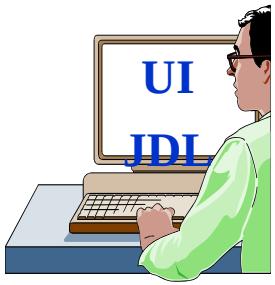
Workload Management System and Job Description Language

Contents

- Reminder of the main grid services
- A closer look at Workload Management System (WMS) and its Resource Broker (RB)

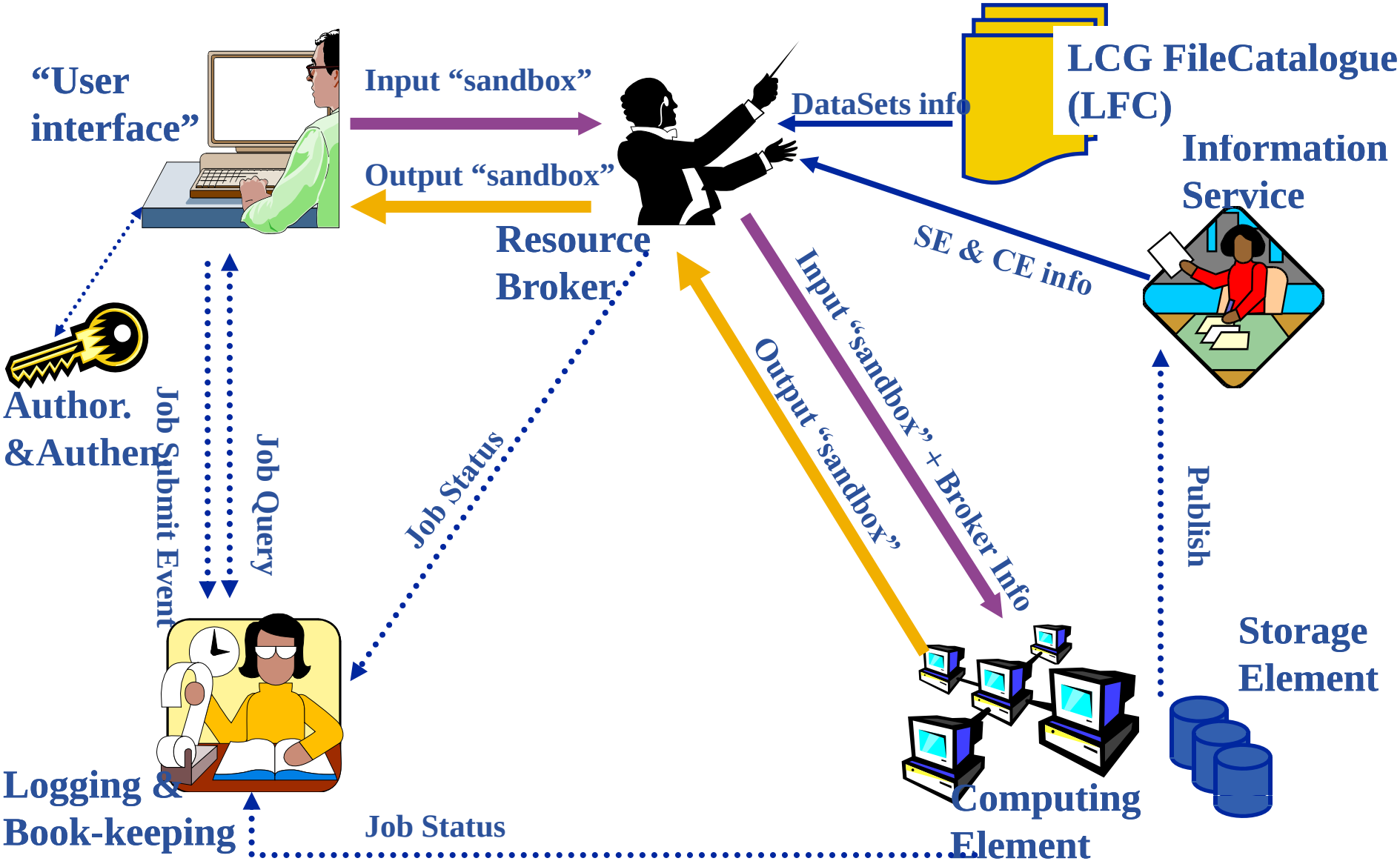
User Interface node

- **The user's interface to the Grid**
- **Command-line interface to**
  - Create proxy with VOMS extensions
  - Job operations – (non-blocking batch mode)
    - To submit a job
    - Monitor its status
    - Retrieve output
  - Data operations on files
  - Other grid services
- **Also C++ and Java APIs**



- **To run a job user creates a JDL (Job Description Language) file**

Current production middleware



Basic JDL example

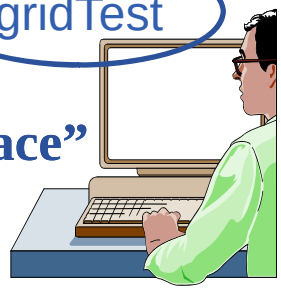
- Submit job to grid via the “resource broker (RB)”,
- edg-job-submit *my.jdl*

Returns a “job-id” used to monitor job, retrieve output

Example JDL file

```
Executable = “gridTest”;
StdError = “stderr.log”;
StdOutput = “stdout.log”;
InputSandbox = {“/home/joda/test/gridTest”};
OutputSandbox = {“stderr.log”, “stdout.log”};
```

