
Grid Portals – A User's Gateway to the Grid

Dennis Gannon, Indiana University

Geoffrey Fox, Indiana University

Beth Plale, Indiana University

Marlon Pierce, Indiana University

Mary Thomas, Univ. of Texas

Charles Severance, University of Michigan

Gregor von Lazewski, Argonne National Labs

Jay Alameda, NCSA

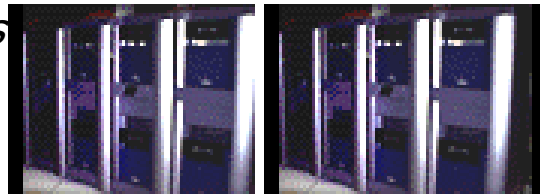
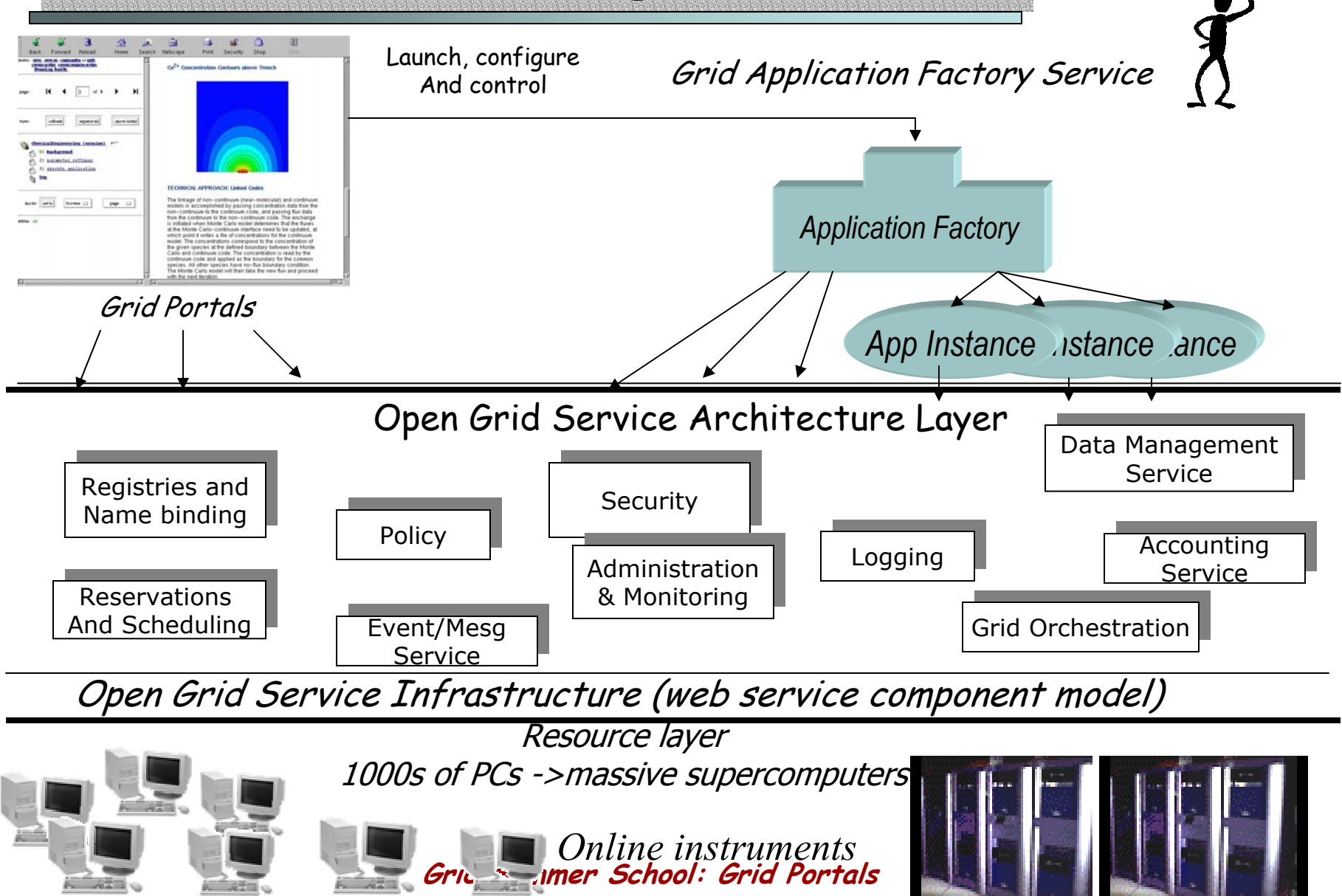
Other Important Contributors

- People who do the "Real" work.
 - **Argonne:** Micheal Hategan, Kaizar Amin, Shashank Shankar
 - **Indiana Grids Lab:** Shrideep Pallickara, Ali Kaplan, Ahmet Topcu, Ahmet Sayar, Fatih Mustacoglu
 - **Indiana CS:** Octav Chipara, Marc Christie, Liang Fang, Matthew Farrellee, Gopi Kandaswamy, Deepti Kodeboyina, Wei Lu, Sriram Krishnan, Yogesh Simmhan, Alek Slominski, Anuraag Sarangi
 - **University of Texas:** Maytal Dahan, Tomislav Urban, Eric Roberts, Akhil Seth
 - **University of Michigan:** Joseph Hardin, Glenn Golden, Jim Eng
 - **NCSA:** Shawn Hampton, Al Rossi, Greg Daues

Outline

- Hour 1 - An overview of Grid Portals
 - The Big Picture
 - Example Applications
- Hour 2 - The standard Architecture
 - Portal servers and portlets
- Hour 3 - Working with Standard Grid Tools
 - MyProxy Server and Gram Job Submission
 - Web and Grid Services
- Hour 4 - Advanced topics

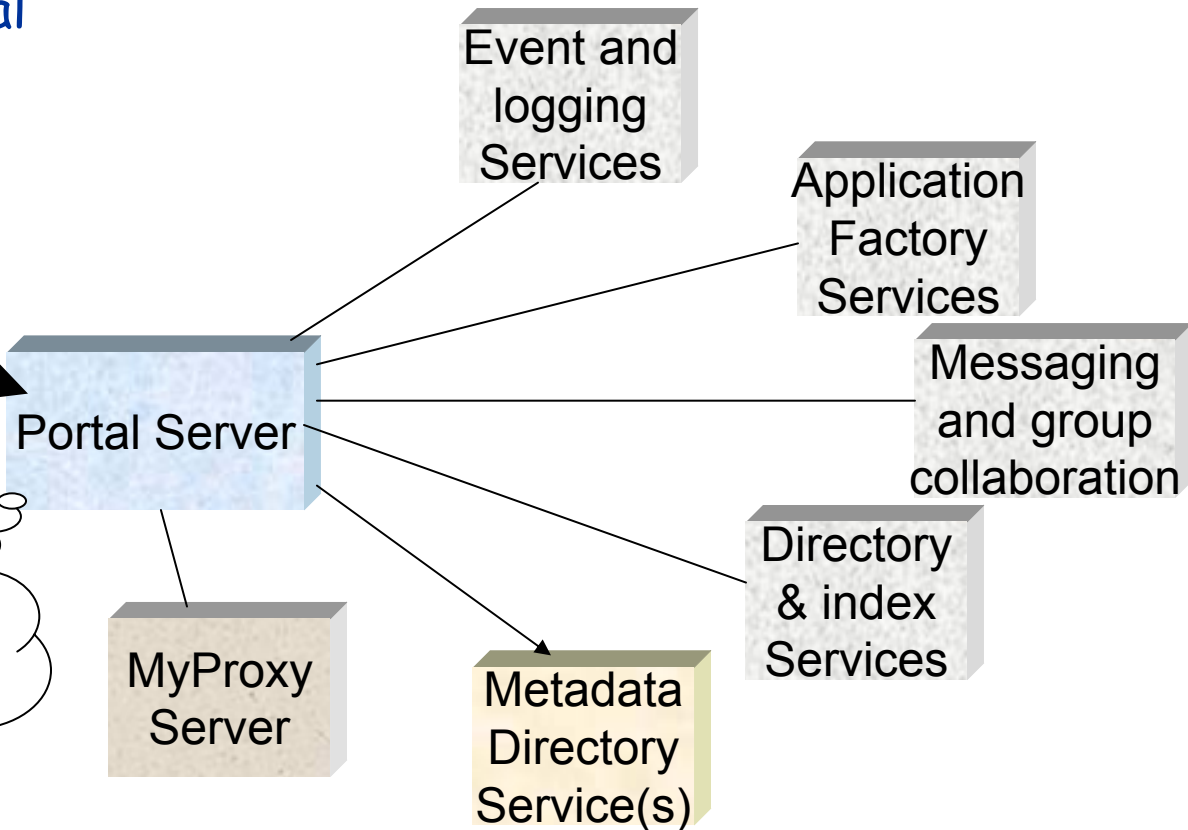
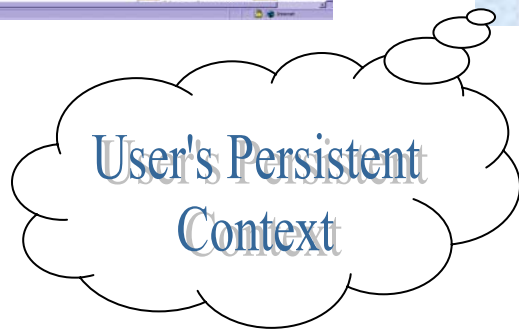
The Big Picture



The Big Picture

- The Grid is defined by a collection of distributed Services
 - For most users the primary interaction with the Grid will be through a portal

The User



What do users want from a portal

- Access to grid services
 - Security services
 - Management of my grid certificates
 - VO management (who is in my group?)
 - Remote File management
 - Access to file and metadata directories
 - Staging files and moving them
 - Remote Job management
 - Simple remote job submission and monitoring
 - Complex workflow management
- Access to information services
 - Directories, index tools, messaging, notification
- Application interfaces
 - Hiding the details of Grid middleware
- Access to collaboration
 - Shared video and applications

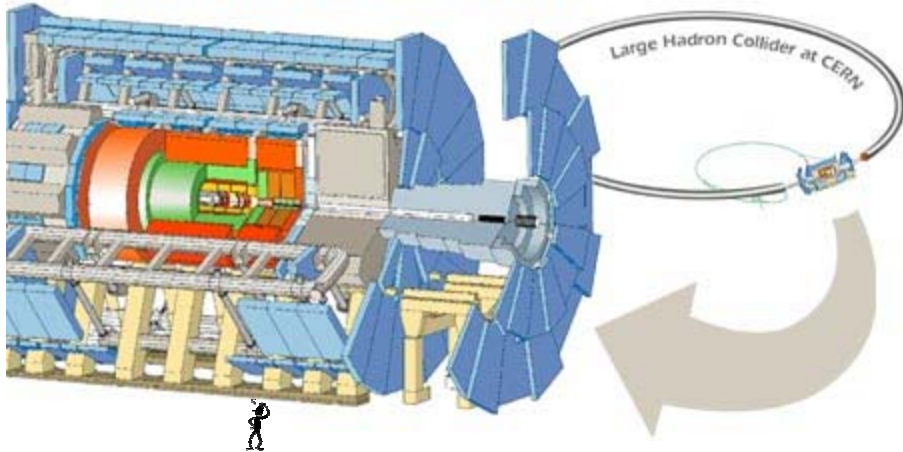
Some Examples

- Grappa
 - High Energy Physics (Indiana & Chicago)
- Gateway Computational Sciences Portal
 - Earthquake Science (Indiana)
- Nees Grid
 - Earthquake Engineering (Michigan)
- Alliance Portal
 - NCSA toolkit (Indiana, NCSA, Argonne)
- Open Grid Computing Environment (OGCE) Portal
 - Part of NSF Middleware Initiative: a standard toolkit of portal tools.
 - Indiana, Michigan, Texas, NCSA collaboration

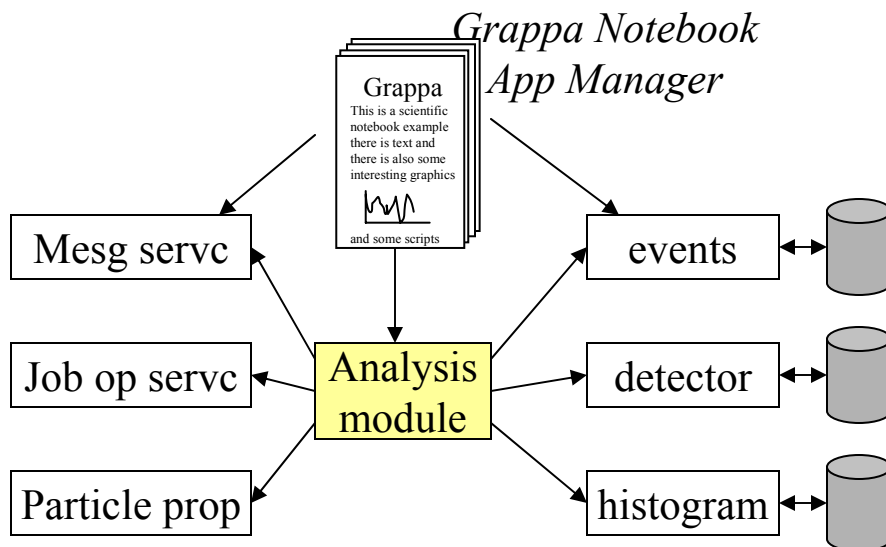
GRAPPA – A Science Portal



The ATLAS Detector



- Atlas Experiment
 - Large Hadron Collider
 - 100 MB/sec (10PB/year)
- Grid Access Portal for Physics Apps
 - To provide Atlas physicists with an point of access to Grid resources
 - Manage submission of Athena and ATSim jobs to condor cluster
 - Manage interaction of analysis with physics data and grid services



NEES – www.neesgrid.org

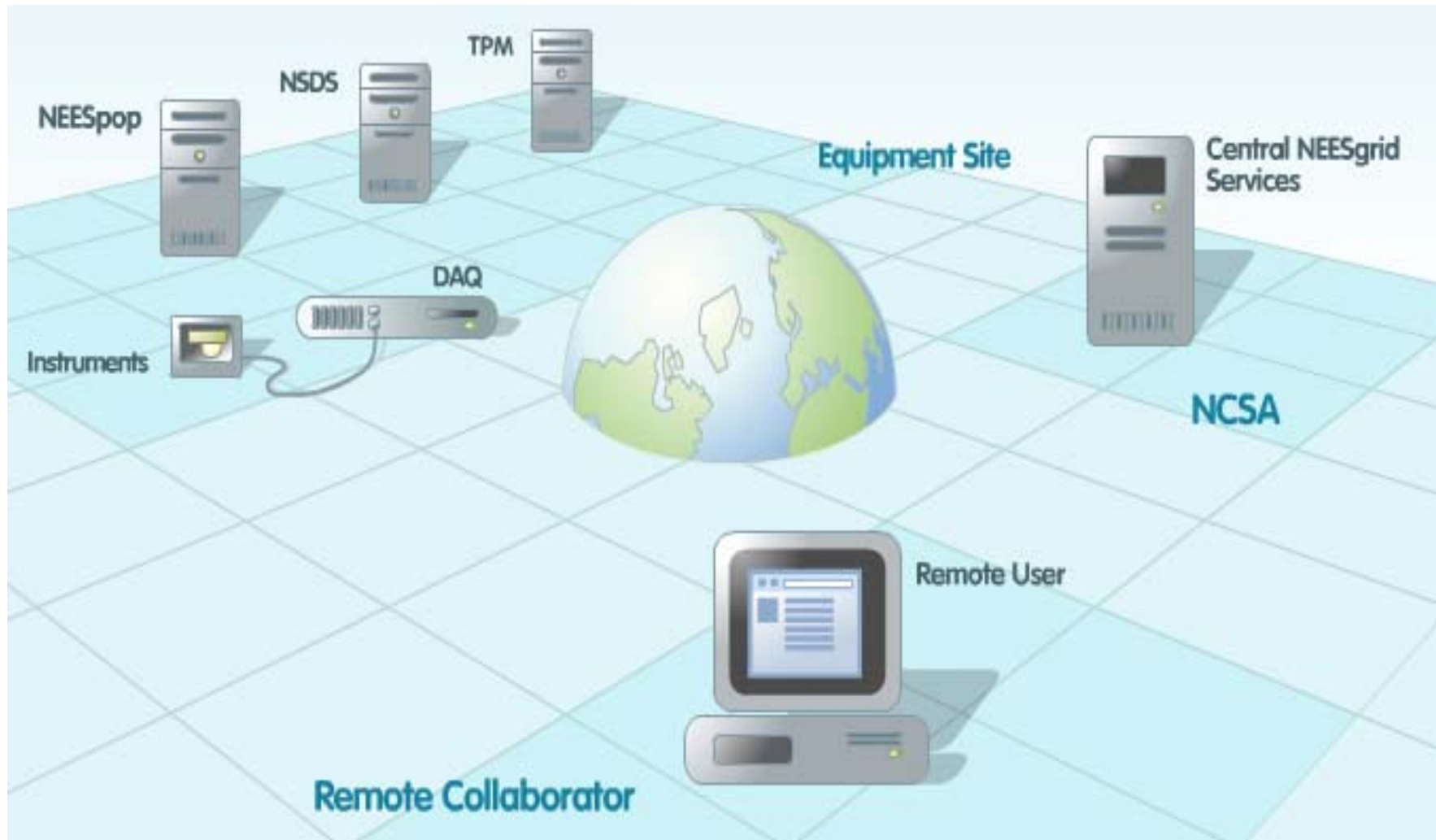
- George E. Brown, Jr. Network for Earthquake Engineering Simulation
 - Large Installations of physical equipment for earthquake experiments and simulations
 - Part of the award is to make equipment available for remote collaborators
- Focus is on collaboration and experimental equipment sharing in addition to access to computation



NEESGrid System Integration Team

- Developing infrastructure to:
 - access collaborative tools for experiment planning, execution, analysis, and publication.
 - perform tele-observation and tele-operation of experiments;
 - publish to and make use of a curated data repository using standardized markup;
 - access computational resources and open-source analytical tools;
- NCSA, ANL, USC/ISI, UMich, USC

NEESGrid Components



NEESgrid

...d Summer School: Grid Portals

NEESGrid Examples

The image displays three screenshots of the NEESGrid web application interface:

- Top Left:** A screenshot of a Microsoft Internet Explorer browser window showing the 'Video Cameras' page. The address bar indicates the URL: `http://neespop.ce.unr.edu:9271/chef/portal/group/NEESgridUNR/page/default.psm1/js_pane/P-f16ab04bfe-10006`. The page title is 'CHEF (dev-local): Worksite - Microsoft Internet Explorer'. The sidebar menu includes options like Home, Schedule, Announcements, Resources, Discussion, Telepresence Server, Video Cameras, ENotebook, Data Browser/Viewer, NEESgrid Repository, Chat, Browse Testbed, Testbed Status, News, Logout, Customize, and Users Present.
- Top Right:** A screenshot of the 'DataViewer' interface. It shows a table of parameters for an event named 'core: ex2 sine1-4'. The parameters include sine1, sine2, sine3, and sine4 with numerical values. To the right, there is a 2D plot showing a dense cluster of lines, likely representing the data being viewed.
- Bottom:** A screenshot of the 'NEES Repository @ UNR' interface. It features a tree view of the repository structure, including folders like 'UNR Demo', 'Data Acquisition', and 'White Noise System'. A detailed view of the 'Data Acquisition' folder is shown on the right, listing metadata such as 'name (string)', 'lastAccessedTimestamp (timestamp)', 'lastModifiedTimestamp (timestamp)', 'originalVersionID (object)', 'versionNumber (integer)', 'creationTimestamp (timestamp)', and 'lastTimestamp (timestamp)'.



Based on CHEF

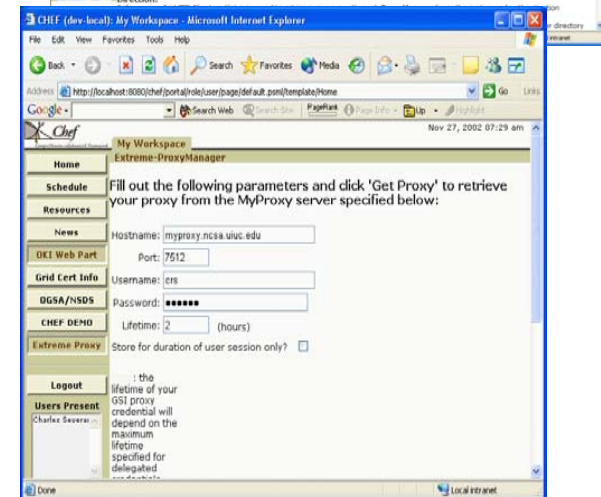
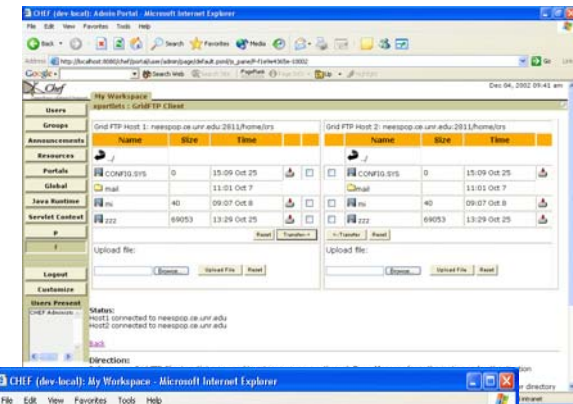
CompreHensive CollaborativE Framework

- Based on Jakarta Jetspeed - Portal Toolkit
 - Portal provided custom personal start pages for individuals
 - Unit of extensibility: Portlet
- CHEF - Portal-based Collaborative Application Framework
 - Adds notion of groups of users
 - Adds notion that groups have customizable portals (i.e. group portal administrator)
 - "Services Soup" - CHEF, OKI, COG, OGSA ...
 - Unit of extensibility: Module (Portlet, Teamlet, Service)



CHEF: Initial Applications

- Common Group-Aware Tools
 - Announcements, chat, calendar, threaded discussion, resource repository, webdav
- Learning Management System
 - Assignment tool, quizzing tool, OKI support
- Grid Infrastructure
 - Grid portal toolkit - used in NEESGrid
- More Later!



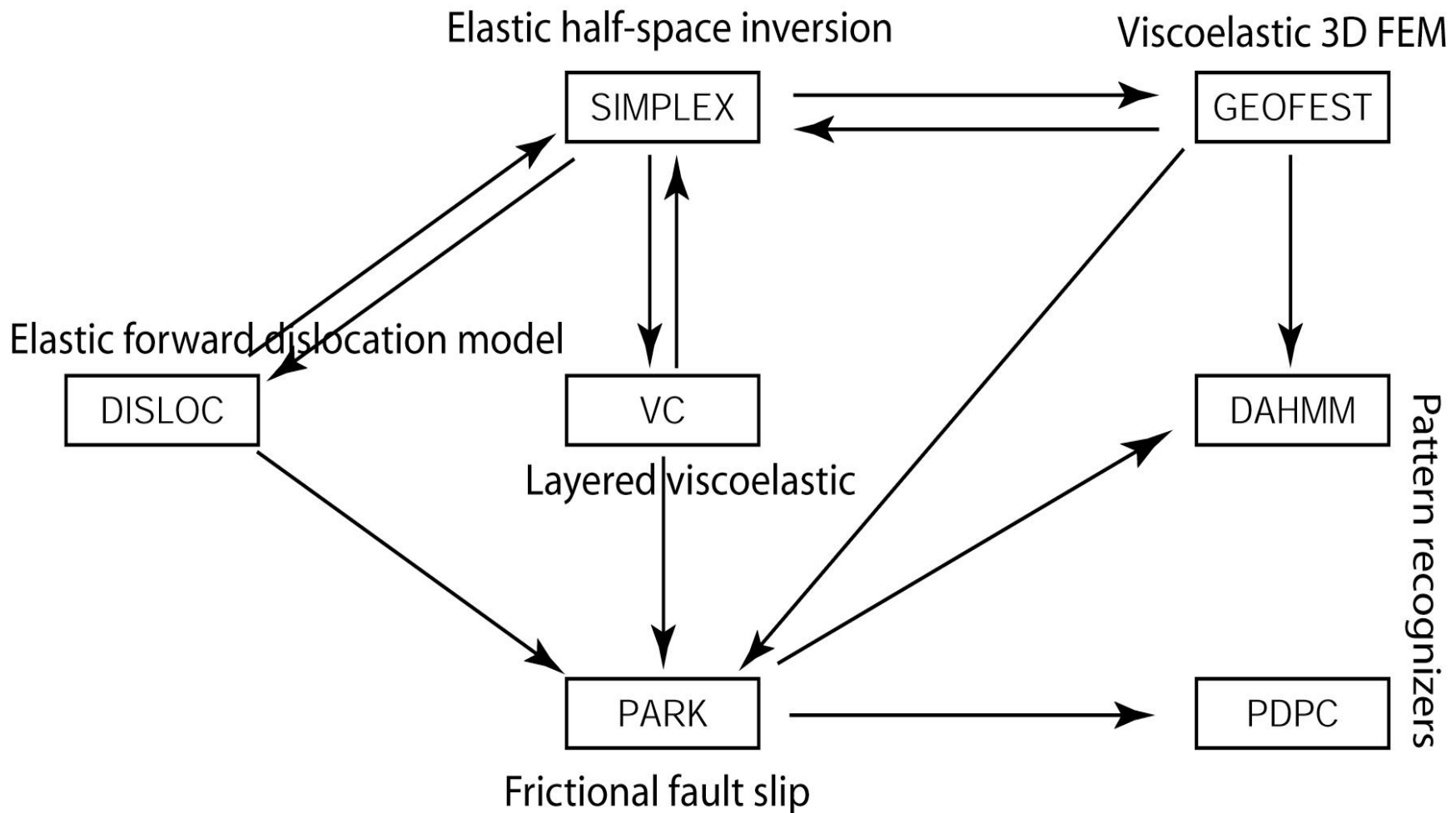
ServoGrid: Web Services and Portlets for Earthquake Science

Using Web Services and portlets to support earthquake simulation and forecasting.

