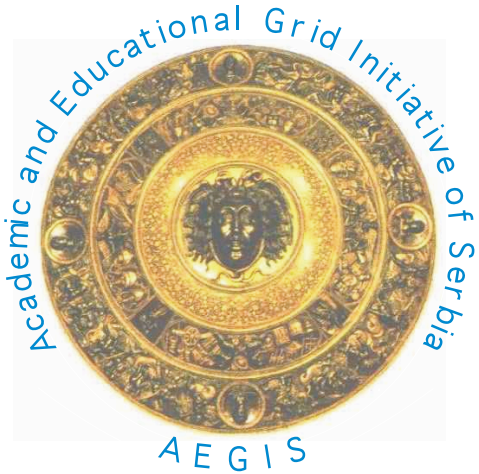
# Basic Job Submission on the Grid

**Antun Balaz**

**Scientific Computing Laboratory**

**Institute of Physics Belgrade**

**http://www.scl.rs/**






SEE-GRID-SCI

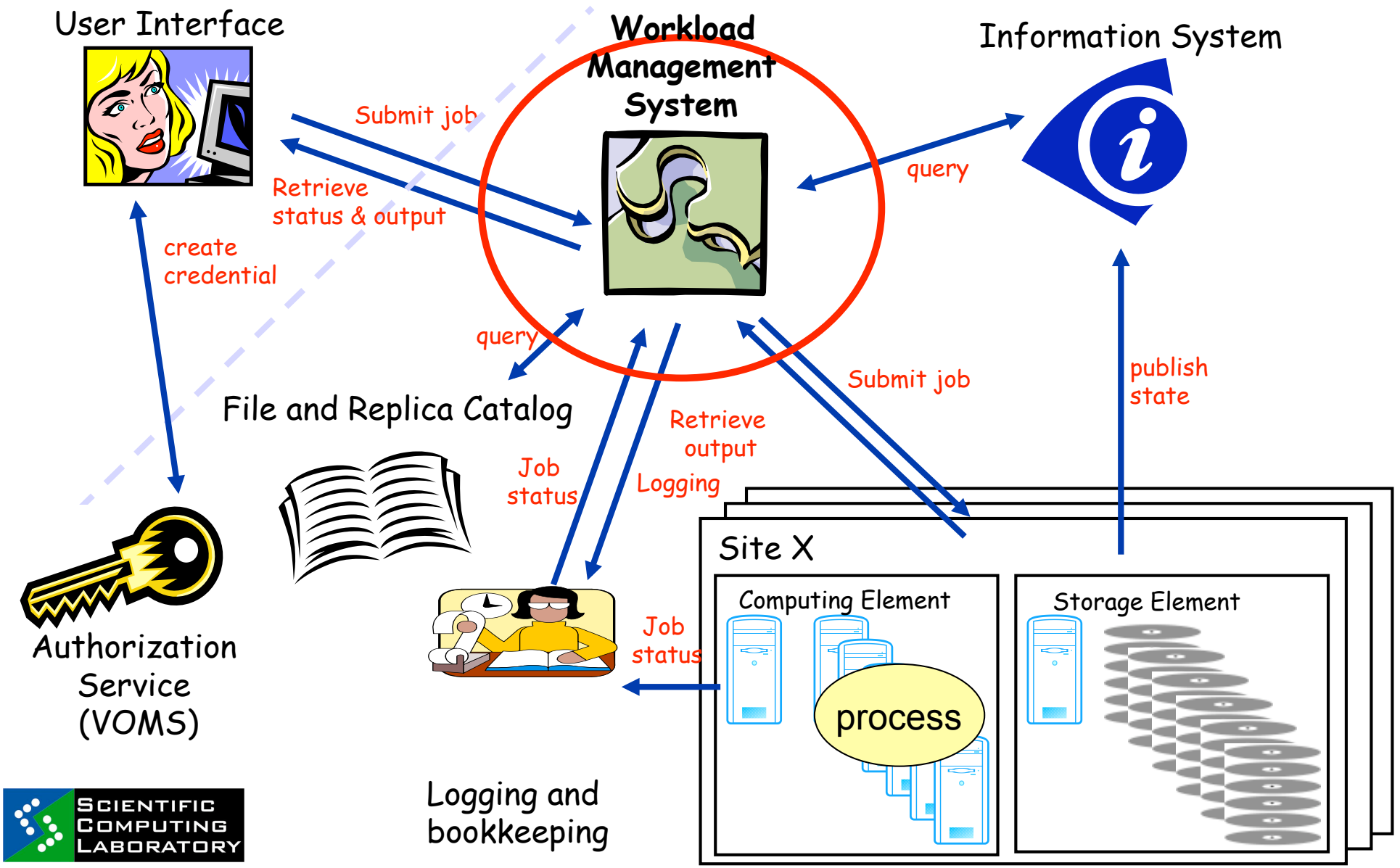
SEE-GRID eInfrastructure for regional eScience



