# 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**

# User-Centric Monitoring SBIR Project

- **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

# Specific Customer, General Applicability

- **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

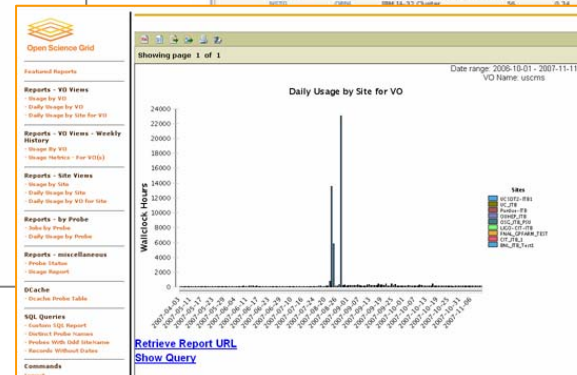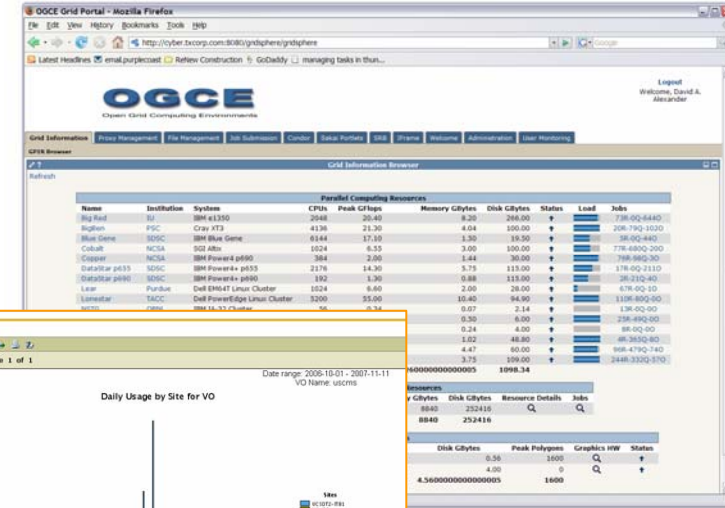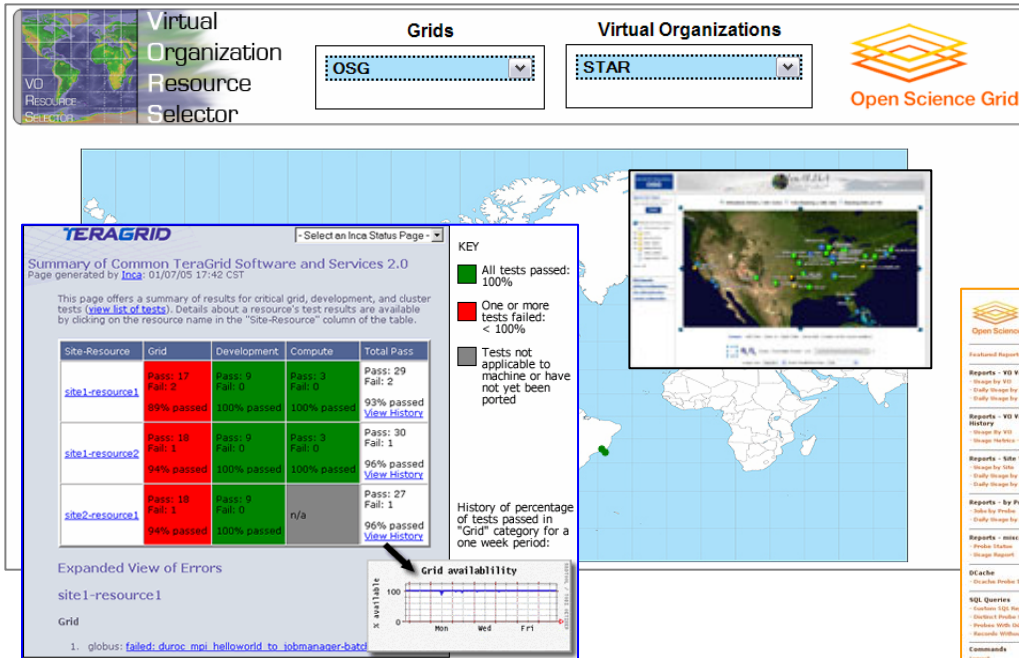# UCM Not Just Another Monitoring Tool

System Administrators have…

Smaller VO needs to get these to their users ….

