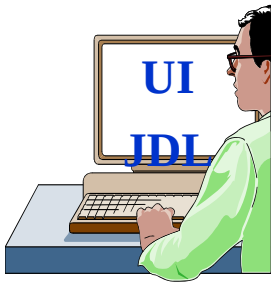
Workload Management System and Job Description Language

Contents

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

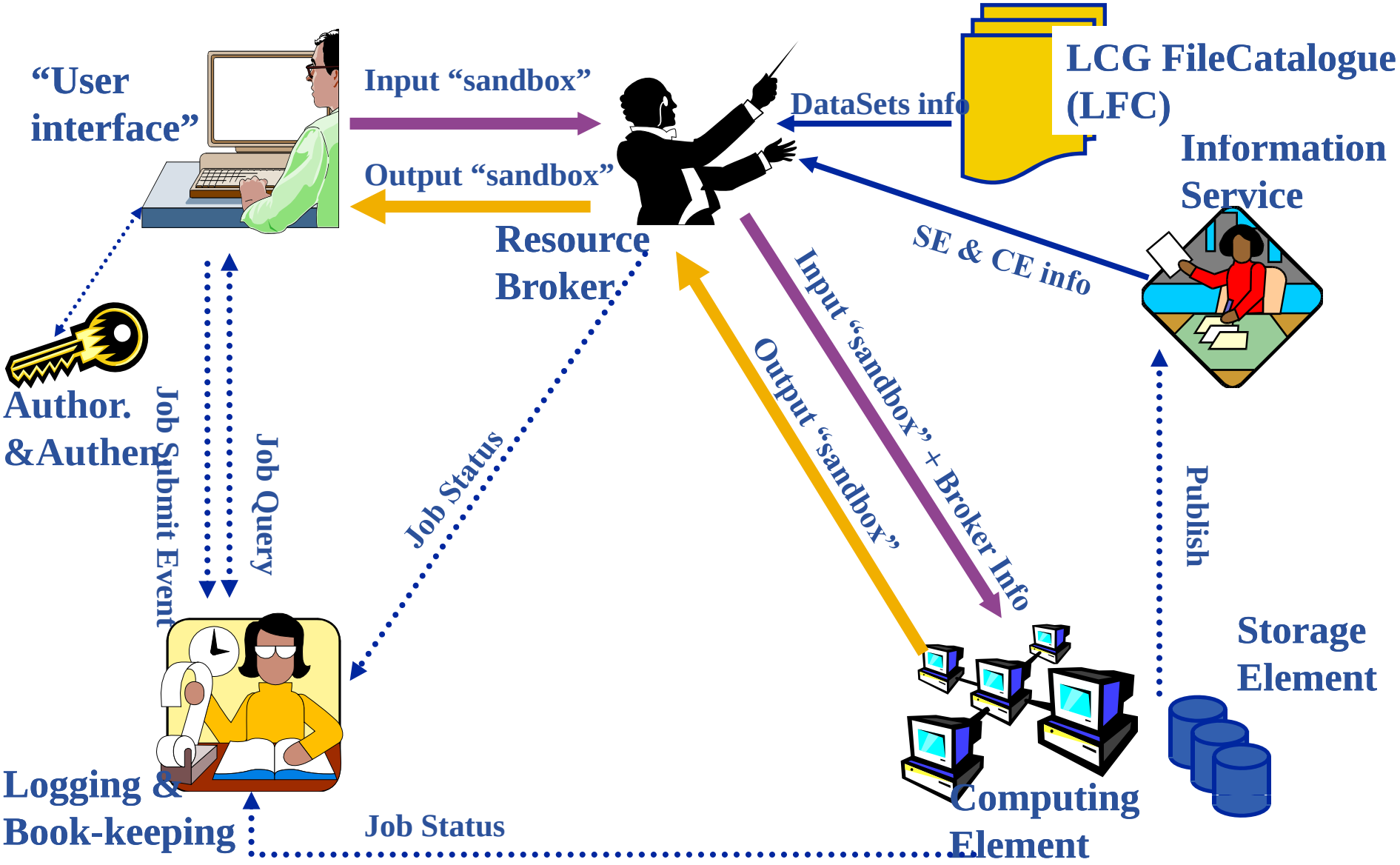
User Interface node

- **The user's interface to the Grid**
- **Command-line interface to**
  - Create proxy with VOMS extensions
  - Job operations – (non-blocking)
    - 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



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”;
DataAccessProtocol = “gridftp”;
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”,
- 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/gridte
OutputSandbox = {“stderr.log”, “stdou log”};
InputData = “lfn:/grid/VOname/mydir/testbed0-00019”;
DataAccessProtocol = “gridftp”;
Requirements = other.Architecture==“INTEL” && \
               other.OpSys==“LINUX” && other.FreeCpus >=4;
Rank = “other.GlueHostBenchmarkSF00”;
```

lfn: logical file name

