



Enabling Grids for E-scienceE

# Workload Management System

*Mike Mineter*

*mjm@nesc.ac.uk*

[www.eu-egee.org](http://www.eu-egee.org)



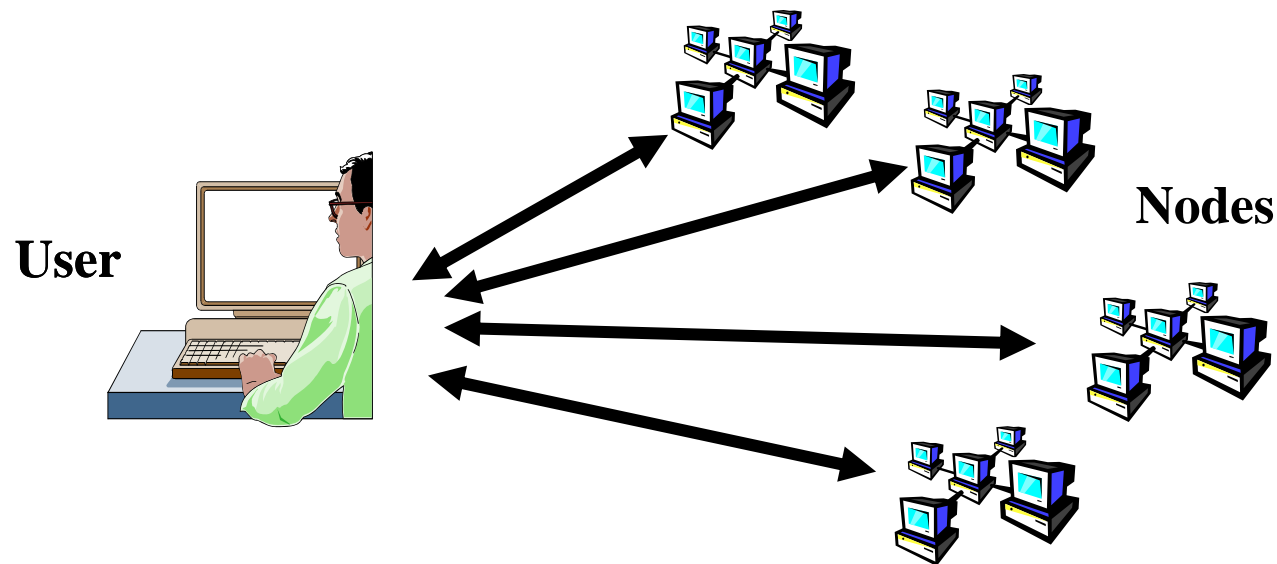
Information Society



INFSO-RI-508833

- **What is the Workload Management System (WMS)?**
- **How do you use it?**
- **Further information**
- **Practicals**





- **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**

## Why does the Workload Management System exist?

- **Grids have**
  - Many users
  - Running 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?

