



Community Call: October 21, 2013

Overview of EMI WMS

**Steve Johnson
Texas A&M University**



Workflow Management

- Schedule thousands of jobs
- Varying parameter spaces and/or input data
- Simple or complex workflows (e.g, DAGs)
- Utilize multiple grid resources simultaneously
- Extend basic Globus commands
- Provide a friendlier end user experience

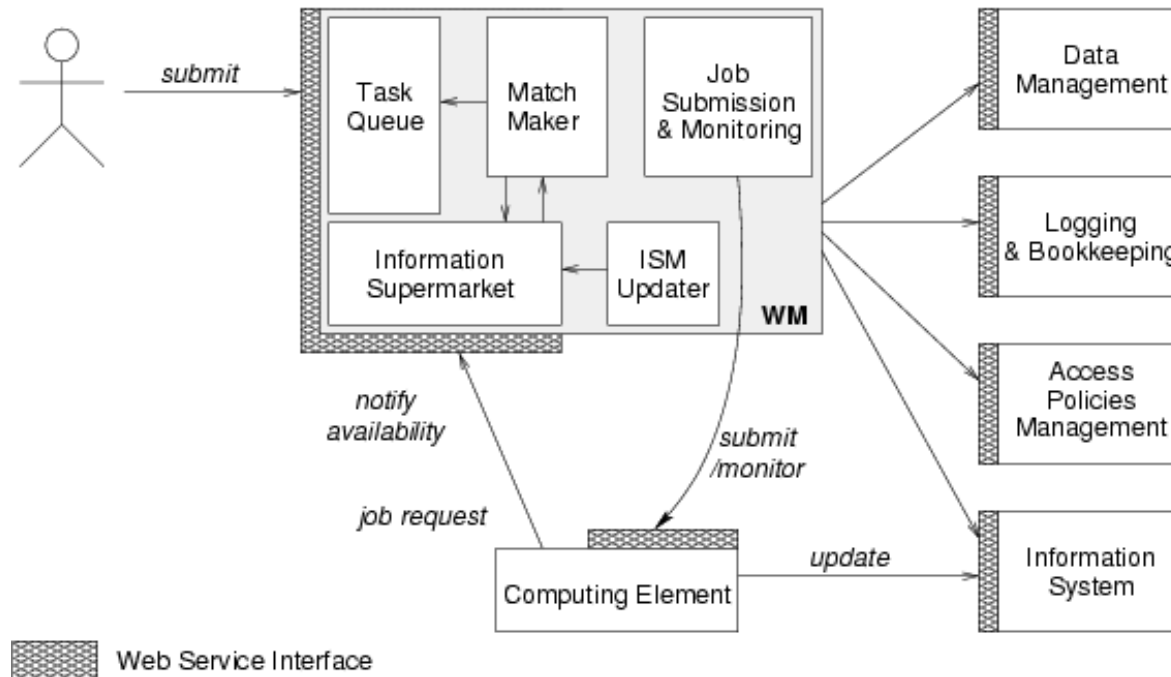
European Middleware Initiative

- www.eu-emi.eu/what-is-emi
- Middleware based largely on LHC activities → homogeneity of data and applications
- Harmonizes existing middleware solutions
ARC + dCache + gLite + UNICORE
- Similar to OSG distribution

EMI WMS

- Workflow Management System scheduling multiple CE's from a single location
- Current version as of Oct 21, 2013, is 3.5.1, released May 21, 2013.
- Workload Manager (WM) is core component
 - Accept and satisfy requests for job management
 - Handles submission and cancellation
 - Evaluates requirements and preferences of job
 - Passes job to appropriate Compute Element (CE) for execution.
- Job Logging and Bookkeeping Service (LB)

Architecture



Source: <https://web2.infn.it/gLiteWMS/index.php/techdoc/howtosandguides/57-archoverview>

Using the WMS

- WMPProxy service provides access to WMS through Web Services interface using CLI or UI.
 - C++ CLI
 - API for C++, Java, Python
- User obtains proxy credential: *voms-proxy-init*
- Credential is delegated to the WMPProxy service
- Job is submitted using the delegated credential
- Job Description Language (JDL)

Using the WMS

- WMproxy is front end web service to WM backend infrastructure.
 - Handles external user interaction
 - Manages user proxy
- JobController backend utilizes CondorG + others
- Job info stored in MySQL

AuthN/AuthZ

- Authentication
 - X.509 Certificates and Proxies, of course
- Authorization
 - Implemented in WMPProxy service via FQAN or DN in allow-deny gacl file, *glite_wms_wmproxy.gacl*
 - Can also use *Argus*

Simple JDL

```
[
  Type = "Job";
  JobType = "Normal";
  Executable = "a.out";
  StdInput = "infile.txt";
  StdOutput = "outfile.txt";
  StdError = "errfile.txt";
  InputSandbox = {"infile1.txt", "/u/myname/a.out"};
  OutputSandbox = {"outfile.txt", "errfile.txt"};
]
```

Submission Commands

- Manually delegate proxy

```
voms-proxy-init -voms suragrid  
glite-wms-delegate-proxy -d mydelegationIDstring
```

- Submit the JDL file

```
glite-wms-job-submit -d mydelegationIDstring -o myidsFile JDL/myjob.JDL
```

- Submit Collection of JDL files from one dir

```
glite-wms-job-submit -d mydelegationIDstring -o myidsFile \  
--collection /u/myname/JDLcoll/
```

- Monitor job status

```
glite-wms-job-status -i myidsFile
```

- Collect Output

```
glite-wms-job-output -i myidsFile -dir /u/myname/joboutput/
```

- Cancel Job

```
glite-wms-job-cancel jobID  
glite-wms-job-cancel -i myidsFile
```

- List all matching CE's for the JDL

```
glite-wms-job-list-match -d mydelegationIDstring -rank JDL/myjob.JDL
```

EMI WMS for SURAgrid?

- Uses different repo from OSG.
 - Can they coexist? Er, no, not on the same machine
 - The EMI WMS should be able to use OSG resources
- Unlike glideinWMS -
 - Does not use pilot job
 - Does not require a service or pilot account on CE
 - Does not require setuid executable to change uid
- Unlike basic CondorG or homegrown scheduler
 - Handles the selection of resources for each job
- SURAgrid would need a WMS & MyProxy server
- We might be able to use OSG BDII for info services

EMI WMS for SURAGrid?

- Users are still bound to a CLI for job control
- Additional burden of a new, but powerful, JDL
- Can this be wrapped in a web interface?
 - Creating simple submission/monitor web interface wouldn't be difficult
 - Supporting complex DAGs obviously more of a challenge
- *There still needs to be interaction between local support staff and researchers!*

Any Takers?

Q?

References

- **EMI WMS User's Guide**
<https://edms.cern.ch/file/674643/1/WMPROXY-guide.pdf>
- **gLite WMS Architecture Overview**
<https://web2.infn.it/gLiteWMS/index.php/techdoc/howtosandguides/57-archoverview>
- **System Administrator Guide**
<https://wiki.italiangrid.it/twiki/bin/view/WMS/WMSSystemAdministratorGuide>
- **Service Reference Card**
<https://wiki.italiangrid.it/twiki/bin/view/WMS/EMIWMSServiceRefCard>
- **JDL Attributes Specification**
<https://edms.cern.ch/file/590869/1/WMS-JDL.pdf>