



Portlets for User Centric Job and Task Monitoring for Open Science Grid Virtual Organizations

D Alexander (presenter), R Pundaleeka, S Tramer

Tech-X Corporation

Boulder CO

(UCM Project Funded by Office of NP, DOE)

J Lauret, V Fine

Physics Department

Brookhaven National Laboratory

GCE07 Workshop, November 11-12, 2007

- **Small Business Innovative Research Project (SBIR)**
 - Middle of a 2 year project
 - Main customer DOE-NP, so focus is Open Science Grid (OSG)
- **Collaborating with STAR Computing Group**
 - Use case is STAR nuclear physics experiment
 - C++ analysis framework
 - Handles interface to data (distributed: BNL, LBNL)
 - Has “Logger” to log application-level messages.
 - STAR Unified Meta-Scheduler has high-level expression of analysis task

- **Keeping Design General**

- Will apply to other Grids than OSG
- Will apply to other domains, not just physics

- **Keeping Tools General**

- Main Code Library implemented in many languages
- Data Store has modular design to be extensible
 - Database Module with MySQL implementation
 - File Module, working with a DOE SciDAC funded trouble-shooting group (Center for Distributed Petascale Science)
- Portlet-Standard presentation layer

Job Monitoring Common - Geared To Administrators & Benchmarking



Virtual Organization Resource Selector

Grids

OSG

Virtual Organizations

STAR

Open Science Grid

TERAGRID

Summary of Common TeraGrid Software and Services 2.0

This page offers a summary of results for critical grid, development, and cluster tests. Details about a resource's test results are available by clicking on the resource name in the "Site-Resource" column of the table.

Site-Resource	Grid	Development	Compute	Total Pass
site1-resource1	Pass: 17 Fail: 2	Pass: 9 Fail: 0	Pass: 3 Fail: 0	Pass: 29 Fail: 2
site1-resource2	Pass: 18 Fail: 1	Pass: 9 Fail: 0	Pass: 3 Fail: 0	Pass: 30 Fail: 1
site2-resource1	Pass: 18 Fail: 1	Pass: 9 Fail: 0	n/a	Pass: 27 Fail: 1

Expanded View of Errors

site1-resource1

Grid

1. globus: failed: duroc_mpi_helloworld to jobmanager-bat0

KEY

- All tests passed: 100%
- One or more tests failed: < 100%
- Tests not applicable to machine or have not yet been ported

History of percentage of tests passed in "Grid" category for a one week period:

Grid availability

Name	Institution	System	CPU's	Peak C/Flops	Memory Gbytes	Disk Gbytes	Status	Load	Jobs
Big Red	SD	IBM x1350	2548	20.40	8.20	286.00	+	178.00	4440
BigRed	SDC	Cray XT3	4138	21.20	4.04	100.00	+	208.190	1030
Blue Gene	SDC	IBM Blue Gene	6144	17.10	1.30	18.50	+	38.00	460
Cobalt	NCSA	SQZ Atlas	1024	8.55	3.00	100.00	+	778.000	200
Copper	NCSA	IBM Power4 p690	384	2.00	1.44	30.00	+	786.860	30
Dallas p415	SDC	IBM Power4 p655	2176	14.30	1.73	113.00	+	176.00	2110
Dallas p690	SDC	IBM Power4 p690	182	1.30	3.88	113.00	+	20.200	40
Leah	Purdue	Dell EMC47 Linux Cluster	1024	8.00	2.00	28.00	+	676.00	10
Lonestar	TACC	Dell PowerEdge Linux Cluster	5200	33.00	10.40	34.90	+	1136.800	90
OSG	OSG	IBM x312	46	0.34	0.07	2.14	+	139.00	0
					0.30	8.00	+	258.800	0
					1.24	4.00	+	66.00	0
					4.47	60.00	+	46.300	40
					4.47	60.00	+	668.470	140
					3.75	100.00	+	248.300	370

Daily Usage by Site for VO

Showing page 1 of 1

Date range: 2008-10-01 - 2007-11-11

VO Name: uscms

Wallclock Hour

Retrieve Report URL

Show Query

Administrator's Perspective

up/down site status

aggregate resource usage information for accounting

overall health of virtual organization's Grid resources

Hope Is To Provide More Service Than Baseline Configuration & Operation



User expresses task as “Run application A on dataset D.” Broker handles the execution, but user needs to know how it goes....

