



Commodity Grid (CoG) Kits

Abstractions

CoG Abs

**Gregor von Laszweski,
Kaizar Amin, Mike Hategan, , ANL
Keith R. Jackson, LBNL**



www.cogkits.org

Dennis Gannon, Charles Severance, Mary Thomas, Jay Alameda, Marlon Pierce, Al Rossi, Shawn Hampton, Nestor Zalupec,



Outline



- What are CoG Kits? (1 minute)
- What are CoG Abstractions?
- What are we doing with CoG Abs? : Gridfaces
(visual components)



Observation (cont.)



- Problem
 - Many application developers desire to program the Grid in familiar higher level frameworks that allow rapid prototyping.
- Solution
 - We propose to reuse a variety of commodity tools, protocols, approaches, methodologies, while integrating Grid software based on the Globus Toolkit
 - Easier development of advanced Grid services
 - Easier and more rapid application development
 - Easier deployment of Grid services
 - Code reuse and use of component repositories
 - Use of Web services as part of the Grids
 - Widespread use of the Grid
 - Use of commodity technology is not limited to the client!



Overview: CoG Kits



- Project Goals: Make Grid programming simple
- Approach: Develop CoG Kits for Java and Python
 - Others are possible, but that's what we focus on
 - Available since 1997
- We have experience in this group dating back to the beginning of Grid and Distributed Computing



