



Enabling Grids for E-science

# EGEE Middleware

*Mike Mineter*

*mjm@nesc.ac.uk*

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

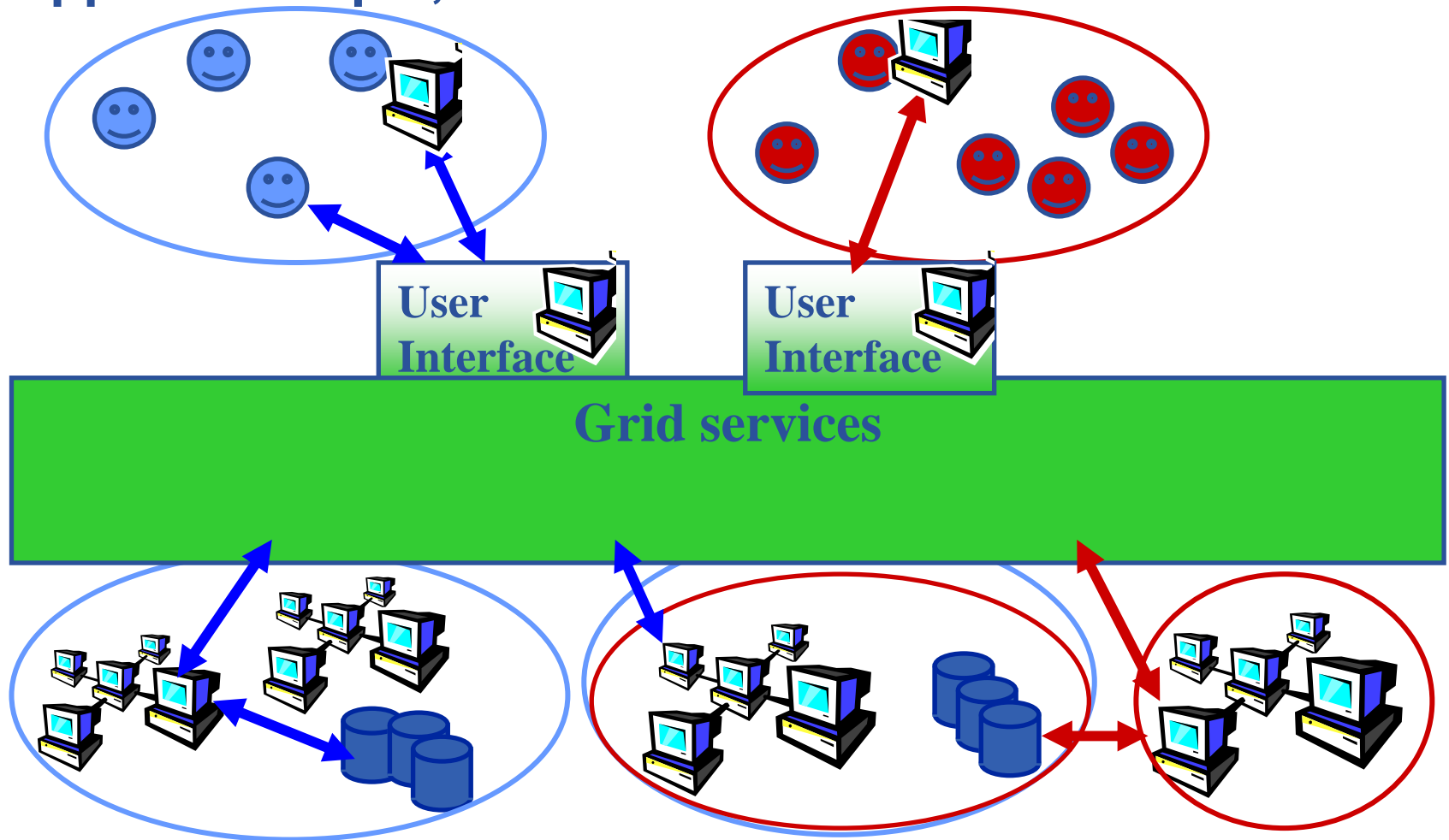


