pierce, and G.C. Fox, *Developing a secure grid computing environment shell engine: containers and services.* Neural, Parallel & Scientific Computations, 2004. **12**(3): p. 379-390.

Backup Slides

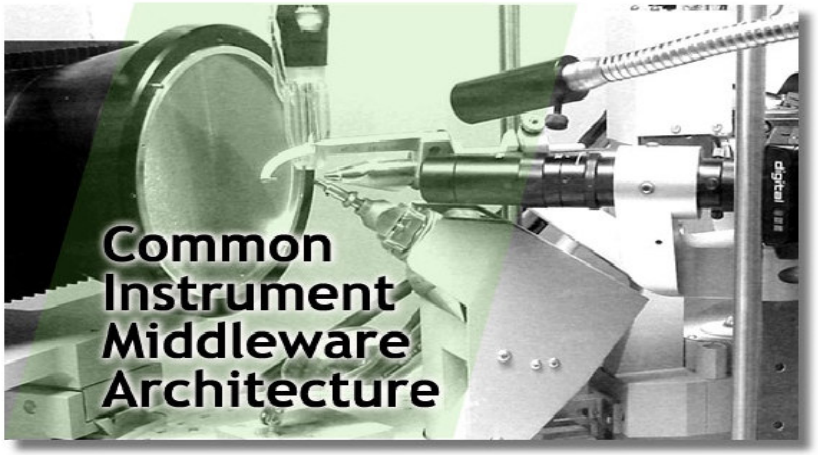
CIMA Crystallography portal



Home | IUMSC | ChemMatCARS | Purdue | Minnesota | Myers Hall | James Cook Univ. | CSAF Sydney | Request an Account

Home | News | About | Contact

- IUMSC Indiana University
- ChemMat-CARS Univ. of Chicago at APS
- Chem. Cryst. Center Purdue University
- Biology Dept. Indiana University
- X-ray Cryst. Lab Univ. of Minnesota
- AAC James Cook Univ., AU
- CSAF University of Sydney, AU
- Other Collaborating Laboratories
- NCS Southampton, UK