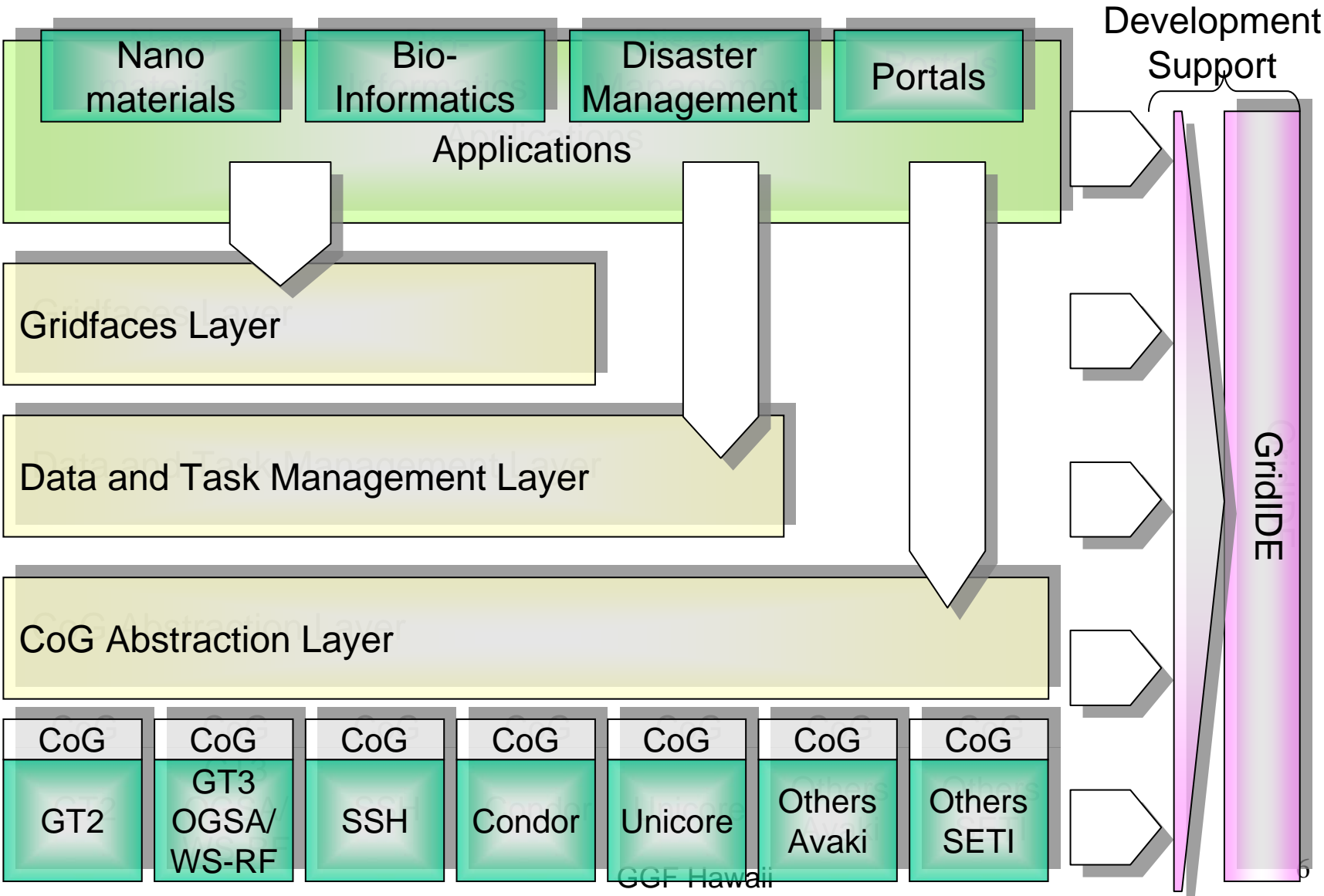
Abstractions



- Hypothesis:
 - With rapidly changing technologies it may be beneficial to have an abstraction that can be assisting in this technical challenge
 - Charter of the Research Group
 - CoG Abs shares this vision