Current production middleware

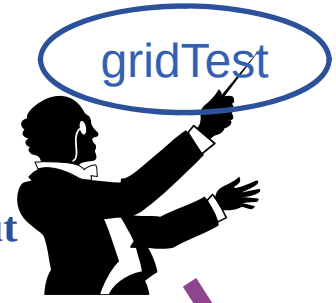


edg-job-submit…

Input "sandbox"

edg-job-get-output

Output "sandbox"



stderr.log
stdout.log

stderr.log
stdout.log

```
Executable = "gridTest";
StdError = "stderr.log";
StdOutput = "stdout.log";
InputSandbox = {"/home/joda/test/gridTest"};
OutputSandbox = {"stderr.log", "stdout.log"};
```

Input "sandbox"

Output "sandbox"

Computing Element

A worker node is allocated by the local jobmanager

- *STD input stream is read from file*
- *STD out and err. streams are redirected into files*

stderr.log
stdout.log
gridTest

Building on basic tools and Information Service

- Submit job to grid via the “resource broker (RB)”,
- edg-job-submit *my.jdl*

Returns a “job-id” used to monitor job, retrieve output

Example JDL file

```
Executable = “gridTest”;
StdError = “stderr.log”;
StdOutput = “stdout.log”;
InputSandbox = {“/home/joda/test/gridTest”};
OutputSandbox = {“stderr.log”, “stdout.log”};
InputData = “lfn:/grid/gilda/training/testbed0-00019”;

Requirements = other.Architecture==“INTEL” && \
               other.OpSys==“LINUX”;
Rank = “other.GlueHostBenchmarkSF00”;
```

Building on basic tools and Information Service

- Submit job to grid via the “resource broker (RB)”,
- edg-job-submit *my.jdl*

Returns a “job-id” used to mo

```
                    Example JD
Execu
StdEr
StdOu
InputSandbox = { /      joda/test
OutputSandbox = {“s   err.log”, “stdout   g”};
InputData = “lfn:/grid/VOname/mydir/testbed0-00019”;

Requirements = other.Architecture
               other.OpSys==“LINUX”
Rank = “other.GlueHostBenchmarkSF
```

lfn: logical file name

RB uses File Catalog to find file location

The file itself is NOT transferred by the middleware! Your binary must transfer input/output grid files!

Higher level tools can tranfer the file for you. E.g. P-GRADE

Building on basic tools and Information Service

- Submit job to grid via the “resource broker”,
- edg-job-submit *my.jdl*

Returns a “job-id” used to monitor job, retrieve output

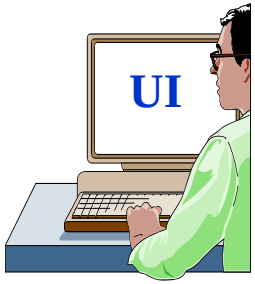
Example JDL file

```
Executable = “gridTest”;
StdError = “stderr.log”;
StdOutput = “stdout.log”;
InputSandbox = {“/home/joda/test/gridTest”};
OutputSandbox = {“stderr.log”, “stdout.log”};
InputData = “lfn:/grid/VOname/jdir/testbed0.00019”;

Requirements = other.Architecture==“INTEL” && \
               other.OpSys==“LINUX” && other.FreeCpus >=4;
Rank = “other.GlueHostBenchmarkSF00”;
```

Uses Information System

Job submission



RB node

Network Server

Workload Manager

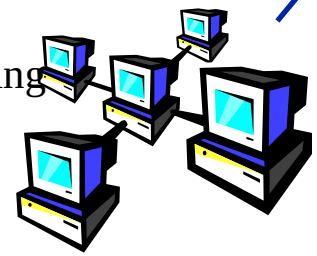
Job Contr. - CondorG

LFC

Inform. Service

CE characts & status

SE characts & status

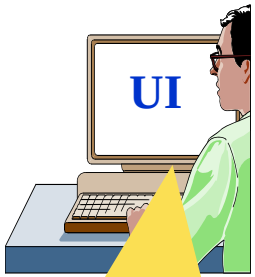


Computing Element

Storage Element

Job Status

RB node



Network Server

Workload Manager

Job Contr. - CondorG

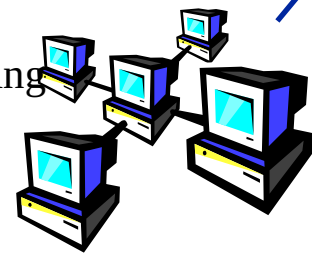
LFC

Inform. Service

UI: allows users to access the functionalities of the WMS (via command line, GUI, C++ and Java APIs)
WMS: Workload Management System

CE characts & status

SE characts & status



Computing Element

Storage Element

edg-job-submit myjob.jdl

my.jdl

```
Executable = “gridTest”;
StdError = “stderr.log”;
StdOutput = “stdout.log”;
InputSandbox = {“/home/joda/test/gridTest”};
OutputSandbox = {“stderr.log”, “stdout.log”};
InputData = “lfn:/grid/VOname/mydir/testbed0.00019”;

Requirements = other.Architecture==“INTEL” && \
               other.OpSys==“LINUX” && other.FreeCpus >=4;
Rank = “other.GlueHostBenchmarkSF00”;
```