RB uses Catalog to find replica locations

Building on basic tools and Information Service

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

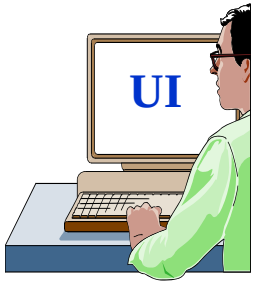
Returns a “job-id” used to monitor job, retrieve output

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

Uses Information System

Job submission



RB node

Network Server

Workload Manager

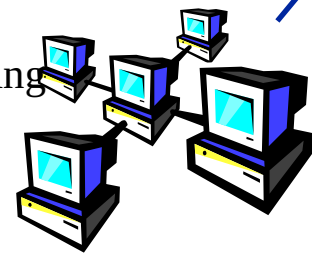
Job Contr. - CondorG

LFC

Inform. Service

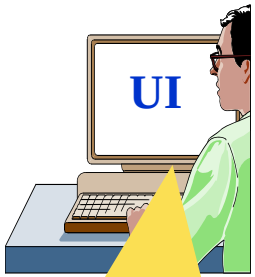
CE characts & status

SE characts & status



Computing Element

Storage Element



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

RB node

Network Server

Workload Manager

Job Contr. - CondorG

Job Status

LFC

Inform. Service

CE characts & status

SE characts & status

Computing Element

Storage Element

edg-job-submit myjob.jdl

Job Status

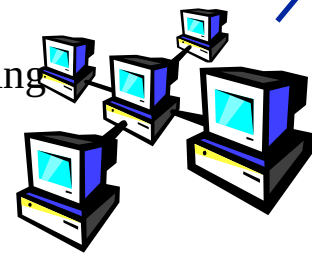
submitted

Myjob.jdl