# Job Monitoring Common - Geared To Administrators & Benchmarking



**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….**

IDEALLLY, 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?
  - 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
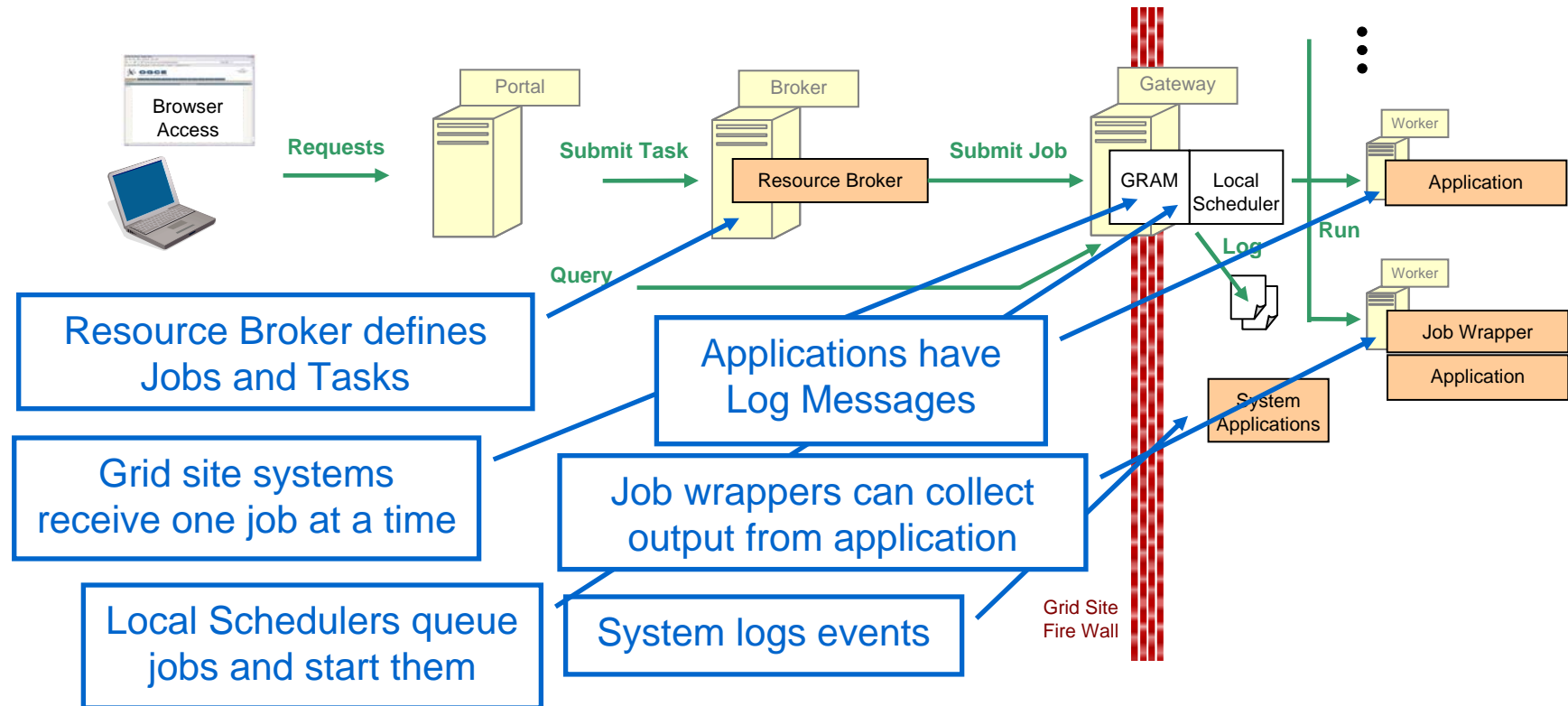