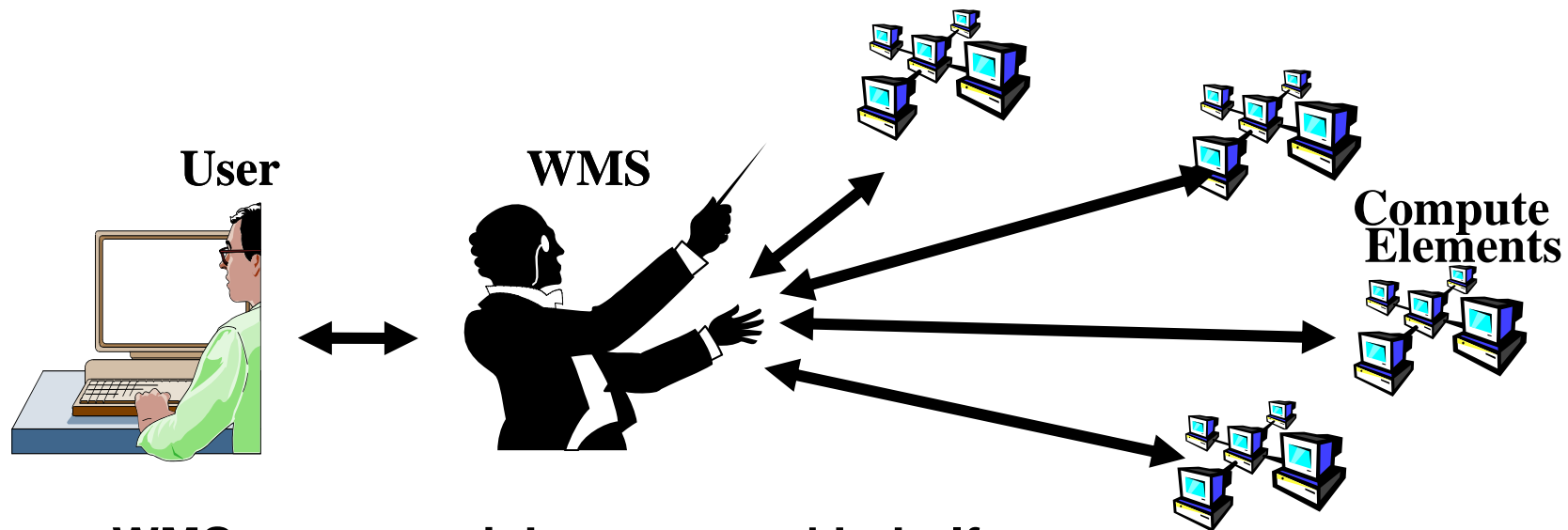
- Without the WMS, use the Information System to see what's available, then choose...