INFSO-RI-508833

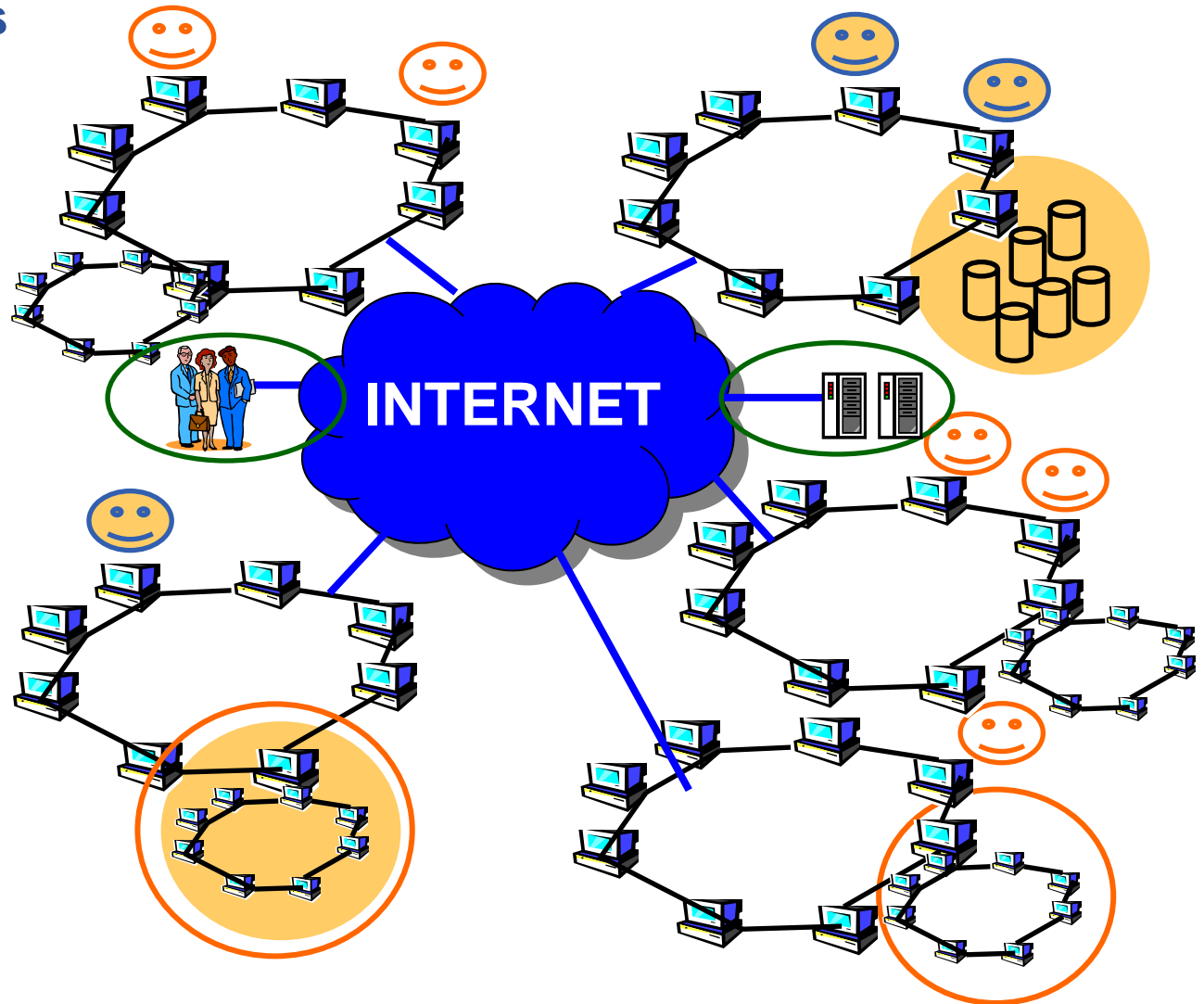
- **Short review of concepts**
- **Requirements of the applications communities**
- **Overview of the main grid services**
- **A closer look**