IDEALLY, users should not be burdened with...

- How many jobs are needed (may depend on optimization keyed on dataset distribution)**
- The details of which sites exist (if all goes well, user doesn't care where jobs go)**
- Which sites are up or down (as long as some sites are available to do task)**

Service to the Users Means Access to Specific Monitoring Information

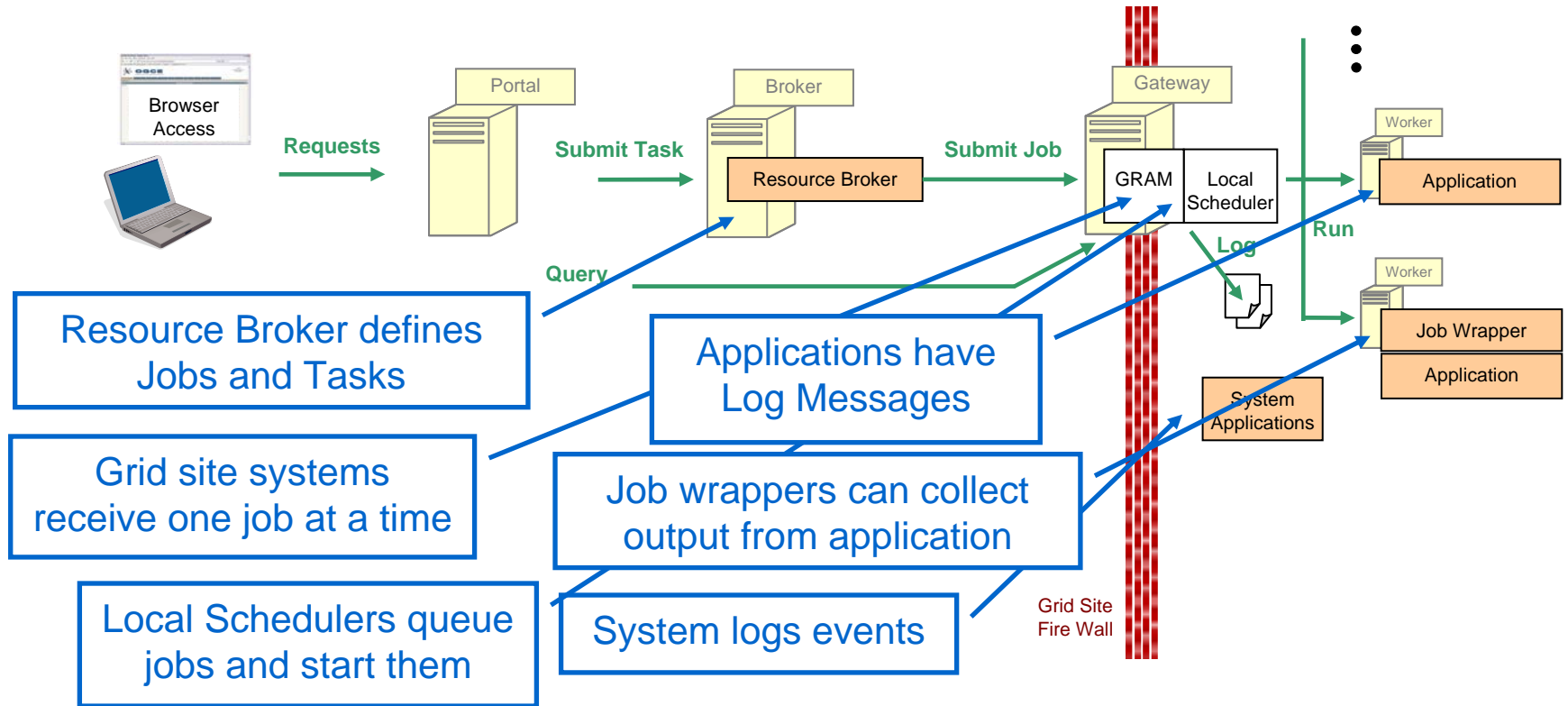


- **Users want to know if Grid is “working for them”**
 - What % of the task is completed?
 - Are my Grid jobs in the queue?
 - What position are the jobs? How long is the line?
 - What priority do the jobs have?
 - When the jobs start? Stop? How long?
 - What are the standard output and error messages?
 - What broad context or section of code is executing?
 - What are the values of some of the important variables?
 - Detailed job failure information? What part of Grid? When?

A BOTTOM LINE