Job Status

submitted

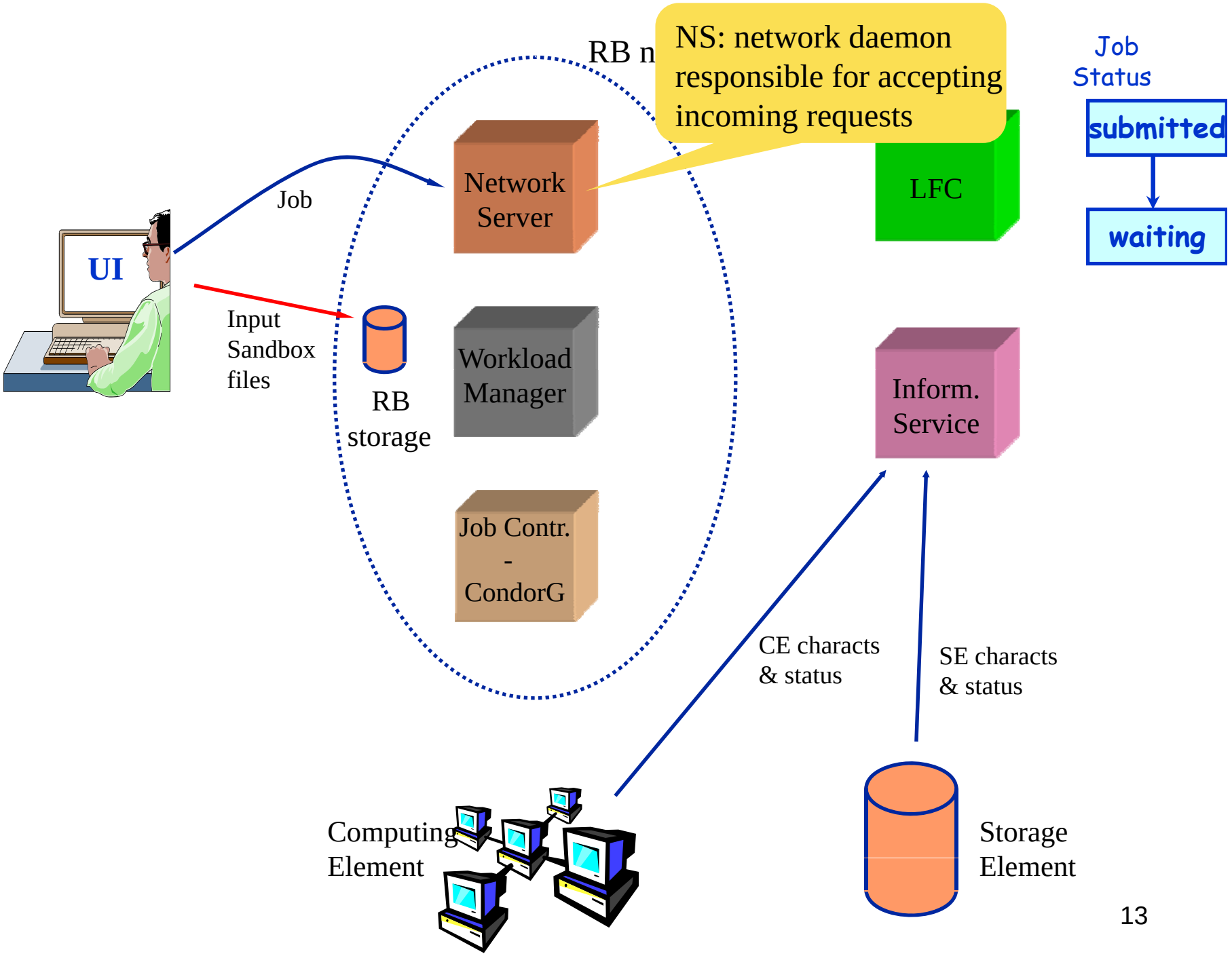
CondorG

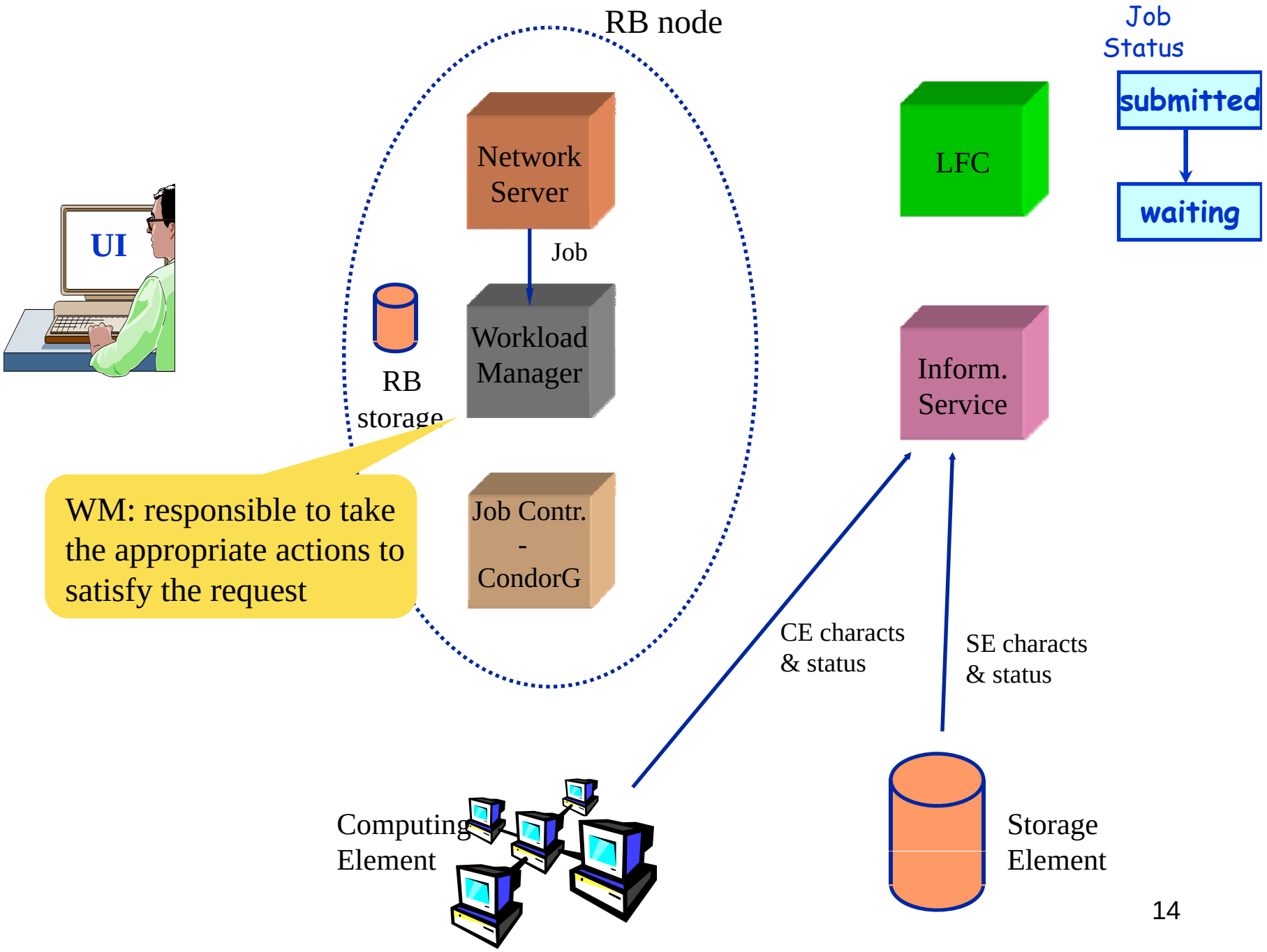
CE characts & status

Job Description Languag (JDL) to specify job characteristics and requirements

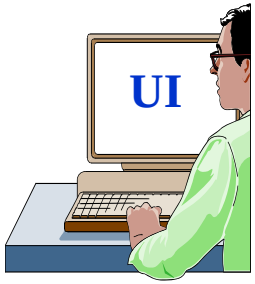
Computing Element

Storage Element


RB n
NS: network daemon responsible for accepting incoming requests
Job Status
submitted
waiting
Network Server
LFC
Job
UI
Input Sandbox files
RB storage
Workload Manager
Inform. Service
Job Contr. - CondorG
CE characts & status
SE characts & status
Computing Element
Storage Element


RB node
Job Status
submitted
waiting
UI
Network Server
Job
LFC
Workload Manager
RB storage
Inform. Service
WM: responsible to take the appropriate actions to satisfy the request
Job Contr. - CondorG
CE characts & status
SE characts & status
Computing Element
Storage Element

Job submission



RB node

Network Server

Match-Maker/ Broker

LFC

Job Status

submitted

waiting

RB storage

Workload Manager

Where must this job be executed ?

Inform. Service

Job Contr. - CondorG

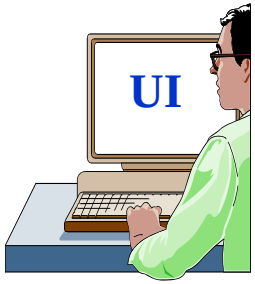
CE characts & status

SE characts & status

Computing Element

Storage Element

Job submission



RB node

Matchmaker: responsible to find the “best” CE where to submit a job