Login

User Name

Password

Remember my login

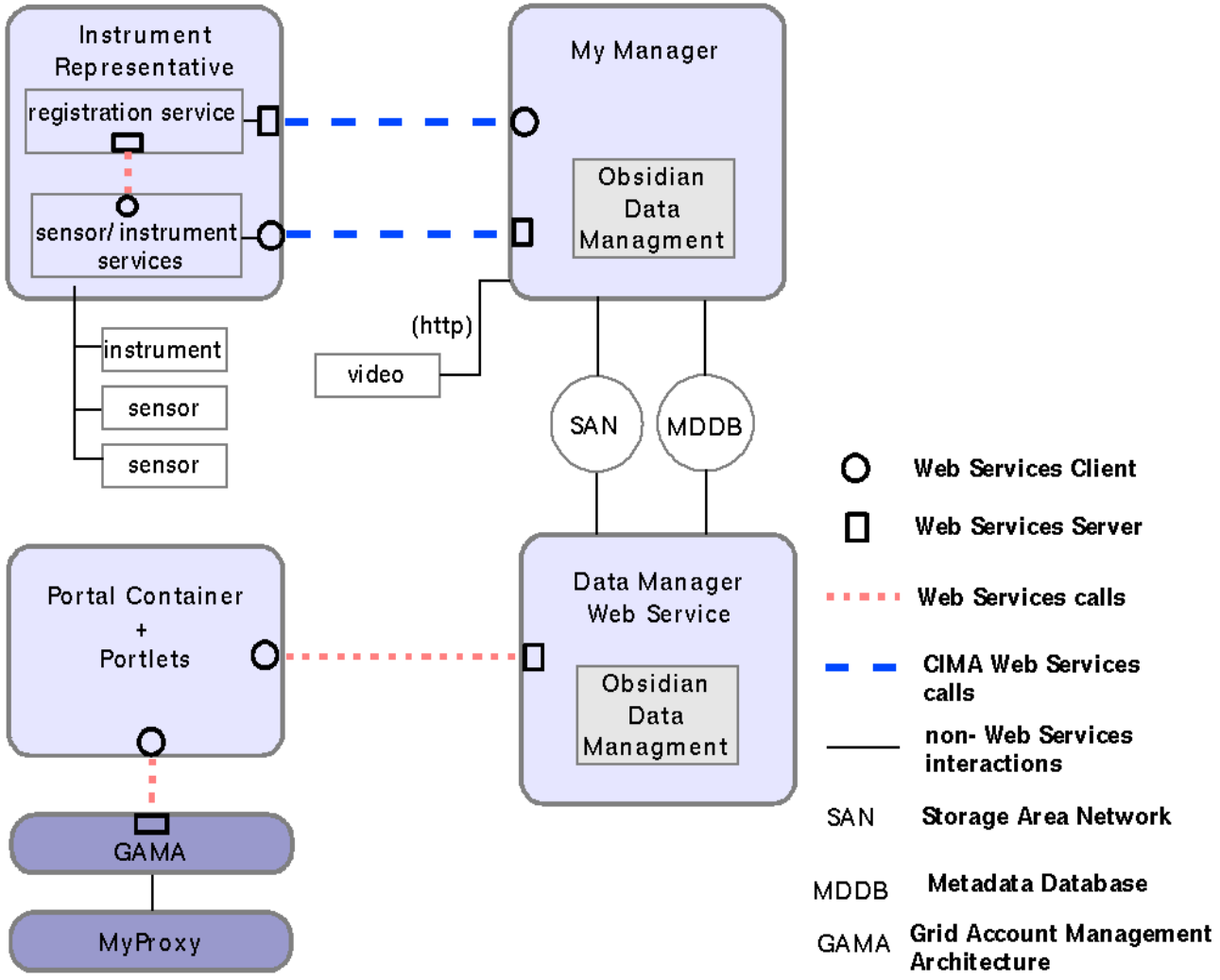
Login

Forget your password?

This portal demonstrates some of the functionality that will be available to the crystallographers collecting data in participating laboratories (left hand side menu). Researcher can follow the progress of an experiment using a simple web browser connected to the Internet. The IUMSC instrument is being used as a test system for the Instrument Middleware project, see
http://www.instrumentmiddleware.org

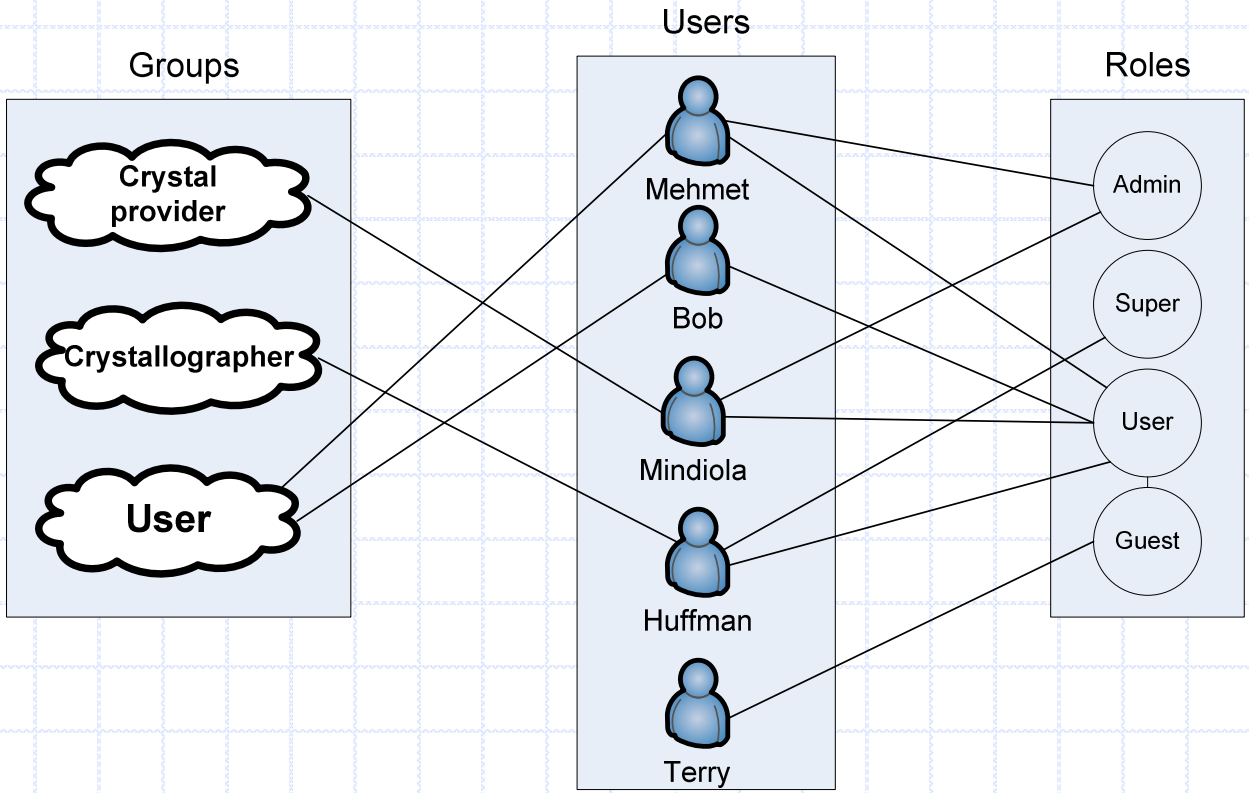
The work being described on these pages are supported by NSF Cooperative Agreement OCI-0330568 and MRI CDA-0116050