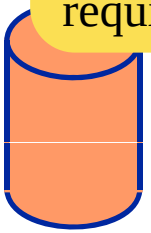
```
JobType = “Normal”;
Executable      = "$(CMS)/exe/sum.exe";
InputSandbox = {"/home/user/WP1testC","/home/file*”,
"/home/user/DATA/*"};
OutputSandbox = {“sim.err”, “test.out”, “sim.log"};
Requirements    = other. GlueHostOperatingSystemName ==
“linux" &&
other. GlueHostOperatingSystemRelease == "Red Hat 7.3“ &&
other.GlueCEPolicyMaxCPUTime > 10000;
Rank  = other.GlueCEStateFreeCPUs;
```

CE characts & status

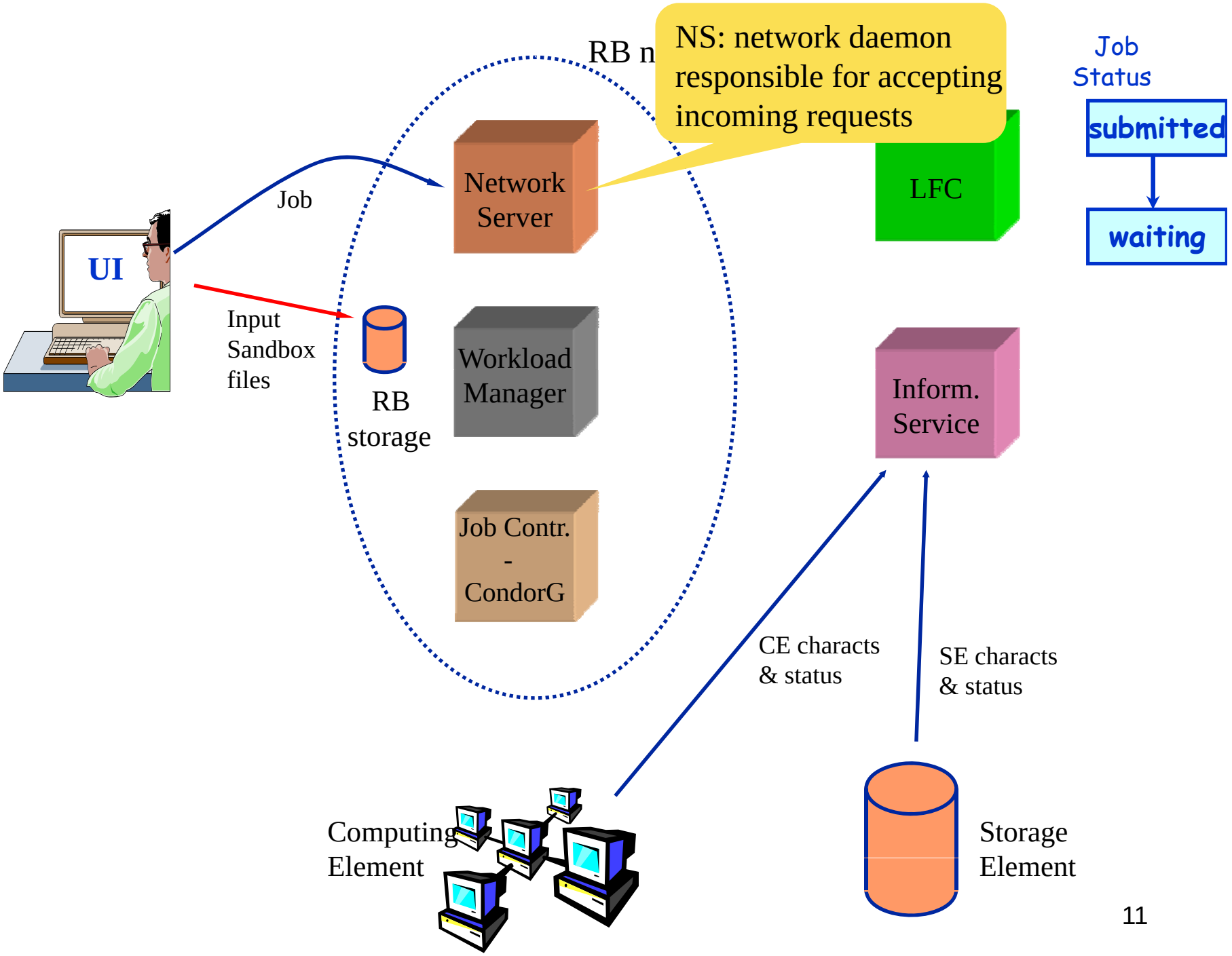
Job Description Languag (JDL) to specify job characteristics and requirements

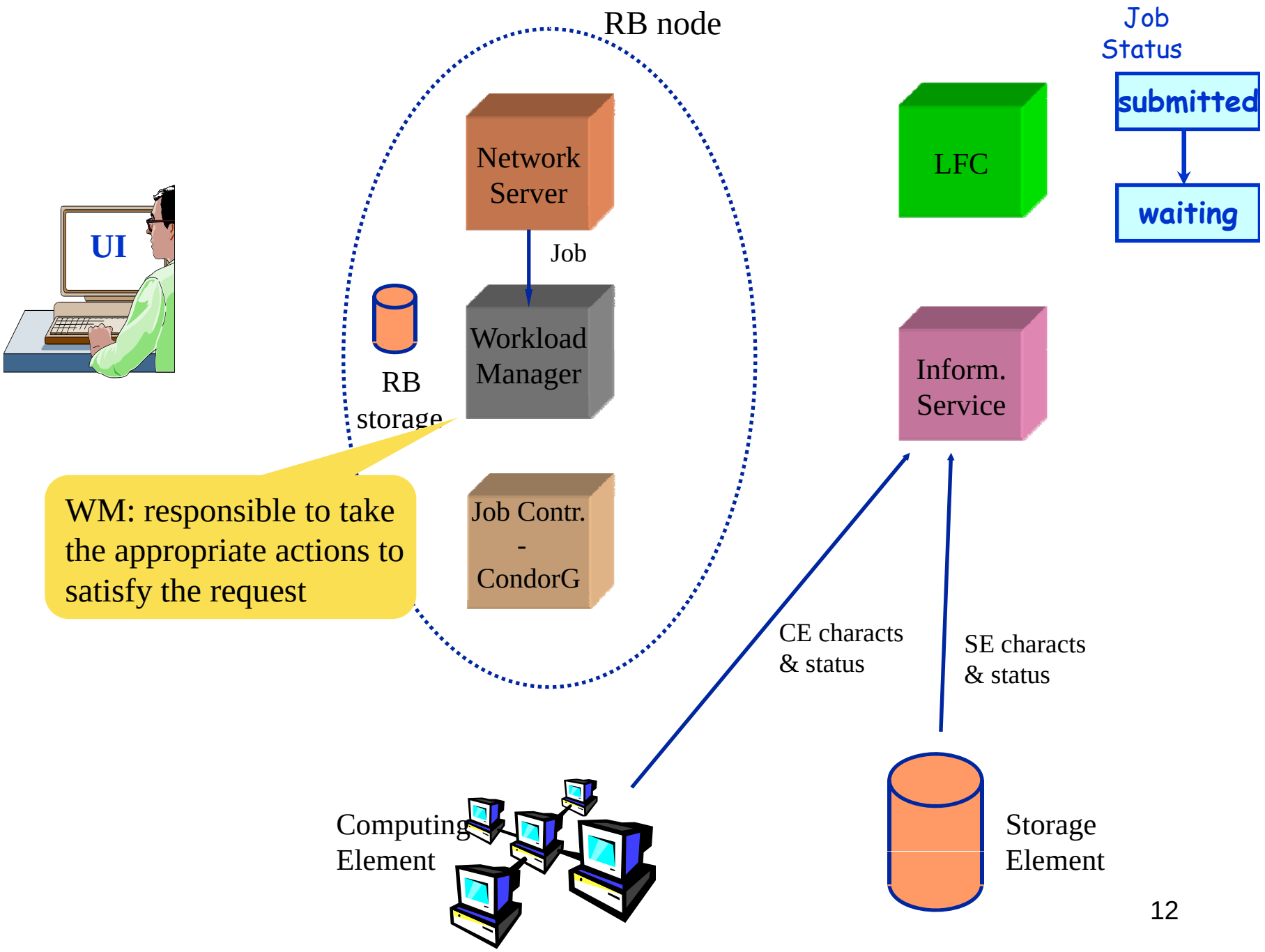


Computing Element

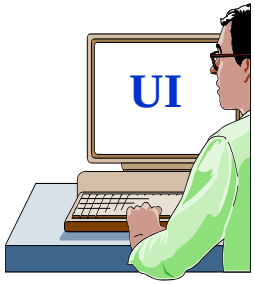


Storage Element


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

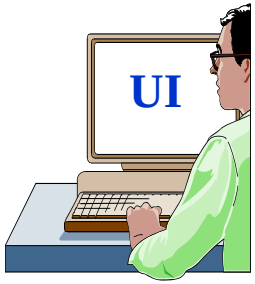
CE characts & status

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

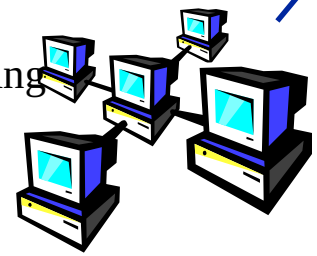
Workload Manager

Inform. Service

Job Contr. - CondorG