**Need fine-grain information about a user's scientific task composed of many jobs. This involves data aggregation from many unrelated sources.**
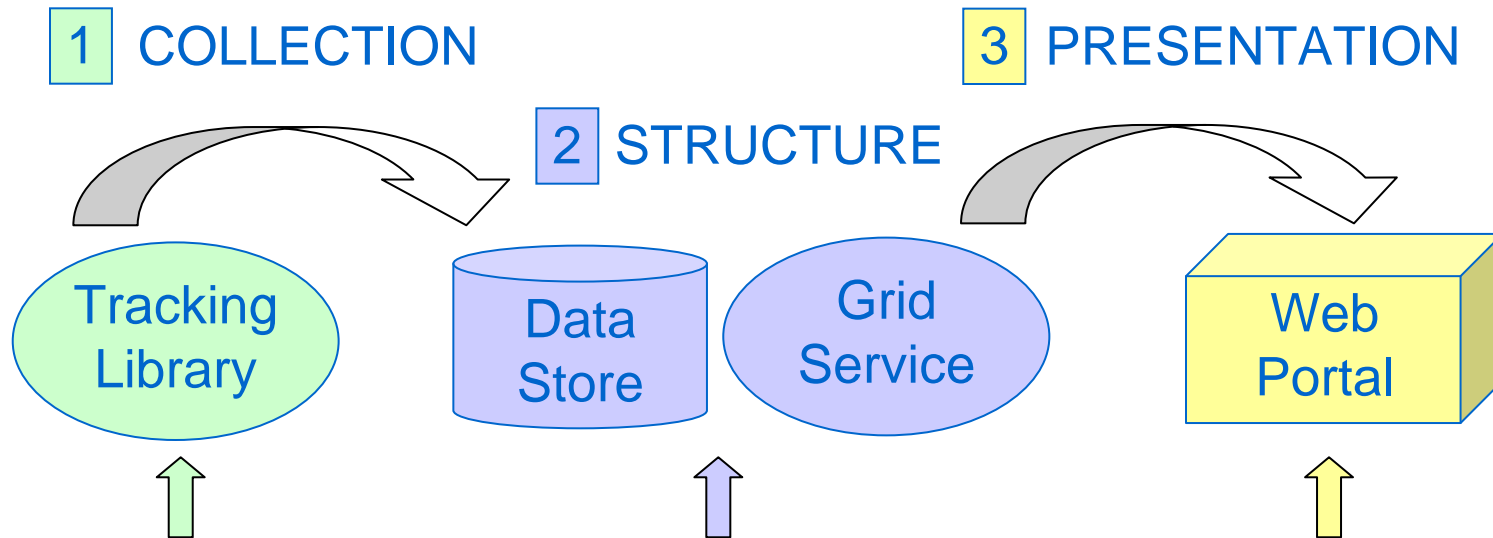
# VO Needs Help to Help Users