CIMA (Common Instrument Middleware Architecture)



- Primarily a data portal to online instruments
- Crystallographers collect data in participating laboratories and collaborate on samples.
- Portlets have to access data with group privileges.

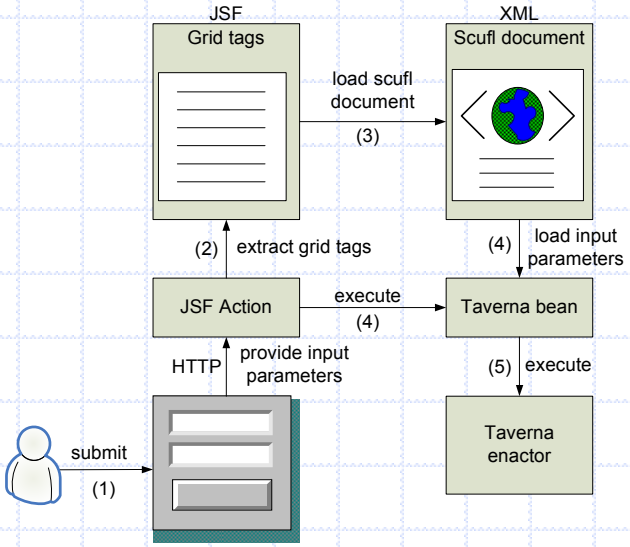
Research issue: Access control of portlets



Continue

- Portal users have roles and groups
  - RBAC provides this for portlets in portal framework
- We need to have access control over portlet data
- We need to map identities between portal and data manager service
- We have used an additional database to map portal users and service users
- CIMA sample portlets are customized to work with user groups
- This customization is done by adding new database tables to the existing portal framework

Taverna use case



- A user interacts with a workflow portlet to utilize Taverna enactor.
- User provides parameters by submitting a web form that start the chain of events in order.