If individual users from small Virtual Organizations are going to find value in the Grid, then they are going to need service with User-Centric Monitoring.

Aggregating Monitoring Information is Difficult for Any Organization



Need fine-grain information about a user's scientific task composed of many jobs. This involves data aggregation from many unrelated sources.

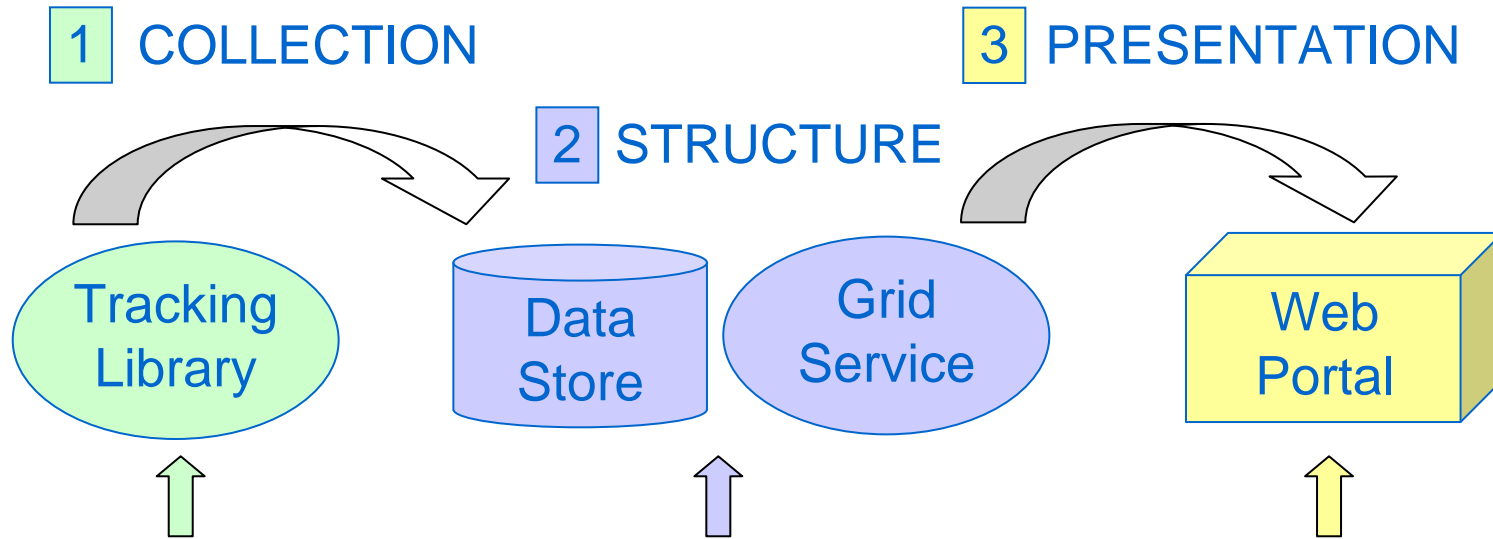
VO Needs Help to Help Users



- **Especially smaller members of OSG like STAR need help**
 - Must rely on OSG Software Stack and other tools because they can not afford large development efforts
- **Current Grid monitoring tools for Open Science Grid organizational members can be improved**
 - Focus is on collective job information for accounting & operations (MonALISA, Gratia, etc.)
 - Current monitor very valuable for running the Grid, but not as valuable for using it
 - ARDA Dashboard notable exception (works well for the LHC experiments, CERN Oracle DB)
 - Many monitoring systems do not have access to application-level logging