Match-Maker/ Broker

LFC

Job Status

submitted

waiting

RB storage

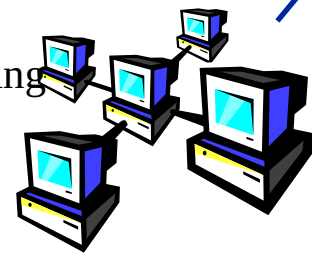
Workload Manager

Inform. Service

Job Contr. - CondorG

CE characts & status

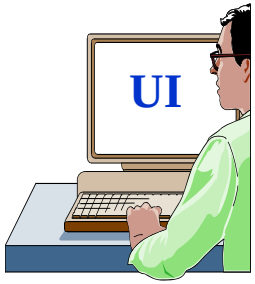
SE characts & status



Computing Element

Storage Element

Job submission



RB node

Network Server

Workload Manager

RB storage

Job Contr. - CondorG

Match-Maker/ Broker

Where is the needed InputData ?

LFC

Job Status

submitted

waiting

Inform. Service

What is the status of the Grid ?

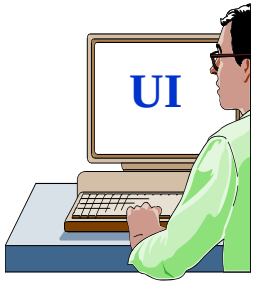
CE characts & status

SE characts & status

Computing Element

Storage Element

Job submission



RB node

Network Server

Match-Maker/ Broker

Workload Manager

CE choice

RB storage

Job Contr. - CondorG

LFC

Inform. Service

Job Status

submitted

waiting

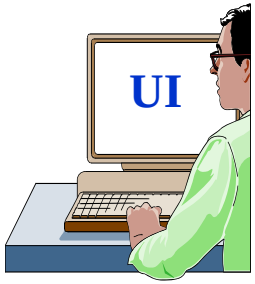
CE characts & status

SE characts & status

Computing Element

Storage Element

Job submission



RB node

Network Server

Workload Manager

RB storage

Job Contr. - CondorG

Job Adapter

LFC

Inform. Service

Job Status

submitted

waiting

JA: responsible for the final “touches” to the job before performing submission (e.g. creation of wrapper script, etc.)