**lcg-infosites --vo gilda ce**

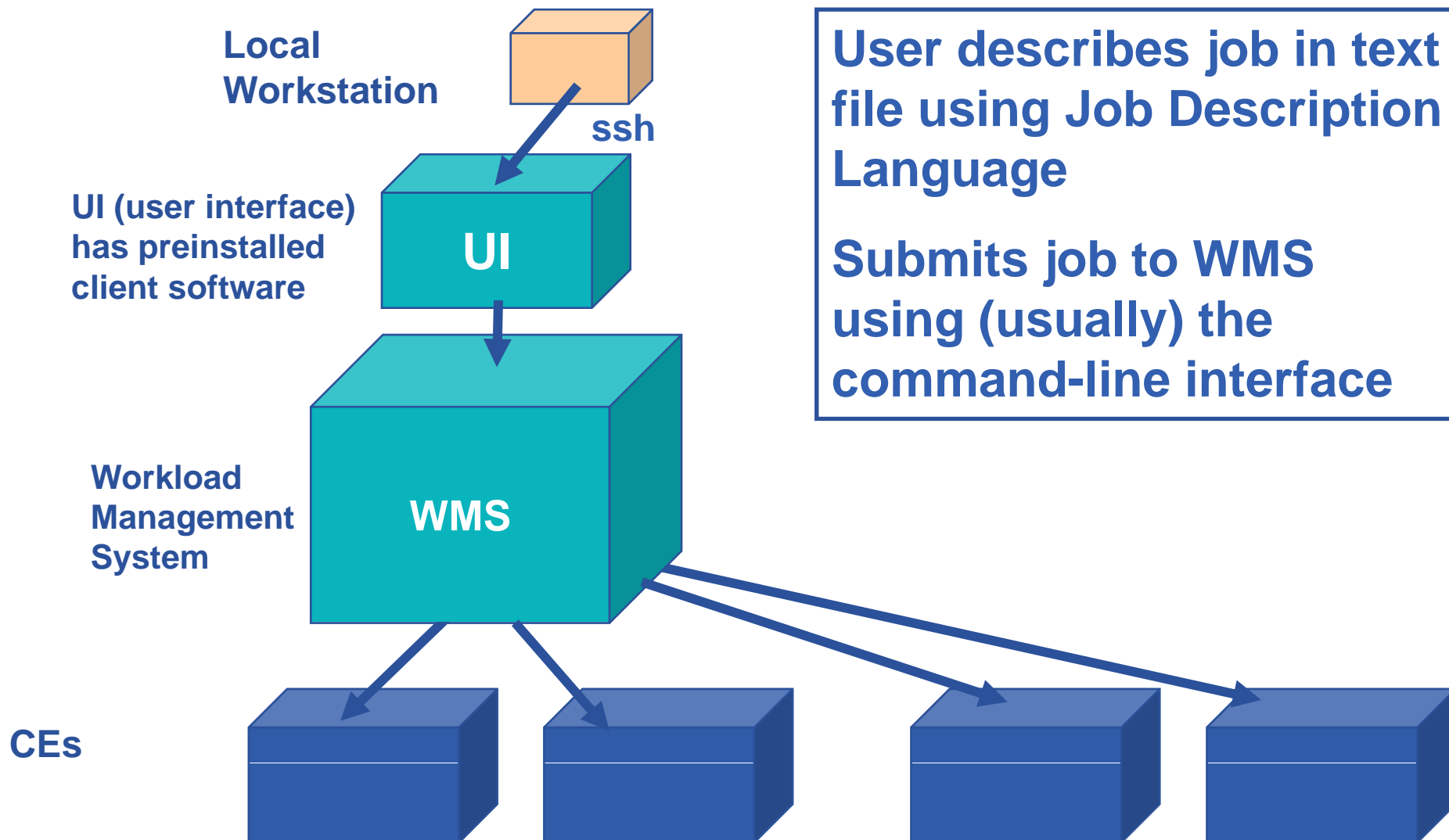
#CPU	Free	Total Jobs	Running	Waiting	ComputingElement
10	10	1	0	1	grid011f.cnaf.infn.it:2119/jobmanager-lcgpbs-short
10	10	0	0	0	grid011f.cnaf.infn.it:2119/jobmanager-lcgpbs-long
10	10	2	0	2	grid011f.cnaf.infn.it:2119/jobmanager-lcgpbs-infinite
48	48	0	0	0	grid010.ct.infn.it:2119/jobmanager-lcgpbs-short
48	48	0	0	0	grid010.ct.infn.it:2119/jobmanager-lcgpbs-long
48	48	0	0	0	grid010.ct.infn.it:2119/jobmanager-lcgpbs-infinite

.....[30% shown].

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



- **WMS manages jobs on users' behalf**
  - User doesn't decide where jobs are run
  - User defines the job and its requirements, WMS matches this with available CEs
- **Effect:**
  - Easier submission
  - Users insulated from change in Compute elements
  - WMS – can optimise your jobs – e.g. which CE?