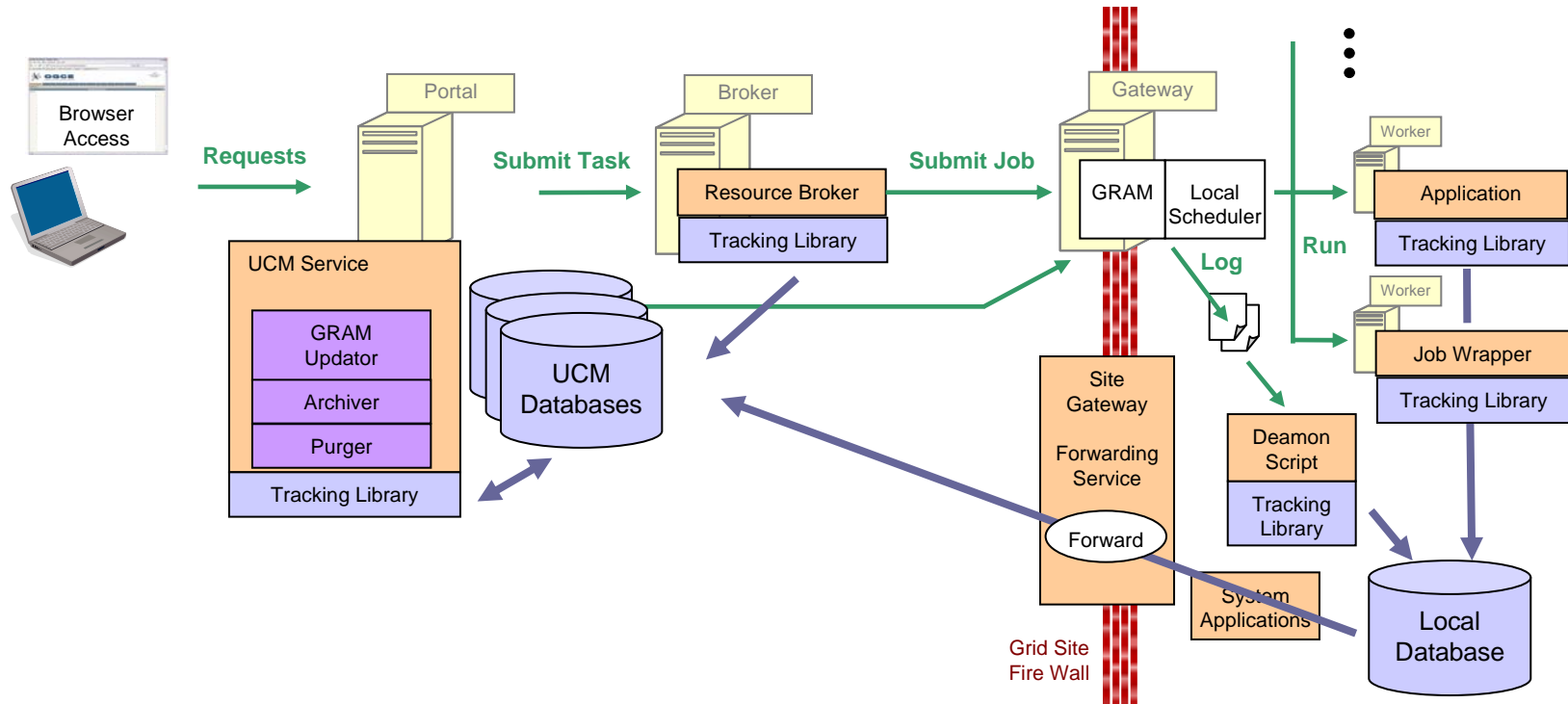


Tools For and End-to-End Solution



- Versatile library to set UCM information, usable in many places
- Written in C++ wrapped/translated in other languages
 - Flexible data model to include both database and file storage
 - Distributed data storage, scalability test done during development
 - UCM Information Service is a Globus Grid Service with OSG credentials
 - Also can collect other information (GRAM status) and maintain store
 - UCM Service information can be obtained through any Grid client or portal
 - Tools to display the information in user-friendly ways

(1) COLLECTION: Versatile Tracking Library



Tracking Library Hides Data Store Behind High-Level Interface



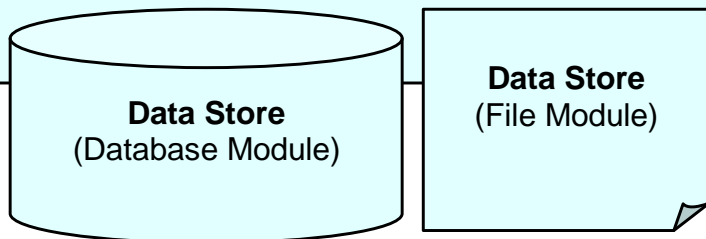
Interface Tier

Handles the calls to the application programming interface (API) and has logic to relate them to the data structures in the Data Tier. Three API sections are provided for Science Applications, Resource Monitoring Programs, or Job Submission Brokers.

Data Tier

Handles the calls from the Interface Tier and holds a particular data store module implementation

Modularized Data Store Types



Data Tier Holds Data Store Module

API for Science Applications

```
TxEvtLog myLog = TxEventLog()

myLog.logSystemEvent(enum stage, enum level,
string systemContext, string systemMsg)

myLog.logUserEvent(enum stage, enum level,
string userContext, string userKey, string
userValue)
```

API for Resource Monitoring

```
TxJob myJob = TxJob(int brokerTaskID, int
brokerJobID)

myJob.setProperty(enum propID, string value)

string val = myJob.getProperty(enum propID)
```

API for Job Submission Brokers