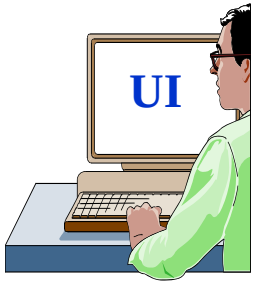
aracts
us

SE characts & status

Computing Element

Storage Element

Job submission



RB node

Network Server

RB storage

Workload Manager

Job

Job Contr. - CondorG

JC: responsible for the actual job management operations (done via CondorG)

LFC

Inform. Service

CE characts & status

SE characts & status

Computing Element

Storage Element

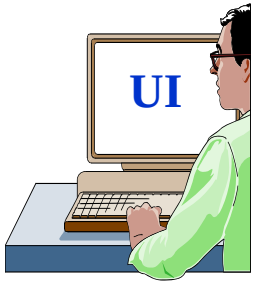
Job Status

submitted

waiting

ready

Job submission



RB node

Network Server

Workload Manager

RB storage

Job Contr. - CondorG

Input Sandbox files

Job

Computing Element

LFC

Inform. Service

CE characts & status

SE characts & status

Storage Element

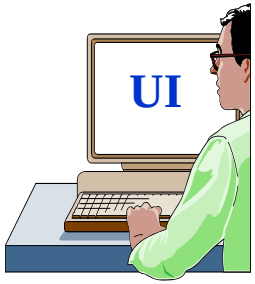
Job Status

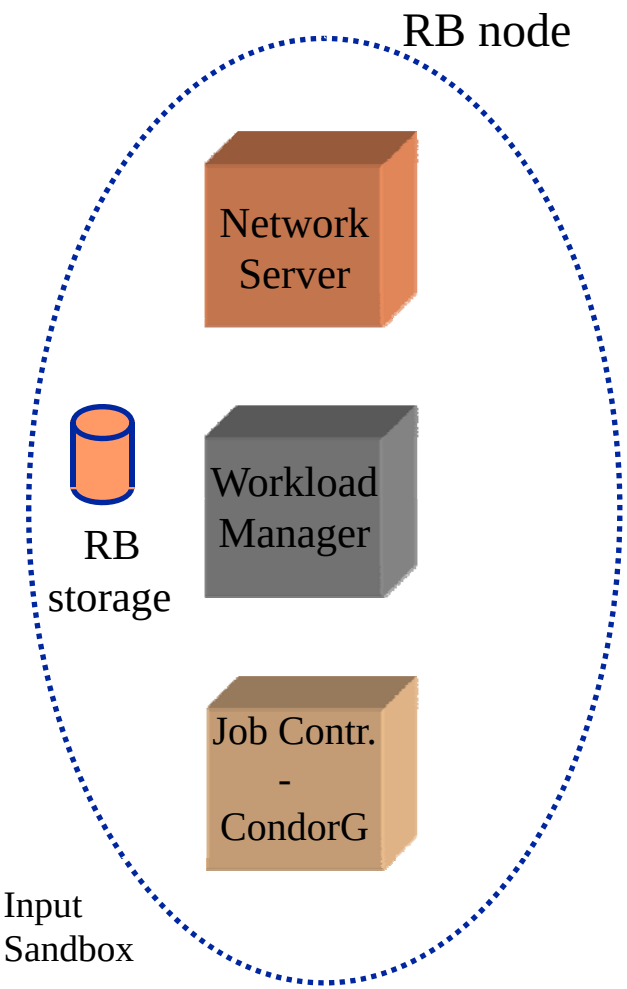
submitted

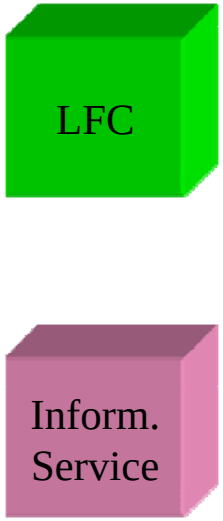
waiting

ready

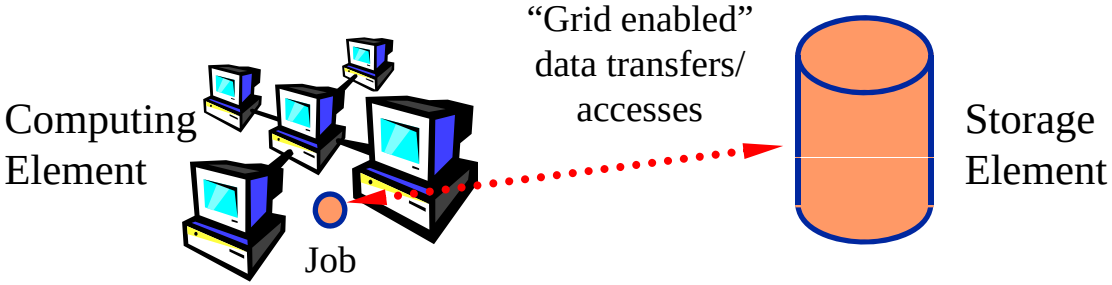
scheduled

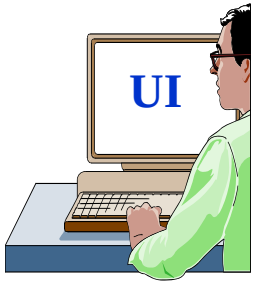

UI

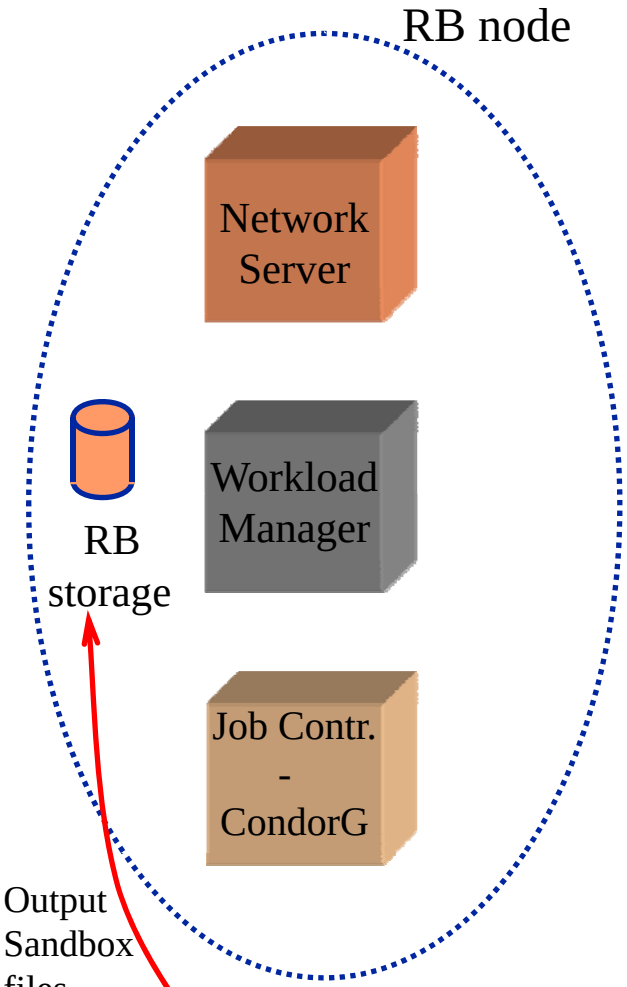

RB node
Network Server
RB storage
Workload Manager
Job Contr. - CondorG
Input Sandbox

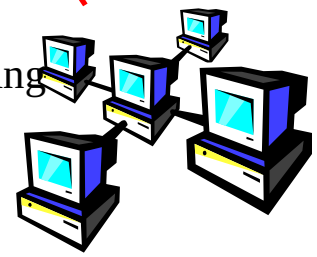

LFC
Inform. Service

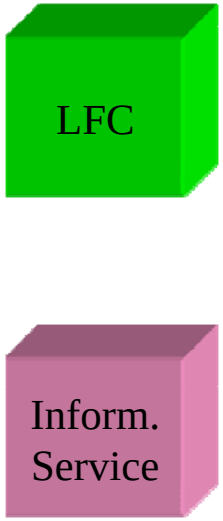
Job Status
submitted
waiting
ready
scheduled
running