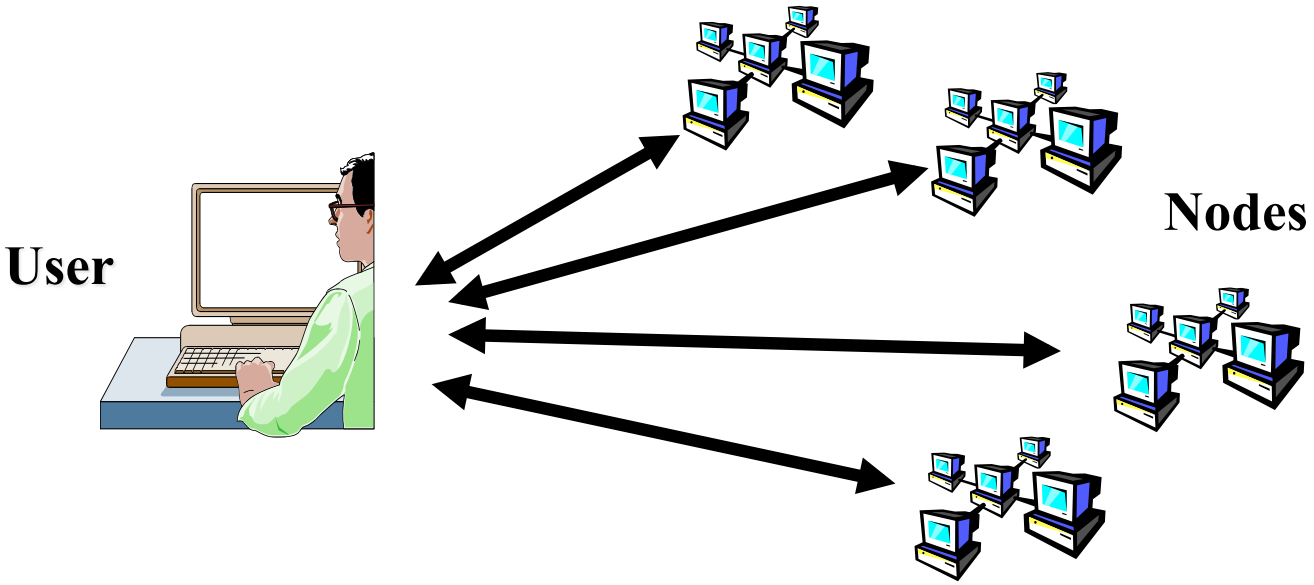

21 Jan – 01 Feb 2009

www.eu-egee.org

# Scope


User Interface
Workload Management System
Information System
Submit job
Retrieve status & output
query
create credential
query
File and Replica Catalog
Submit job
publish state
Retrieve output
Job status
Logging
Site X
Computing Element
Storage Element
process
Job status
Authorization Service (VOMS)
Logging and bookkeeping

# Without WMS…




- **Without the WMS, need direct interaction with nodes**
  - Need to know resource addresses, capabilities
- ***Usually* want a higher level abstraction – submit a job to a Grid not to a CE**

# Basics

## Why does the Workload Management System exist?

- **Grids have**
  - Many users
  - Many jobs – a "job" = an executable you want to run
  - Where many compute nodes are available
  - Workload Management System is a software service that makes running jobs easier for the user
- **It builds on the basic grid services**
  - E.g. Authorisation, Authentication, Security, Information Systems, Job submission
- **Terminology: "Compute element": defined as a batch queue - One cluster can have many queues**

# Which CE do you want to use?

- **Without the WMS, use the Information System to see what's available, then choose…**

  **lcg-infosites --vo aegis ce**

| #CPU | Free | Total Jobs | Running | Waiting | ComputingElement |
|---|---|---|---|---|---|
| 13 | 13 | 0 | 0 | 0 | grid01.rcub.bg.ac.yu:2119/jobmanager-pbs-aegis |
| 85 | 1 | 1836 | 27 | 1809 | ce-atlas.phy.bg.ac.yu:2119/jobmanager-pbs-aegis |
| 3 | 3 | 0 | 0 | 0 | grid01.elfak.ni.ac.yu:2119/jobmanager-pbs-aegis |
| 689 | 41 | 1979 | 200 | 1779 | ce64.phy.bg.ac.yu:2119/jobmanager-pbs-aegis |
| 42 | 15 | 0 | 0 | 0 | cluster1.csk.kg.ac.yu:2119/jobmanager-pbs-aegis |
| 27 | 27 | 0 | 0 | 0 | rti29.etf.bg.ac.yu:2119/jobmanager-pbs-aegis |

- **WMS does this for you!**
  - chooses CE for each job, balances workload, manages jobs and their files

# Information service

- VO-specific information on existing Grid resources

```
lcg-infosites --vo <vo> <option> -v <verbosity> -f <site> --is <bdii>
```