CE characts & status

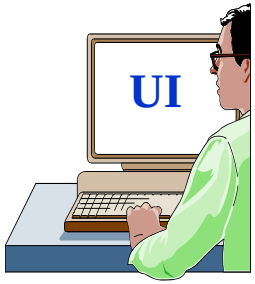
SE characts & status



Computing Element

Storage Element

Job submission



RB node

Network Server

Workload Manager

Job Contr. - CondorG

RB storage

Match-Maker/ Broker

Where are (which SEs) the needed data ?

LFC

Inform. Service

What is the status of the Grid ?

CE characts & status

SE characts & status

Computing Element

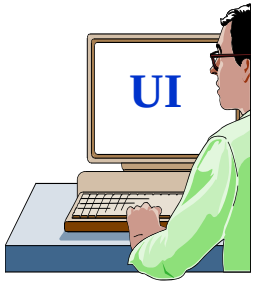
Storage Element

Job Status

submitted

waiting

Job submission



RB node

Network Server

Match-Maker/ Broker

LFC

Job Status

submitted

waiting

RB storage

Workload Manager

CE choice

Inform. Service

Job Contr. - CondorG

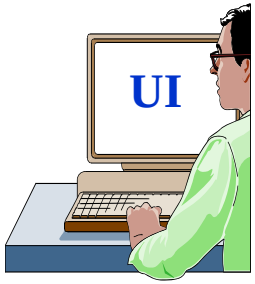
CE characts & status

SE characts & status

Computing Element

Storage Element

Job submission



RB node

Network Server

Workload Manager

RB storage

Job Contr. - CondorG

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.)