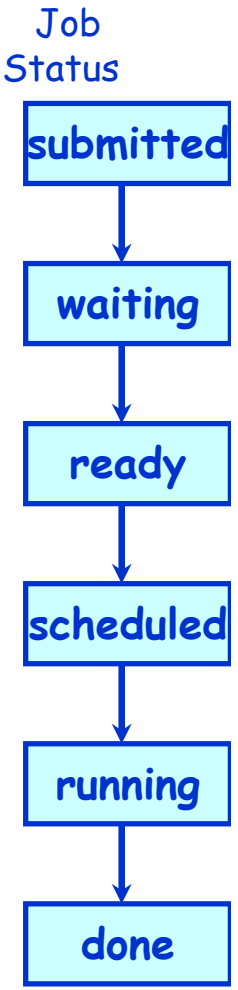

Computing Element
Job
"Grid enabled" data transfers/ accesses
Storage Element

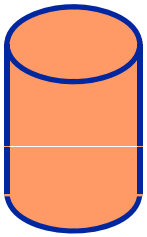

UI

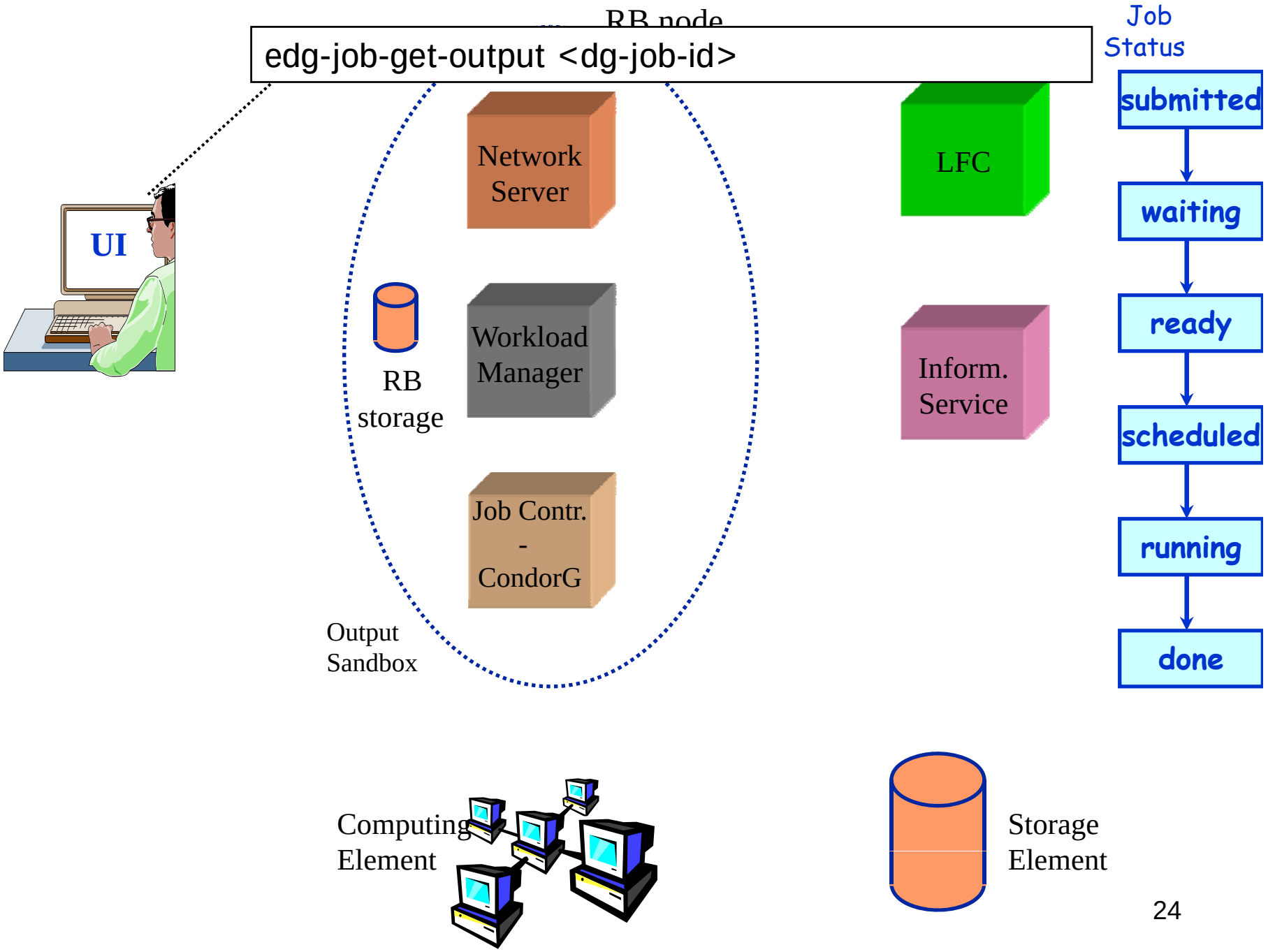

RB node
Network Server
Workload Manager
Job Contr. - CondorG
RB storage
Output Sandbox files