```
TxTask myTask = TxTask(int brokerTaskID, int
brokerID, int requesterID, string name="",
string description="")

myTask.addJob(int brokerJobID)
```

Interface for System Wide Use

STAR Experiment = Good Test Case



- **User's do not have the feedback to feel comfortable running jobs on the Grid**
 - STAR Experiment Computer Support team say user's experience too much like throwing jobs over a fence
 - A few researchers have run on various Grids, but the details generally hidden
- **Users define jobs with SUMS broker (Java)**
- **Bulk of analysis with root4star application (C++)**
- **Data fetching wrappers (Python)**

C++ Tracking Library Prototype
working with nightly tests of
root4star framework

SWIG in place to generate
Java and Python versions
of library

(2) STRUCTURE: Three Levels



TASK / JOB / EVENTS

Job Submission
Broker



Defines Jobs in Task

Grid Resource
Scheduler



Job Level Status Updates

Scientific
Application



Application Event Messages

DATA STORE COLLECTIONS

Tasks

- Task ID (assigned by Broker)
- Broker Task ID (assigned by Broker)
- Broker ID
- Requester ID
- Name
- Description
- Size (number of jobs)
- Remaining size (number of jobs left)
- Submit Time
- Update Time (last time size was updated)

Jobs

- Job ID (assigned by Broker)
- Job ID (assigned by Grid)
- Job ID (assigned by Local Scheduler)
- Task ID (for task that holds this job)
- Grid Submit Time
- Local Scheduler Submit Time
- Site Location
- Queue
- Queue Position
- Node Location
- Start Time
- Update Time (execution state last updated)
- Execution User (local system user)
- State ID (current execution state, 9 defined states)