CoG Abstraction Layers



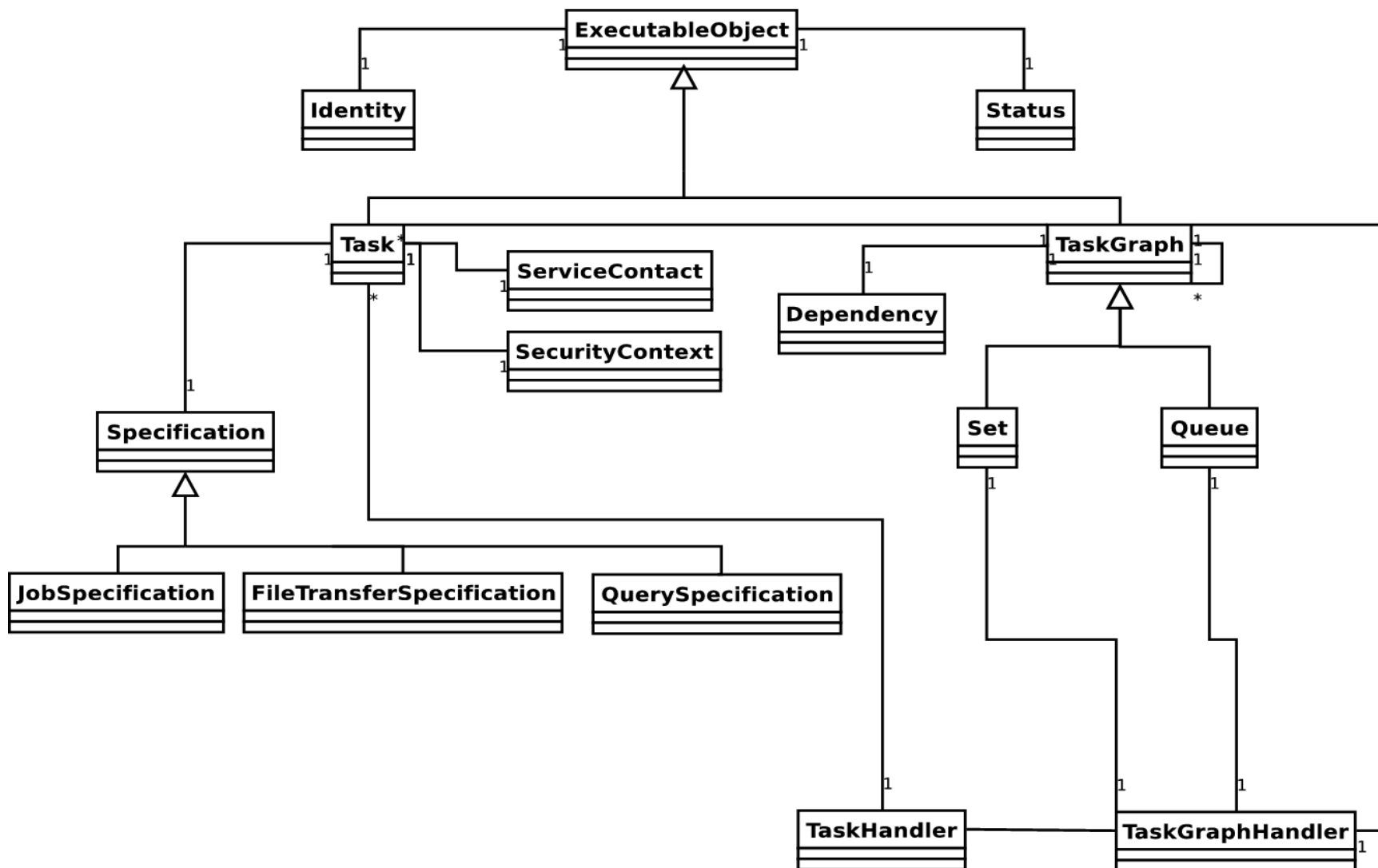


Grid Abstractions



- Hides the underlying toolkit
 - GT2, GT3, ssh, Unicore,
 - others such as Condor could be possible
- Has advanced data structures and services
 - Such as graphs, workflows, queues, ...
- Provides elementary API's to
- Security management
- Job management
 - Including Workflow management
- File management
 - Including Queues

Task Abstractions are part of CoG





Directory Management Abstractions



- public interface DirectoryManagementResource extends GridResource {
- ...
- // set/get Host, port, protocol, URI, SecurityContext
- void setCurrentDirectory(String directory);
- String getCurrentDirectory();
- Enumeration list (String directory)
- void rmdir(String directory, boolean force)
- void getFile (String remoteFilename, String localFileName)
- void putFile(String localFilename, String remoteFileName)
- void getDir(String remoteDirname, String localDirName)
- void putDir(String localDirname, String remoteDirName)
- void chmod(String filename, int mode)
- boolean isReadable(String filename)
- void makeWritable(String filename, boolean writable)
- boolean isWritable(String filename)
- long size(String filename)
- boolean exists(String filename)
- ...
- boolean isDirectory(String dirName)
- ...
- }



Using CoG Abstractions





Build Higher Level Abstractions: Grid Resource With a Queue



- // pseudocode
- interface QueueEnabledExecutionResource extends GridResource {
 - void setQueue (Queue queue);
 - Queue getQueue ();
 - void submit (String provider, ExecutableObject executableObject);
 - void remove (Identity identity);
 -
 - void setServiceContact (ServiceContact serviceContact);
 - ServiceContact getServiceContact (String provider, int serviceType);
 - void setSecurityContext (String provider, SecurityContext securityContext);
 - SecurityContext getSecurityContext (String provider);
 - Enumeration getAllSubmittedTasks ();
 - void setAttribute (String name, Object value);
 - Object getAttribute (String name);
- }