**User describes job in text file using Job Description Language**

**Submits job to WMS using (usually) the command-line interface**

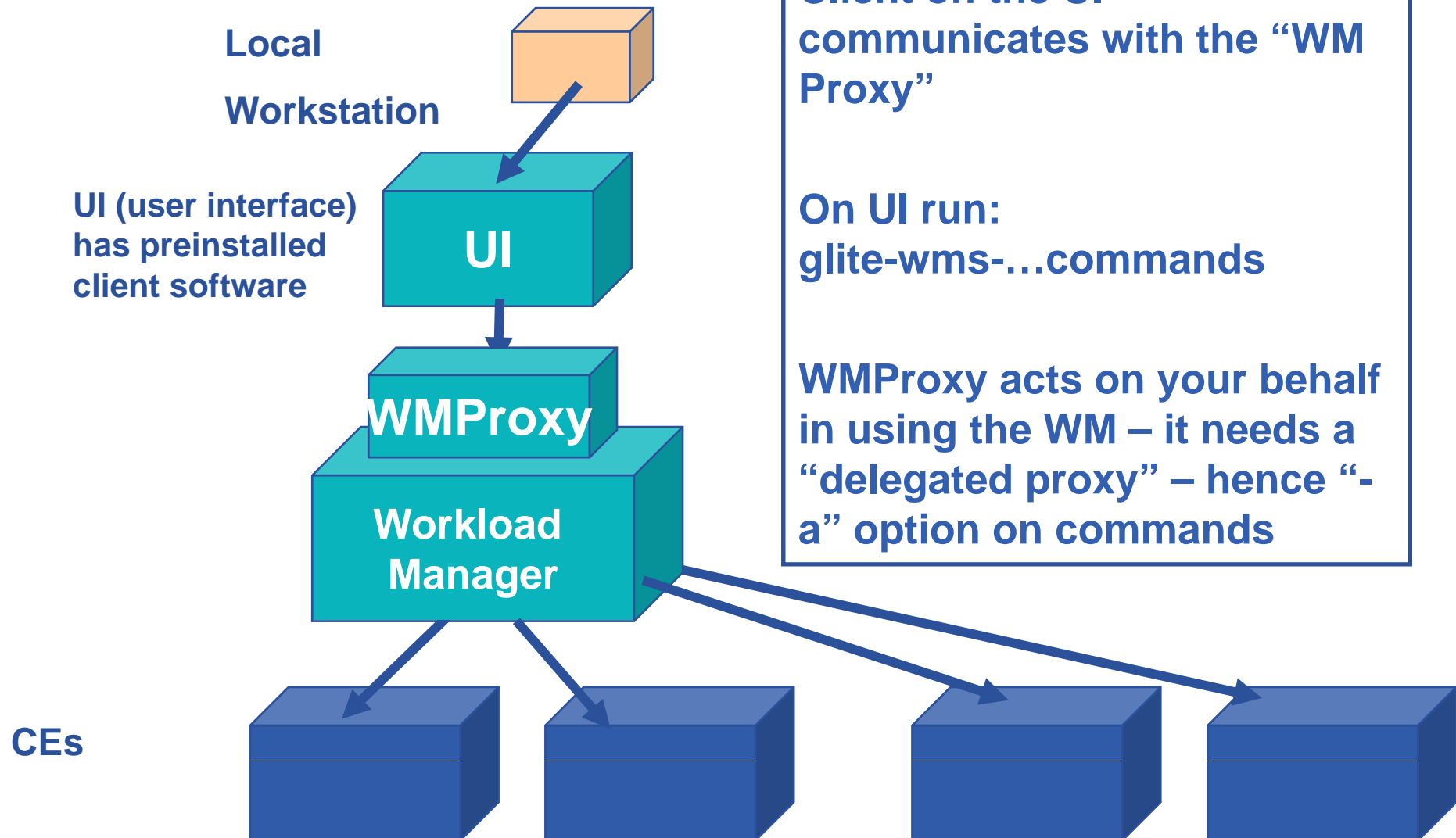
- **Jobs run in batch mode on grids.**
- **Steps in running a job on a gLite grid with WMS:**
  - 1. Create a text file in “Job Description Language”**
  - 2. Optional check: list the compute elements that match your requirements (“list match” command)**
  - 3. Submit the job ~ “glite-wms-job-submit myfile.jdl”  
Non-blocking - Each job is given an id.**
  - 4. Occasionally check the status of your job**
  - 5. When “Done” retrieve output**