- Other information on existing Grid resources

```
ldapsearch -x -h <hostname> -p 2170 -b "mds-vo-name=local, o=grid"
ldapsearch -x -H ldap://bdii.phy.bg.ac.yu:2170
           -b mds-vo-name=AEGIS01-PHY-SCL,mds-vo-name=local,o=grid
ldapsearch -x -H ldap://ce64.phy.bg.ac.yu:2170
           -b mds-vo-name=AEGIS01-PHY-SCL,o=grid
ldapsearch -x -H ldap://ce64.phy.bg.ac.yu:2170
           -b mds-vo-name=resource,o=grid
ldapsearch -x -H ldap://bdii.phy.bg.ac.yu:2170
           -b mds-vo-name=local,o=grid
           -x "GlueSAAccessControlBaseRule=aegis" GlueChunkKey
ldapsearch -x -H ldap://bdii.phy.bg.ac.yu:2170
           -b mds-vo-name=local,o=grid
           -x "GlueSAAccessControlBaseRule=aegis"
           GlueChunkKey GlueSAStateAvailableSpace
```

SCIENTIFIC COMPUTING LABORATORY

# Job Description Language

- JDL file

```
attribute = expression;
```

- Simple example

```
[
Type = "Job";
Executable = "/bin/hostname";
Arguments = "";
StdOutput = "stdout.txt";
StdError = "stderr.txt";
OutputSandbox = {"stdout.txt","stderr.txt"};
]
```

- Additional attributes

```
InputSandbox = {"test.sh", "fileA", "fileB", ...}
InputSandbox = {
"gsiftp://lxb0707.cern.ch/cms/doe/data/fileA","fileB"};
VirtualOrganisation = "cms";
RetryCount = 0;
MyProxyServer = "myproxy.phy.bg.ac.yu";
```

SCIENTIFIC COMPUTING LABORATORY

# Job Description Language

- **Requirements**

```
Requirements =  RegExp("ce64.phy.bg.ac.yu*”,other.GlueCEUniqueID);
Requirements = Member("VO-cms-CMSSW_2_0_0",
      other.GlueHostApplicationSoftwareRunTimeEnvironment);
Requirements = (other.GlueHostArchitecturePlatformType == "x86_64”);
```

# Advanced job types (tomorrow)

- **Job Collections**
  - Type = "Collection";
- **DAG jobs (Direct Acyclic Graphs)**
  - Type = “Dag";
- **Parametric jobs**
  - JobType = "Parametric";
- **Interactive Jobs**
  - JobType = "Interactive”;
- **MPI Jobs (Message Passing Interface)**
  - JobType = ”MPICH”;

# UI <-> WMS

- ## Single Job Submission

```
glite-wms-job-list-match -a <jdl file>
glite-wms-job-delegate-proxy -d <delegID>
glite-wms-job-submit -a <jdl file>
glite-wms-job-status <jobID>
glite-wms-job-cancel <jobID>
glite-wms-job-output <jobID>
glite-wms-job-logging-info <jobID>
```

# Job submission (1)

## http://wiki.egee-see.org/index.php/SG_Running_Jobs_WMProxy_CLI

- **To delegate a proxy**

```
$ glite-wms-job-delegate-proxy -d <dlgID>
```

- **Delegation of a proxy can be automated, using "-a"**
  - Not a very good idea for submitting a lot of jobs – delegation of the proxy takes time, so using the one delegated can speed up the submission process for many jobs
- **Listing CE that match a job description**

```
$ glite-wms-job-list-match -d <dlgID> <jdl_file>
```

# Job submission (2)

- **To submit a job**

```
$ glite-wms-job-submit -d <dlgID> <jdl_file>
$ glite-wms-job-submit -d <dlgID> -o <job_ID_file>
<jdl_file>
$ glite-wms-job-submit -d <dlgID> -r <CE_ID>
<jdl_file>
```

- **Retrieving status of a job**

```
$ glite-wms-job-status <job_ID>
$ glite-wms-job-status -i <job_ID_file>
```

- **Retrieving the output of a job**

```
$ glite-wms-job-output <job_ID>
$ glite-wms-job-output -i <job_ID_file>
$ glite-wms-job-output -dir <path> <job_ID>
```

- **Canceling a job**

```
$ glite-wms-job-cancel <job_ID>
$ glite-wms-job-cancel -i <job_ID_file>
```

SCIENTIFIC COMPUTING LABORATORY

# User's todo list

1. **Create JDL file**
2. **Create proxy**
3. **(Delegate proxy)**
   - glite-wms-job-delegate-proxy
4. **Check some CEs match your requirements:**
   - glite-wms-job-list-match
5. **Submit job**
   - glite-wms-job-submit
6. **Do something else for a while!**
   **gLite is not written for short jobs!**
7. **Check job status - occasionally**
   - glite-wms-job-status
8. **When job is "done", get output**
   - glite-wms-job-output

SCIENTIFIC COMPUTING LABORATORY

# Practical on job submission

- Create a simple JDL file
- Delegate proxy
- List the CEs that can accept a JDL
- Submit the job (save JobID to a file)
- Check its status until its done
- Retrieve output
- If you have time, try to add some requirements