Big Red Portal

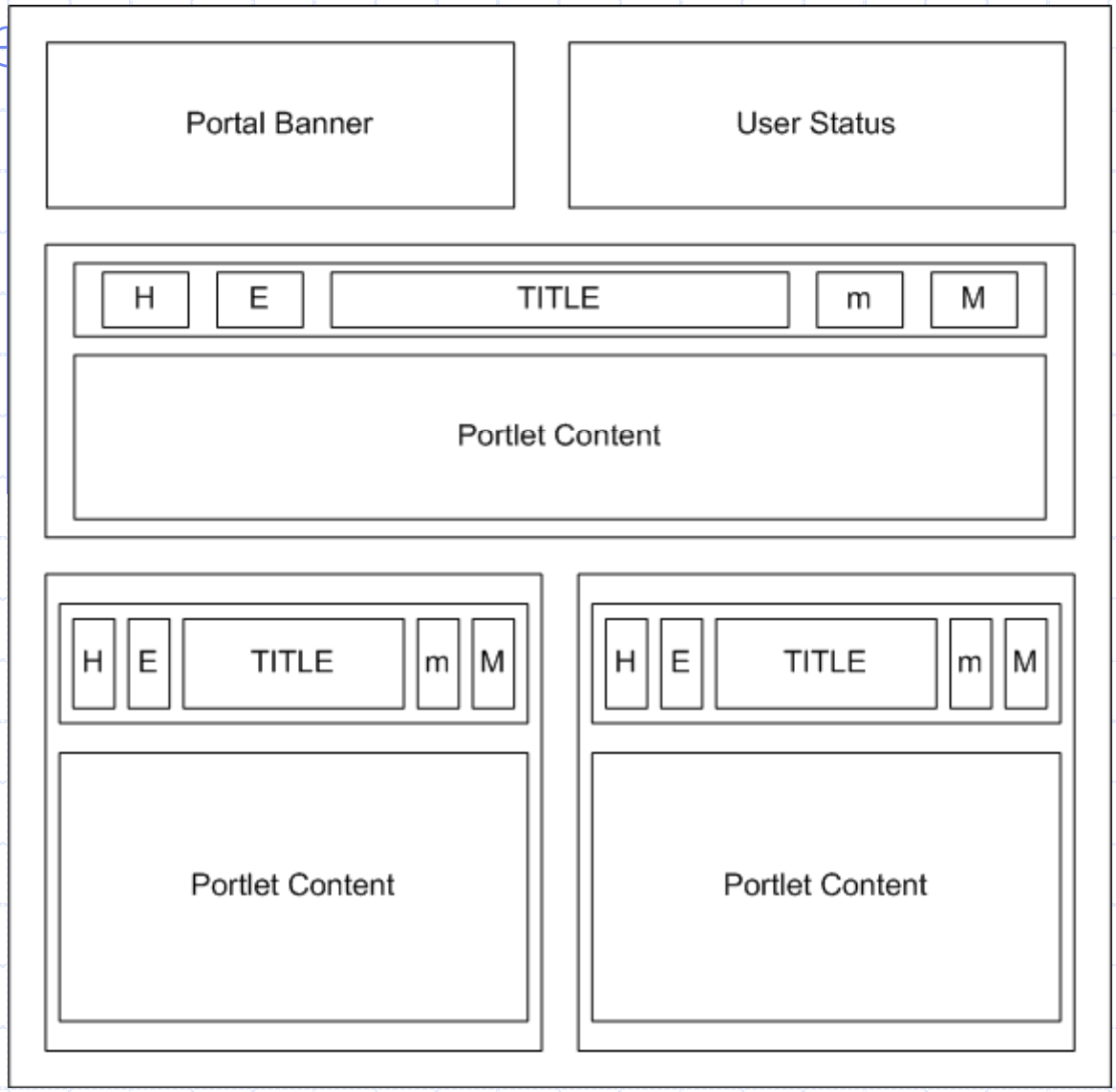
- Big Red supercomputer is part of TeraGrid at Indiana University
  - 2048 cluster nodes, 4 terabyte memory
- We have developed several portlets for Big Red portal submission
  - MEME job submission
    - Interactive and batch
    - Includes both GridFTP and GRAM clients
    - Job tracking.
  - MOAB queue monitoring for both entire machine and the specific user.
- Although utilizing Big Red, the portlets can be used with any other gateway in TeraGrid
  - We tested across IU and NCSA resources.

Research issues: Grid components

- Grid portlet applications require dynamic interaction and fine-grained components.
  - Portlets themselves need to be built out of components
  - Grid services mostly use request/response paradigm
  - Grid portlets use web forms heavily
    - Compared to wikis, blogs, RSS-driven news portals, which have a different problem of content management.
  - Grid widgets can provide components for:
    - Proxy credential
    - GridFtp operations
    - Job submissions, multi-staged jobs
  - Using widgets as tag libraries help to encapsulate reusable Grid components


Web Images Video News Maps Gmail more ▼
manacar@gmail.com | Classic Home | My Account | Sign out
iGoogle
Google Search
I'm Feeling Lucky
Advanced Search
Preferences
Language Tools
Welcome to iGoogle, your home on the web. Start by selecting some of our most popular content.
Save
Cancel
Popular
Gmail
Google Calendar
CNN.com
Weather
How to of the Day
News - Top Stories
Date & Time
Tools
Sticky Note
World Clocks
Google Docs & Spreadsheets
Google Notebook
Dictionary
To-Do List
Hotmail
News
NYT > Home Page
BBC News | News Front Page | World Ed...
FOXNews.com
Reuters: Top News
NPR Topics: News
Google News
Full print edition
Fun & Games
Bejeweled
Mario is Back!
PacMan v2.4
Fish
Idiot Test
Bowling
Mini Golf
Google Calendar
August 2007
Quick Add
Create Event
Show Agenda
Weather
Bloomington, IN
87°F
Clear
Wind: N at 0 mph
Humidity: 30%
Today
Tue
Wed
Date & Time
Mon
AUG 27
CNN.com
Vick pleads guilty
Gonzales announces his resignation
Robot may offer peek inside Utah mine
How to of the Day
How to Break a World of Warcraft Addiction
How to Make Fried Kasseri Cheese with Cucumber Sauce
Top Stories
Go-Go Gonzales
Swampland - all 1938 related »
Afghan Opium Production Sets Record Highs, says UN Report
Voice of America - all 259 related »
Sarkozy Cautions Against Attack on Iran
Forbes - all 187 related »
Smoking linked to Deutsche Bank blaze
United Press International - all 123 related »
Bush praises Iraqi deal, but says more to be done
Reuters - all 353 related »