Solid Earth Research Virtual Observatory Grid

- A number of simulation methods for studying earthquakes are being developed by the ServoGrid team including:
 - Disloc (JPL): Calculates surface displacements for model faults in an elastic half-space.
 - Simplex(JPL): Inverts Disloc to calculate model fault parameters from given surface displacements (→GPS data)
 - GeoFEST: 3D finite element code for calculating nodal displacements and tractions using realistic fault models.
 - Virtual California (UC-Davis): Simulates interactions between fault segments. Models time evolution of fault friction buildup and release on known California fault segments.
 - PARK (Brown): A boundary element program to calculate fault slip velocity history.

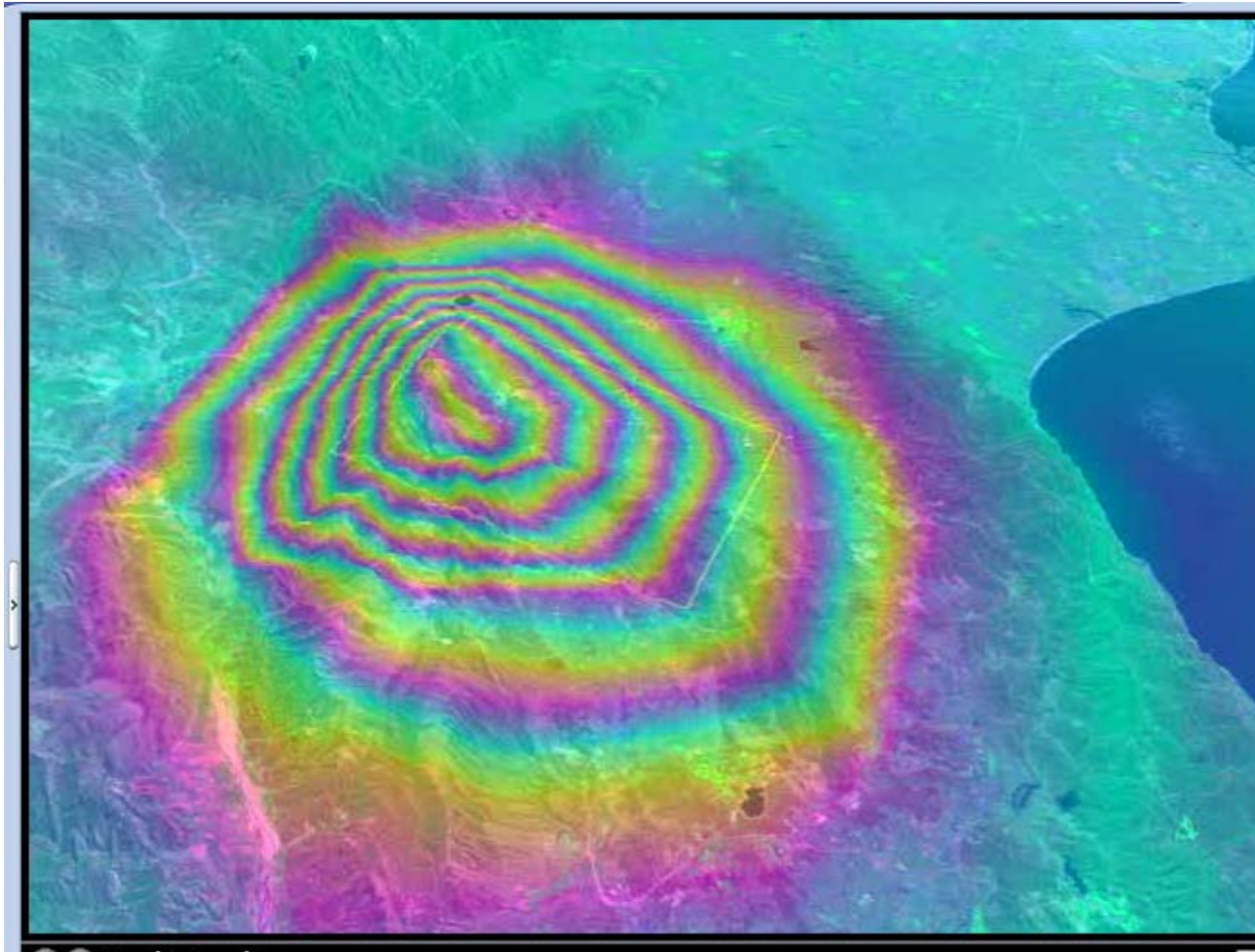
Interoperability through Data Sharing



What the Previous Slide Means

- SERVOnGrid codes need to interoperate through shared data.
- EX: Simplex surface observation points may be used as input to Disloc.
 - "What if I had more observations stations or another satellite? Would it decrease model uncertainties?"
- EX: Simplex can be used to create model faults need by Virtual California and Simplex.
 - "Would my model be better if I had two fault segments?"
- Data sharing may be either
 - Synchronous: link inputs and outputs
 - Asynchronous: store simulation results in databases with real observations.

GeoFEST Simulation of 1994 Northridge Earthquake



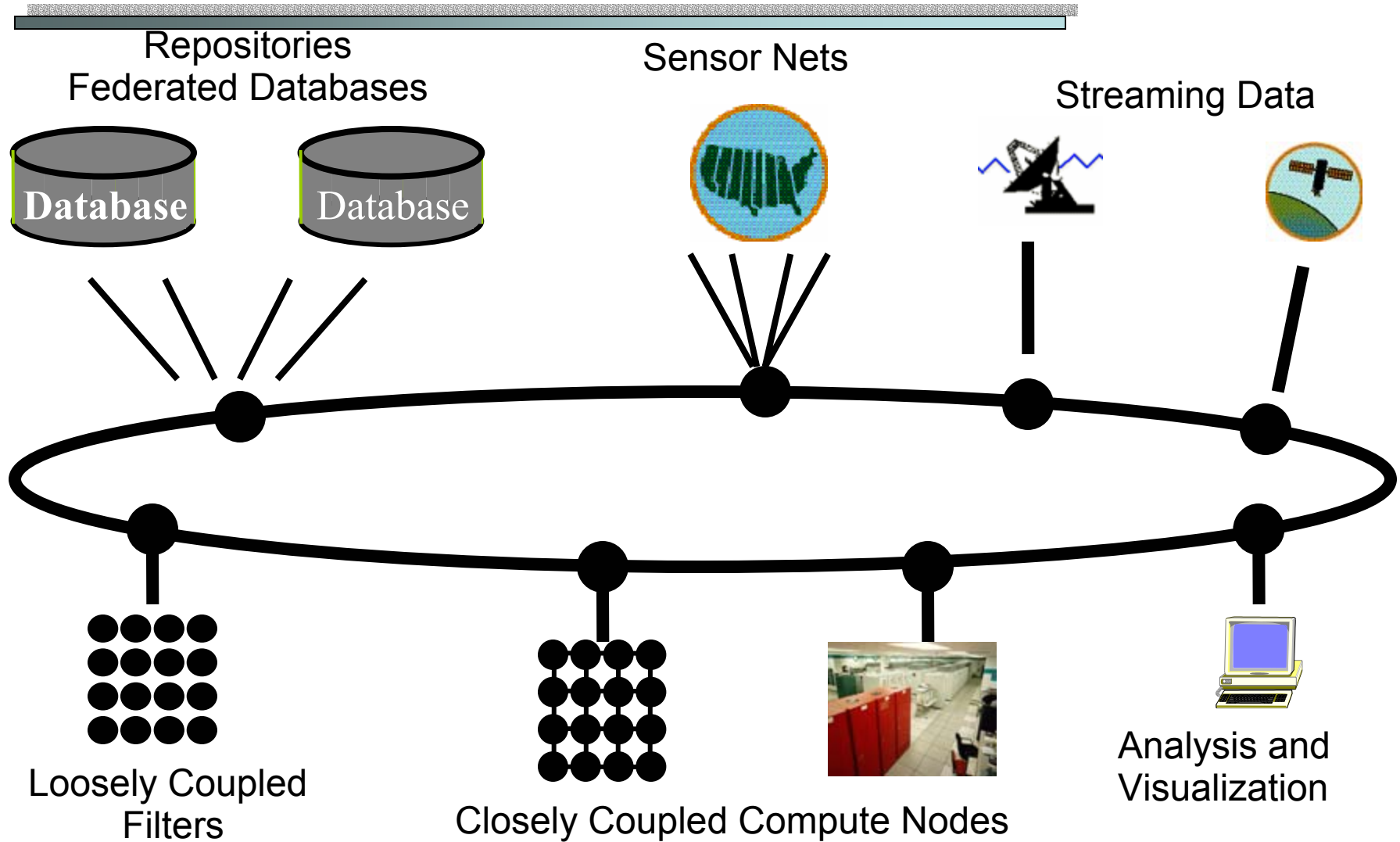
ServoGrid: Portal Access to Codes

- Codes are written independently, will need to run on workstations, clusters, and NASA supercomputers.
- As codes become more robust and accepted, problems emerge:
 - Need to manage information about distributed data sources: multiple databases, sensors, simulated data.
 - Need to organize, manage information about multiple code installation sites.
 - Need to simplify access to data, use of codes, and use of visualization/analysis tools for broad range of users
 - Need to link together

SERVOGrid: Accessing Data

- Codes also need access to a variety of data, including
 - Synthetic data generated by other codes
 - Historical fault and earth mantle data obtained from journals and other references
 - Web accessible GPS and Seismicity data.
- So in addition to application services, we need data services

SERVOGrid Architecture



SERVOGrid Services

- We have developed a number of core services to support
 - Access to codes through browser interfaces
 - Bind services to data sources
 - Link several dependent services into a single service
- Site Dependent Services
 - Job Submission
 - File Transfer
 - Job Monitoring
 - Host Monitoring
 - Visualization services

More ServoGrid Services

- Site Independent Services
 - Context Management for metadata
 - Batch Generation
 - Session archiving Services
 - Data services to access Fault data.
- Coupling services
 - We use Apache Ant as a web service to link several dependent tasks (such as steps needed to run visualization).
 - Ant tasks themselves may be Web service clients
 - More on this later

Computing Portal Grid Web Services

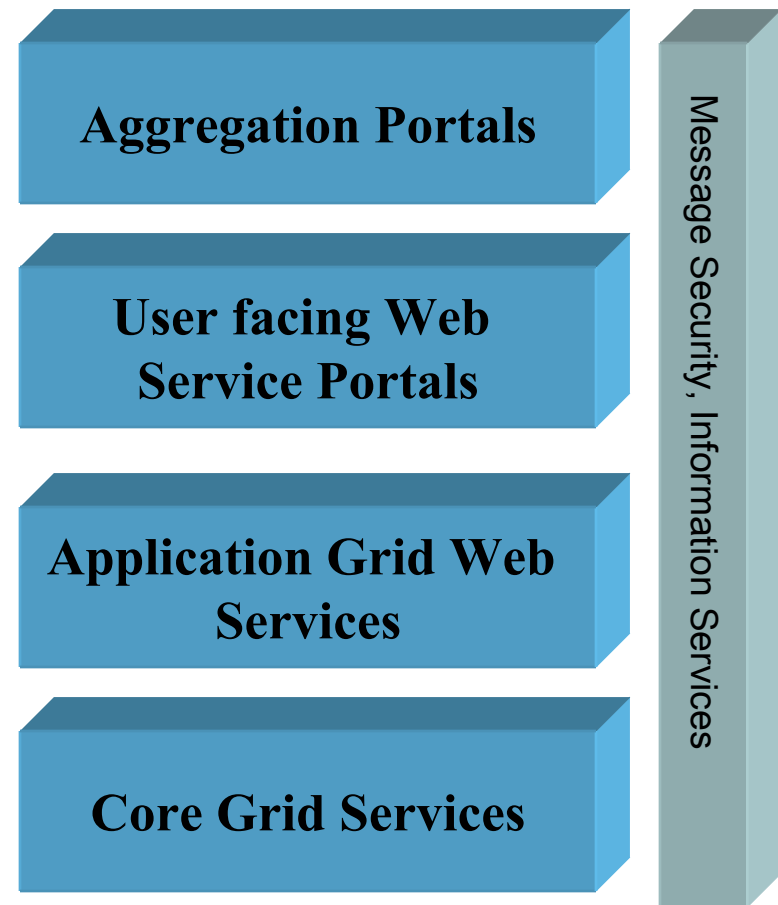
- We have built a suite of general purpose Grid Web services for managing distributed applications.
- **Core Computing services** define general purpose functions:
 - Ex: job submission, file transfer, job monitoring, management of jobs and results
 - Described as a **GridShell** as plays same role to Grid that Shell does for UNIX on a single machine
- **Application Grid Web services** include metadata about applications.
 - Built on top of core services.
 - Original application NOT changed
- We have developed a **toolkit** that allows one to convert general software packages into Grid Web Services and manage application collections

Application Grid Web Services

- AGWS are designed to make scientific applications (i.e. earthquake modeling codes) into Grid Resources
- AGWS services are described by two XML Schemas:
 - **Abstract descriptors** describe application options. Used by the application developer to deploy his/her service into the portal.
 - **Instance descriptors** describe particular user choices and archive them for later browsing and resubmission.

SERVOGrid Portlets and Portal Stacks

- User interfaces to SERVOGrid services (Code Submission, Job Monitoring, File Management for Host X) are all managed as portlets.
- Users, administrators can customize their portal interfaces to just precisely the services they want.



SERVOGrid Portal

- SERVOGrid codes, data, and services are managed on a testbed of IU, USC, and JPL resources.
 - Hosts include danube, noahsark, solar, grids
- We provide access to these with a Jetspeed-based portal.
 - Portlet components are Web service clients to remote services
 - Portlets can also be easily shared with other portals, like the Alliance Portal

GEM Computational Portal - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Home Search Favorites Media Refresh Print Stop

Address http://complexity.ucs.indiana.edu:8282/jetspeed/portal Go Links

Google john rundle davis Search Web Search Site News PageRank Page Info Up Highlight john rundle davis

Welcome to Jetspeed -> [Home](#)
[Customize](#) [Logout](#) [Edit Account \(gateway\)](#)
Server date: Jul 10, 2003 3:56:33 PM

GEM Applications **Noahsark** Admin Danube Solar Grids SERVO Files Fault DB

GEM Portal for Data Services

Welcome to the
Gateway Computational Science Portal

The Gateway Science Portal is a tool for remotely accessing computational science applications. For more information on the Gateway Project, please visit the our [web site](#).

Select Code **Application Selection:**
Choose the application and host you wish to use and submit your job.

View Archive **Problem Archive:**
View previously submitted problems, revise and resubmit.

Testbed hosts assigned to separate tabs

Community Grid Labs, Indiana University

GEM

Done Internet

Example: Host Solar

File Browser Web Service - Microsoft Internet Explorer

Address: http://complexity.ucs.indiana.edu:8282/jetspeed/portal/user/gateway/page/default.psm/js_pane/P-efb46991d1-10034

Welcome to Jetspeed -> [Home](#)
[Customize](#) [Logout](#) [Edit Account \(gateway\)](#)
Server date: Jul 10, 2003 4:06:52 PM

GEM Applications **Noahsark** **Admin** **Danube** **Solar** **Grids** **SERVO Files** **Fault DB**

Solar File Browser Index of: /scr/Gateway/GEMdata/Disloc

Filename	Download	Copy	Rename	Crossload
Gemdata1056469740294.out				
Test.o18203				
Test.e18203				
Gemdata1057072857752.xml				
Gemdata1057072857752.txt				
Gemdata1057072857752.out				
Gemdata1056478550228.xml				
Gemdata1056478550228.txt				
Gemdata1056478550228.out				
Gemdata1056479756555.xml				

Solar Job Monitor

Job Monitor

No Submitted Jobs.

File Selector Screenshot

- Users can view files on remote hosts
 - Shown is user file list for host noahsark.
 - Host solar is also shown.
- Files can be uploaded or downloaded between PC and remote host.
- Files may be crossloaded between two remote hosts.
- Same interface definition may be used to access databases.

File management

GEMPortal Solar.uits.indiana.edu Noahsark.ucs.indiana.edu Application Admin

NoahsarkFB NoahsarkJMJ

Noahsark File Browser

Index of: /home/cyoun/GEMCodes/Simplex

Upload File Change Directory

Filename	Download	Copy	Rename	Crossload
ran.c	✓	📄	🔄	📁
disloc.ps	✓	📄	🔄	📁
outfile	✓	📄	🔄	📁
simplex.input	✓	📄	🔄	📁
simplex.tar.gz	✓	📄	🔄	📁
simplex_readme.txt	✓	📄	🔄	📁
simplex4.ps	✓	📄	🔄	📁
factor_symmetric_matrix.c	✓	📄	🔄	📁
inverse_check.c	✓	📄	🔄	📁
invert_upper_triangle_matrix.c	✓	📄	🔄	📁
lower_cholesky_factor.c	✓	📄	🔄	📁
simplexfd.c	✓	📄	🔄	📁
square_upper_triangle_matrix.c	✓	📄	🔄	📁
arithmetic_types.h	✓	📄	🔄	📁
compilation.h	✓	📄	🔄	📁
environment.h	✓	📄	🔄	📁
math_tools_for_matrix_algebra.h	✓	📄	🔄	📁
ran.h	✓	📄	🔄	📁
ran.o	✓	📄	🔄	📁
jaycompile	✓	📄	🔄	📁
simplexfd.o	✓	📄	🔄	📁

Tabs indicate available portlet interfaces.

Lists user files on selected host, noahsark. File operations include Upload, download, Copy, rename, crossload

Application Selection Screen Shots

- Users can select from available codes
 - Simplex, Disloc, various VirtualCalifornia
- Users also select from available hosts for particular code
 - grids, noahsark, solar shown
- Selecting solar (Sun 64 Node E10000) prompts user for information needed to use PBS script generating service.

User Application Selection and Submission

GEMPortal Solar.uits.indiana.edu Noahsark.ucs.indiana.edu Application Admin

GEM Portal

Code Selection

Please select a code and host machine from the following list of applications. When you have made your choice, click the "Make Selection" button at the bottom of the page.

- Disloc
 - solar.uits.indiana.edu
 - grids.ucs.indiana.edu
 - Test2
- Simplex
 - solar.uits.indiana.edu
 - grids.ucs.indiana.edu
 - noahsark.ucs.indiana.edu
 - test2

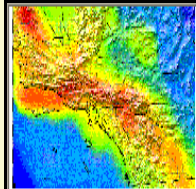
- VC STRESS GREEN
- VC SG COMPRESS
- VC INIT SER
- VC SER

Make Selection

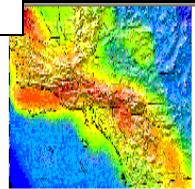
Cancel

Select desired application and host

Generate script for job submission



Welcome to Jetspeed -> [Home](#)
[Customize](#) [Logout](#) [Edit Account \(gateway\)](#)
Server date: Dec 4, 2002 5:11:47 PM



GEMPortal Solar.uits.indiana.edu Noahsark.ucs.indiana.edu Application Admin

GEM Portal

PBS Script Generator

Please provide the following information needed to generate the PBS queue script that will be run on solar.uits.indiana.edu.

Amount of Memory: 32mb

Job Name:

WallTime (hh:mm:ss): 15min

Number of CPUs:

Email: When job begins Email When job ends Email When job aborts Email

The code you have selected takes 1 input file. See the code documentation for details.

Input File:

The application generates 1 outputfile. Please specify the full path name of the directory on the HPC server where you would like this file to be placed.

Output File:

Make Selections

Cancel

Application Administrator Interfaces

- Application Administrators deploy and manage applications
- Update screen shows various application and host parameters that are set by the user.
- These are used to generate user interfaces for the selected codes.

Administer Grid Portal

The screenshot displays the 'Admin' interface for the GEM Application Admin Portal. At the top, navigation links include 'Customize', 'Logout', and 'Edit Account (gateway)'. The main content area is titled 'Form' and contains a section for updating application information. This section includes 'Input parameters, set 1' and 'Output parameters, set 1', each with fields for 'Input Field Handle', 'InputDescription', 'InputMechanism', 'Output Field Handle', 'Output Description', and 'Output Mechanism'. A 'Host Computers and Queues' section lists two hosts: 'solar.uits.indiana.edu' (Queue Type: PBS) and 'grids.ucs.indiana.edu' (Queue Type: CSH). Each host entry includes fields for Host Name, Host IP, Queue Type, Scratch Directory, Executable Path, Qsub Path, and Verify options. A sidebar on the left shows a list of application codes with radio buttons: Disloc, Simplex, VC_STRESS_GREEN, VC_SG_COMPRESS, VC_INIT_SER, and VC_SER. A 'GEMPortal Solar.uits.indiana.edu' header is visible at the top left. A 'GEM Application Admin Portal' header is also present. A 'Please select a code from the following list' instruction is followed by the radio button options. An 'Update Info' button is located below the list. The browser's address bar shows 'Internet'.

Provide information about application and host parameters

Select application to edit

Update Info

Host Computers and Queues:

Host Name	Queue Type
solar.uits.indiana.edu	PBS
grids.ucs.indiana.edu	CSH

Context Management

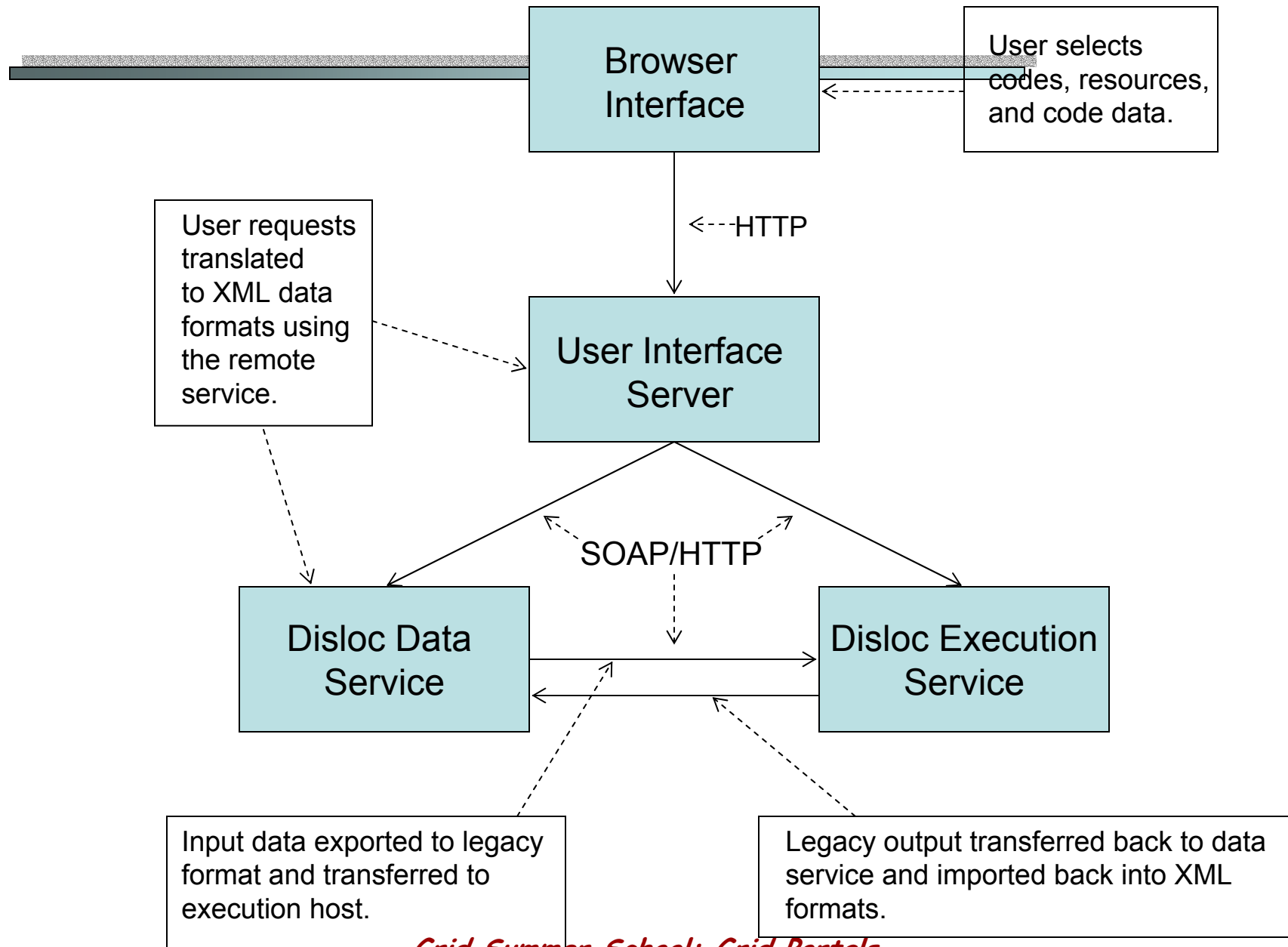
- Common problem of portals is to store all of the metadata associated with user sessions.
- Context Management service provides simplest possible data model
 - Context manager provides an easy interface to data trees.
 - Context data nodes are defined by recursive schema that hold optional, unbounded name/value pairs and child nodes.
- We use CM to store locations of job scripts, miscellaneous file URIs, etc.
- CM metadata stored on file systems, XML-native databases,
 - Actual data may be anywhere.
- Searched with XPath queries.