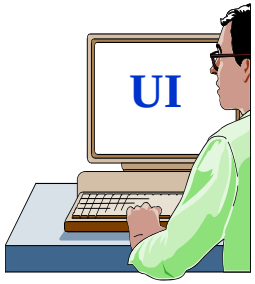
aracts
us

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

Job Status

submitted

waiting

ready

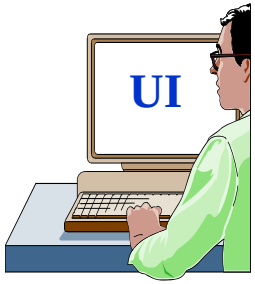
CE characts & status

SE characts & status

Computing Element

Storage Element

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

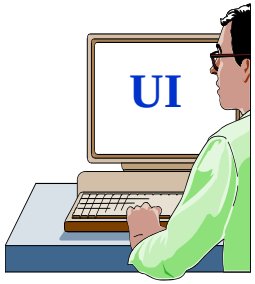
Job Status

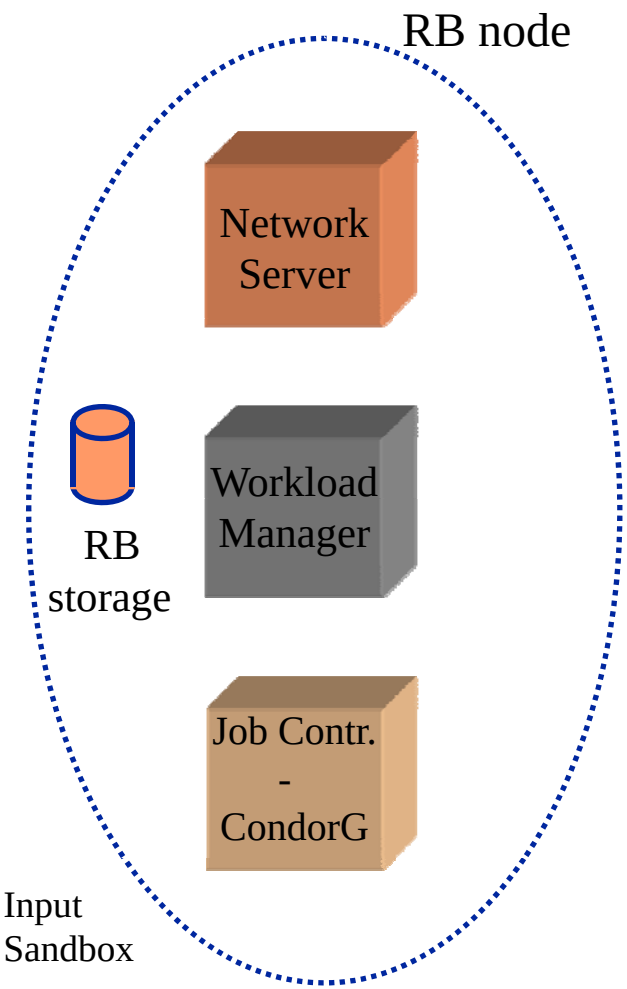