Build Higher Level Abstractions: Grid Resource With QoS



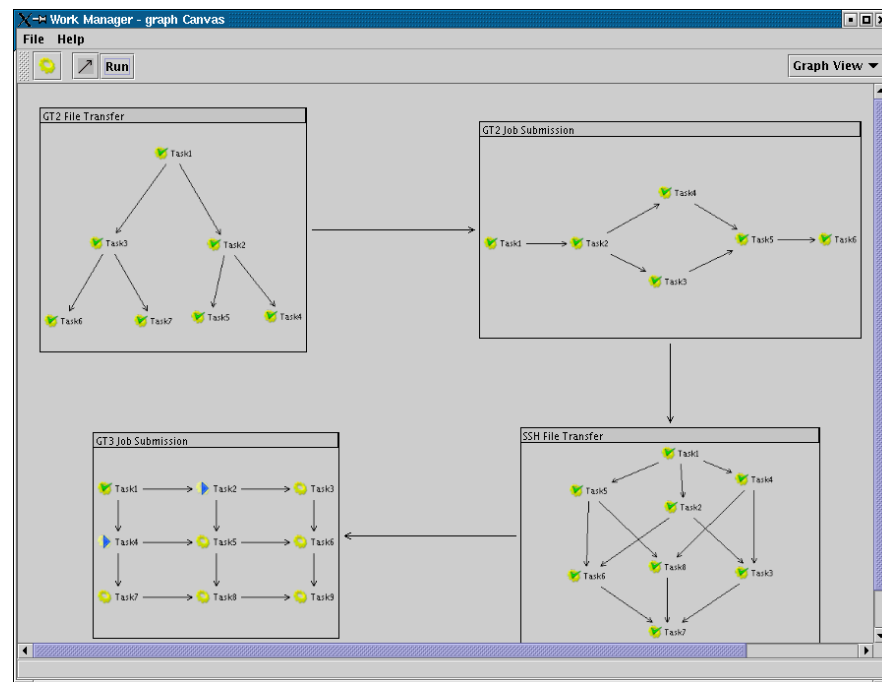
- // pseudocode
- interface QoSEnabledExecutionResource extends GridResource {
 - void setQueue (Queue queue);
 - Queue getQueue ();
 - void submit (Date startTime, Date endTime, String provider, ExecutableObject executableObject,);
 - void remove (Identity identity);
 -
 - void setServiceContact (ServiceContact serviceContact);
 - ServiceContact getServiceContact (String provider, int serviceType);
 -
 - void setSecurityContext (String provider, SecurityContext securityContext);
 - SecurityContext getSecurityContext (String provider);
 -
 - Enumeration getAllSubmittedTasks ();
 -
 - void setAttribute (String name, Object value);
 - Object getAttribute (String name);
 - }



GridAnt and Karajan



- Workflow visualization
- We can submit to
 - GT2
 - GT3.02
 - SSH
 - (Unicore)



Java CoG Kit User Interface Components

The screenshot displays the Java CoG Kit user interface, which includes a central desktop area with various application icons and several auxiliary windows.

Central Desktop Icons:

- Four gear icons labeled Ls1, Ls2, Date, and filetransfer.
- Four icons with the CoG logo labeled GT3wiggum, GT3arbat, GT2arbat, and GT2wiggum.
- Icons for FTPServer (a penguin), Form (a computer monitor), Graph (a network diagram), Notepad (a document), Raswin (a gear with a document), Portal (a CoG logo), and MS-WORD (a document with a green chart).

File Transfer Component Window:

This window is open on the left side of the desktop. It features a menu bar with File, Connect, Security, RFT, and Options. Below the menu bar are icons for a lock, gridFTP, and ftp. The main area is divided into sections for Transfer Requests, Current Transfers, and a File Transfer Message Window.

Current Transfers Table:

Name	JobID	From URL	To URL
Copy 1		ftp://an...	file:///C:1...
Copy 2		ftp://an...	file:///C:1...
Copy 3		gsiftp://...	file:///C:1...
Copy 4		gsiftp://...	file:///C:1...
Copy 5		gsiftp://...	gsiftp://...
Copy 6		gsiftp://...	file:///C:1...
Copy 7		gsiftp://...	gsiftp://...

File Transfer Message Window:

From: gsiftp://choate.mcs.anl.gov:2811
To: file:///C:/labRoot
Done: successfully
Dragging gridFTPUI file...
Dropping the file...
Copying...
From: gsiftp://wiggum.mcs.anl.gov:2811
To: gsiftp://choate.mcs.anl.gov:2811/wh
Done: successfully

Options Window:

Options

Device
kit

J=mcs.anl.gov,O=Globus,O=Grid

Refresh

Test Remove

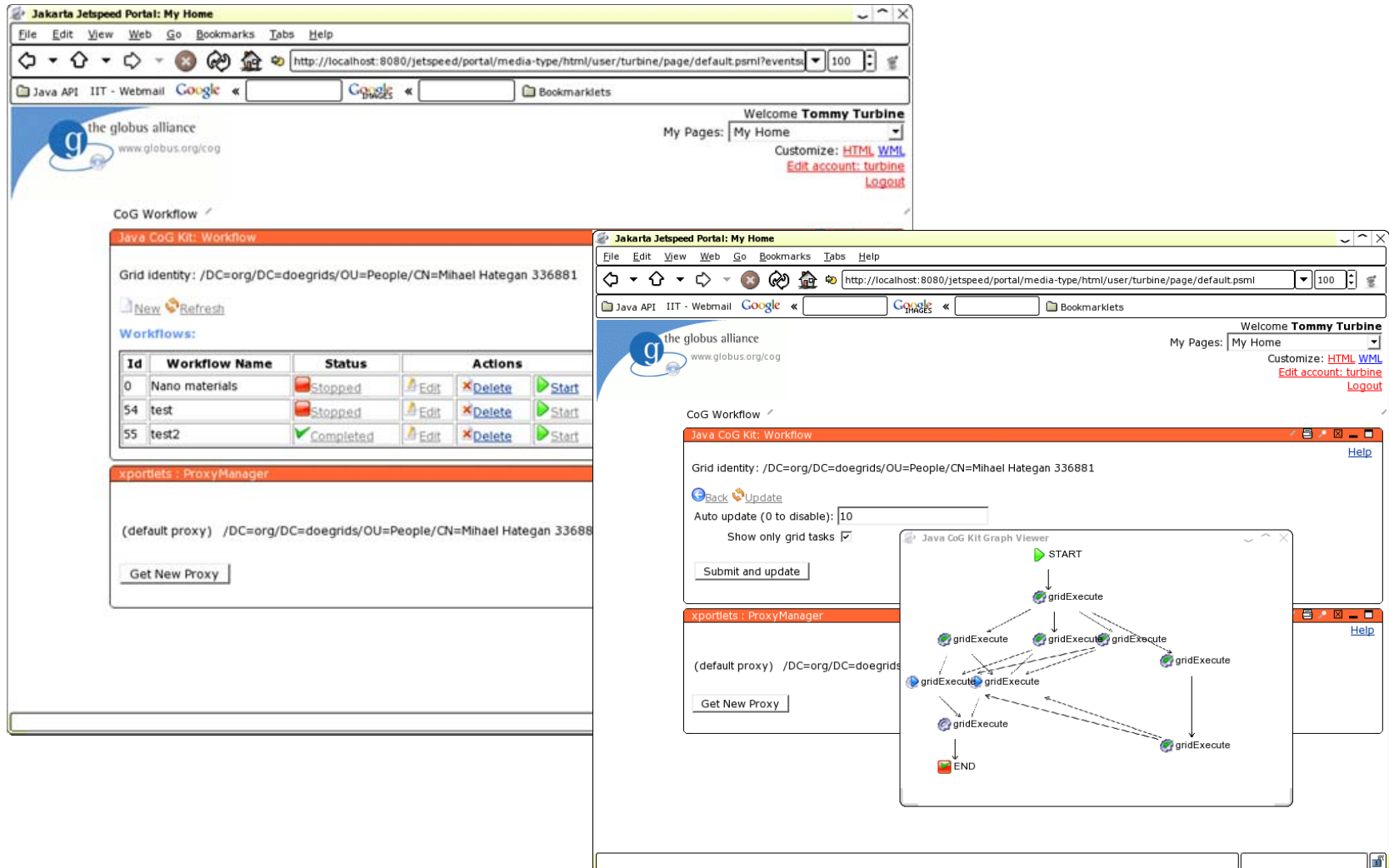
Local file

File
File
File

Status Window:

state..... ACTIVE
jobContact.... https://pitcairn.mcs.anl.gov:59237/23328/1037384976/

OGCE Use Case



The screenshot displays two overlapping browser windows of the Jakarta Jetspeed Portal. The top window shows the 'My Home' page with a welcome message for 'Tommy Turbine' and links for customization and login. The bottom window shows the 'CoG Workflow' section, which includes a table of workflows and a 'ProxyManager' section.