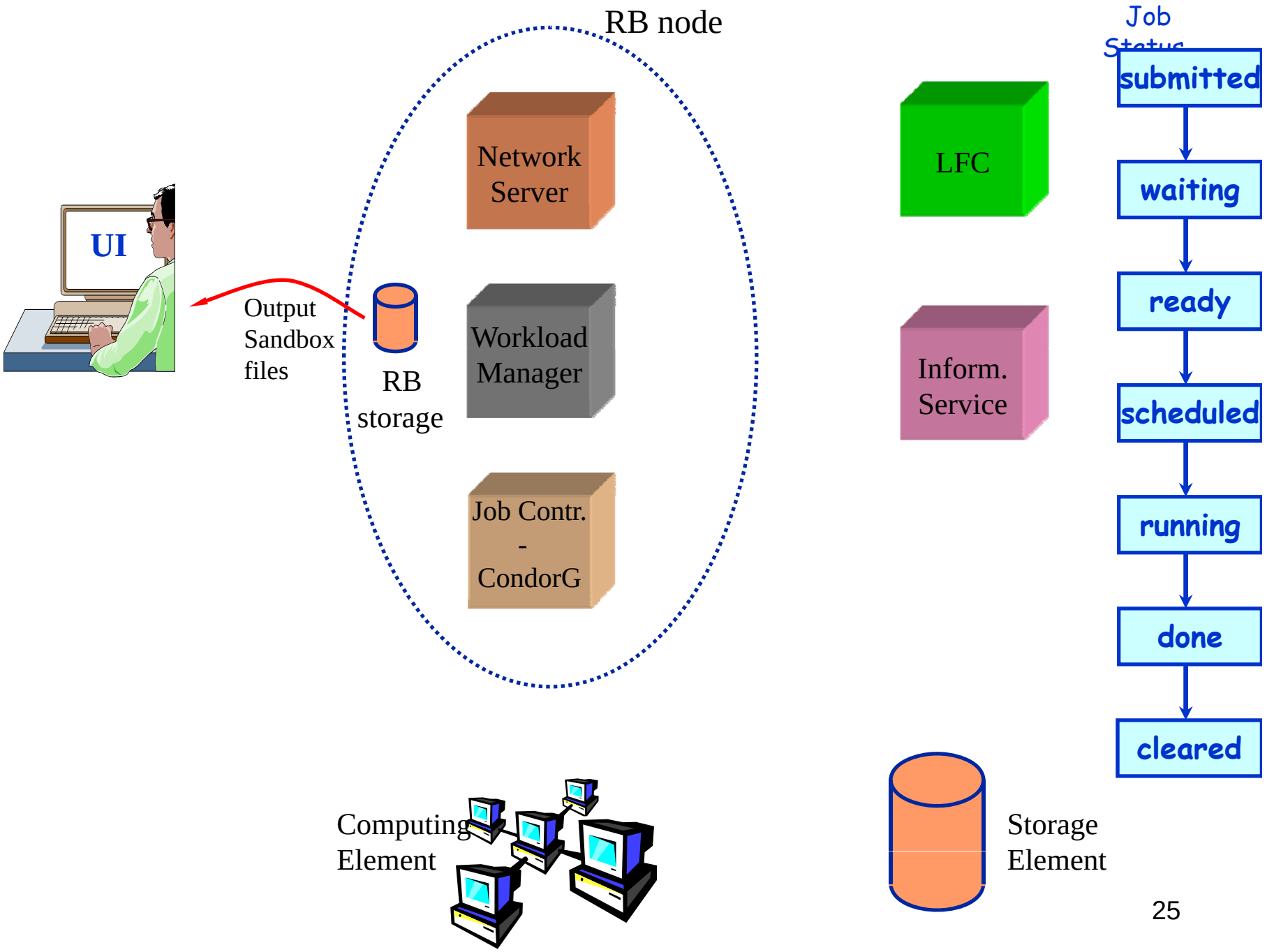

Computing Element


LFC
Inform. Service

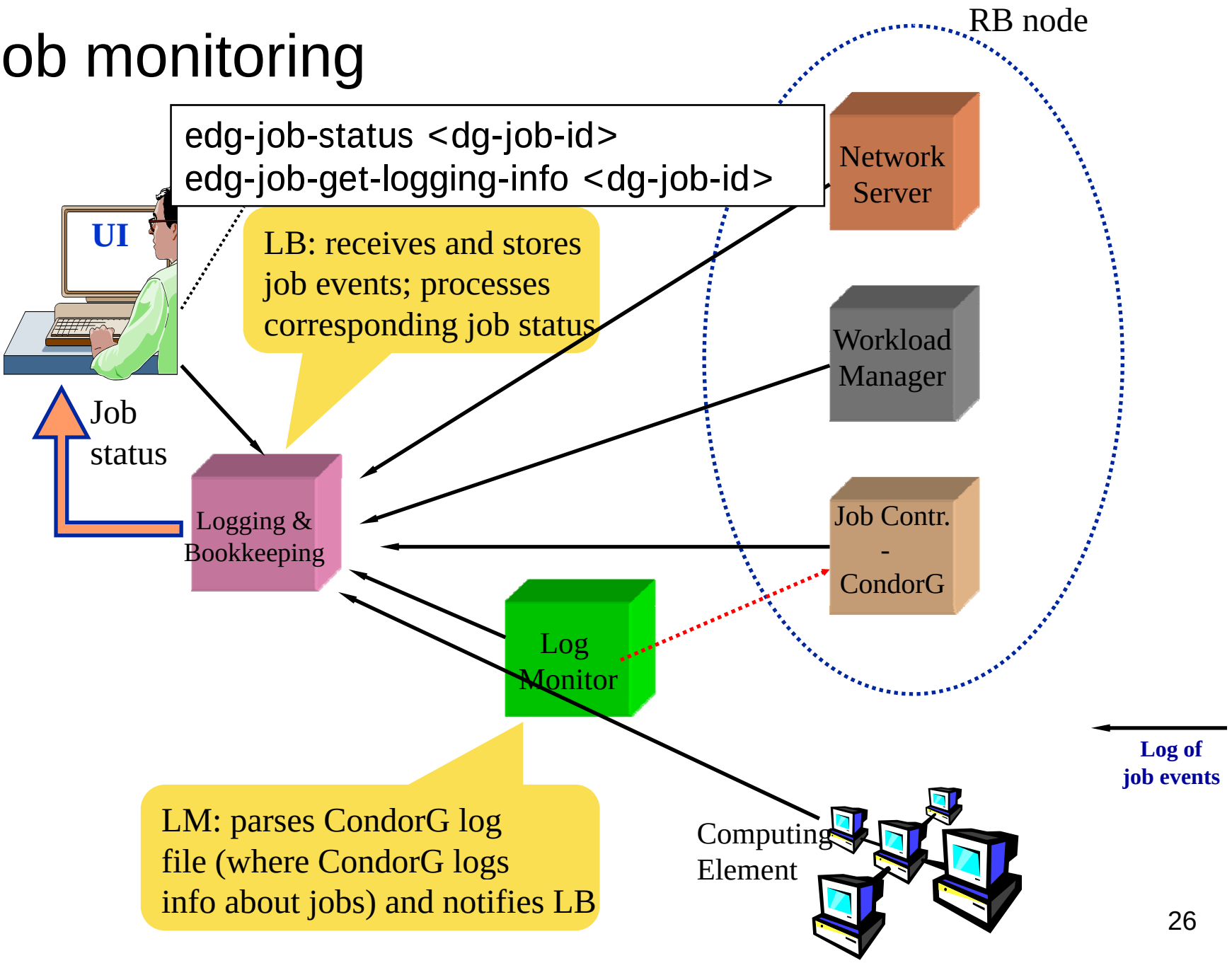

Job Status
submitted
waiting
ready
scheduled
running
done


Storage Element


RB node
edg-job-get-output <dg-job-id>
UI
Network Server
Workload Manager
RB storage
Job Contr. - CondorG
Output Sandbox
LFC
Inform. Service
Job Status
submitted
waiting
ready
scheduled
running
done
Computing Element
Storage Element


RB node
UI
Output
Sandbox
files
RB
storage
Network
Server
Workload
Manager
Job Contr.
-
CondorG
LFC
Inform.
Service
Job
Status
submitted
waiting
ready
scheduled
running
done
cleared
Computing
Element
Storage
Element

Job monitoring



Possible job states

Flag	Meaning
SUBMITTED	**submission logged in the LB**
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 WN of the selected CE queue**
DONE	**job terminated without grid errors**
CLEARED	**job output retrieved**
ABORT	**job aborted by middleware, check** *reason*

JDL: some relevant attributes

- JobType
  - *Normal* (simple, sequential job), *Interactive*, *MPICH*, *Checkpointable*
  - Or combination of them
- Executable **(mandatory)**
  - The name of the binary executable (absolute path)
- Arguments **(optional)**
  - Job command line arguments
- StdInput**,** StdOutput**,** StdError **(optional)**
  - Standard input/output/error of the job
    (stdin absolute path; stdout & stderr relative path)
- Environment (optional)