- **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


Open Science Grid

- **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



**1** COLLECTION     **3** PRESENTATION

**2** STRUCTURE

Tracking Library    Data Store    Grid Service    Web Portal

- 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
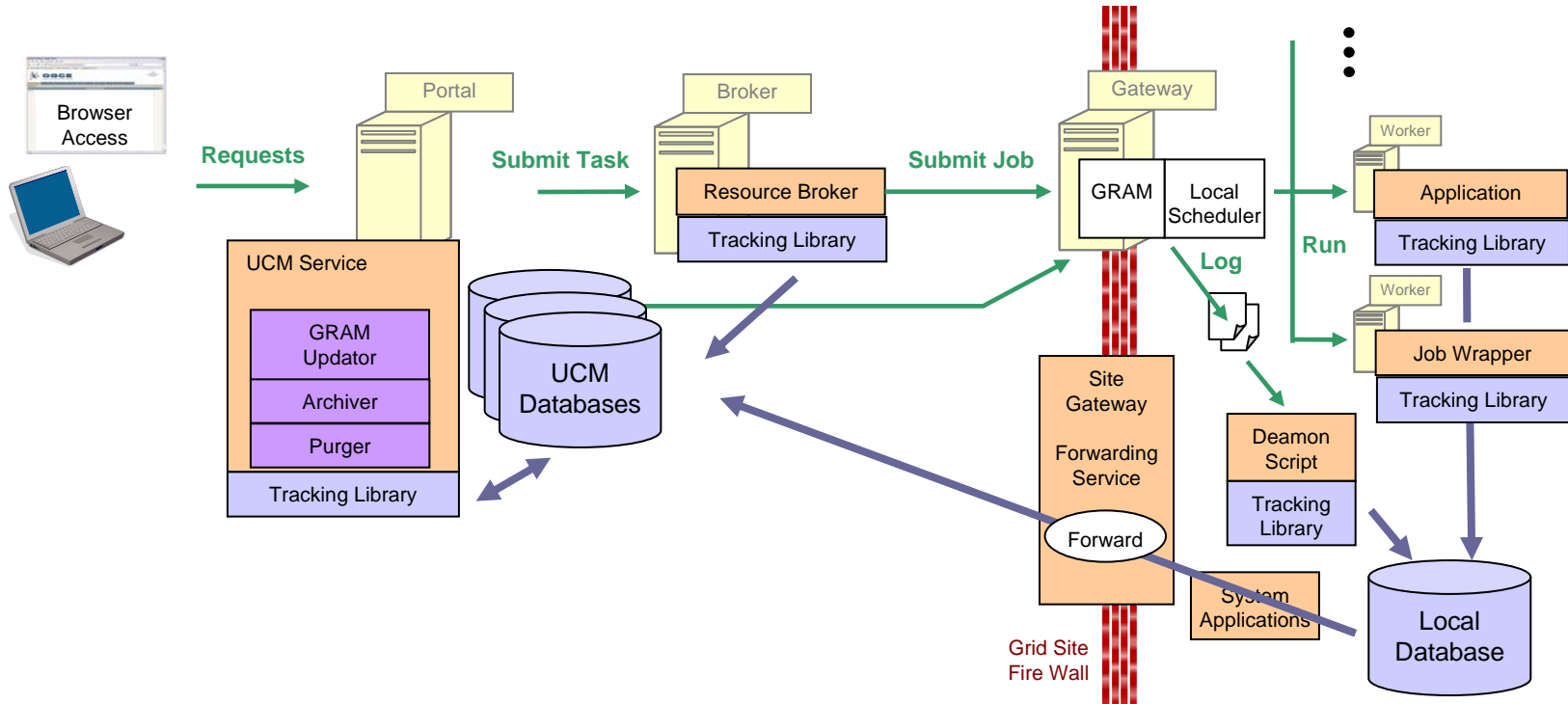
# Tracking Library Hides Data Store Behind High-Level Interface