CoG Workflow Table:

Id	Workflow Name	Status	Actions
0	Nano materials	Stopped	Edit Delete Start
54	test	Stopped	Edit Delete Start
55	test2	Completed	Edit Delete Start

Java CoG Kit Workflow View:

Grid identity: /DC=org/DC=doe grids/OU=People/CN=Mihael Hategan 336881

Auto update (0 to disable): 10

Show only grid tasks ☒

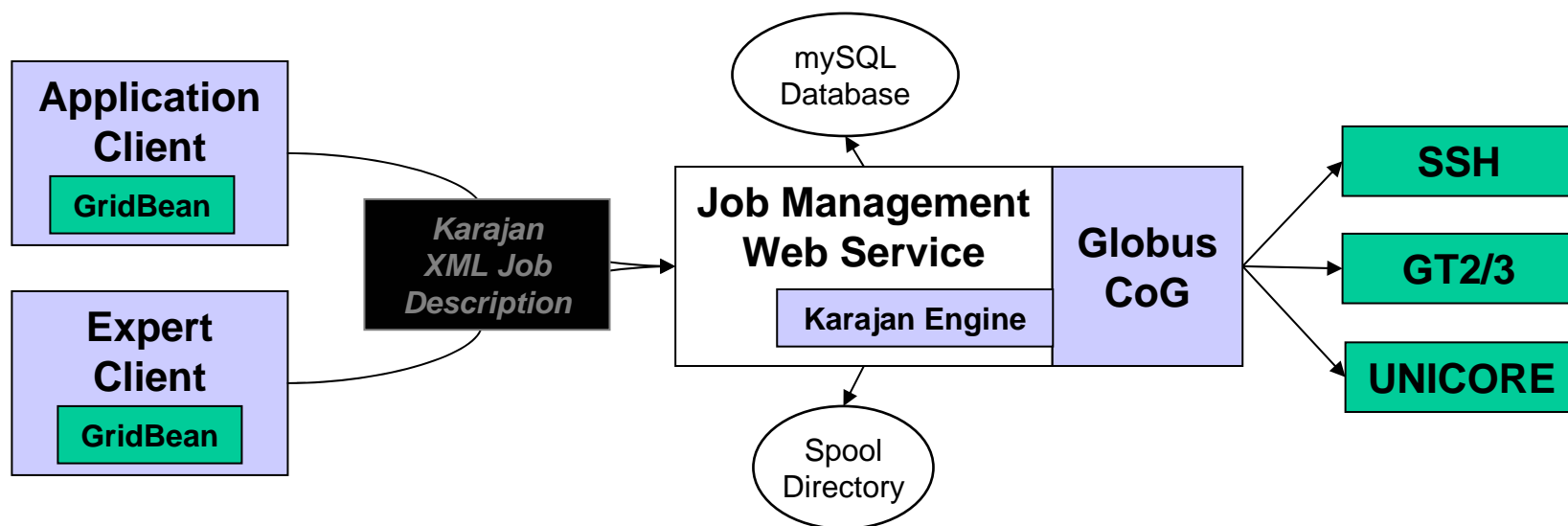
Submit and update

Java CoG Kit Graph Viewer:

```

graph TD
    START --> GE1[gridExecute]
    GE1 --> GE2[gridExecute]
    GE1 --> GE3[gridExecute]
    GE2 --> GE4[gridExecute]
    GE3 --> GE4
    GE4 --> GE5[gridExecute]
    GE5 --> END[END]
  
```

Intel Grid Programming Environment: Proof of Concept Implementation



- Database keeps track of jobs and files
- Files are spooled at web service
- Service and two Clients implemented



Further Information



- **URL**

- <http://www.cogkits.org/>

- **Contacts**

- KRJackson@lbl.gov

- Gregor@mcs.anl.gov

- **Acccknowledgement**

- DOE, NSF

- Mary Anne Scott, Ian Foster, Dennis Gannon