Application Codes as Web Services

- Scientific applications consist of several core Web services.
 - Get files to right place, script submission instructions, submit the job, get notified at various states.
- We need a meaningful metadata model for applications
 - Describe application-specific requirements
 - Describe bindings of applications to host environments and to Web services

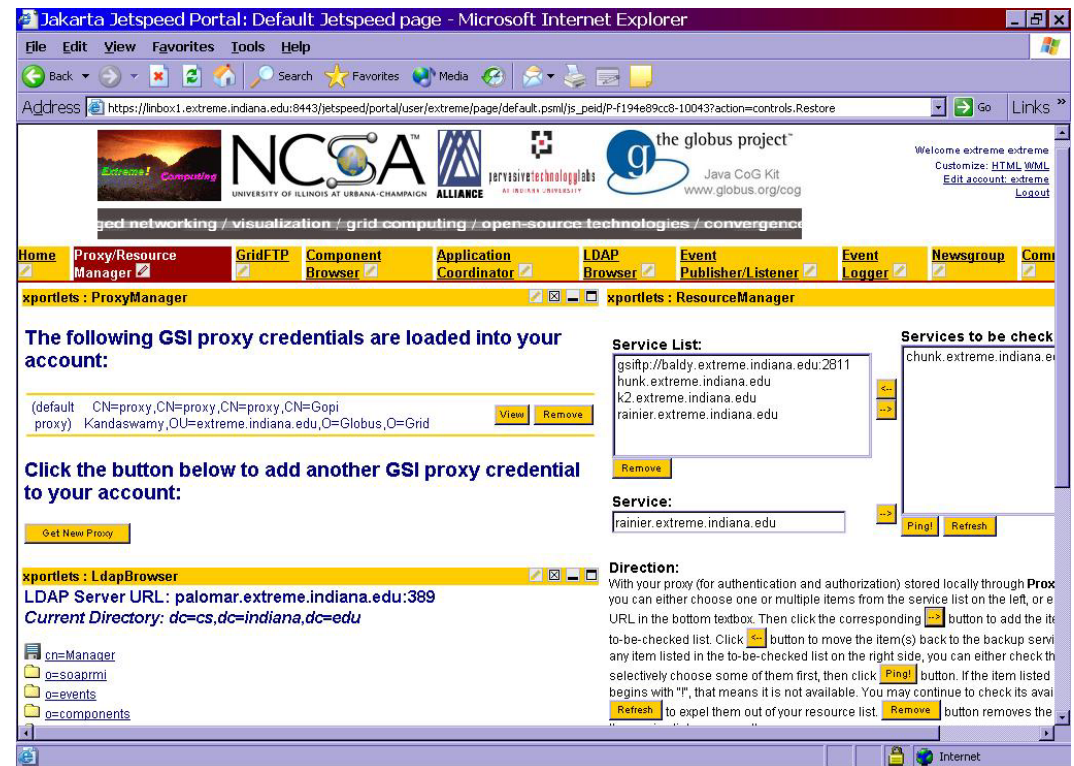
Data Sharing Services

- Codes have many common data types:
 - Faults
 - Mantle layers
 - Surface observation points
- But they all express these differently (custom/legacy formats).
- We have defined XML common data formats for these.
 - Services need translators to convert between legacy and common formats.
 - We've built these for Simplex, Disloc
- Currently converting schemas to Geography Markup Language (GML) → vast superset of what we need.
- Schemas available from <http://www.servogrid.org/GCWS/Schema>.



The Alliance Grid Portal

- Provides Portlets for
 - Management of user proxy certificates
 - Remote file Management via Grid FTP
 - News/Message systems
 - for collaborations
 - Grid Event/Logging service
 - Access to OGSA services
 - Access to directory services
 - Specialized Application Factory access
 - Distributed applications
 - Workflow
 - Access to Metadata Index tools
 - User searchable index



Open Grid Computing Environment Portal Project

- A collaboration of portal design groups to build a common set of tools for building Grid portals.
 - Indiana, NCSA, Michigan, Texas
 - Funded by
 - NSF New Middleware Initiative
 - NCSA, NPACI, DOE, NASA
 - Based on
 - Jetspeed and Chef
 - OGSI/OGSA
 - Evaluating GridShpere

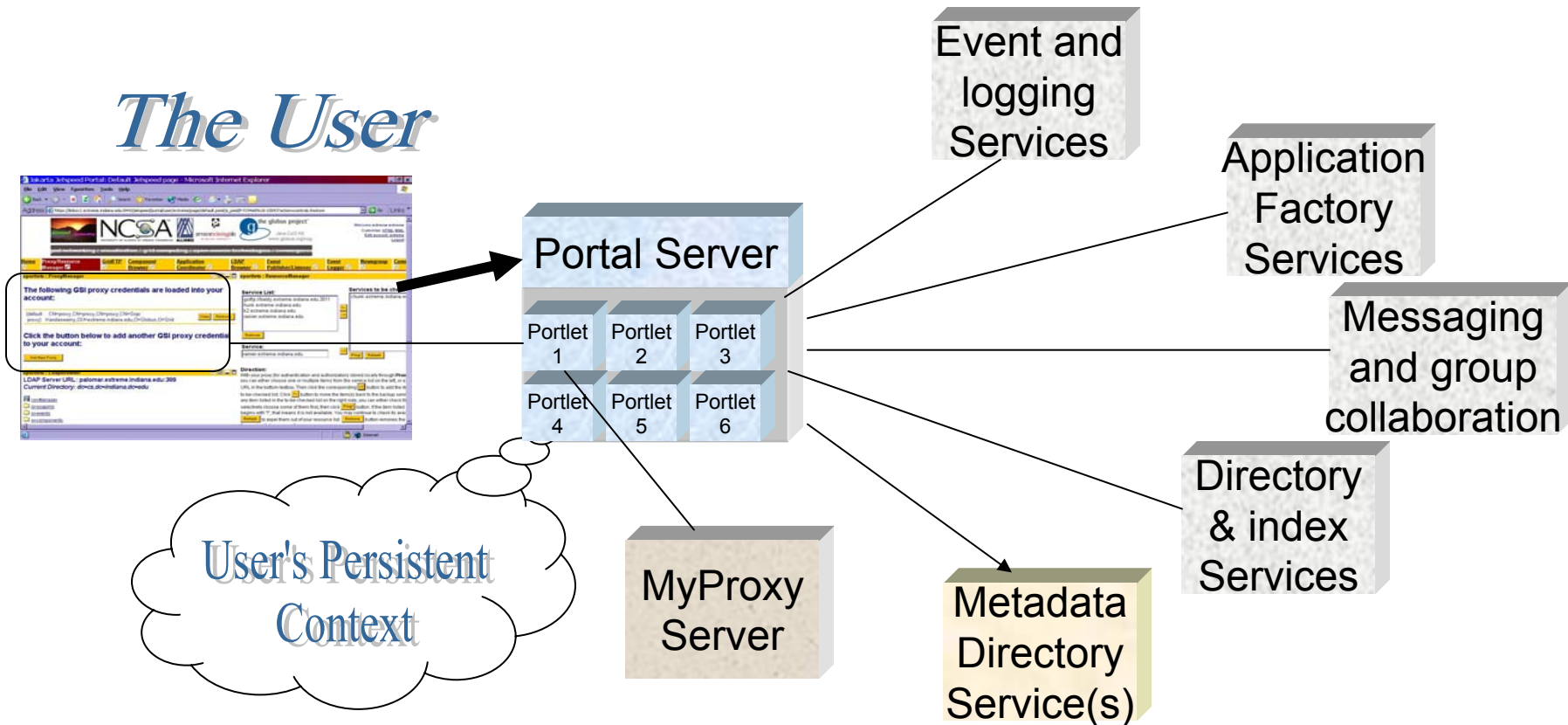
Portal Interaction Model

- How should the user interact with the Grid using the portal?
 - Logging onto the portal should bring up you current "Grid Context"
 - The set of tools (portlets) you use to access remote services
 - Configured into groups you access from tabbed panes the way you want them and the way you last left them.
 - Log files of events and annotations that for a persistent record of your grid transactions
 - Access to your group identity within a collaboration to give you access to group news and communications

OGCE based on a Portlet Model

- A Portlet is a portal server component that provides a basic functionality rendered in a user configurable window in a portal pane.

The User



Advantage of portlet architecture

- Each Grid service can be associated with a unique portlet
 - Very easy to add new services
 - Many different Groups can contributed portlets which can be plugged into a portal.
 - Currently Indiana, Argonne, Michigan, NCSA, Texas + GridSphere portal group following this model.
 - Jetspeed is Apache standard. Also basis for IBM websphere portal and others. Now moving to Java Standard
- Each user can select and configure the portlets he/she wishes to use
 - Selection becomes part of the persistent context

The Hard Questions

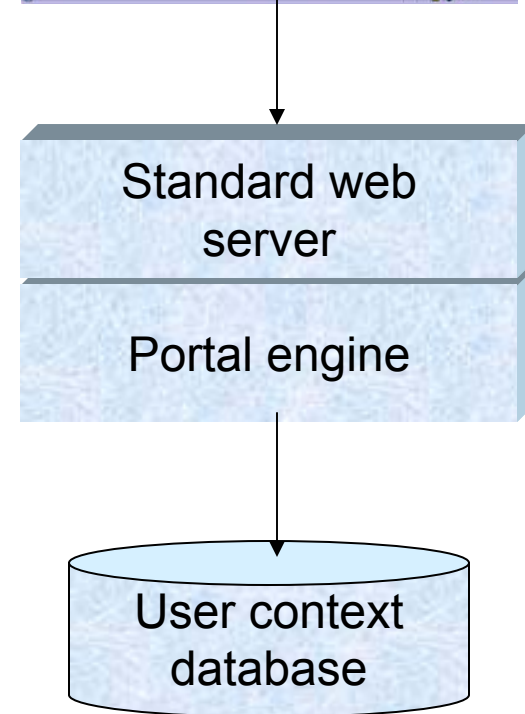
- If the Grid consists of thousands of services, do we need a portlet for each one?
 - Clearly not. Who will write/use all those portlets?
 - Is there a set of "generic portlets" that can adapt to any Grid/Web service?
- Most Grid/Web services are designed to be invoked by other services... not humans
 - What is the right model for the user interaction?
 - Should some services be our "agent services" that talk to us while managing our interaction with remote services?
 - If a service wants to interact with a human, should it not supply its own "portlet"?

Hour 2. Portal Programming

The basic architecture
How to write a portlet

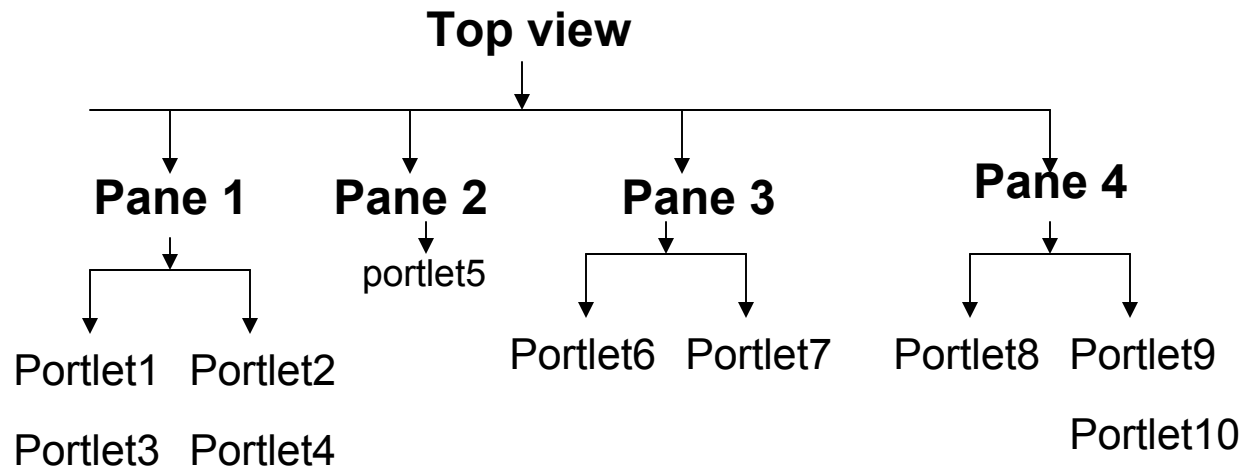
Standard Portal Architecture

- User access:
 - A standard browser
 - User provides login id & password
- Portal engine
 - Retrieves user context from DB
 - User's configuration of the portal interface and other persistent data.



The “Standard” Implementation

- Based on the Java “Portlet” model.
 - IBM Websphere, Oracle, HP, Sun, BEA Weblogic, ...
 - Open Source:
 - Apache Jakarta Jetspeed, Chef, Gridsphere, JPortlet ...
- User’s View of the portal:
 - A user configurable set of “panes” each containing a set of small web applications called portlets.



A Look at a pane and portlet

The current Visible pane

Proxy Manager Portlet

The screenshot shows the Jakarta Jetspeed Portal interface. At the top, there is a navigation bar with menu items: Home, Proxy/Resource manager, GridFTP, Component Browser, Application Coordinator, LDAP Browser, Event Publisher/Listener, and Event Logger. Below the navigation bar, there are several portlets. The 'Proxy Manager' portlet is highlighted with a black box and a label 'Proxy Manager Portlet'. It displays the following content:

portlets : ProxyManager

The following GSI proxy credentials are loaded into your account:

(default proxy) CN=proxy,CN=proxy,CN=proxy,CN=Gopi Kandaswamy,OU=extreme.indiana.edu,O=Globus,O=Grid	<input type="button" value="View"/>	<input type="button" value="Remove"/>
--	-------------------------------------	---------------------------------------

Click the button below to add another GSI proxy credential to your account:

Other portlets visible include 'Resource Manager' (Service List) and 'Ldap Browser' (LDAP Server URL: palomar.extreme.indiana.edu:389, Current Directory: dc=cs,dc=indiana,dc=edu).

Direction:
With your proxy (for authentication and authorization) set you can either choose one or multiple items from the service list. Then click the corresponding checkbox in the to-be-checked list. Click button to move the item(s) from the to-be-checked list on the right side to the service list. You may selectively choose some of them first, then click begins with "!", that means it is not available. You may

User Configuration

- At any time the user can add a new pane
- In that pane, he/she can add portlets
 - The system administrator adds new portlets as they become available.
- configure them on the page
 - Single column or multi-column
 - Other interface options

Jetspeed Computing Portal: Choose Portlets



Welcome to Jetspeed ->

[Contact](#) [Home](#)

[Customize HTML](#)

[Customize WML](#)

[Edit account: \(GeoffreyFox\)](#)

[Logout](#)

[Home](#)

[Documentation](#)

[API](#)

Customize pane

HTML

[Layout](#)

[Add portlet](#)

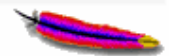
Select	Title	Description
<input type="checkbox"/>	AdminInfo	Unavailable
<input type="checkbox"/>	Ansys	Construct a batch script for ANSYS
<input type="checkbox"/>	Apache News from Apache Week	The essential resource for anyone running an Apache server or anyone responsible for running Apache based services
<input type="checkbox"/>	Apacheweek	Unavailable
<input checked="" type="checkbox"/>	CommunityFileBrowser	File Browser for Community
<input checked="" type="checkbox"/>	CommunityJobMonitor	Runs top on Grids
<input type="checkbox"/>	GridsFileBrowser	File Browser for Grids
<input type="checkbox"/>	GridsJobMonitor	Runs top on Grids
<input type="checkbox"/>	HelloJSP	Simple JSP Portlet Example
<input type="checkbox"/>	HelloTest	Simple hello test
<input type="checkbox"/>	HelloVelocity	Simple Velocity Portlet Example
<input type="checkbox"/>	JSP1_2andJetspeedTagLib	JSP Portlet Example that displays data from the Jetspeed Tag Library and Java Servlet request class
<input type="checkbox"/>	JavaSoft	JavaSoft
<input type="checkbox"/>	Jetspeed	Unavailable
<input type="checkbox"/>	Jetspeed	Jetspeed is a multi-device portal system with simple syndication capabilities

4 available portlets linking to Web Services
I choose two

<< Prev

[Next >>](#)

[Add](#)



Choose Portlet Layout

Welcome to Jetspeed -> [Contact](#) [Home](#)

[Customize HTML](#) [Customize WML](#) [Edit account: \(GeoffreyFox\)](#) [Logout](#)

Customize pane

HTML [Layout](#) [Add portlet](#) - -

Layout : Single column Colors : orange-grey

CommunityFileBrowser

CommunityJobMonitor

Choose 1-column Layout

Welcome to Jetspeed -> [Contact](#) [Home](#)

[Customize HTML](#) [Customize WML](#) [Edit account: \(GeoffreyFox\)](#) [Logout](#)

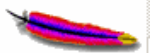
Customize pane

HTML [Layout](#) [Add portlet](#) - -

Layout : Two columns (50/50) Colors : orange-grey

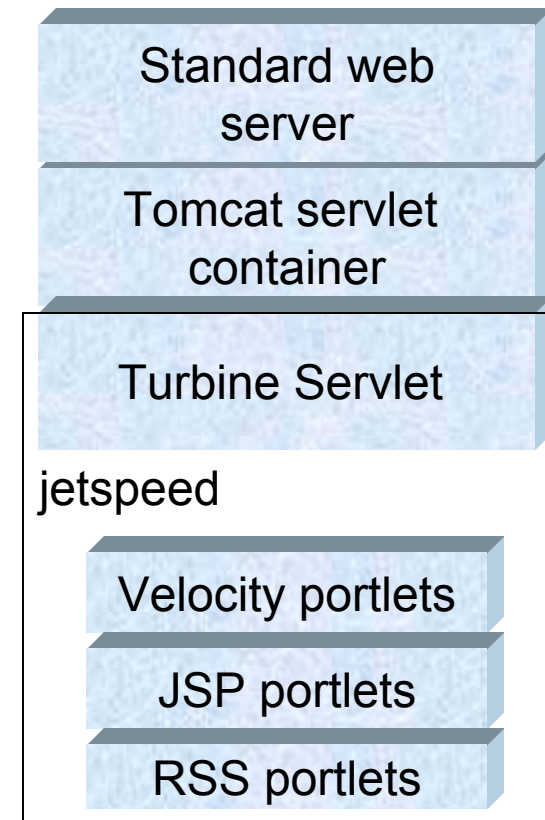
CommunityFileBrowser CommunityJobMonitor

Original 2-column Layout

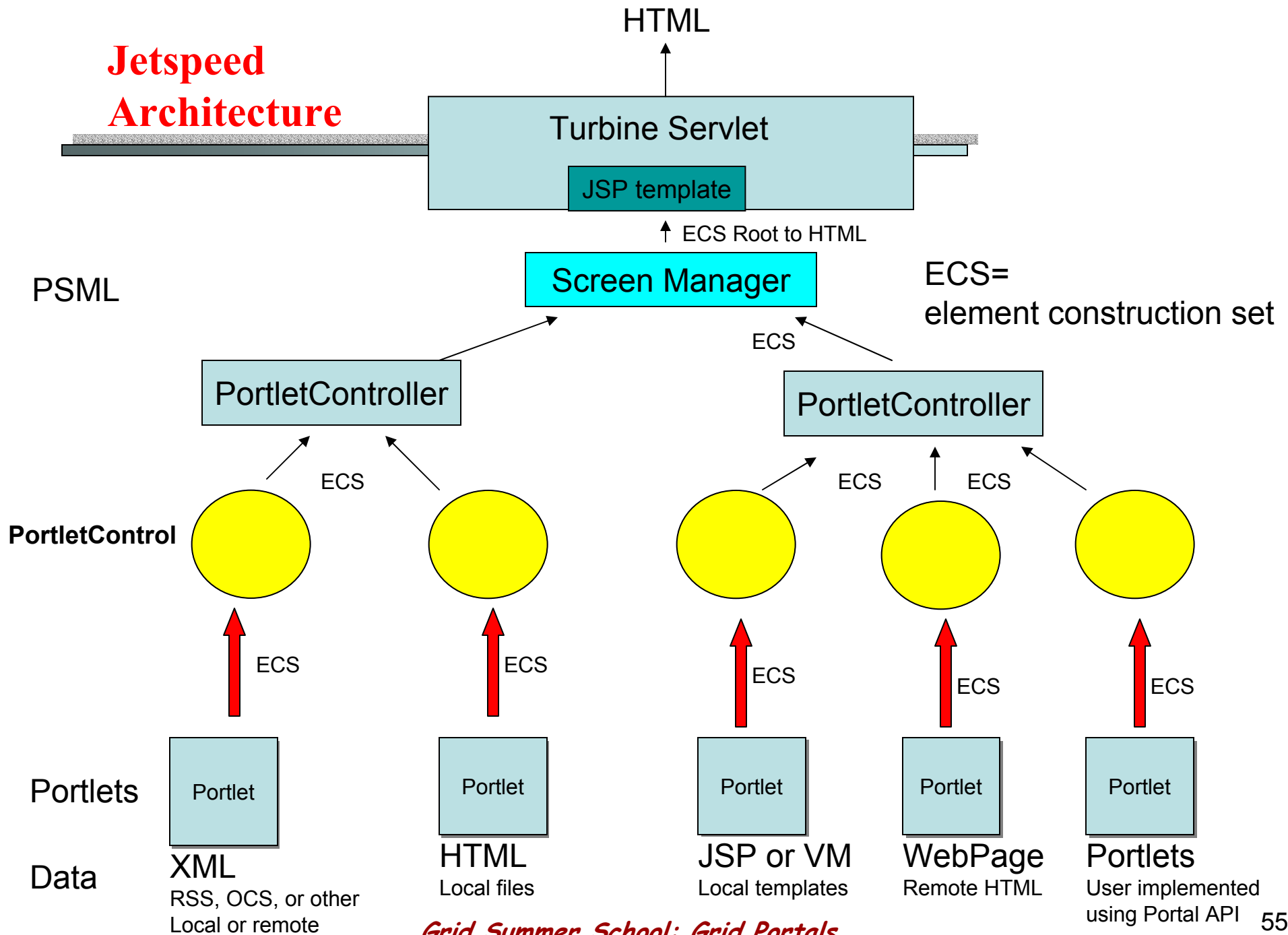


The Jetspeed Architecture

- Jetspeed builds on the Turbine Servlet which
 - Provides the “model-view-controller” framework for portlet execution.
 - 3 basic types of portlets
 - Velocity templates
 - Java Server Pages
 - RSS feeds



Jetspeed Architecture



The Model View Controller Pattern

- The MVC pattern is a way to design applications that separates
 - The View of an application, i.e. its user interface
 - The Model = the business end of the application
 - The controller = the entity that sequences view-model interaction
- For a portlet this means
 - Use a markup/template language to describe the details of how information is presented to the users
 - Use a set of back-end classes to define the way the portlet does its computations
 - Let the control of the two be handled by a third party like Turbine or by a specialized "action" class.