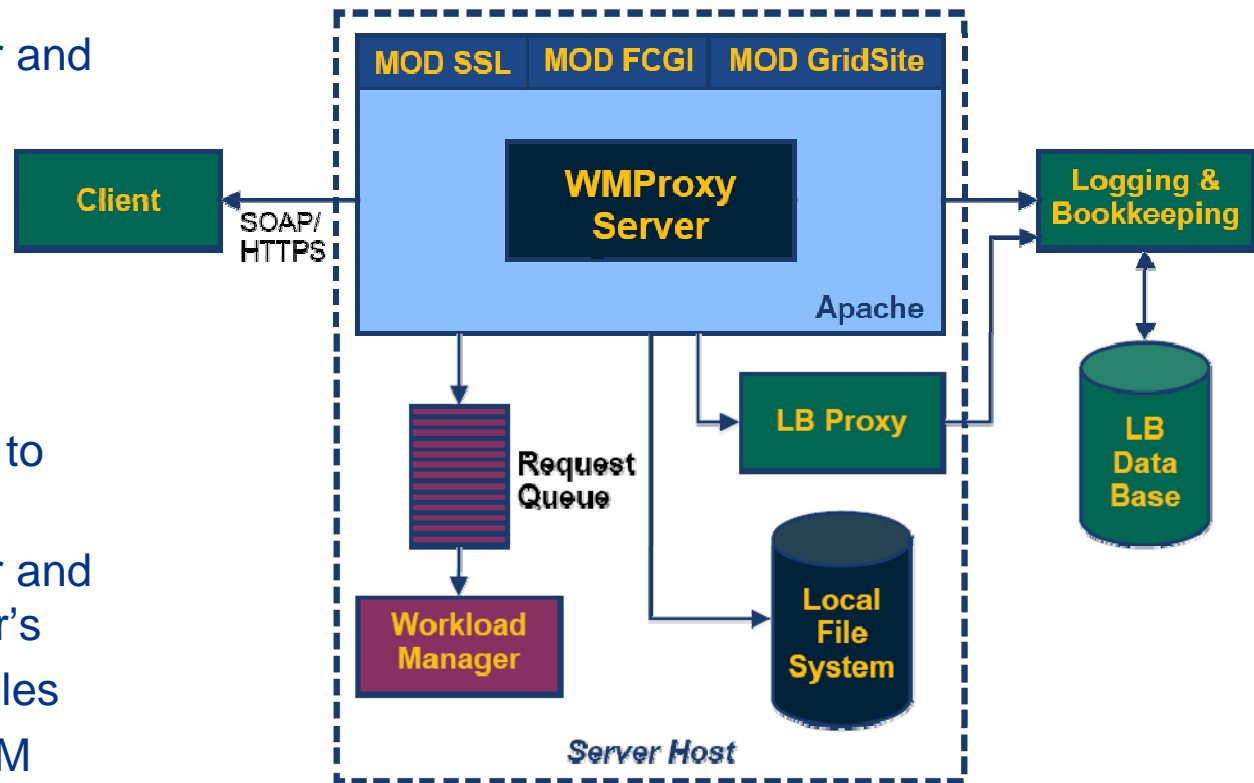
- **Executable** – sets the name of the executable file;
- **Arguments** – command line arguments of the program;
- **StdOutput, StdError** - files for storing the standard output and error messages output;
- **InputSandbox** – set of input files needed by the program, including the executable;
- **OutputSandbox** – set of output files which will be written during the execution, including standard output and standard error output; these are sent from the CE to the WMS for you to retrieve
- **ShallowRetryCount** – in case of grid error, retry job this many times (“Shallow”: before job is running)