  - List of environment variables to be set for the binary
- InputSandbox **(optional)**
  - List of files on the UI local disk needed by the job for running
  - The listed files will automatically staged to the remote resource
- OutputSandbox **(optional)**
  - List of files, generated by the job, which have to be retrieved

JDL: Some relevant attributes 2.

- Input Data **(For the broker, but no data movement is performed)**
- Output Data **{**OutputFile**= [CE path]**
  [StorageElement**= SE]**
  [LogicalFileName **=** lfn**:fileName] }**
  (For the broker, but no data movement is performed)
-
- Requirements **– CE features**
- Rank **– Importance of the different CE features**
- RetryCount **– how many times try to resubmit a failed job**
- MyProxyServer **– where to download proxy from**

Summary

- **Create JDL file**
- **Check some CEs match your requirements:**
  - edg-job-list-match
- **Submit job**
  - edg-job-submit
- **Do something else for a while! – gLite is not written for short jobs!**
- **Check job status - occasionally**
  - edg-job-status
- **When job is “done”, get output**
  - edg-job-get-output

NOTES about the practical

- **“Write a simple JDL file like the following (jlvptest.jdl)”**
  - You already have hostname.jdl – use that!
- **Follow “Practical_1” link in the agenda page**
  - Also try the command edg-job-get-logging-info
  - And follow Practical_2 to explore different JDL options.