submitted

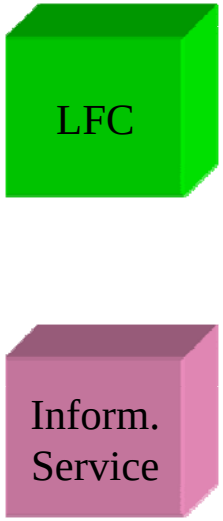
waiting

ready

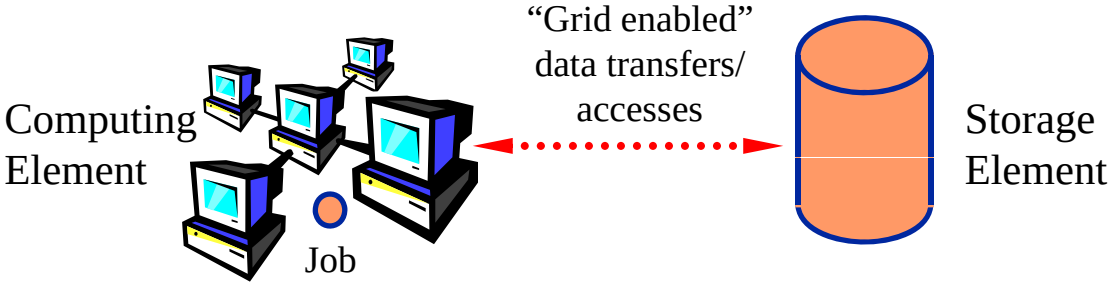
scheduled

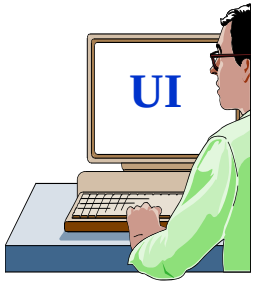

UI

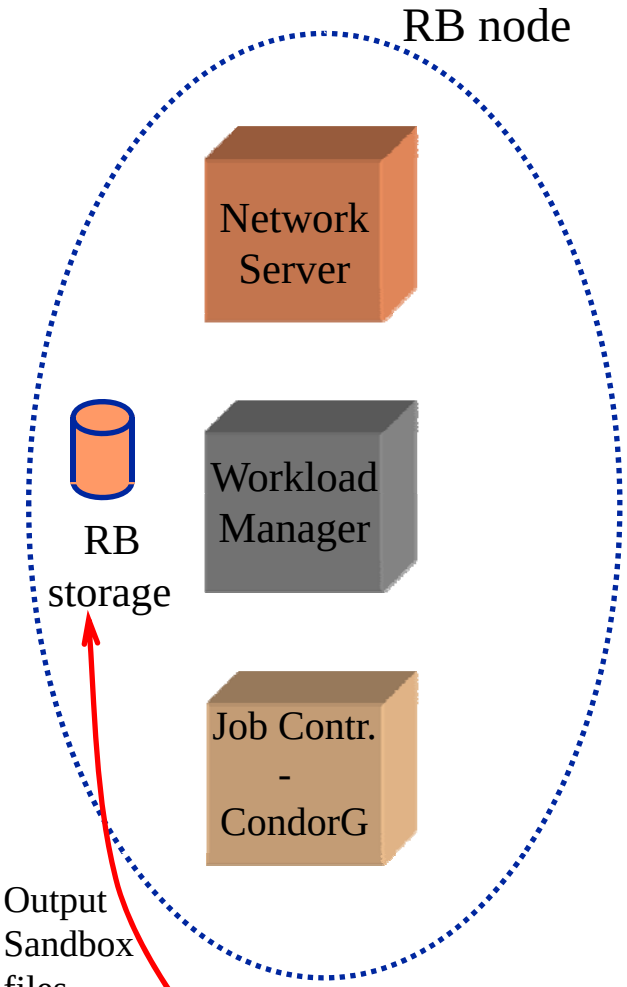

RB node
Network Server
Workload Manager
RB storage
Job Contr. - CondorG
Input Sandbox

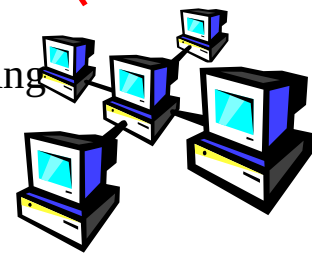

LFC
Inform. Service

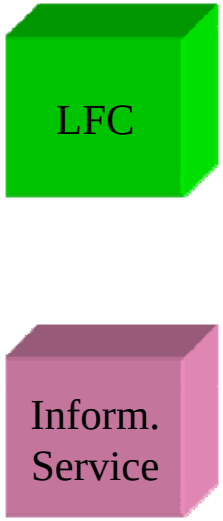

Job Status
submitted