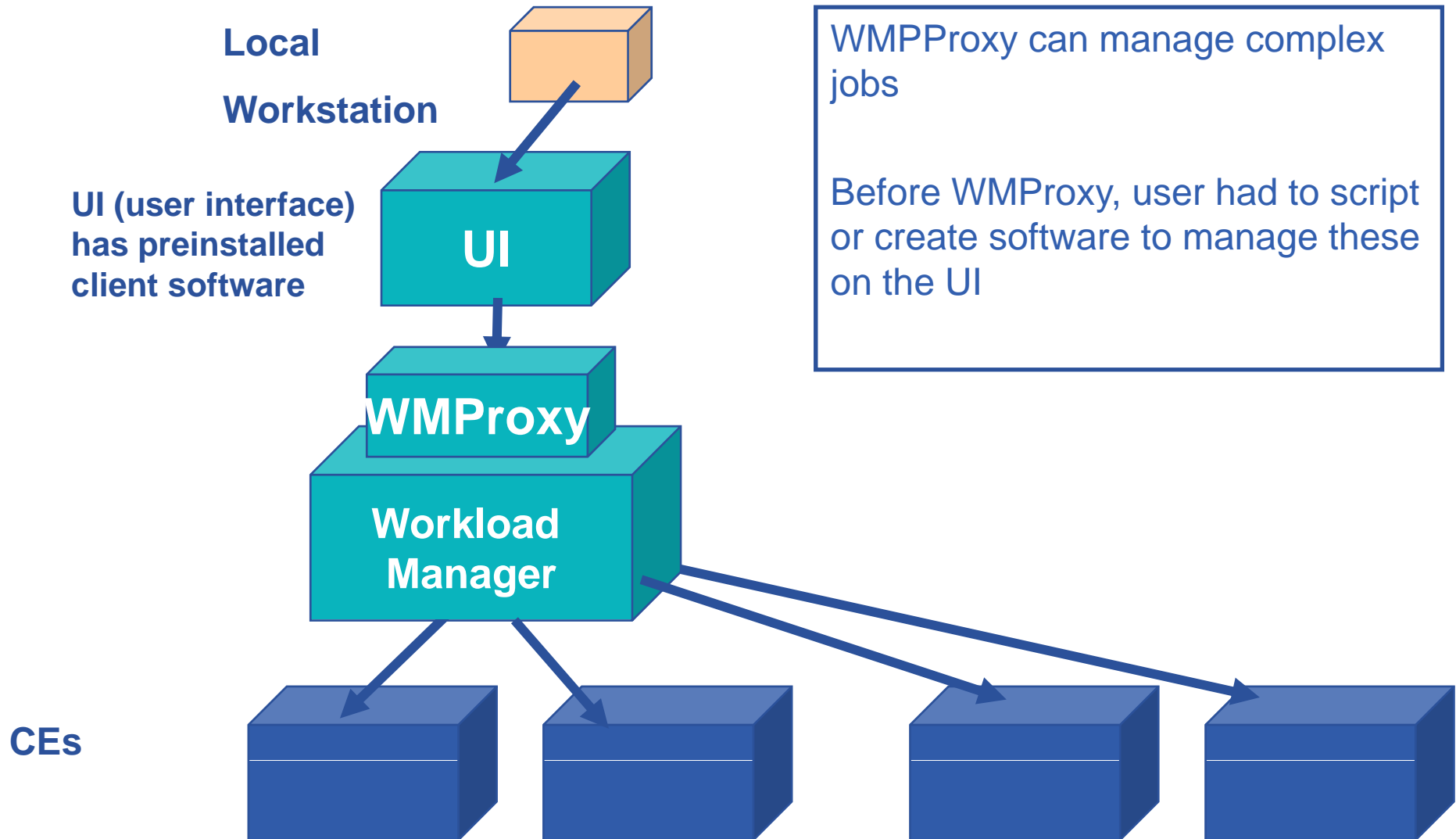
```
Executable = "gridTest";
StdError = "stderr.log";
StdOutput = "stdout.log";
InputSandbox = {"/home/joda/test/gridTest"};
OutputSandbox = {"stderr.log", "stdout.log"};
Requirements = other.GlueCEPolicyMaxCPUTime >
    480;
ShallowRetryCount = 3;
```

Flag	Meaning
SUBMITTED	submission logged in the Logging & Bookkeeping service
WAIT	job match making for resources
READY	job being sent to executing CE
SCHEDULED	job scheduled in the CE queue manager
RUNNING	job executing on a Worker Node of the selected CE queue
DONE	job terminated without grid errors
CLEARED	job output retrieved
ABORT	job aborted by middleware, check <i>reason</i>