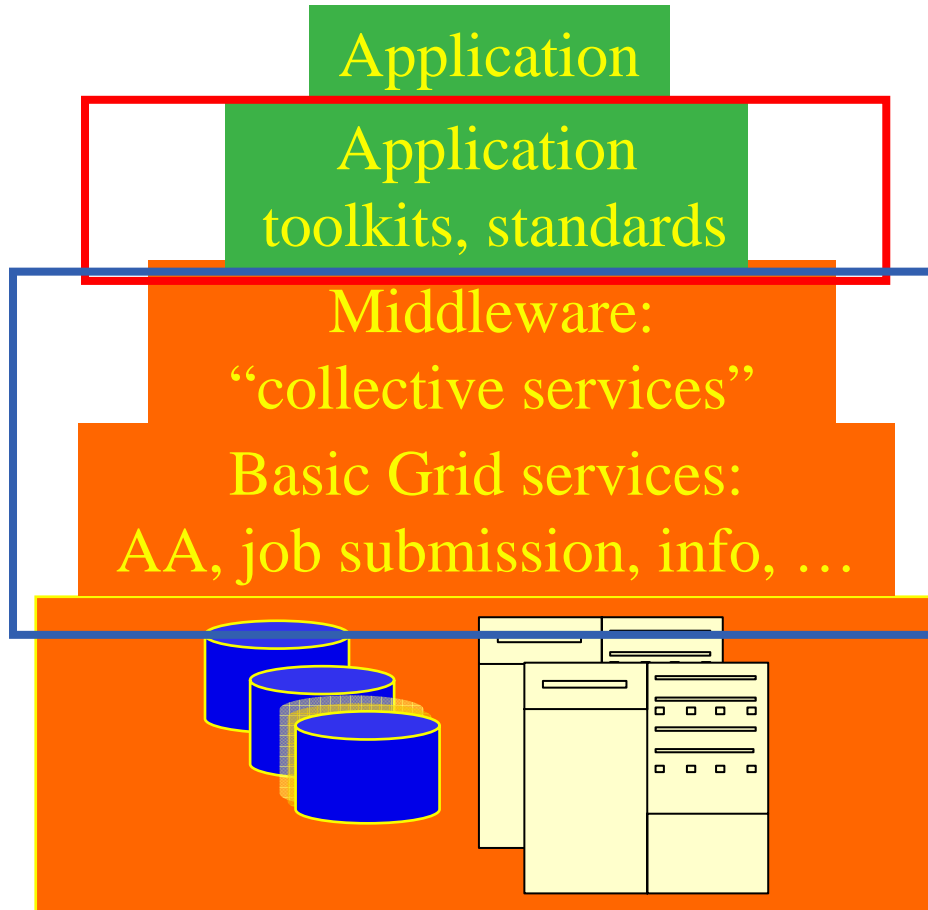


- EGEE is establishing a production grid service to support multiple, diverse VO's



- **Grid middleware runs on each shared resource to provide**
  - Data services
  - Computation services
  - Single sign-on
- **Users join VO's**
- **Virtual organisation negotiates with sites to agree access to resources**
- **Distributed services (both people and middleware) enable the grid for multiple VO's**



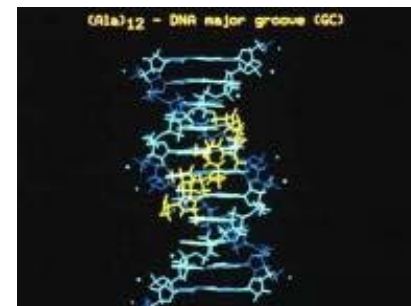
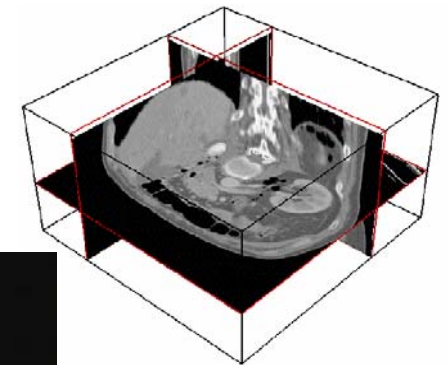