Portal page aggregation



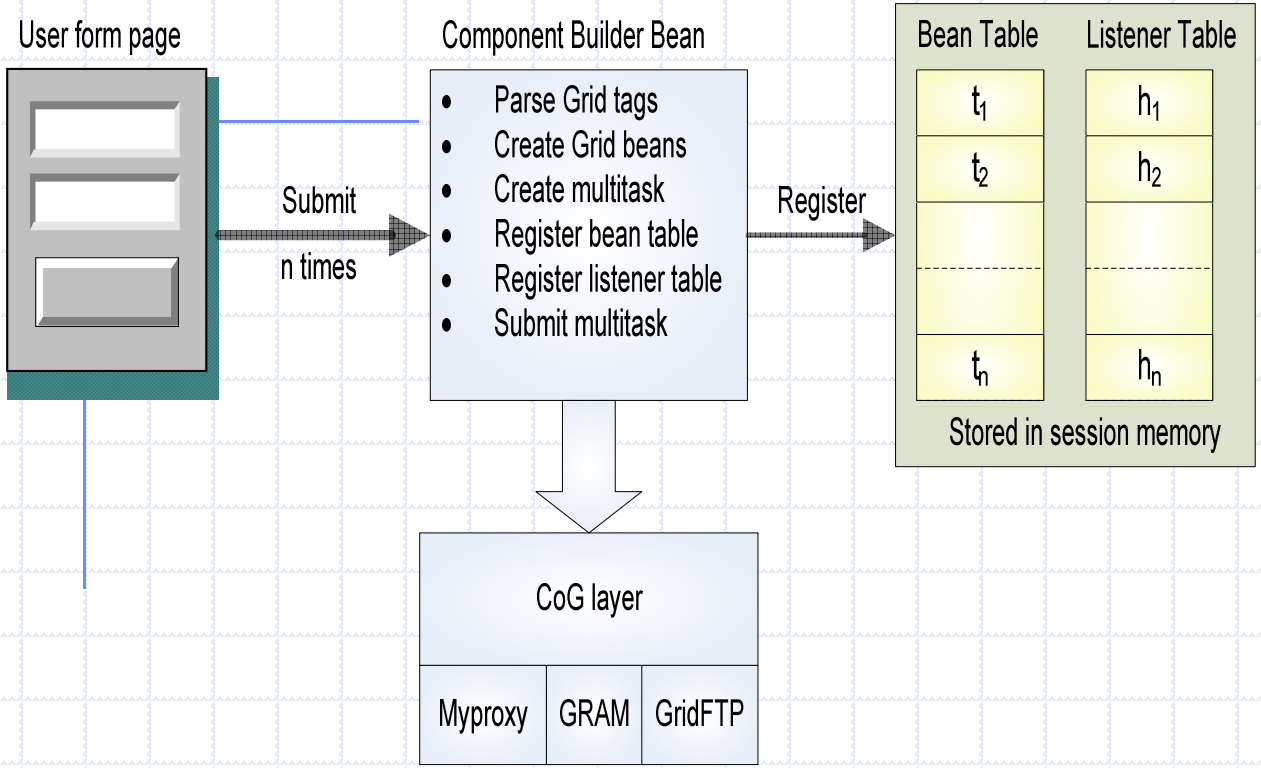
Portlet modes

- H--> Help
- E--> Edit

Window states

- M--> Maximize
- m-->Minimize

Architecture of GTLAB



- Component Builder Bean (CBB) handles user requests using JSF form pages
- CBB parses custom components embedded into JSF view page.

- Constructs the workflow by using Multitask bean with dependencies
- Maintains task handlers and task objects with persistency
- Submit multitasks to Grid services