waiting
ready
scheduled
running

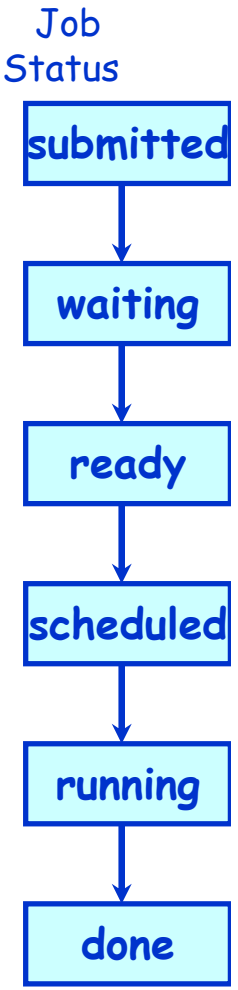

Computing Element
Job
"Grid enabled" data transfers/ accesses
Storage Element

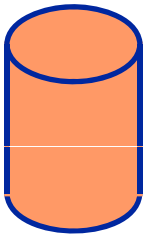

UI

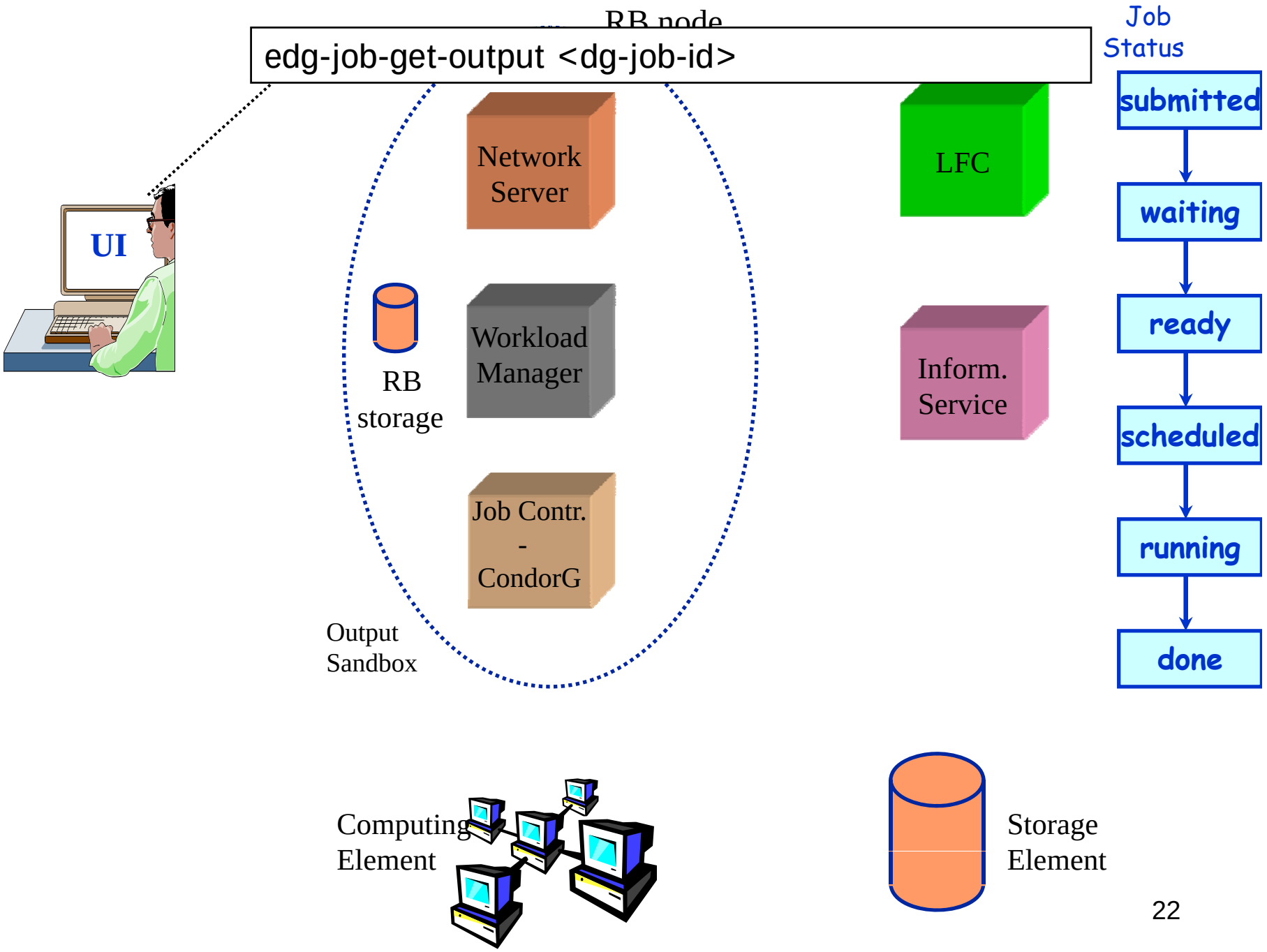

RB node
Network Server
Workload Manager
Job Contr. - CondorG
RB storage
Output Sandbox files

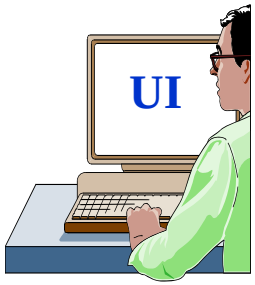

Computing Element