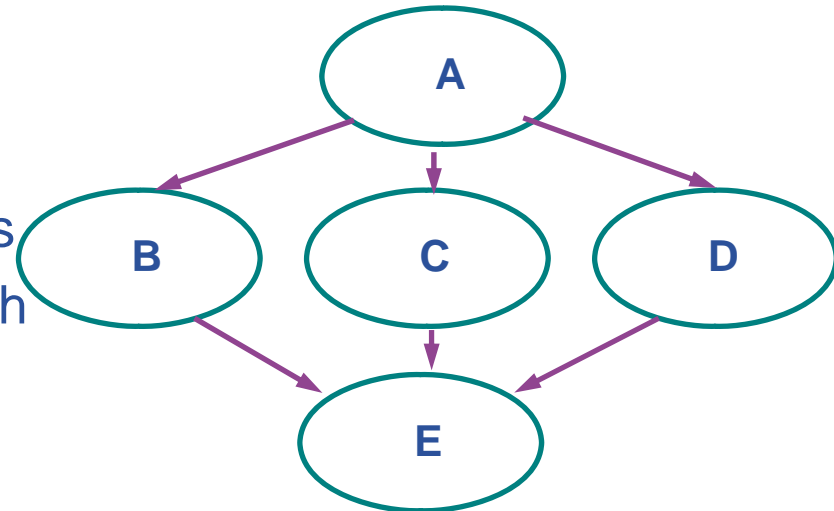


- WMPProxy is a SOAP Web service providing access to the Workload Management System (WMS)
- Job characteristics specified via JDL
  - jobRegister
    - create id
    - map to local user and create job dir
    - register to L&B
    - return id to user
  - input files transfer
  - jobStart
    - register sub-jobs to L&B
    - map to local user and create sub-job dir's
    - unpack sub-job files
    - deliver jobs to WM





- **Direct Acyclic Graph (DAG)** is a set of jobs where the input, output, or execution of one or more jobs depends on one or more other jobs
- **A Collection** is a group of jobs with no dependencies
  - basically a collection of JDL's
  - Can have common sandbox
- **A Parametric job** is a job having one or more attributes in the JDL that vary their values according to parameters
- **It is possible to have one shot submission of a (possibly very large, up to thousands) group of jobs**
  - Submission time reduction
    - Single call to WMPProxy server
    - Single Authentication and Authorization process
    - Sharing of files between jobs
  - Availability of both a single Job Id to manage the group as a whole and an Id for each single job in the group



- For simple jobs: **glite-wms-...** becoming the recommended way to use the WMS
- **History:**
  - Before the **glite-wms-** commands we had **glite-** commands
    - used the WMS without WMPProxy
  - Before the **glite-** commands we had
    - **edg-** commands (edg-job-submit....)
      - *European Data Grid – project before EGEE*
    - Used the “resource broker”
    - Still very widely used
  - You might see these commands still in use.
- **Status**
  - Complex jobs with WMPProxy: not yet in routine production use
  - Watch for news!



- **gLite Users Guide**
  - Follow <http://www.glite.org> and “Documentation”
- **GILDA wiki**
  - We are using some of these pages
  - <https://grid.ct.infn.it/twiki/bin/view/GILDA/>
- **EGEE Digital Library** <http://egee.lib.ed.ac.uk/>

- **You will need to be know UNIX (Linux) a bit – edit files and commands**
  - Work with someone if this is new to you
- **Follow links on the agenda page**
  - “Practical\_1a”
    - Create a simple JDL file
    - List the CEs that can accept it
    - Submit it
    - Check its status until its done
    - Retrieve output
  - “Practical\_1b”
    - Uses a script – time for that one exercise only from this page
  - Practical 2: complex jobs - see the benefit of WMPProxy