## 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 Store**
(Database Module)

**Data Store**
(File Module)
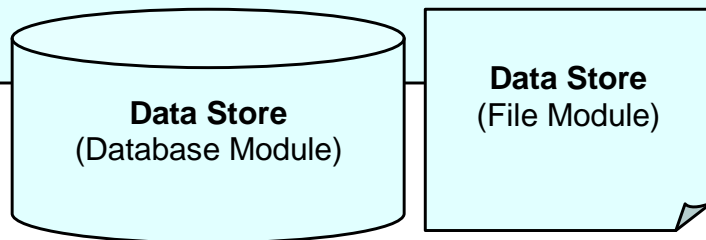
**Data Tier Holds Data Store Module**

## API for Science Applications

```
TxEventLog 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

## TASKS / JOBS / EVENTS

---

**Job Submission Broker** → **Defines Jobs in Task**

**Grid Resource Scheduler** → **Job Level Status Updates**

**Scientific Application** → **Application Event Messages**

# Abstracted Data Store Concept

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
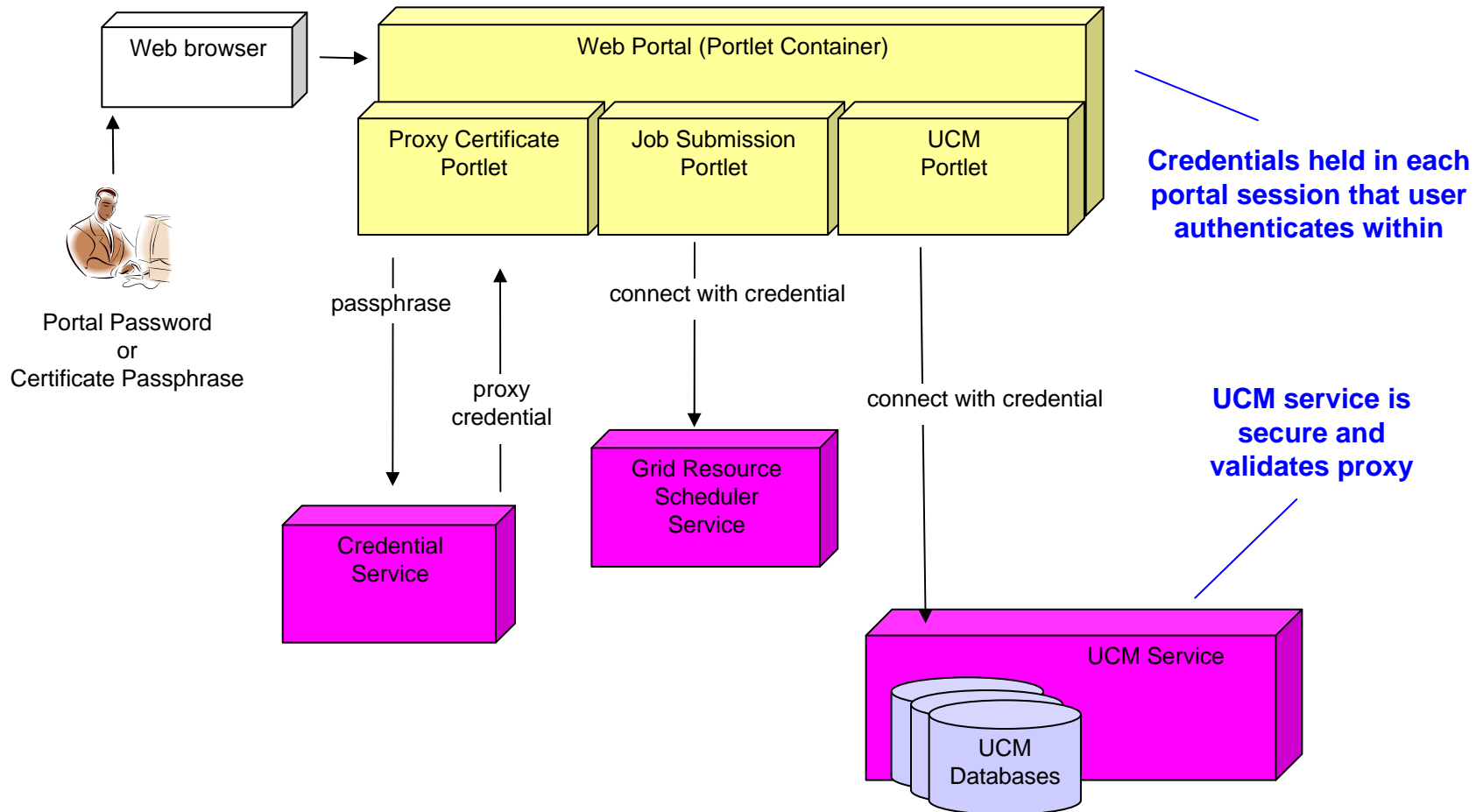- Using dictionary tables

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
- **Prototype portlets be tested with submission to a resource scheduler with the Open Science Grid software stack installed at Tech-X**
- **Access in prototype is currently based on requester's Grid certificate, but access could be hidden by portal**

# UCM Portlet Can Work Along Side Other Portlets



Web browser

Portal Password
or
Certificate Passphrase

Web Portal (Portlet Container)

Proxy Certificate Portlet

Job Submission Portlet

UCM Portlet

**Credentials held in each portal session that user authenticates within**

passphrase

proxy credential

connect with credential

connect with credential

Credential Service

Grid Resource Scheduler Service

**UCM service is secure and validates proxy**
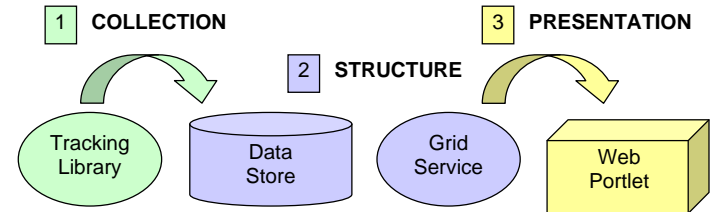
UCM Service

UCM Databases

# AJAX Portlet Prototype Underway

# Summary
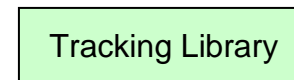
- **User-Centric Monitoring is Needed**

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



- **COLLECTION:**
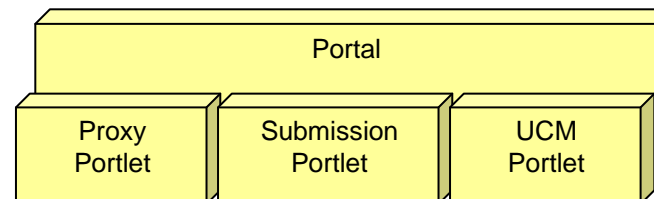  – Versatile Tracking Library



- **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**