Job Events

- Job ID (job which generated message)
- Level ID (info, warning, error, etc. 9 defined states)
- Context (bulk category description of event)
- Stage ID (start, status, or end)
- Content (long string of message)
- Event Time

First module implemented is a MySQL database

Adding File Module soon to work with logs from resource schedulers

Optimization is Key:

- Scaling tests underway
- Using dictionary tables

OTHER DICTIONARY COLLECTIONS

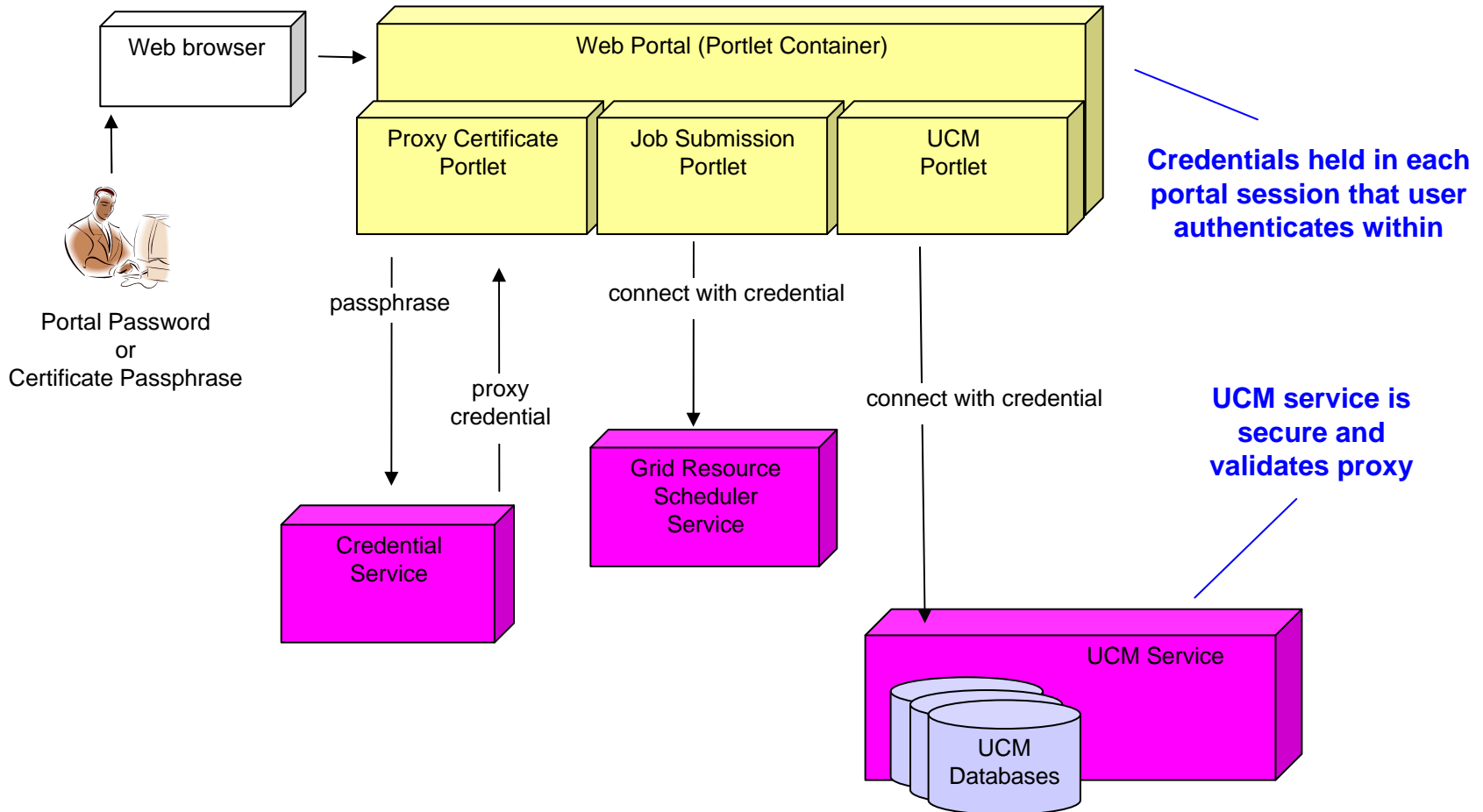
Broker, Requester, Stage, Level, State

(3) PRESENTATION: Portlet



- **We are developing JSR 168 compliant portlets to display information in database**
 - Allows our work to be easily integrated into portals
 - Virtual Organizations can put a portal together from a repository of portlets, ours being one of many
- **Access based on requester name, but user has to be authenticated either by portal or with Grid certificate**
- **Testing submission to a resource scheduler with the Open Science Grid software stack installed**

UCM Portlet Can Work Along Side Other Portlets



AJAX Portlet Prototype Underway



The screenshot displays the OGCE Grid Portal interface in a Mozilla Firefox browser. The page features a navigation bar with 'User Centric Monitoring', 'Welcome', and 'Administration' tabs. The main content area is divided into two portlets:

- ProxyManager Portlet:** Displays the text 'The following GSI proxy credentials are loaded into your account: No proxies currently loaded.' Below this, it says 'Click the button below to add a GSI proxy credential to your account:' with a 'Get New Proxy' button.
- User Centric Monitoring Portlet:** Shows a 'Select Database:' dropdown set to 'TechUCM Database'. It contains a table with columns 'Task/Job', 'State', and 'Progress'. The table lists seven tasks, all with a state of 'Done' and 100% progress. A 'Refresh' button and an 'Auto refresh every 10 seconds' checkbox are located below the table.

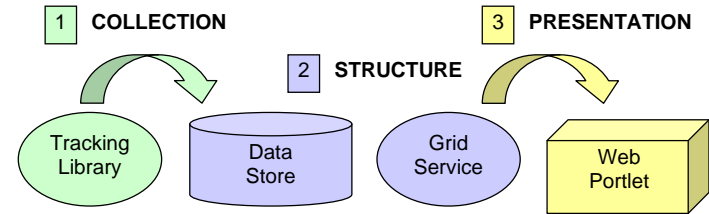
At the bottom of the page, there is a 'Job Submission' form with fields for Host, Port, Username, Executable, Arguments, Standard Input, Standard Output, Standard Error, Directory, CPU Count, Environment Variables, and Wall Clock Time (min). Below the form is a 'Job Status' section with a table header 'Job Handle' and 'Status', and a message: 'YOU CURRENTLY HAVE NO SUBMITTED JOBS.'

Summary



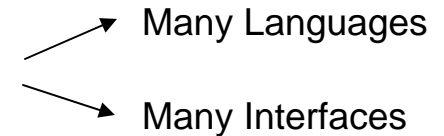
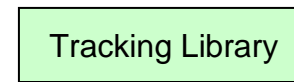
- **User-Centric Monitoring is Needed**

- **UCM end-to-end approach**



- **COLLECTION:**

- Versatile Tracking Library



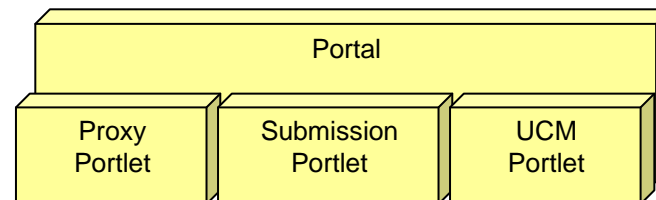
- **STRUCTURE:**

- Abstract Data Store



- **PRESENTATION:**

- Portlet



For More Information



**Tech-X Booth (#190 – back left of exhibit hall)
Demo Wed 4pm or anytime I'm at the booth**

**David Alexander
Principal Investigator, UCM project
+1 (303) 448-7751
alexanda@txcorp.com
<http://ice.txcorp.com/trac/ucm>**