A Velocity based Example

- A simple portlet that keeps a list of names and allows the user to add to the list

The last name added was: **Bill Jones**
Here is the list so far:
Charlie Catlett
Beth Plale
Bill Jones

Do you want to add somebody?

List of People Example

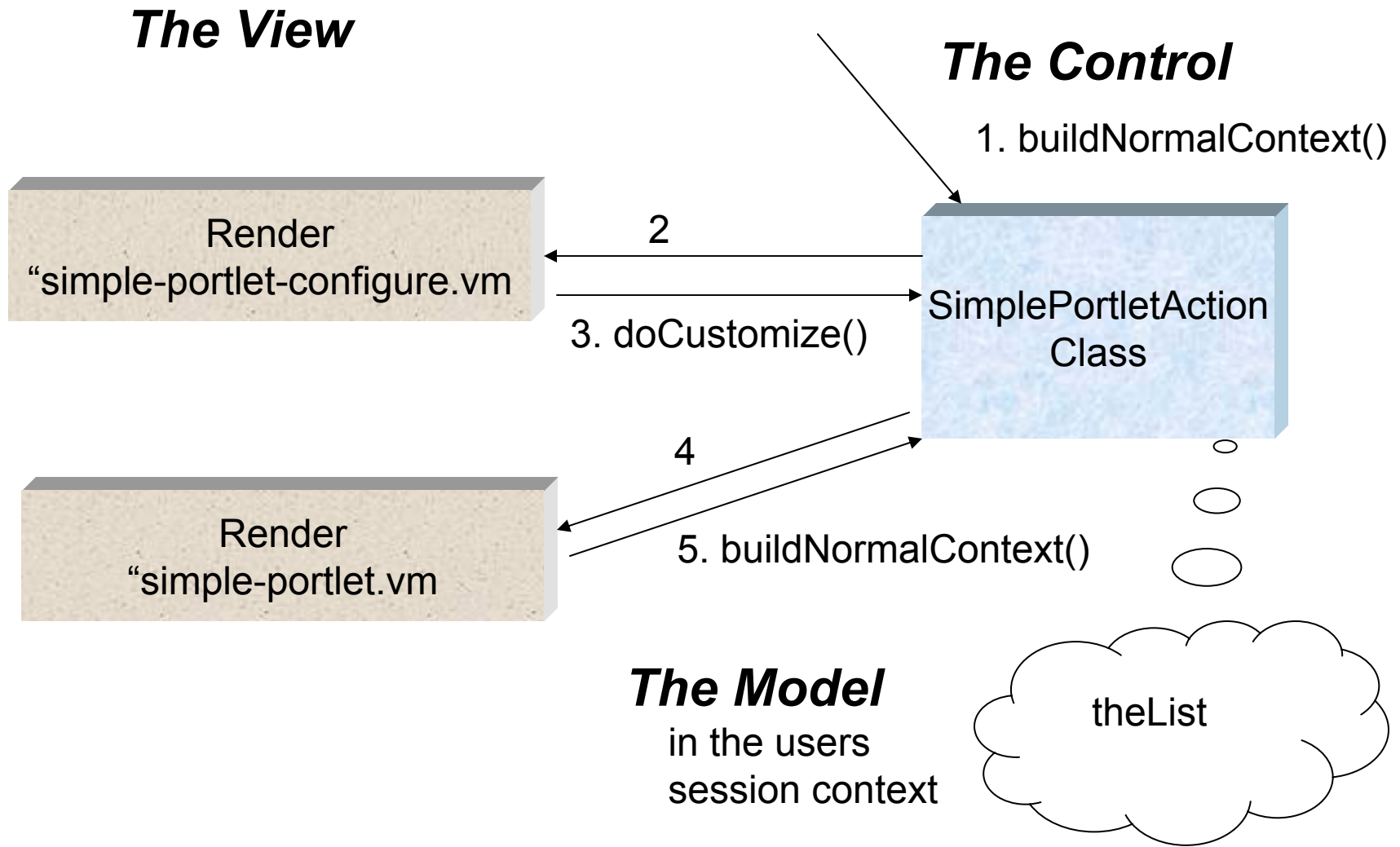
- But when it first comes up we want to configure it with a first name.

The first name is fred
He is the default to get the list started.

Would you like to start with a different name?

Show List!

MVC Picture



Simple-portlet.vm

```
<b>The last name added was: $person </b>
<br/>Here is the list so far: <br/>
#if($links)
  #foreach ($name in $links)
    <font style="font-size:12">person: $name </font></b>
    <br/>
  #end
#end
<br/>Do you want to add somebody?
<form action="$jslink.Template" method="post">
  <font color="$!{skin.Color}">
    <input type="text" name="person" value="$!person" size="50">
    <br/>
    <input type="submit" name="eventSubmit_buildNormalContext"
      value="refreshlist"/>
  </font>
</form>
```

Simple-portlet-configure.vm

```
#if (!$person)
  #set ( $person="fred" )
#end
```

The first name is \$person
He is the default to get the list started.

```
<form action="$jslink.Template" method="post">
  <font color="$!{skin.Color}">
    would you like to start with a different name?:
    <input type="text" name="person" value="$!person" size="50">
    <br/>
    <input type="submit" name="eventSubmit_doCustomize"
      value="Show List!"/>
  </font>
</form>
```

The Action Class

```
public class SimplePortletAction extends VelocityPortletAction{
    private String PERSON = "person";
    private String OLDVEC = "xportlets.simpleportlet.thelist";
    private String LINKS = "links";
    private String PARAMS = "params";
    private String START = "xportlets.simpleportlet.start";

    protected void buildNormalContext( VelocityPortlet portlet,
                                       Context context,
                                       RunData rundata ) {
        String start = (String)rundata.getUser().getTemp(START, "false");
        if(start.equals("false")) {
            setTemplate(rundata, "simple-portlet-customize");
            rundata.getUser().setTemp(START, "true");
        } else {
            setTemplate(rundata, "simple-portlet");
            doGetinfo(rundata, context);
        } }
}
```

Clicking the customize button

```
public void doCustomize(RunData data, Context context)
{
    String person = data.getParameters().getString(PERSON);

    VelocityPortlet portlet = (VelocityPortlet)context.get("portlet");

    portlet.setAttribute(PERSON, person, data);
    data.getUser().setTemp(PERSON, person);
    Vector vec = new Vector();
    data.getUser().setTemp(OLDVEC, vec);

    buildNormalContext(portlet, context, data);
}
```

The Action: doGetinfo

```
public void doGetinfo( RunData data, Context context )
{
    String person = data.getParameters().getString(PERSON);
    if(person == null) data.getUser().getTemp(PERSON);
    context.put(PERSON,person);

    Vector vec;
    Vector oldvec = (Vector) data.getUser().getTemp(OLDVEC);

    if(oldvec != null) vec= oldvec;
    else vec = new Vector();
    if(person != null) vec.add(person);

    data.getUser().setTemp(OLDVEC, vec);
    context.put( LINKS, vec );
}
```


Configuring the Portlet in the Portal

- We add an "simpleportlet.xreg" file

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<registry>
```

```
  <portlet-entry name="xportlets:simpleportlet" hidden="false"  
    type="ref" parent="CustomizerVelocity" application="false">
```

```
    <meta-info>
```

```
      <title>xportlets : simple portlet </title>
```

```
      <description>tutorial demo</description>
```

```
    </meta-info>
```

```
  <classname>
```

```
    org.apache.jetspeed.portal.portlets.CustomizerVelocityPortlet
```

- ~~Continued~~ next slide

Xref file continued

- Binding the templates to the action class

```
<parameter name="template" value="simple-portlet" hidden="false"
  cachedOnName="true" cachedOnValue="true"/>
<parameter name="customizeTemplate"
  value="simple-portlet-customize" hidden="false"
  cachedOnName="true" cachedOnValue="true"/>
<parameter name="action" value="xportlets.SimplePortletAction"
  hidden="false" cachedOnName="true" cachedOnValue="true"/>
<media-type ref="html"/>
<url cachedOnURL="true"/>
</portlet-entry>
</registry>
```

Portlets for Legacy and Remote Web Content

How to pull content into
Jetspeed from external servers.

Limitations of Local Portlets

- Typical portlets are created from templates (either JSP or Velocity) that are located on the Portal (Jetspeed) server.
- But there are lots of portals (and an infinite amount of web content) that you may want to add.
 - You want to minimize conversion time for legacy portals.
- Jetspeed does not make this easy
 - Excludes non-Java technology.
 - HTML Frames not supported

Two Solution Methods

- These problems can be solved by “proxy” portlets
 - Connect to remote sites and download content.
 - Need also to navigate remote content in the portal
- IFramePortlet use IFrames to load remote sites.
 - Standard solution available with Jetspeed.
- WebFormPortlet developed for the Alliance Portal
 - Provides lower level control over remote sites.
 - Not discussed in detail here. See Marlon Pierce for more details.

Using IFramePortlet

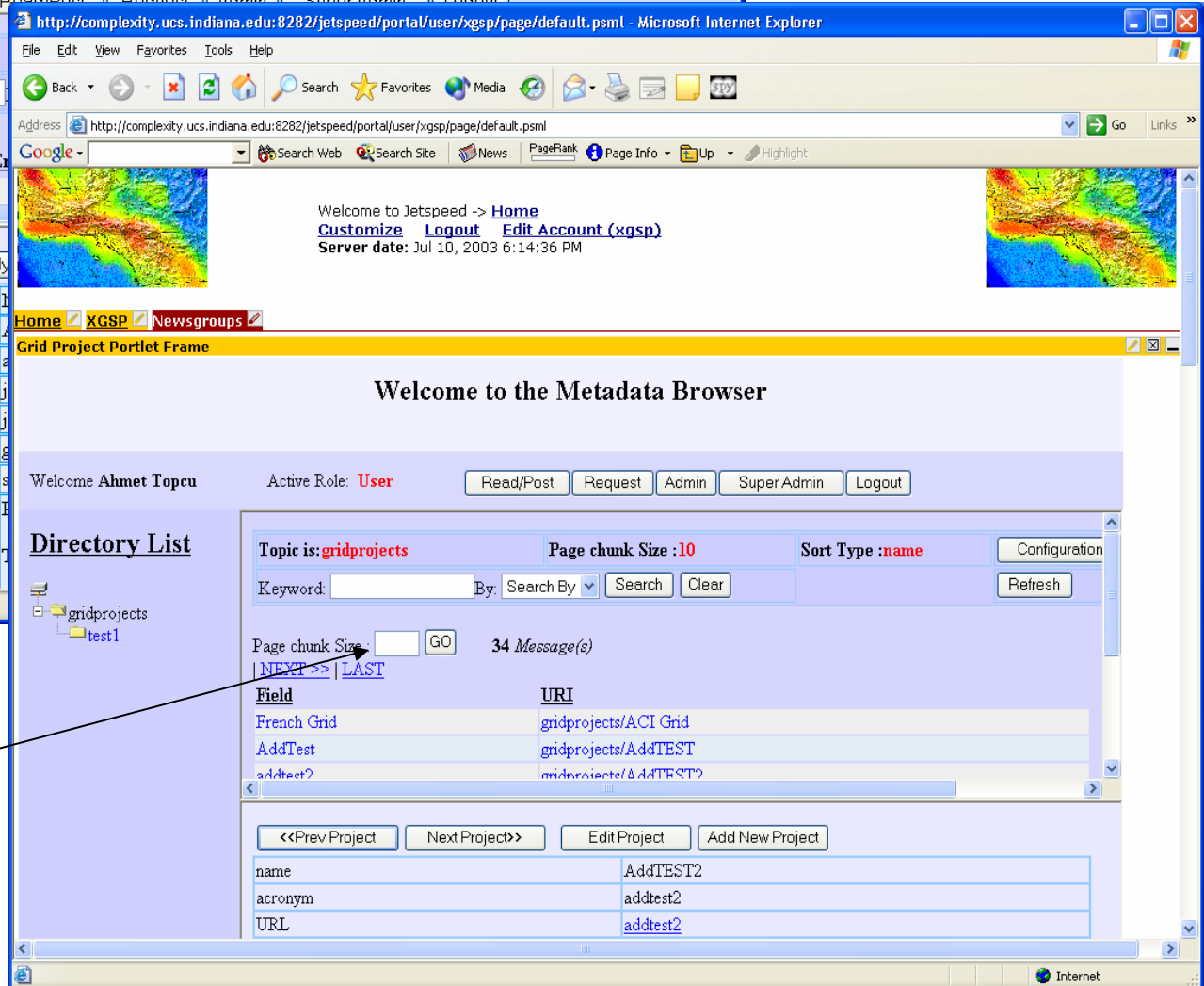
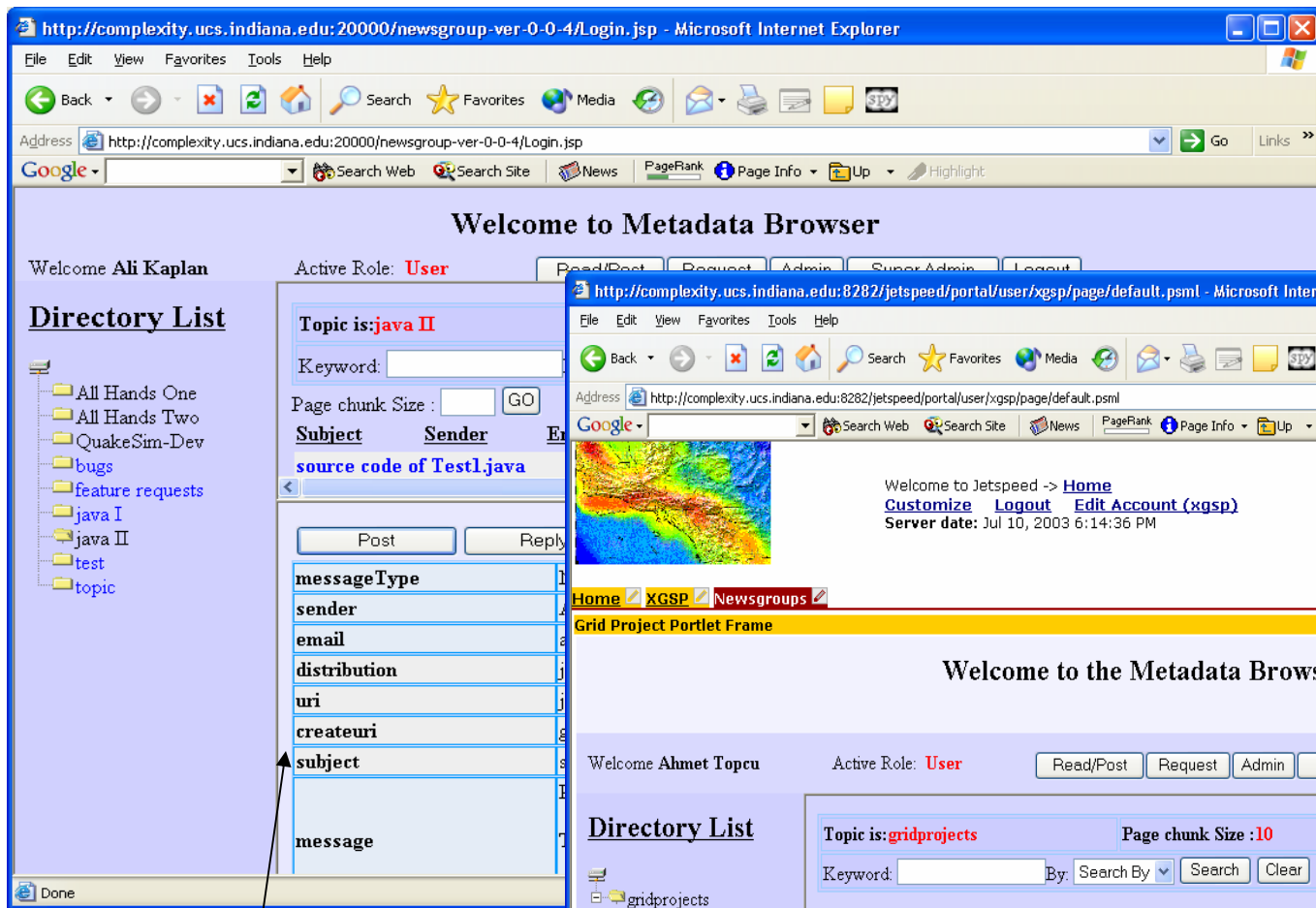
- This provides the simplest solution for adding legacy/remote content.
 - Load pages and inherit Jetspeed features like customization.
- Works very well
 - Browsers will manage cookies, SSL, HTTP request/response messages with form parameters, Javascript that they encounter in IFrames.
 - Can also display pages with HTML Frames.
 - Supports popup windows.
- Drawbacks
 - You must specify the frame size in configuration files.
 - No low level control.

Adding an IFrame

- Shutdown Tomcat.
- Edit one of the .xreg files in the `$JETSPEED/WEB-INF/conf` or create a new one.
 - See next slide for a sample entry.
- Restart Tomcat.
- Login to Jetpeed and customize the display to add the new IFrame portlets.

Sample XREG Entry for IFrame