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



Output Sandbox files

RB node

Network Server

Workload Manager

Job Contr. - CondorG

RB storage

LFC

Inform. Service

Job Status

submitted

waiting

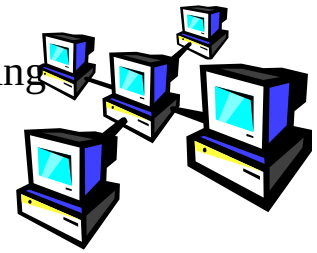
ready

scheduled

running

done

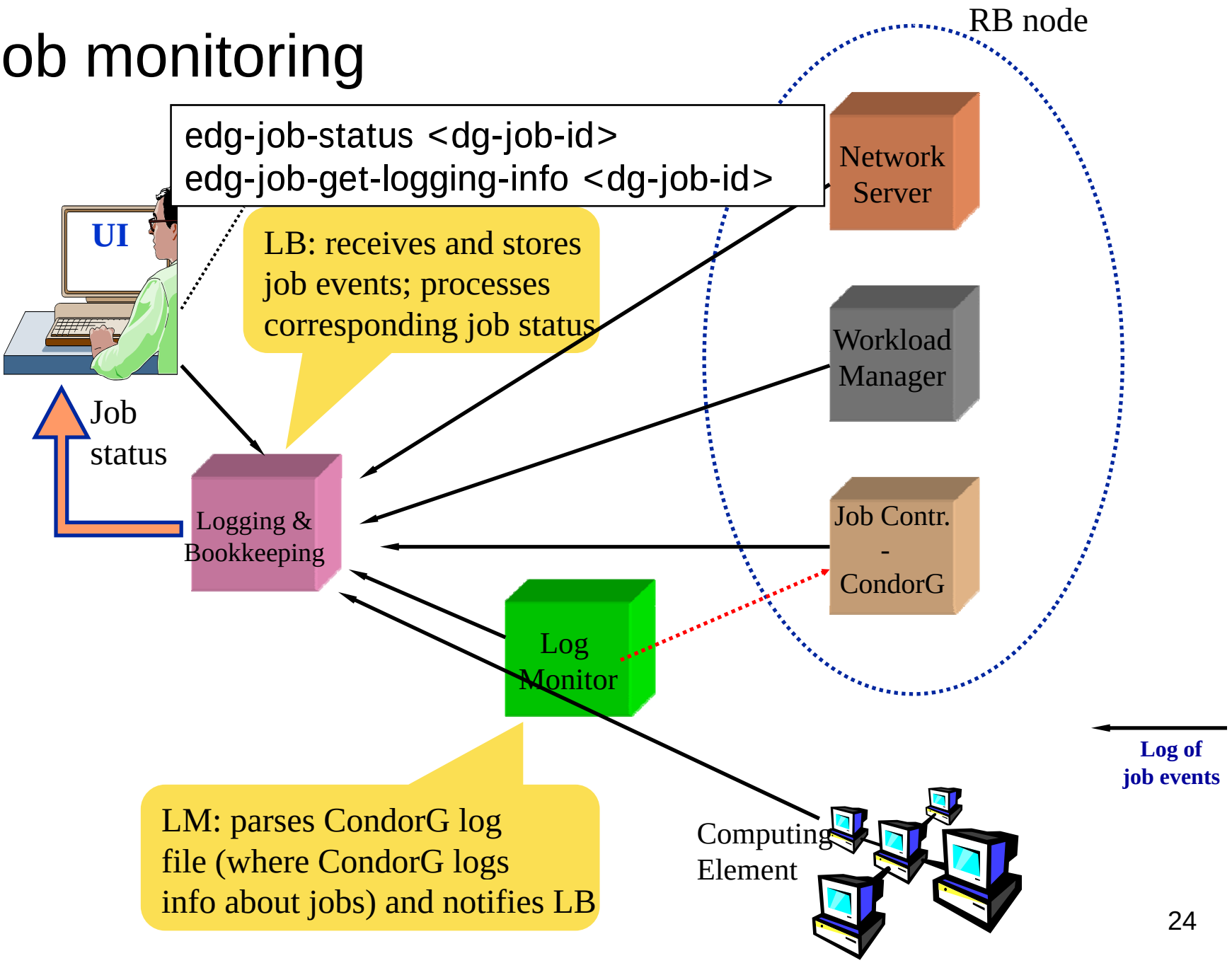
cleared



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*

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.