```
<portlet-entry name="XMLNuggets" hidden="false" type="ref"
  parent="IFramePortlet" application="false">
  <meta-info>
    <title>CommGrids Newsgroup System</title>
    <description>Load the CGL's Newsgroup system as a
  portlet</description>
  </meta-info>
  <parameter name="source"
    value="http://complexity.ucs.indiana.edu:20000/newsgroup-
  ver-0-0-4/Login.jsp" hidden="false"/>
  <media-type ref="html"/>
</portlet-entry>
```

Remote portal content pulled into a Jetspeed portal.

Some Pros and Cons of IFrame Portlets

- IFrames are standard HTML supported by IE and Netscape 6 and 7.
- Very simple, reliable
 - Browsers rather than portlets manage SSL connections, cookies, etc.
 - Works well with JavaScript, Web forms.
- Netscape 4 does not support.
- IFrame Portlets don't give you low level control of content
 - Can't filter out or add to content.
- Currently, you can't dynamically size the portlet window.
- Solution: Use webform portlet.

A Closer Look at Chef

A Portlet-based architecture
designed for collaboration

The Chef Portal Architecture

- Comprehensive collaborative Framework (CHEF) is a portal
- Collaboration between UMich School of Information and Media Union, OKI, CMCS, NEES
- Goal is to provide portal tools for web-based collaborative working communities

Functionalities

- Administration
 - Protected by CHEF security
- Users
 - Authenticates to various communities (portals)
- Presence
 - Active user listing
- Communication
 - Through persistent channels depending on the tool being used
 - Tools can be announcements, discussions, email, chat and IM
- File Hosting and Manipulation
 - Stored as collection per portal

Functionalities (contd.)

- Resource Description
 - Asserting qualities to resources and relations among different resources.
- Search tools
 - Use filters
- Notification
 - When certain events occur
 - Immediate login notification, or by email
 - Stores a history of past notifications
- Synopsis
 - Dynamically updating views of the filter results from search/notification in real time

Functionalities (contd.)

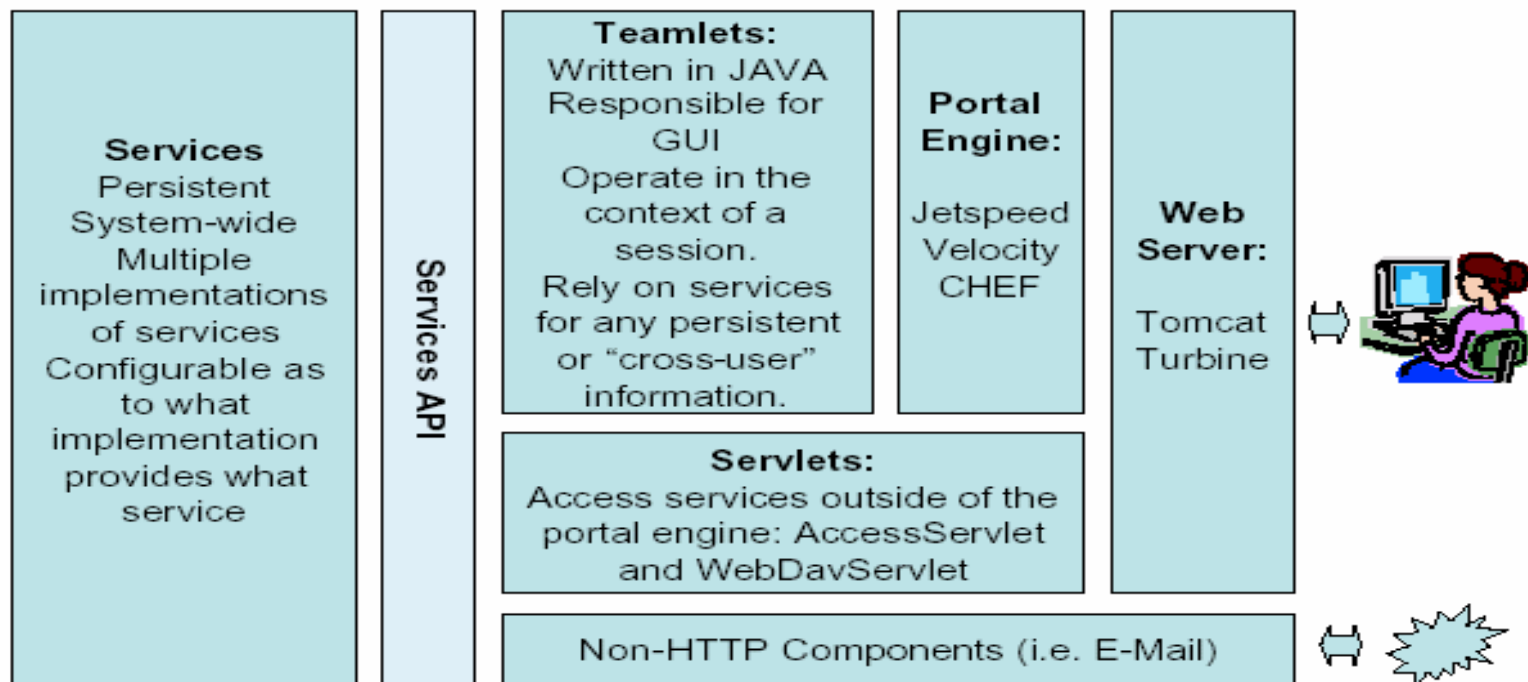
- Help
 - Topics available for each CHEF tool and the CHEF framework
- Calendar
 - Maintenance of events
- Tasks
 - Creation, progress report, milestone and assignments
- Survey
 - Creation, modification, assignment and completion

Functionalities (contd.)

- Record
 - Can store permanent record entries using digital signatures
- Presentations
 - Online presentation using graphics and simultaneous messaging
- Data Browsing
 - Monitoring and exploring visualizations of real-time scientific data channels
- Globus Grid Client
 - Access Globus Grid resources (still to be determined)

Architecture

CHEF Architecture



"will be like a tutorial on portal development using jetspeed"

Architecture (contd.)

- APIs
 - Services are defined by APIs
 - Calls are made to service component instance - dynamically bound and found by a service manager
 - Open Knowledge Initiative (OKI) servlet APIs merging with CHEF
 - CHEF tools built as standard portlets on Jetspeed

Architecture (contd.)

- Teamlets
 - Portlets specially aware of multi-user group work nature of task
 - Special features:
 - Resource access security
 - Automatic updating of displays
 - Multi-user safe
 - Presence
 - Notification
 - Are stateless - rely on services for long-term persistence and on Tomcat/Jetspeed for session persistence

Architecture (contd.)

- Access Servlet
 - Handles non-HTML (binary, etc.) requests
- Web Browsers
 - "thin client" mode - no other special software required
 - Some tools may require java applets or flash component plug-ins
- WebDAV Clients
 - To access certain resources in CHEF like, tool/content management service's folders and files

Architecture (contd.)

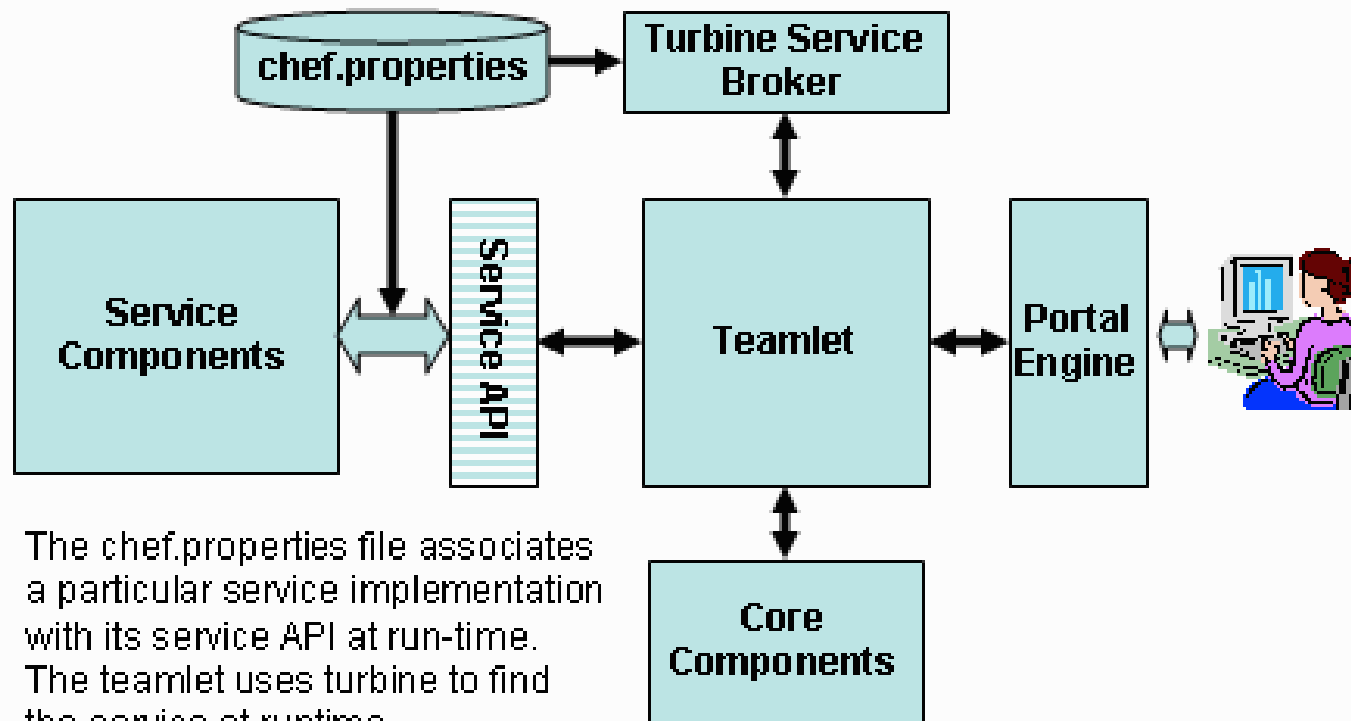
- Email Clients
 - Can be used for discussion messages and notification
- Configuration
 - Done in various "properties" files
- Sessions
 - Users have to authenticate to start a CHEF session which can later be used to store state
- Presence
 - List of active users - automatically updated per portal page
 - Can indicate "tool presence" also

CHEF Services

- Three elements with which users of a service (a tool or another service) interact:
 - Service API - defines properties and methods of service
 - Service Component - actual implementation of service
 - Core Components - objects exchanged between the services and teamlets
- Provide long term persistence

CHEF Services (contd.)

Services and Components



CHEF Services (contd.)

Service APIs (package `org.chefproject.service`)

[AnnouncementService](#)

AnnouncementService is a MessageService that handles Announcements: channels of messages posted by the authoritative few to be read by many, stored in the channel in a list in date order.

Core Objects (package `org.chefproject.core`)

[AnnouncementChannel](#)

AnnouncementChannel is a collection of AnnouncementMessages.

Teamlets

Announcements

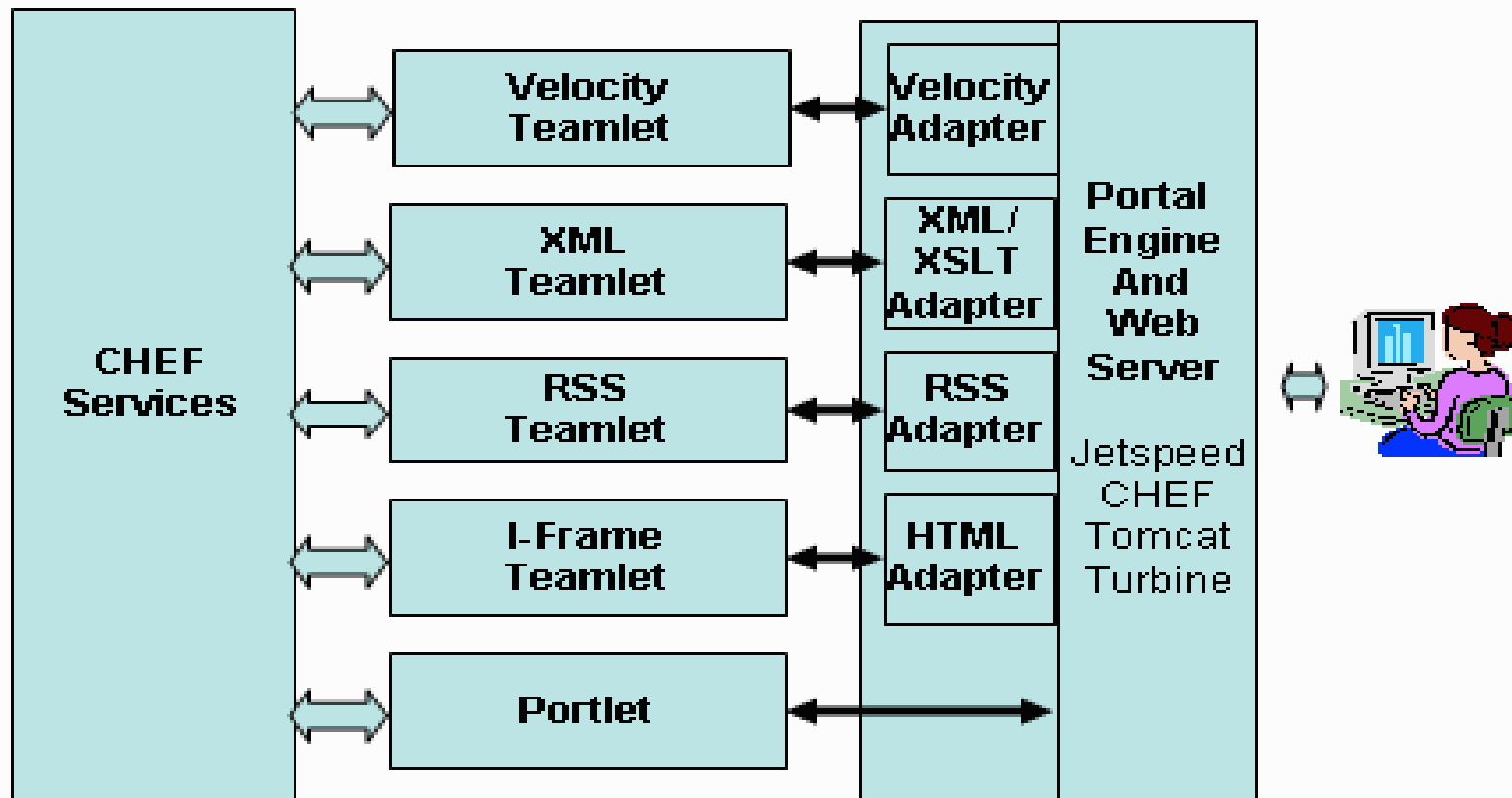
Post and read announcements relevant to your class or group.

Inside Tools

- Tool
 - Controls the user interface for some specific CHEF feature
 - Calls upon various services for information modeling, persistence and important application logic needs

Inside Tools (contd.)

Teamlet Implementation Choices

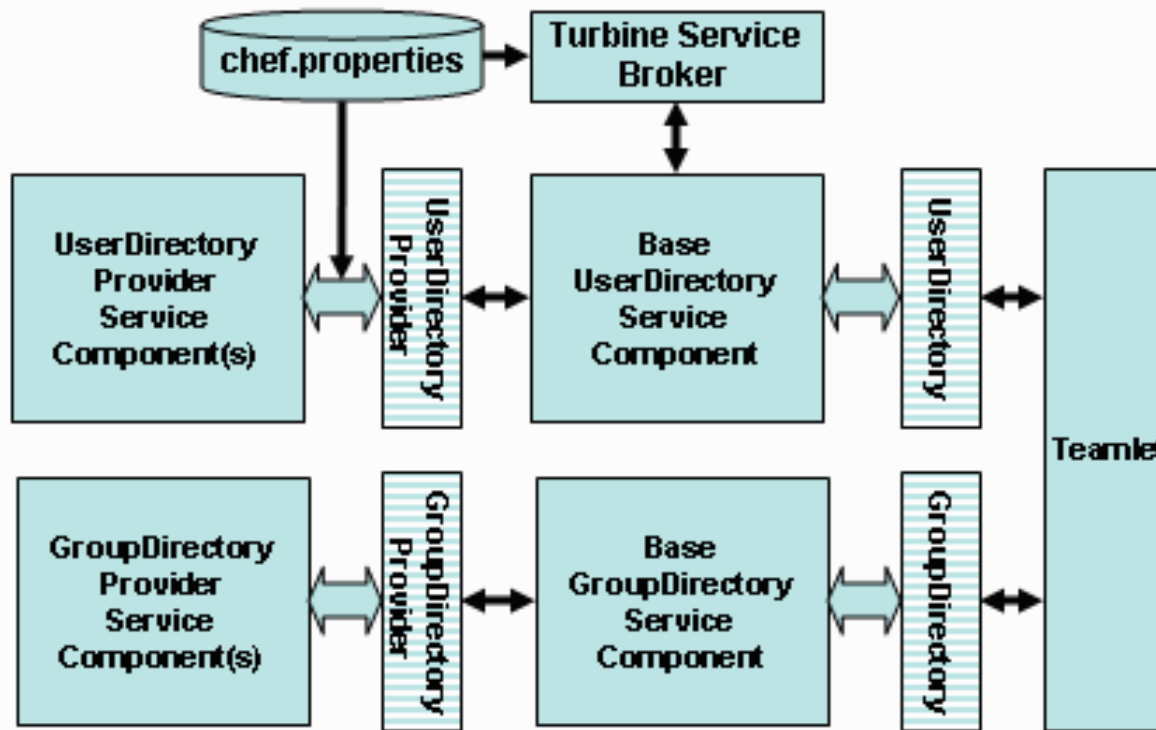


Inside Services

- Service components
 - Maintain information and provide important application logic for tools
 - Security and event tracking also handled uniformly
 - Implemented as Turbine Services - provides a way to configure, maintains a life cycle, brokers runtime requests
 - Can create different service components that satisfy the same service API
 - Example - UserDirectory Service for authentication can use different components, like Kerberos, or Certificates

Inside Services (contd.)

User and Group Service Providers



- Services can use other services or make use of external services

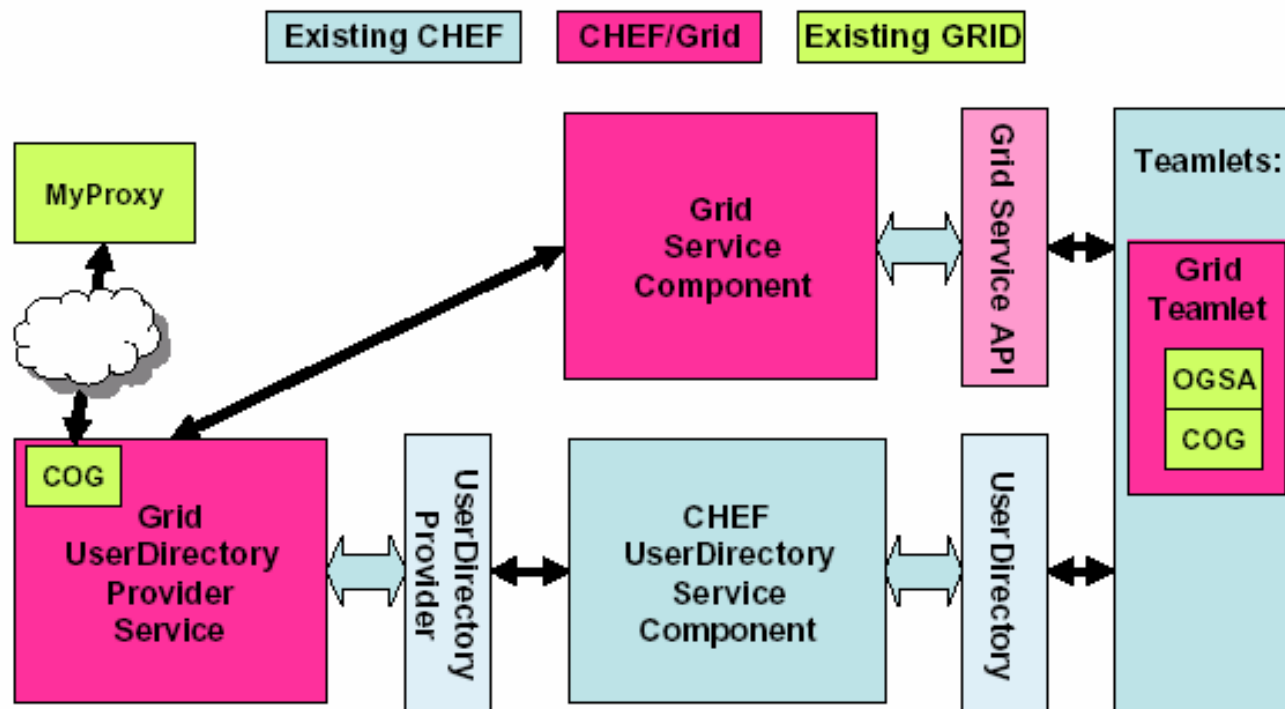
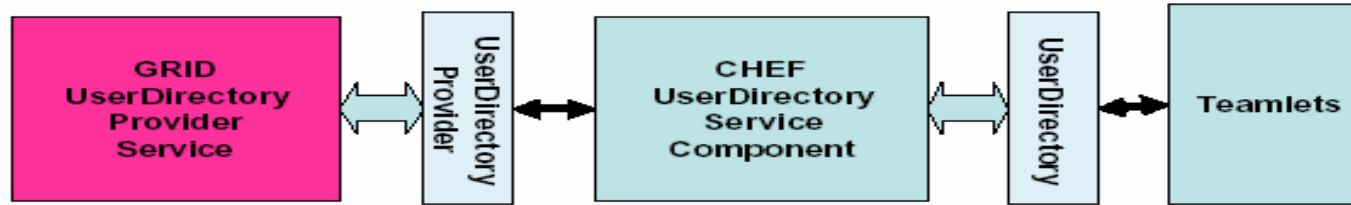
CHEF special needs

- Requires special integration to make certain features work:
 - Each request must be bracketed with calls to the security service to identify the security context (user of the request)
 - The courier must be part of the site level layout (top.vm)
 - The presence display must be part of the site level layout (or the menu, for example)

CHEF and Grid interface

- Primary modification to CHEF for Grid capabilities is:
 - Grid authentication/authorization and security model
- User information accessed by UserDirectoryService
 - Can delegate basic authentication operations to UserDirectoryProvider Service
 - Uses MyProxy - Grid enabled

CHEF and Grid interface (contd.)



CHEF and Grid interface (contd.)

- To specifically support Teamlets that make use of Grid Services using COG and/or OGSA - provide Grid Service Component
 - Can be used to retrieve certificate by the teamlet

CHEF and Grid interface (contd.)

The following is a simplified version of the code elements needed in the above Teamlet. The Teamlet uses Turbine to find the system-configured Grid Service and then uses the Grid Service API to retrieve and print the proxy information for the user who is logged into this session.

```
import org.chefproject.service.GridService;

import org.globus.security.*;
import org.globus.myproxy.*;

// Use turbine to find the CHEF Grid Service
GridService m_Grid = (GridService)TurbineServices
    .getInstance().getService(GridService.SERVICE_NAME);

// Retrieve the proxy for this user from the service
proxy = m_Grid.getCurrentUserGlobusProxy();

System.out.println("Cert:" + m_Grid.toString(proxy));
```

CHEF and Grid interface (contd.)

For a Teamlet to make use of OGSA services, it must import the additional classes from OGSA and provide the proxy to the OGSA tool using the appropriate interface:

```
import org.globus.ogsa.samples.NsdsServiceLocator;
import org.globus.ogsa.samples.NsdsPortType;
import java.net.URL

NsdsServiceLocator locator = new NsdsServiceLocator();
NsdsPortType nsds = locator.getNsdsPort(new URL(uri));

// Provide the credentials
((Stub)nsds)._setProperty(GSIConstants.GSI_CREDENTIALS, proxy);
String channel = nsds.createChannel();
```

CHEF and Grid interface (contd.)

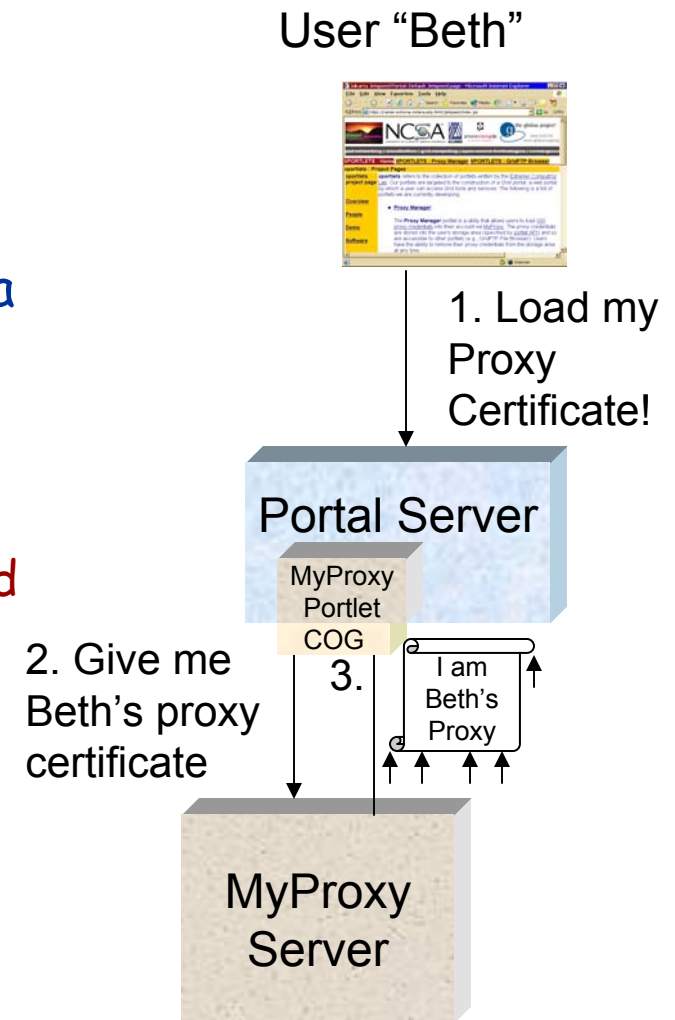
- Blending *Grid/MyProxy* and *CHEF* Security
 - *MyProxy* using signed X.509 certificates
 - Now, moving to *GT3* security
 - User ID and password in *CHEF* align with *MyProxy* ID and passphrase
 - Then it can retrieve the proxy using the subject string in the certificate
 - Authentication is denied if :
 - A proxy cannot be retrieved, or
 - The subject field of the retrieved proxy does not match subject field stored in *CHEF*
- Need not necessarily have *Grid* accounts

Hour 3. Interacting with Web & Grid Services

Security certificate management
Interacting with Generic Web Services
Browsing Grid Services

Gaining Proxies for Users

- The MyProxy Manager
 - The user contacts the portal server and asks it to do "grid" things on behalf of the user.
 - To make this possible the server needs a "Proxy Certificate"
 - The user has previously stored a proxy cert in a secure MyProxy Server stored with a temporary password.
 - User give the portal server the password and the portal server contacts the proxy server and loads the proxy.
 - The portal server will hold the proxy for the user for a "short amount of time" in the user's session state.



Interacting with MyProxy Server

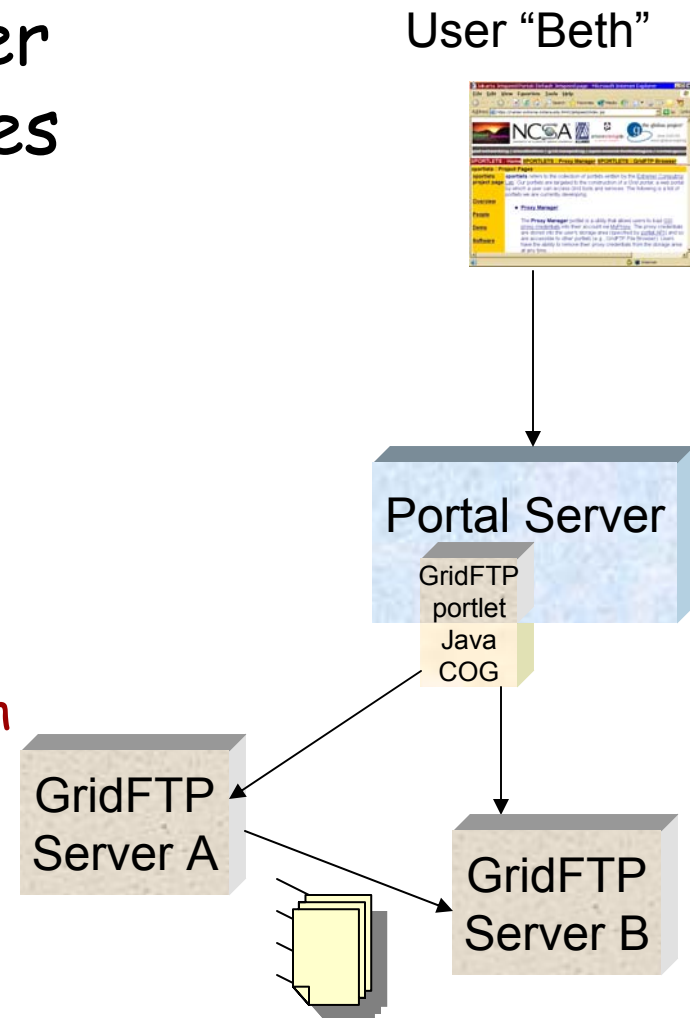
- Through the Java COG Kit.
- Code for your Action Class:

```
GSSCredential proxy = null;
int port = Integer.parseInt(params.port);
MyProxy myproxy = new MyProxy( params.hostname, port );
int lifetime = Integer.parseInt( params.lifetime ) *
                SECS_PER_HOUR;
proxy = myproxy.get(params.username, params.password,
                    lifetime);
proxies.put( proxy, true or false ); // do or don't store
```

- Another portlet can call "proxies.get()" // permanently

Portal Operations: File Management

- Grid FTP portlet- Allow User to manage remote file spaces
 - Uses stored proxy for authentication
 - Upload and download files
 - Third party file transfer
 - Request that GridFTP server A send a file to GridFTP server B
 - Does not involve traffic through portal server



Other Service Portlets

- Resource Verification
 - Verify that the resources and services that I use still recognize me.
- LDAP Browser
 - Browse the contents of an LDAP server
- Grid Job Launch & Monitor
 - Specify a job you would like to launch on some remote resource and launch it.
 - Monitor the progress of the execution

Collaboration: news and mail

- A portlet for accessing a topic based news and notes archive
 - Interfaces to NaradaBroker and Xmessages
 - Interface to email
 - Based on Java JMS standard API with XML content.

Web Services

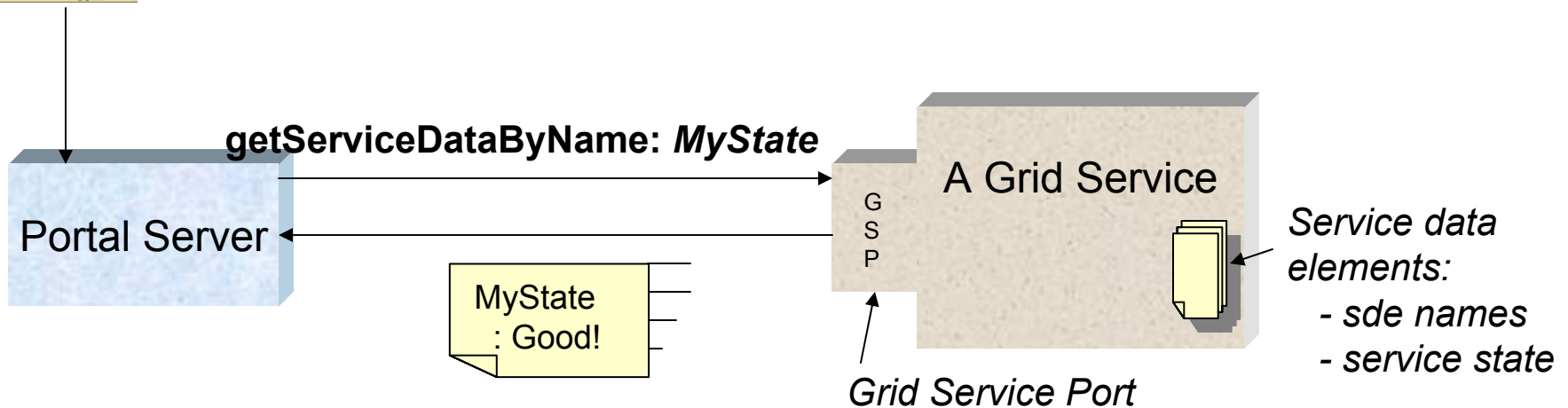
- A distributed computing infrastructure based on XML and some basic Web Ideas
 - A web service is a server process that exposes typed ports to the network
 - Described by the Web Services Definition Language. An XML document that contains
 - The type of messages the service understands and the types of responses and exceptions it returns
 - The associated "methods" are bound together as "port types"
 - Port types are bound to protocols and endpoints as "ports"
 - A WSDL document completely defines a service and how to access it.

Grid Services

- As defined by the Open Grid Service Infrastructure
 - A web service that also adheres to some simple behavior patterns and implements the GridService Port Type and other ports it needs.

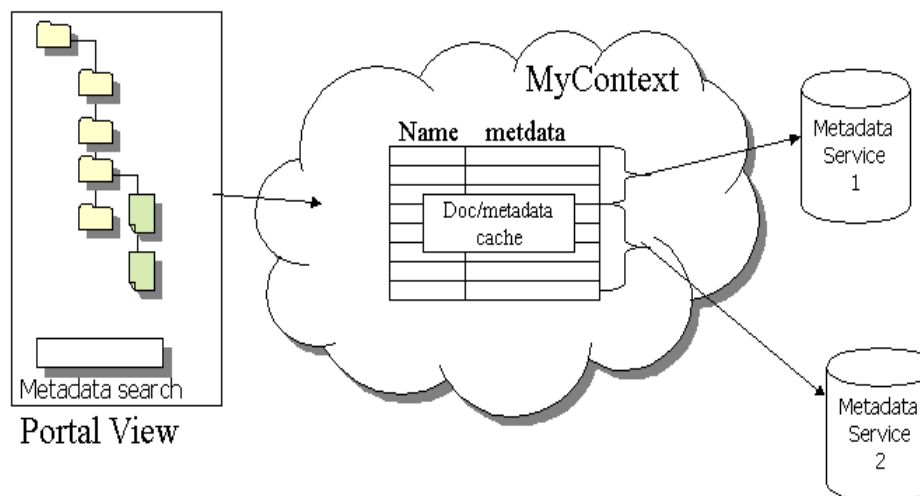
Grid Services Browser

- OGSII defines a Grid Service as a web service that supports the Grid Service port.
 - Grid Service Port provides access to the "service data elements" of the service
 - Includes service metadata and state information
 - Encoded as XML document
 - Any service's state can be examined by the portal.



XDirectory: A Grid Context Service

- XDirectory is itself a Web Service that is access by the portal.
 - An index over a relational database
 - Each node is either a "directory node" or a leaf.
 - Leaf nodes are xml elements which contain metadata as well as html annotations.

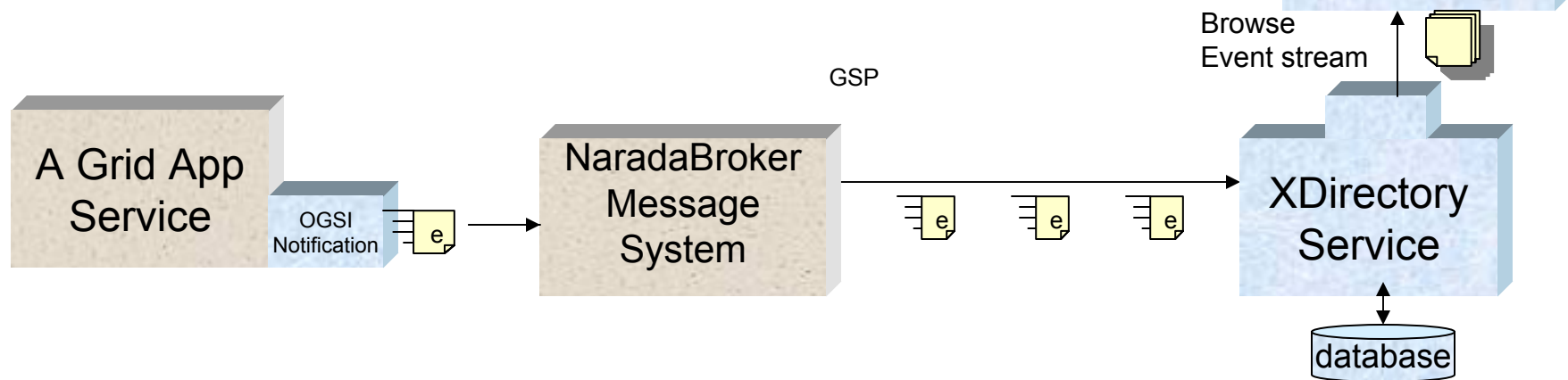


The Dual Roles of an Leaf Node

- The metadata
 - a reference to something else
 - An entry in a meta data directory defined by a specific query (you find x by this query: ...)
 - History information
- The html data
 - An annotation. User can edit it.
 - A reminder of what this node is about.
 - A view: for example a bit of graphics
 - A link to another service that has a web interface.
- A user, or her agent can insert a new node or directory.
 - For example, log events from a program execution can be inserted into the directory for the user to study later.

Events and Messages

- Messaging and Notification are critical
 - Use NaradaBrokering and "Xmessages"
 - Narada provides for ubiquitous delivery
 - More on Narada later.
 - Xml based message elements.
 - Easy to transform events into directory nodes.
 - Example: an event records the location of an application output file. That can be stored in the directory and discovered later.



Application Factories

- An App Factory is a persistent web/grid service I can use to launch Apps on my behalf.
 - Each application instance can be realized as "private" service
 - This service may execute the workflow of my job.
 - The new service instance can register its interface (WSDL) with the xdirectory so the user has access to its control interface
 - And/Or it may post events to the event service which are eventually stored in the channel.

QuickTime™ and a TIFF (Uncompressed) decompressor are needed to see thi:

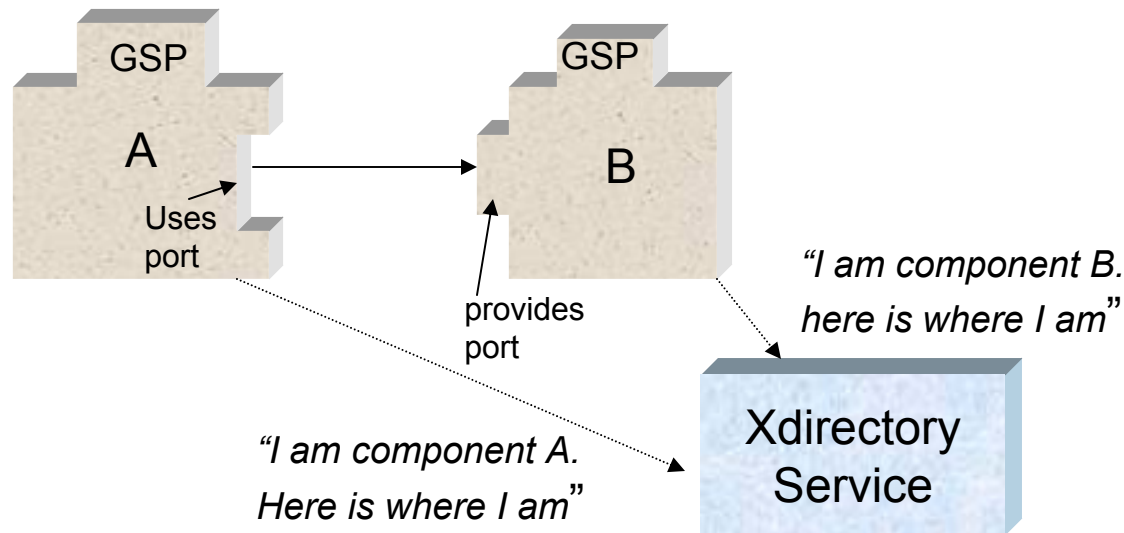
B

Future Work: Semantic Grids

- Investigating Semantic Web
 - To be explored in the context of Earth Sciences.
 - This community already has established a sizable set of metadata ontologies and services
 - We will explore the problem of doing intelligent service discovery.

XCAT Grid Services

- Grid services that are also CCA components
 - Have "uses" and "provides" ports.
 - A uses port on one component may be connected to a provides port on another.
 - XCAT service components register themselves with LDAP server



Xydra

A form generator for Web Services

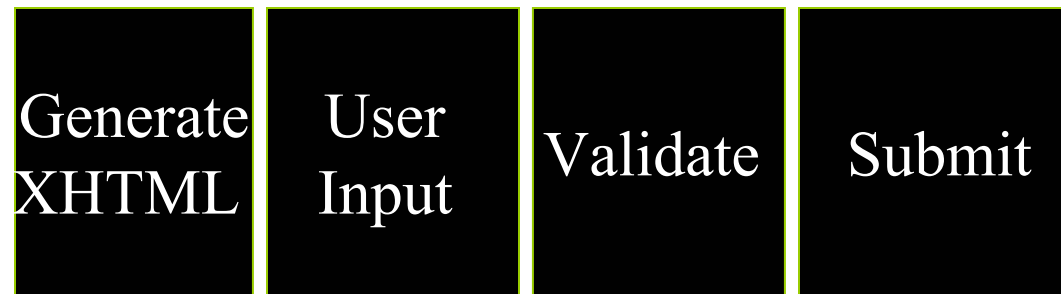
Octav Chipara

Aleksander Slominski

How to Invoke a Web Service?

- Designed to be invoked by machines.
 - Not humans
- But many can have easy-to-use human interfaces
 - To perform simple queries of a web service
- WSRP is one approach
 - The service provides markup information to the portal server which composes this into a portlet that can be used to get user responses and then call the remote service.
 - More on WSRP later.
- WSDL provides a description of each port of the service, so
 - Why not translate the ports directly into XHTML forms that can be invoked by the user?

“Filling a form”



Generation of XHTML

- Input:
 - XML Schema and/or WSDL
- Output
 - XHTML
 - A model describing the XML Schema/WSDL and allowed XML instances

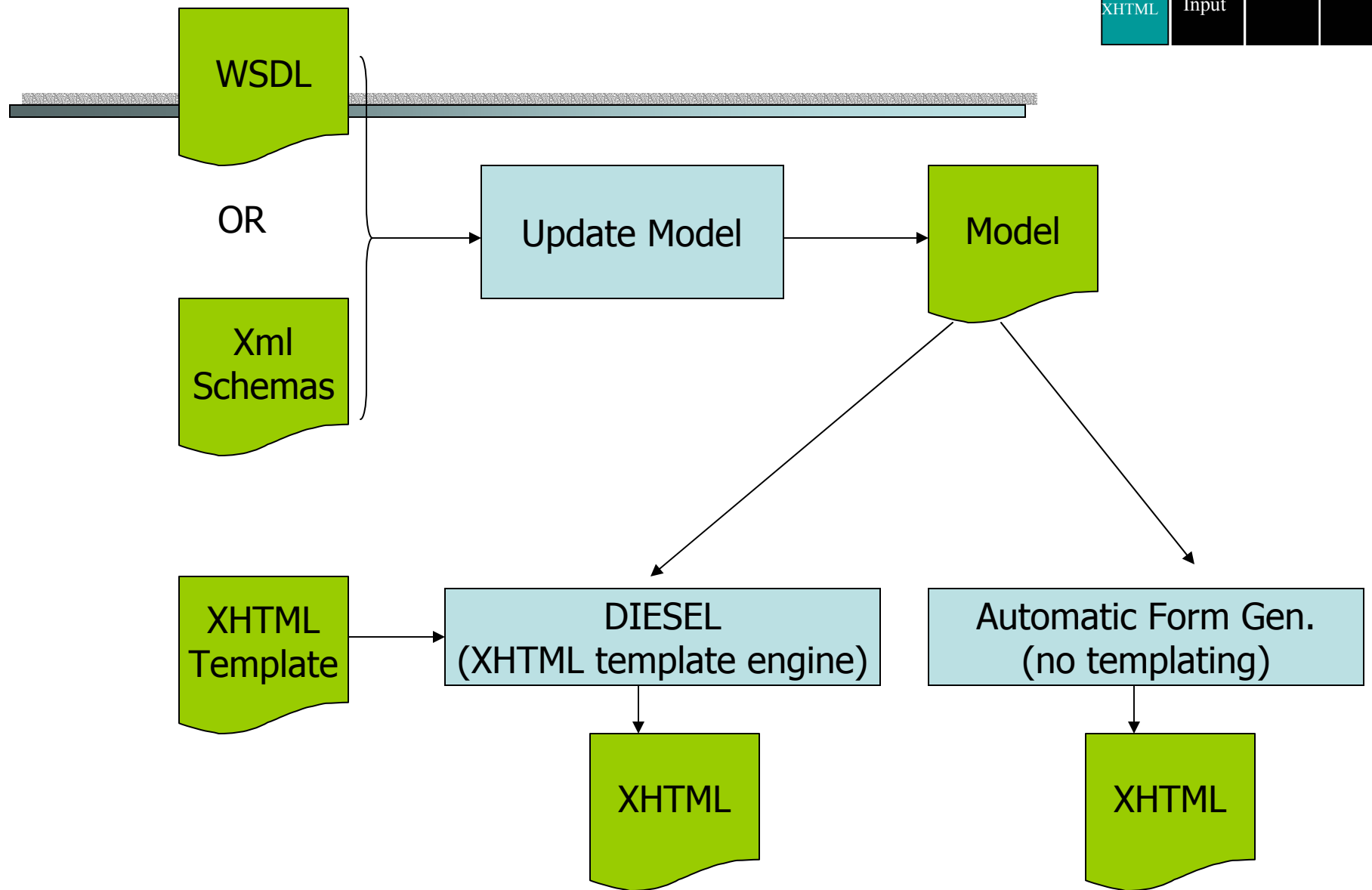
Model requirements

- Capture the constraints from XML Schema
 - Type constraints (mainly)
 - Additional constraints defined in XML Schema (e.g. maxOccurs/minOccurs)
- Capture constraints that cannot be expressed in XML Schema (relationship constraints)
- The model can be translated in name-pair values
 - This is how XHTML forms work
- Store enough information to be able generate an xml instance that is valid relative to the defined constraints

Type and Relationship Constraints

- Type constrains
 - Equivalent to XML Schema
 - a phone number is a string following the pattern XXX-XXX-XXXX
 - a state is a 2 letter string having as value of the strings IN, NY, NE, etc.
- Relationship constrains
 - Examples:
 - Air-line tickets: arrival time > departure time
 - Health-care: if gender=male then cannot fill pregnant section
 - Beyond XML Schema
 - Require domain knowledge
 - Imposed during the filling of the form

Generate XHTML	User Input	Validate	Submit
----------------	------------	----------	--------



Browse WSDL

- The first step is to get the
 - Service
 - Port
 - and Operationthat will be invoked
- From the operation definition we figure out how the form should look based on the form

WSDL – operation

- <operation name="SendMSN">
- <input message="s0:SendMSNSoapIn"/>
- <output message="s0:SendMSNSoapOut"/>
- </operation>
-
- <message name="SendMSNSoapIn">
- <part name="parameters" element="s0:SendMSN"/>
- </message>
- <message name="SendMSNSoapOut">
- <part name="parameters" element="s0:SendMSNResponse"/>
- </message>

WSDL - types

```

<types>
  <s:schema elementFormDefault="qualified" targetNamespace="http://www.bindingpoint.com/ws/imalert/">
    <s:element name="SendMSN">
      <s:complexType>
        <s:sequence>
          <s:element name="FromName" type="s:string"/>
          <s:element name="ToUserID" type="s:string"/>
          <s:element name="Message" type="s:string"/>
        </s:sequence>
      </s:complexType>
    </s:element>
    <s:element name="SendMSNResponse">
      <s:complexType>
        <s:sequence>
          <s:element minOccurs="1" maxOccurs="1" name="SendMSNResult" type="s0:SendResult"/>
        </s:sequence>
      </s:complexType>
    </s:element>
  ..

```

User Input

- Uses HTML Browser for UI interactions
 - Renders the HTML form
 - Gathers user input as name-value pairs
 - Send them as HTTP Post request
- HTTP Post is processed by servlet
- Servlet uses the name-values pairs to update the model

Validation

- Works only with OntoBrew
 - OntoBrew is an implementation of the Model that uses ontologies
- The constraints are captured in the KB and a constraint engine is used to validate them

More about OntoBrew

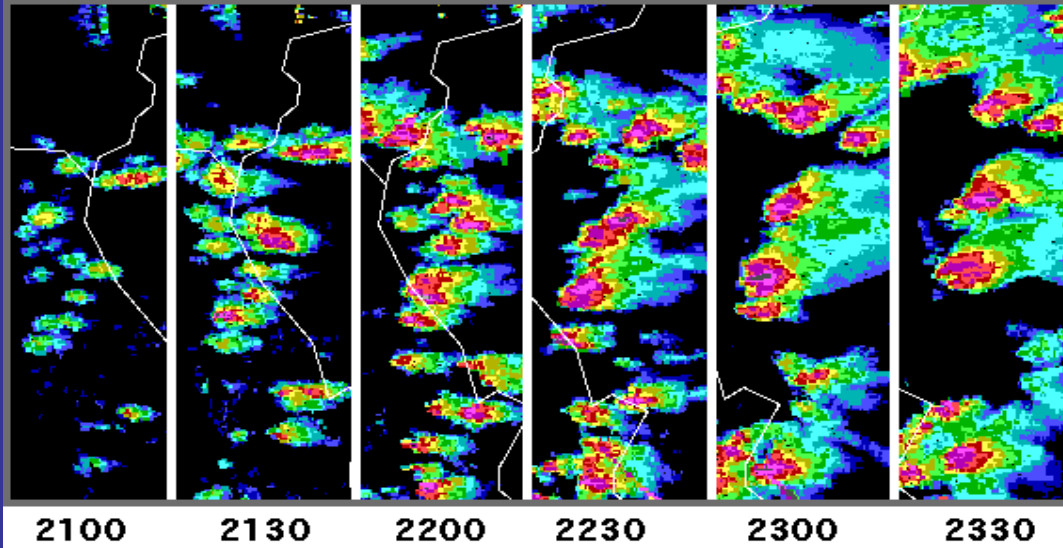
- The model is implemented by ontologies
- The model describes
 - Web services (WSDL)
 - XML Schemas
 - XML Instances
- The model allows content integration from using multiple services based on schemas
- Allows the integration with other ontology tools

Managing Grid Workflows

An example and a simple Grid
Web service based on Ant.

Composite Radar Reflectivity 19 April 1996

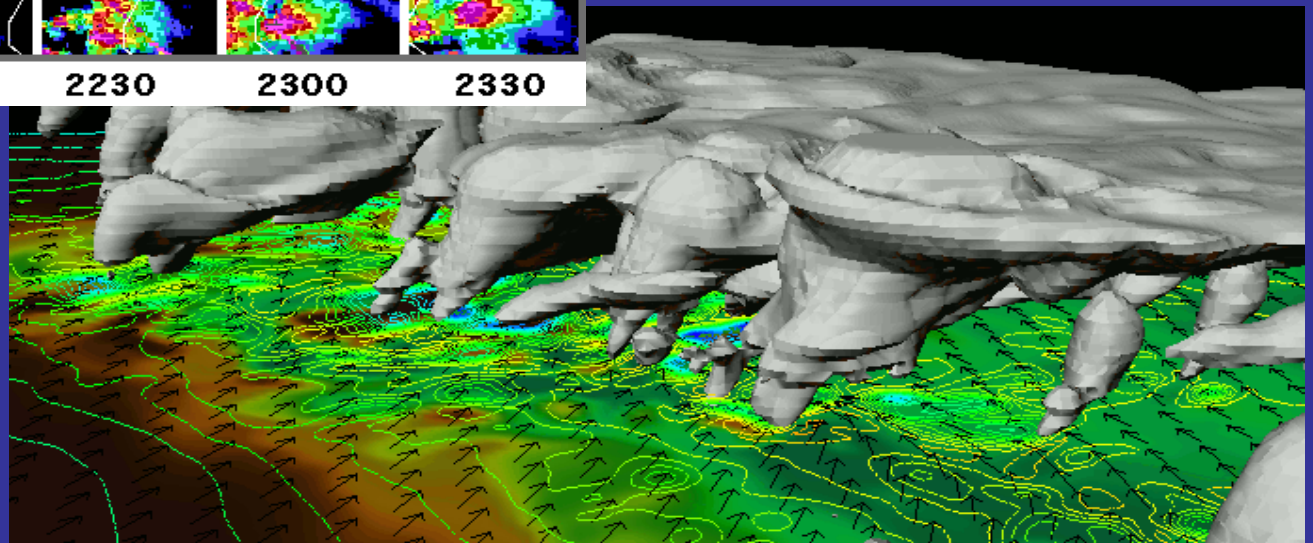
All times UTC



← Real World

1 km MM5, real data initialization

Model →



Robert Wilhelmson, Brian Jewett

Grid Summer School: Grid Portals

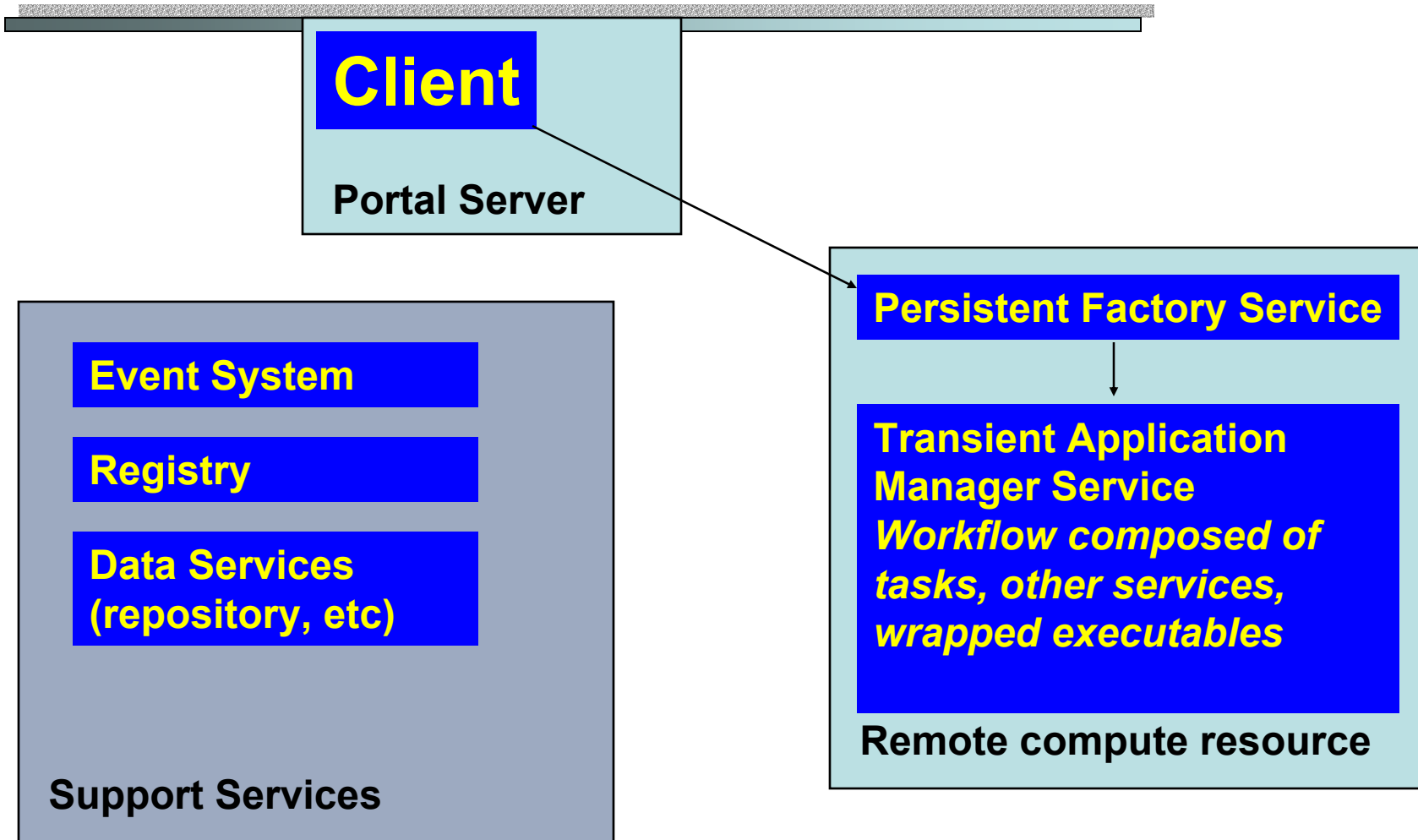
Severe Storm Simulation Challenges

- *Make & manage many simulations*
 - Define and configure the parameter space
 - Create/archive job configuration metadata
 - Seamlessly (?) submit, execute & monitor all jobs without manually working through (around? despite?) the batch system
 - Treat post-analysis, visualization, feature detection and assessment as an integral part of the job workflow
 - Save metadata associated with results & analysis

More Challenges

- Keep track of the work
- Effectively interrogate the data
 - Metadata tied to and permanently associated with the data
 - Searching by input & output characteristics
 - Improved data organization & retrieval

The Model



NCSA Portal Interface Tools

The screenshot displays the NCSA Grid Desktop interface with several tool windows:

- GIF Viewer:** Shows a meteorological plot titled "T(m/w, color) and horizontal winds (m/s)". The plot displays vertical velocity (m/s) with a color scale from -4 to 8. The axes range from 0 to 80. A reference vector is shown at the bottom right.
- WRF Data:** A grid-based visualization with red dots on a black background, representing data points.
- User Notebook:** A central panel with sections for Information, Metadata, Timestamps, Properties, Runs, and Documents. It includes fields for Name (WRF-SC02), ID (35bf727317f45b7c000000f17e0004b9), and Author. It also features buttons for Add, Remove, Edit, Import, New, and View.
- Grid Broker:** A table showing the status of grid operations.

Type	Start	Duration	Status	Progress
GridFTP	Nov 19, 200...	Unknown	Finished	1 of 1

An “Extended” ANT Workflow Interpreter

Objective

Extend the Ant Engine to include task execution in which data (input/output) is shared between tasks and which is Turing-complete.

Method

Incorporate at least the following interpreter-like functionality:

ENVIRONMENT	contains both a flat, globally scoped environment and a stack
REFERENCE	points into the ENVIRONMENT
CONTAINER	mediates between Task and ENVIRONMENT through the WRAPPER = <i>Sequential, Parallel, Synchronized, Loop, Condition Block</i>
ASSIGN	side-effects the ENVIRONMENT
WRAPPER	wraps a Task with its PARAMETERS and RETURN VALUES
PARAMETER	provides either a literal or a REFERENCE for the value of a Task mutator method
RETURN VALUE	tells which Task accessors should be called to return values to the ENVIRONMENT

Using the Workflow Interpreter

Built into the “Application Manager” [Workflow Engine] Transient Service is a simple call to Ant.Main to run a workflow script.

- This script can be a normally written Ant build file or can make use of our extensions.
- The extended workflow language itself further allows:
 1. embedding of normal Ant Tasks;
 2. simple wrapping of Ant Tasks;
 3. use of descriptors for complex wrapping of Ant Tasks.
- The package will also provide special tasks for:
 1. managing events;
 2. executing native code, tracking the creation/updating of files and handling the process’s output and error streams;
 3. GridFtp;
 4. specialized services (like our registry or tuple space);
 5. general web–service lookup and port invocation.

Another Approach From Argonne

- Gregor von Lazewski's
GRID ANT tools
 - Capable of executing
XML Ant workflow specs
and animating the
execution process

QuickTime™ and a TIFF (Uncompressed) decompressor are needed to see this picture.

QuickTime™ and a TIFF (Uncompressed) decompressor are needed to see this picture.

Hour 4. A Web Services Framework for Collaboration and Videoconferencing

**Geoffrey Fox, Wenjun Wu
Ahmet Uyar, Hasan Bulut**

Abstract

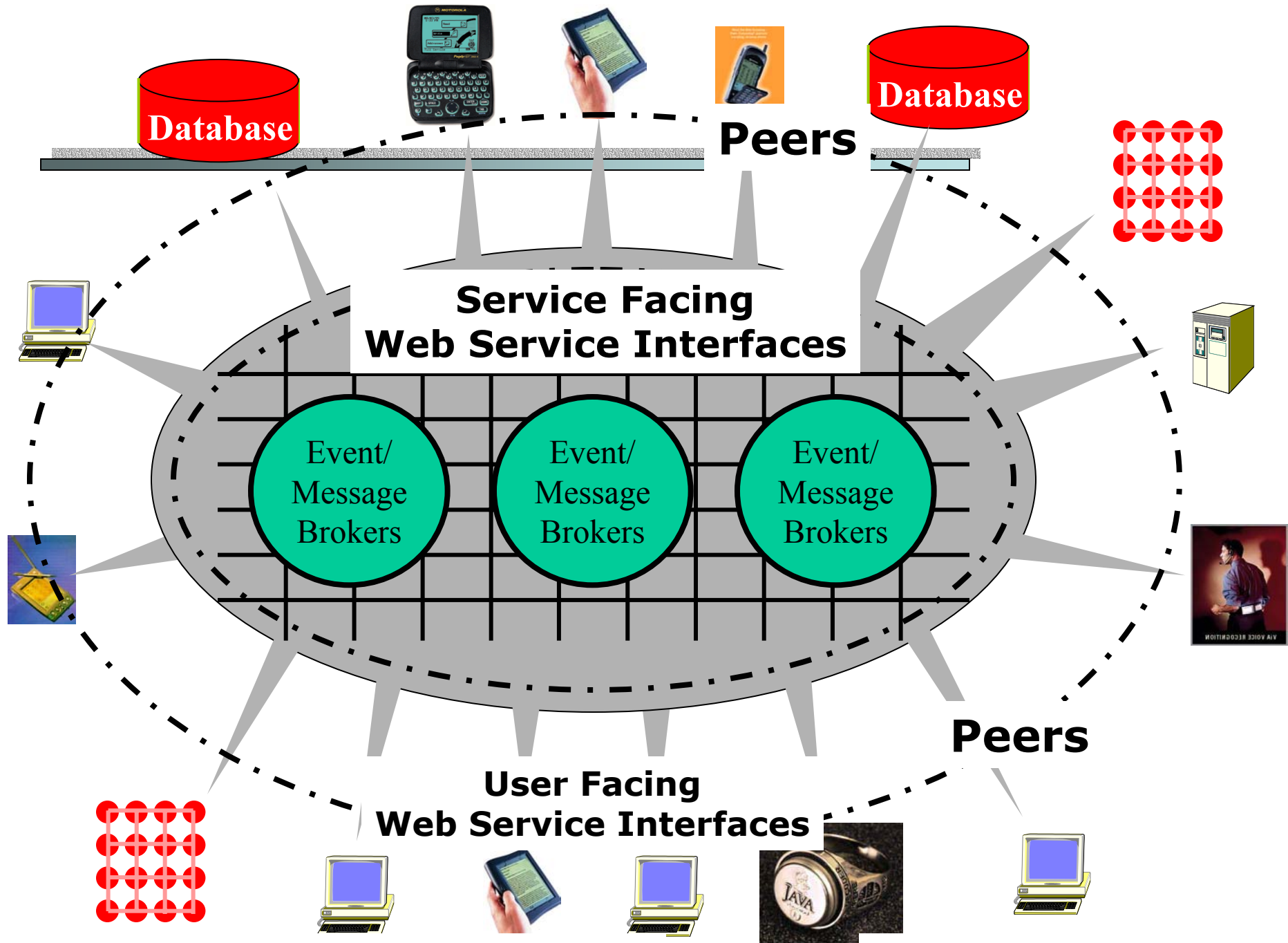
-
- We define such a common, interoperable framework called XGSP (XML based General Session Protocol)
 - Web services technology for creating and controlling videoconferences
 - We developed a common dynamic messaging environment (NaradaBrokering) for the collaboration applications
 - Based on the web-services framework and NaradaBrokering messaging environment, we are developing Global Multimedia Collaboration System (Global-MMCS)
 - Integrates various services including videoconference, instant messaging and streaming, and supports multiple videoconferencing technologies and heterogeneous collaboration environment.

Collaboration and Web Services

- Collaboration has
 - A) Mechanism to set up members (people, devices) of a “collaborative sessions”
 - B) Shared generic tools such as text chat, white boards, audio-video conferencing
 - C) Shared applications such as Web Pages, PowerPoint, Visualization, maps, (medical) instruments
- b) and c) are “just shared objects” where objects could be Web Services but rarely are at moment
 - We can port objects to Web Services and build a general approach for making Web services collaborative
- a) is a “Service” which is set up in many different ways (H323 SIP JXTA are standards supported by multiple implementations) – we should make it a WS

Shared Event Collaboration

-
- Collaboration involves sharing resources and synchronous collaboration involves coordinating a common view of a resource between multiple clients
 - All collaboration is about sharing some sort of event
 - Audio/Video conferencing shares events specifying in compressed form audio or video
 - Shared display shares events corresponding to change in pixels of a frame buffer
 - Instant Messengers share updates to text message streams
 - Microsoft events for shared PowerPoint (file replicated between clients)
 - Using Web services makes universal as exposes updates of all kinds as messages
 - Group communication service is needed for the delivery of the update events
 - Using Event Messaging middleware makes messaging universal



A democratic organization : *Grid Portals* Peer to Peer Grid

Solutions to Problems from current collaboration systems

- Networks were unreliable and firewalls are a problem
 - Not a lot of progress with QoS at network level
 - Some QoS problems are due to different collaboration streams interfering
 - *Use application level QoS with highly robust managed messaging*
- Many different standards H323, SIP, Access Grid, T120 ...
 - *Unify as single XML standard*
 - *Make the conference control services into Web Services*
- Very hard to customize each application in “shared state event model”
 - *Offer shared display*
 - *Convert Applications to Web Services*
- Inconvenient to customize user interfaces
 - *Use portlet technology supporting desktop and PDA clients*

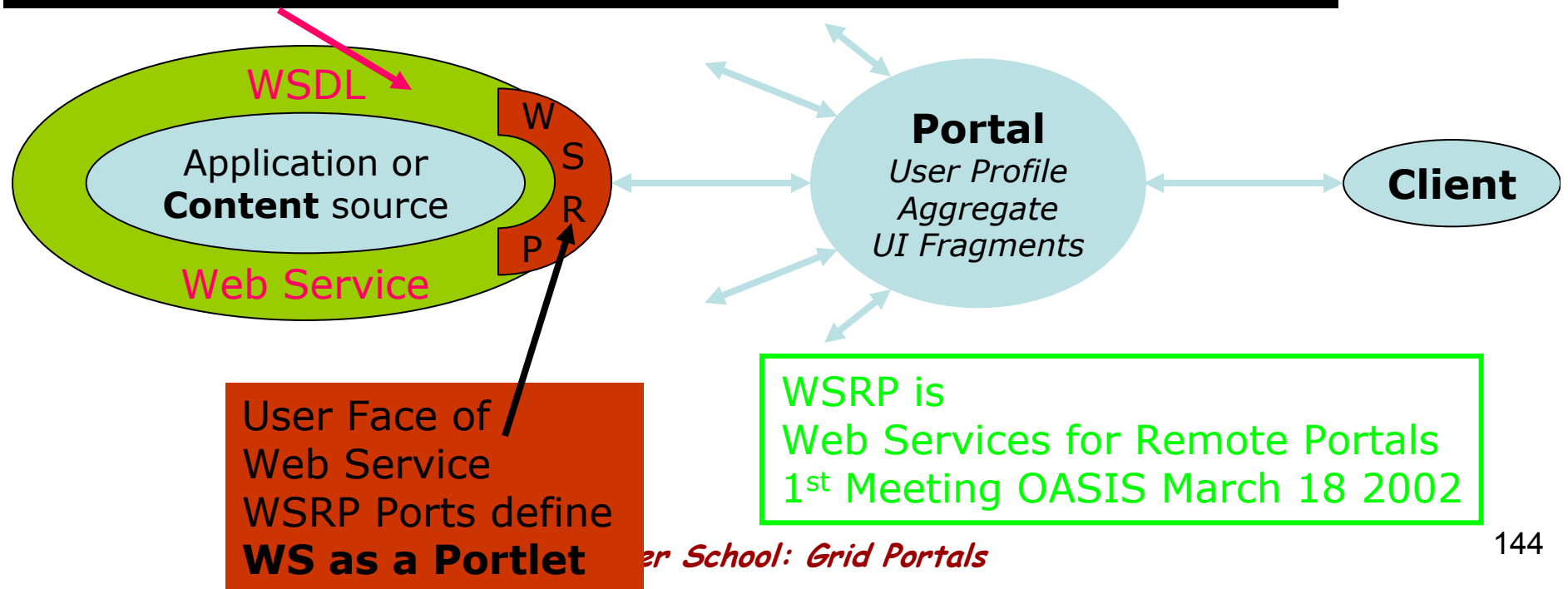
Portals and Web Services

- Web Services allow us to build a component model for resources.
- Each resource naturally has a user interface (which might be customized for user)
- Web Service <--> Portlet
- Natural to use a component model for portal building displayed web page from collection of portlets
 - So can customize each portlet and customize which portlets you want

WSRP Structure of a Portlet

- Each Web Service naturally has a user interface specified as “just another port”
- This gives each Web Service a Portlet view specified (in XML as always) by WSRP (Web services for Remote Portals)
- So component model for resources “automatically” gives a component model for user interfaces
 - When you build your application, you define portlet at same time

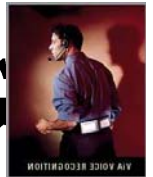
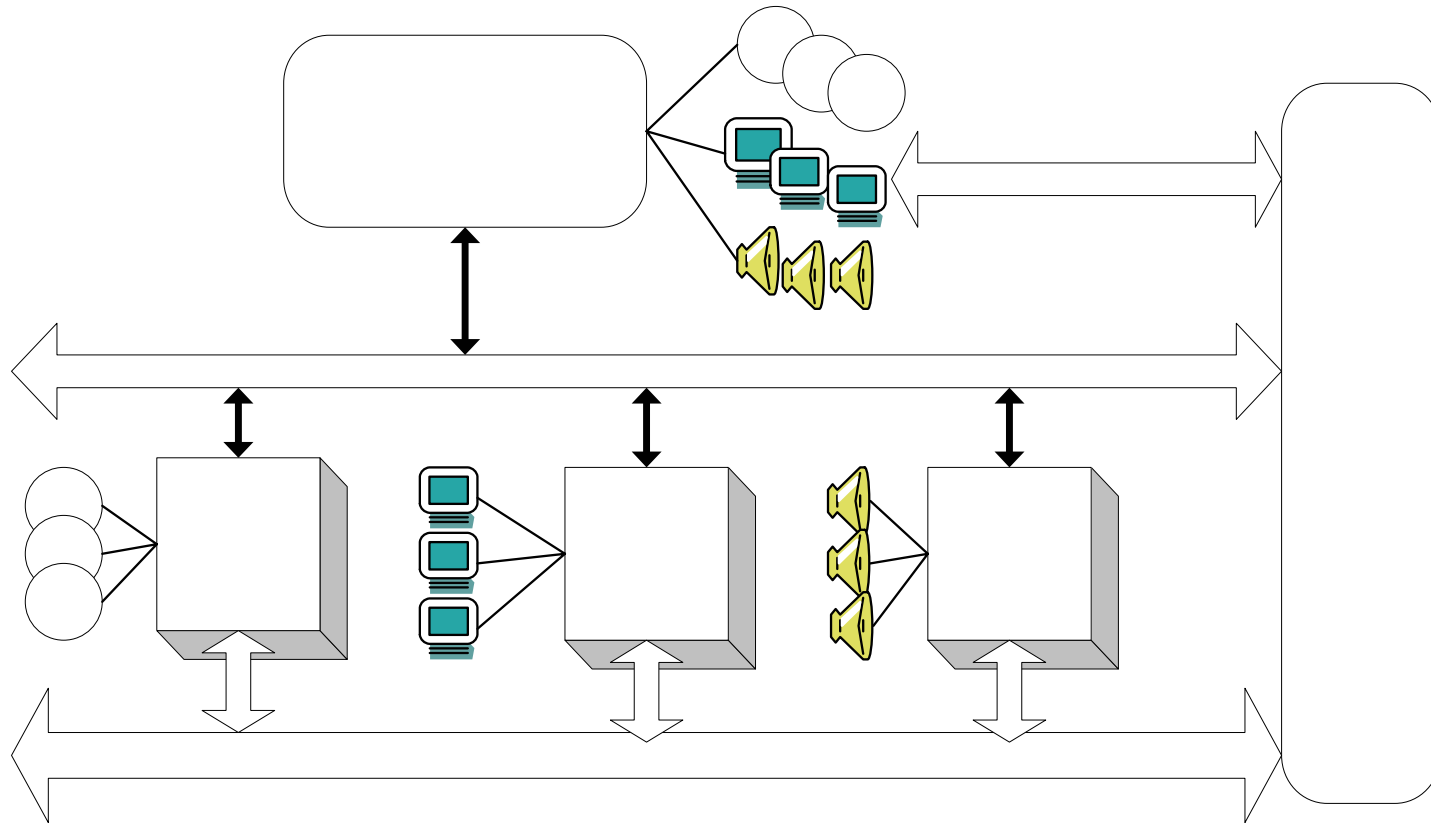
General Application Ports Interface with other Web Services



XGSP Framework

- To integrate heterogeneous systems into one collaboration system, we need to reach the following goals:
- (1) Different kinds of application endpoints should join leave in the same collaboration session.
- (2) Different providers for multipoint A/V and data collaboration should be connected together to build unified A/V and data multipoint channels.
- (3) A common user interface should be present for all the collaboration participants using different A/V and data application endpoints.

XGSP Conference Control Framework



Grid summer School: Grid Portals

XG Co
Manag

XGSP Conference Control Framework

- User session management
 - User session management supports user sign-in, user create/terminate/join/leave/invite-into XGSP sessions.
- Application Session Management
 - XGSP application session management provides the services to A/V and data application endpoints and communities, controlling multipoint A/V RTP and data channels.
- Floor Control
 - Floor control manages the access to shared collaboration resources.

XGSP Application Session Management

- XGSP signaling protocol for
 - H.323 signaling protocols (H.225, H.245)
 - SIP signaling protocol (Invite, Bye Message)
 - Access Grid (Unicast VIC & RAT)

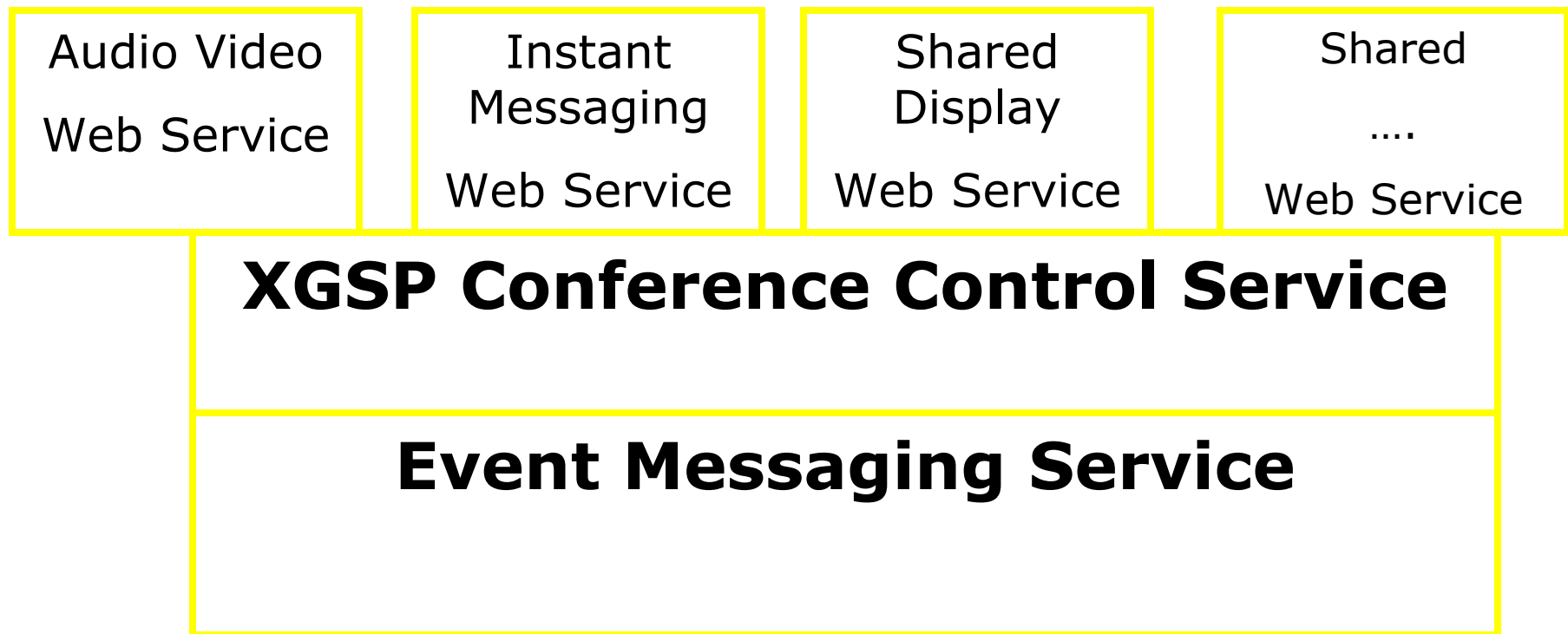
*Join XGSP Session, Leave XGSP Session,
Invite into XGSP Session, Expel from XGSP Session*
- Activate the XGSP session
 - *the XGSP session server will link all the “rooms” in the session together by connecting multipoint A/V and data channels from different communities to the XGSP A/V Media and Data Channel Services.*

Link/Disconnect XGSP SubSession

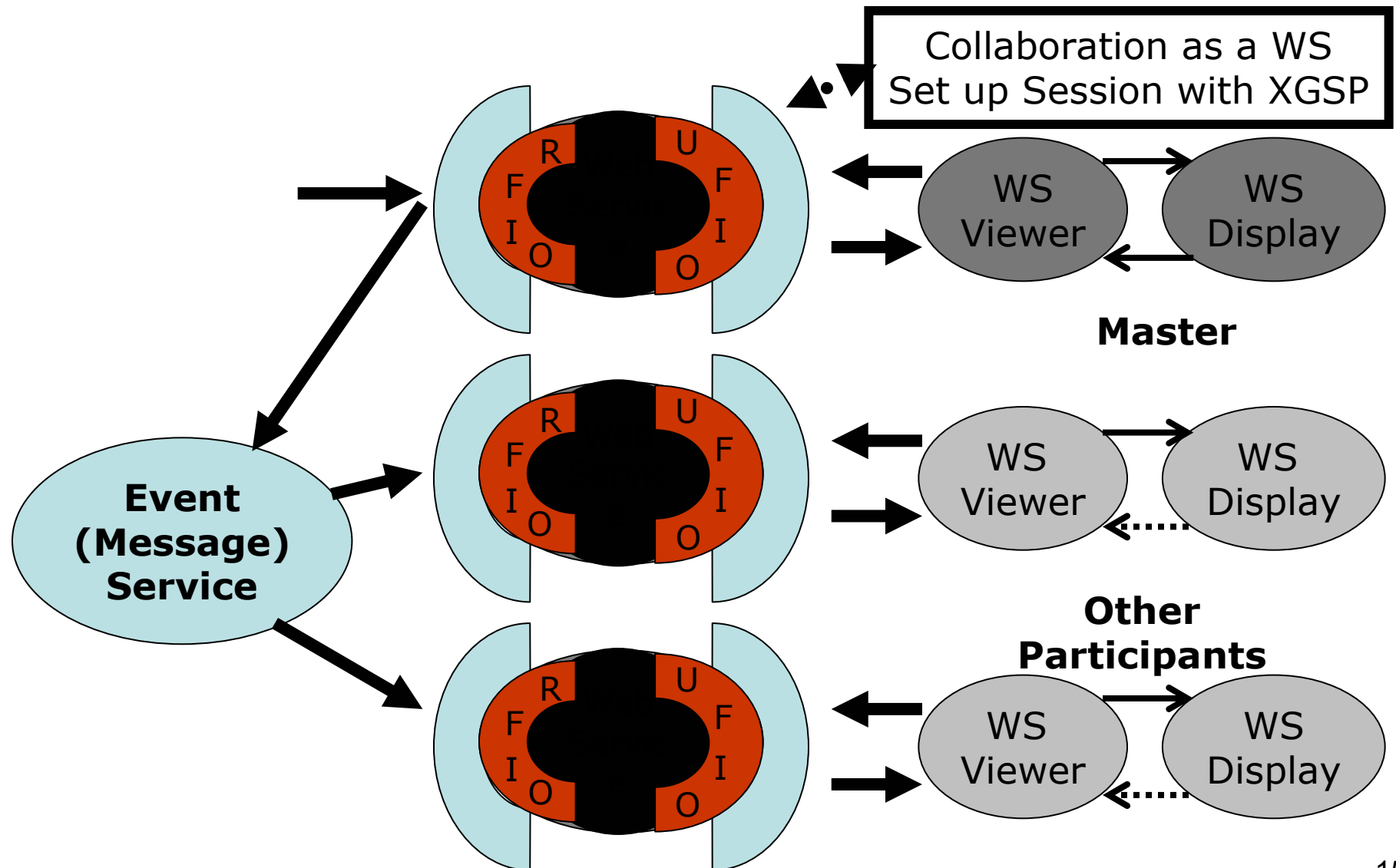
XGSP Floor Control

- XGSP should provide:
 - Floor control primitives, including: *request floor, release floor, grant floor, cancel floor, remove floor request*
 - mediator-controlled floor control: to support the mediator control policy
 - Collaboration applications have to define their own roles in the XGSP registration so that the mediator could assign the role of the application to each user.
for example, a shared PowerPoint application should define master/slave role.

Collaboration Web services System



Shared Input Port (Replicated WS) Collaboration



Collaboration service → portlets

- Portlets are collaboration components which implement “RFIO” and “UFIO” web services interface
- A portlet provides a presentation logic for user interface
- A portlet can be downloaded and instanced when a user joins the conference
- Each portlet provides client-side services to the XGSP portal for application session management and floor control.

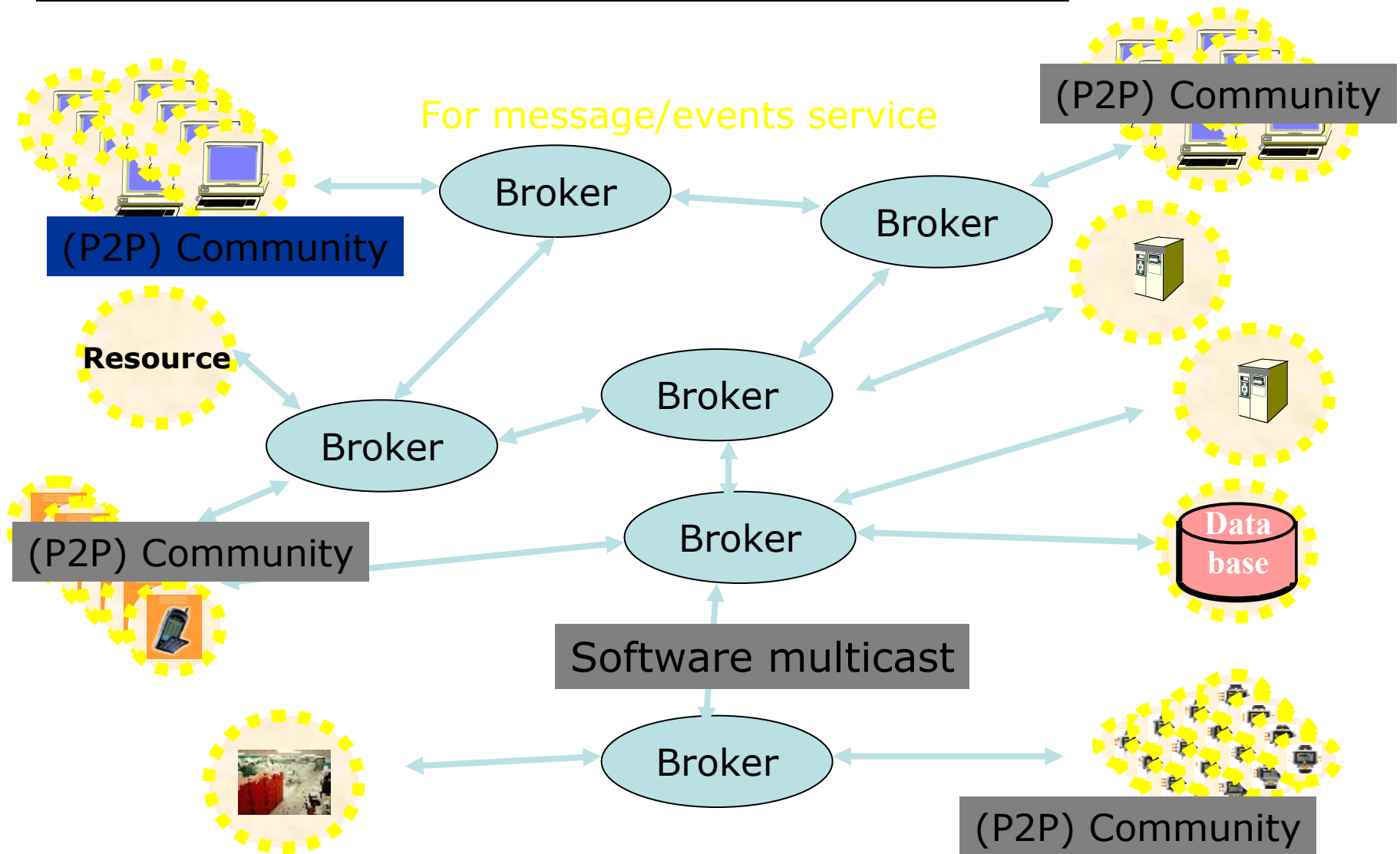
XGSP Collaboration Portal

- XGSP collaboration portal
 - *The aggregation of different collaboration services*
 - *The portal is a container of various collaboration portlets*
- Advantages:
 - *XGSP users can customize their collaboration portals by adding, removing collaboration portlets and changing the layer out of the portals.*
 - *It is very easy to integrate various collaboration services such as A/V, whiteboard, shared display in XGSP framework.*
 - *Other Grid portals can reuse these collaboration portlets for their purposes*

NaradaBrokering

- Based on a network of cooperating broker nodes
 - Cluster based architecture allows system to scale to arbitrary size
- Originally designed to provide uniform software multicast to support real-time collaboration linked to publish-subscribe for asynchronous systems.
- Now has five major core functions
 - Message transport (based on performance measurement) in heterogeneous multi-link fashion
 - General publish-subscribe including JMS & JXTA and support for RTP-based audio/video conferencing
 - Distributed XML data-base using P/S XPATH metaphor
 - Filtering for heterogeneous clients
 - Federation of multiple instances of Grid services as illustrated by JXTA peer-group linkage

Narada Broker Network



Advantages of deploying NaradaBrokering for XGSP group communication services

- Covers the heterogeneity of network transportation and provides unified multipoint transportation API
 - *Software multicast*
 - *Communication over firewalls and proxy boundaries*
 - *Communication over multiple transports*
 - *Application level Quality of Service*
 - *Filter messages to slow (collaborative/real-time) clients*
- Provides robust, scalable and high efficient multipoint transportation services
 - *Availability and scalability*
 - *Efficient routing and bandwidth utilizations*

Global-MMCS 1.0

The first prototype of this system includes:

- A XGSP media server
 - *provides the services of bridging multicast and unicast, video-switching, video-mixing and audio-mixing to H.323, SIP as well as AG endpoints.*
- H.323, SIP and Real Servers for A/V clients
- XGSP A/V Session Server
 - *manages real-time A/V sessions, receiving messages from gateways and the web server, and performing appropriate actions on the media server.*
- The web server
 - *provides an easy-to-use web interface for users to join multimedia sessions and for administrators to perform administrative tasks.*

H323 Client (Polycom) in XGSP Session

The image displays three overlapping screenshots of the XGSP video Conferencing Administration Panel in a Microsoft Internet Explorer browser window. The browser's address bar consistently shows the URL: `http://ararat.ucs.indiana.edu:8083/jspwebapp/portal` or `admin`.

Top Screenshot: Main Portal

- Navigation links: [About XGSP](#), [Make Meeting Reservation](#), [Join](#), [Log In](#)
- Meeting Calendar: Year 2002 Month 8
- Calendar grid (Sun-Fri):

Sun	Mon	Tue	Wen	Thu	Fri	Sat
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31

Middle Screenshot: Session Work Page

- Session show : Room : ag:lob
- Member List:

wwj	wenjunwu
-----	----------
- Stream List:

Stream ID	Stream Name	Stream Type	Stream Description	Stream ID
<input type="radio"/>	caolan	video	this is not a test	1112
<input checked="" type="checkbox"/>	caolan	audio	up side	11120
<input type="checkbox"/>	caolan	audio	he he	5550
<input type="radio"/>	wwj	video	right side	111
<input type="radio"/>	wwj	video	left side	1113

Bottom Screenshot: XGSP MCU (Control) User Interface

- Navigation menu: [System Configuration](#), [User Management](#), [Room Management](#), [Community Management](#), [Session Management](#)
- Welcome to XGSP videoconferencing system!!
- Meeting Calendar: Year 2002 Month 9
- Calendar grid (Sun-Fri):

Sun	Mon	Tue	Wen	Thu	Fri	Sat
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31
- Room List:

test	lobbery
xxx	mars
black	china
fjk	usa
dfjk	solar



vic and RealVideo views of multiple streams

The screenshot shows a web browser window displaying a Polycom video conference. The main content area is a 'ViaVideo' window with four video streams. The top-left stream shows a person in a red shirt. The top-right stream shows a person in a dark jacket. The bottom-left stream shows a blue cartoon character. The bottom-right stream shows a person in a white shirt. The interface includes a menu bar (File, Edit, View, Window, Help), a toolbar with various icons, and a list of participants. The list includes columns for IP address, video status, name, and ID.

IP Address	Video Status	Name	ID	Thumbnail	
104.65	video	224.10.10.10	1051164473		
104.17	video	224.10.10.10	1051697609		
<input type="radio"/>		wu@156.56.104.52	1051690308		
<input checked="" type="radio"/>		MixedVideo	video MixedVideo	-1569365611	
<input type="checkbox"/>		Administrator@156.56.104.48	audio	260587608	

Polycom view of multiple video streams

File Edit View Window Help

224.10.10.10/5554

```

E:\Work\prj3\polycom-issue\mbone-caller\mbone-caller.exe
get session addr
get session ID
session ID : ourtestroom session_addr : 156.56.104.53
reply: <?xml version="1.0" encoding="UTF-8"?><SessionSignalResponse
@156.56.104.52</ClientID><ResponseType>JoinReply</ResponseType>P
k</ResponseResult><
><IPAddress>156.56.
mediaFormat>PCMU</Media
transport><IPAddress
sport><MediaFormat>
sponse>
Start
Start e
remote
remote
  
```

76 xgsp@murray.ucs.indiana...

76 156.56.104.53/11000

xgsp@murray.ucs.indiana.edu
156.56.104.53/h261
0 f/s 0 bps (0%)
 mute color info...

xgsp@murray.ucs.indiana.edu
156.56.104.53/h261
0 f/s 0 bps (0%)
 mute color info...

xgsp@murray.ucs.indiana.edu
156.56.104.53/h261
0 f/s 0 bps (0%)
 mute color info...

xgsp@murray.ucs.indiana.edu
156.56.104.53/h261
10.0 f/s 251 kb/s (0%)
 mute color info...

Pixrate Autoplace
Menu Help Quit

VIC v2.8ucl-1.1.3-nlsde-2002-0305a

Sign In My Yahoo! News Health Yahoo! Games

About XGSP Join Session Enter Room Registration Log In

Meeting Calendar
Year 2003 Month 3

Sun	Mon	Tue	Wen	Thu	Fri	Sat
						1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31					

Year 2003 Month 4

Sun	Mon	Tue	Wen	Thu	Fri	Sat

<input type="radio"/>	UNKNOWN@156.56.104.17	video	224.10.10.10	1051697609	
<input type="radio"/>	wu@156.56.104.52	video	224.10.10.10	1051690308	
<input checked="" type="radio"/>	MixedVideo	video	MixedVideo	-1569365611	
<input type="checkbox"/>	Administrator@156.56.104.48	audio		260587608	

Discussions Subscribe... Discussions not available on http://elkhart.ucs.indiana.edu:8080/

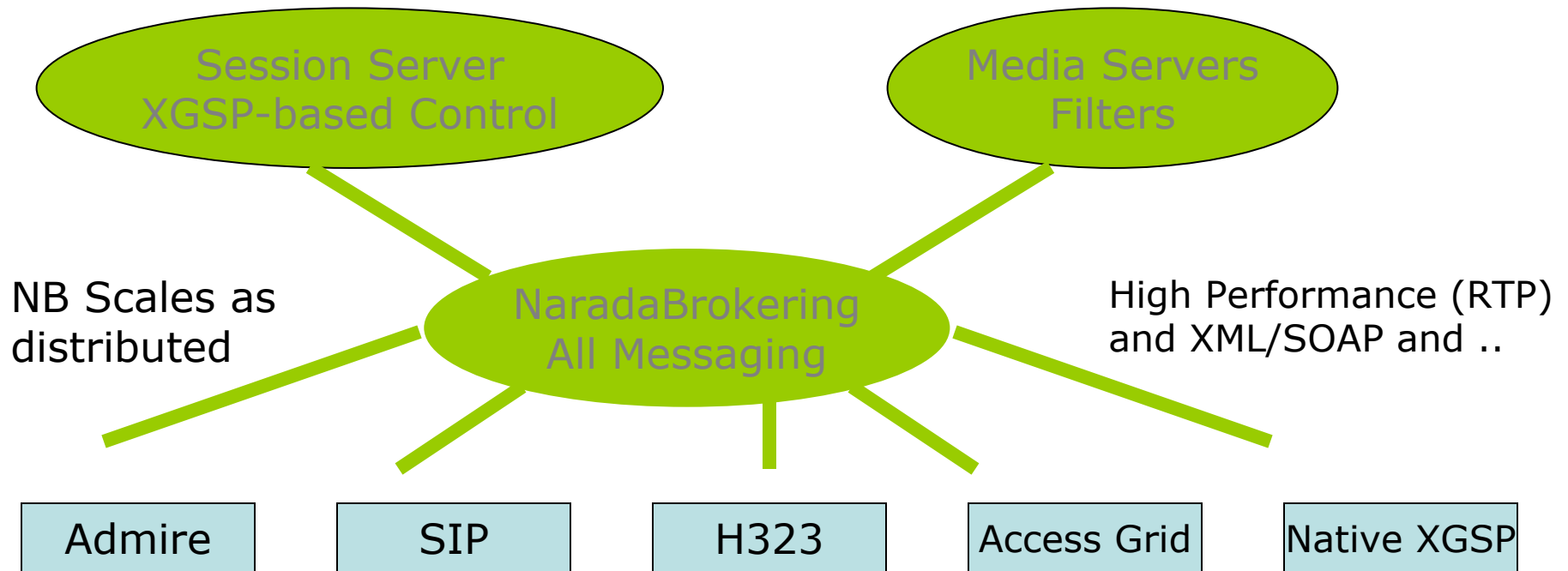
vic views of multiple video streams

Global-MMCS 2.0 (1) XGSP MCU

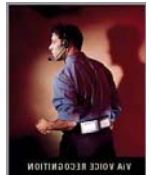
- ~~We are building an open source protocol independent~~ “MCU” which will scale to an arbitrary number of users and provide integrated collaboration services.
- We will deploy it globally and test with thousands of simultaneous users later this year.
- The function of the A/V media server will be distributed in NaradaBrokering architecture.
- Open XGSP MCU based on the following open source projects
 - openh323 is basis of H323 Gateway
 - NIST SIP stack is basis of SIP Gateway
 - NaradaBrokering is open source messaging from Indiana
 - Java Media Framework basis of Media Servers

XGSP MCU Architecture

Use Multiple Media servers to scale to many codecs and many versions of audio/video mixing



Gateways convert to uniform XGSP Messaging



Grid Summer School: Grid Portals

Global-MMCS 2.0 (2) Portlets

- Collaboration clients will be built into portlets by creating *Java Applet* or *ActiveX* controls for these clients and adding them into HTML pages.
- A collaboration portlet opens local services for XGSP application session management and floor control.
 - *Node Manager portlet invoke the service to control local portlets*
- *Apache Jetspeed* seems good open source technology supporting this model
- *Portlets such as Access Grid portlet can be reused by Grid Portal Developers*

Unicast AG Portlet

Work Page
ourtestroom : Room : ourtestroom

Member List	
wvj	wenjmvu

Stream List				
istrator@156.56.104.65	video	224.10.10.10	1056581974	
56.56.104.52	video	224.10.10.10	1056217142	
istrator@156.56.104.48	video	224.10.10.10	1056514355	
istrator@ger	audio	224.10.10.10	-517816191	

Media Player
00:00:42:01

Hello User
Hello Wenjun!

IP	Type	Address	Port	Thumbnail
ag@144.167.32.100	video	224.2.177.155	1056747774	
ag@144.167.32.100	video	224.2.177.155	1056848838	
ag@144.167.32.100	video	224.2.177.155	1056717051	
ag@144.167.32.100	video	224.2.177.155	1056377985	
ag@144.167.32.100	video	224.2.177.155	1056125605	
ag@206.75.91.24	video	224.2.177.155	1056933370	
ag@206.75.91.24	video	224.2.177.155	1056910086	
ag@206.75.91.24	video	224.2.177.155	1056248160	

00:00:27:00

Global-MMCS 2.0 (3)

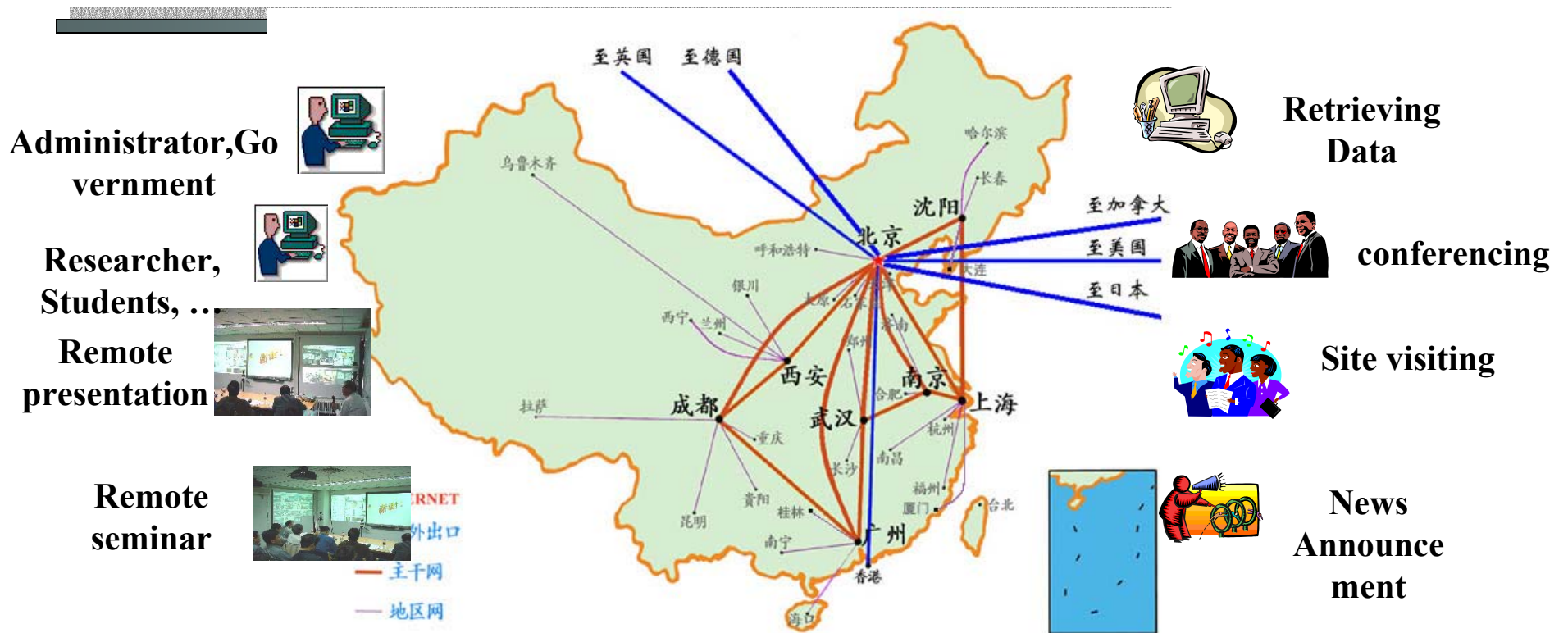
- Use web services to integrate the communities
 - Web-services for Admire in China will be fully implemented in the new prototype.
 - Global-MMCS will integrate Access Grid and Admire as well as other H.323 and SIP communities to build a global collaboration platform.
 - Make full use of conferencing resource and create larger collaboration communities

for example, there is no stable multicast link between China and US, our system can provide a “bridge” for that.

Admire Project in China

- Admire(Advanced Multimedia Interactive Real-time Environment)
A videoconferencing project in China similar to Access Grid.
- It is deployed in many sites across China and provides audio, video, and data sharing tools.
- Admire also provides “Admire Media Gateway Server” which plays the role of the bridge between multicast and unicast networks.

National video conference system for Scientific Research

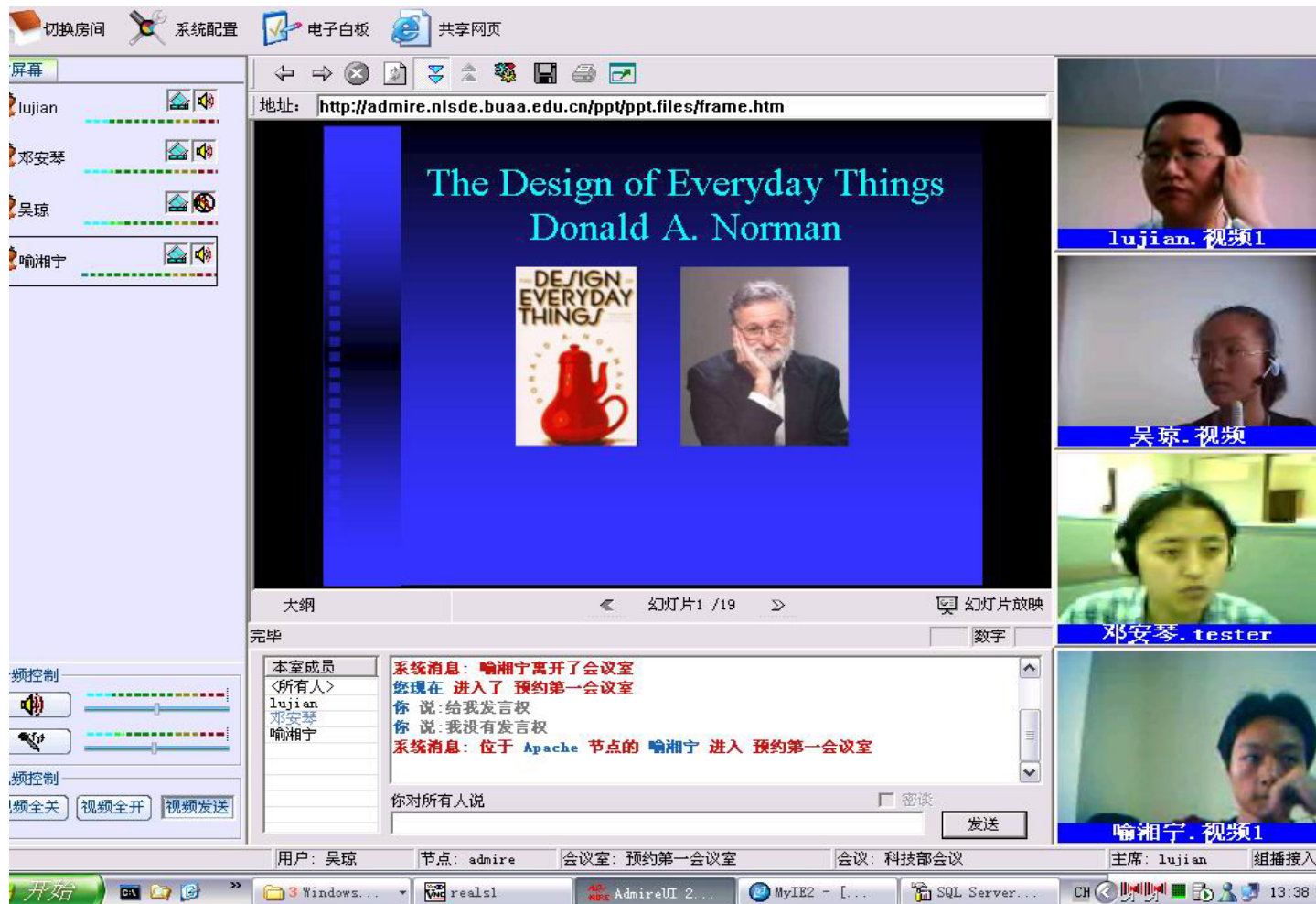


National conference center/regional access points

Open standard for connecting to "National A/V system for Research"

High speed network environment CERNET

Admire Demo Pictures





Grid Summer School: Grid Portals

Integrate Admire into Global-MMCS

- **Goals:**
 - Access Grid users in USA can communicate with Admire users in China
 - H.323 and SIP users can attend the Admire conference
- **Approaches**
 - Admire provides XGSP Web Service Interface
 - Connect Admire Media Gateway Server with NaradaBrokering infrastructure

Related work and comparison

- Global-MMCS provides opportunities for those
 - either use H.323 and SIP clients such as polycom, windows messenger
 - only have unicast network and NAT firewalls.
- Compared to VRVS : different focuses
 - open source scalable “MCU” based on messaging middleware
 - integration with other communities
 - portlet for user interface, providing more collaboration tools

Questions?