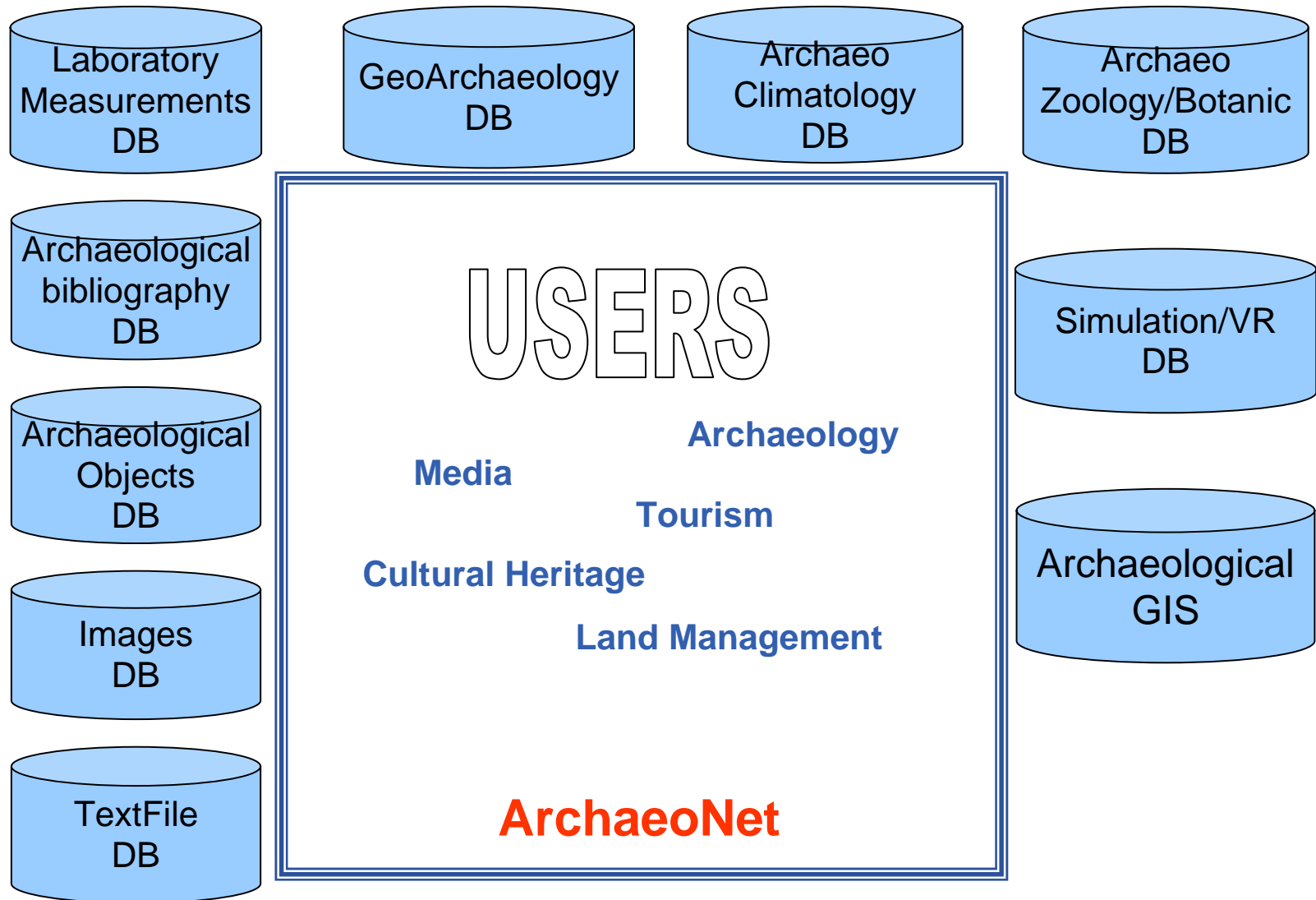
- Application development environment, portals, workflow
- Semantics, ontologies

## EGEE middleware:

- Grid services for many collaborations

- **High-Energy Physics**
  - Provides computing infrastructure  
Large Hadron Collider (LHC)  
LHC Compute Grid (LCG)
  - thousands of processors  
world-wide
  - 1 gigabyte per second achieved  
from CERN to 20 sites
  
- **Biomedical Applications**
  - Less “centralised” VO’s  
& data flows
  - More sporadic demands
  - Interactive applications needed
  - **security & privacy**





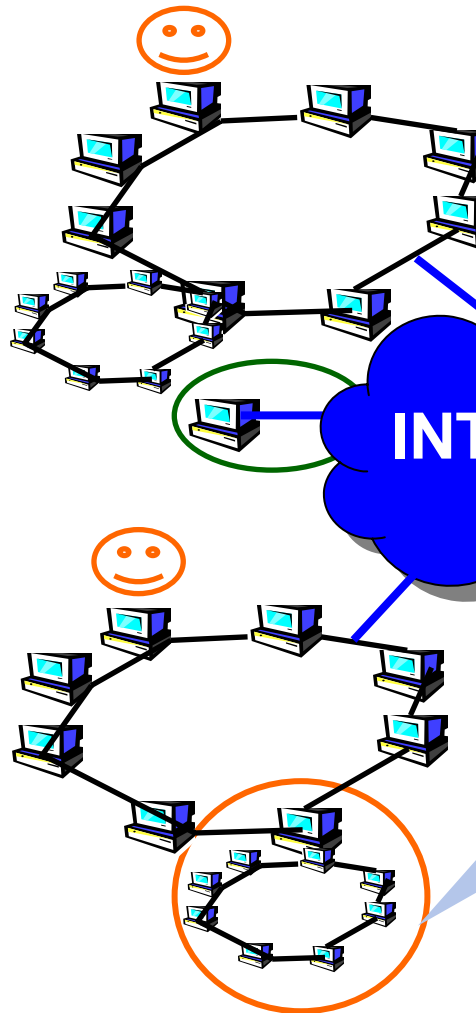
## Grid services

**How can EGEE middleware support collaboration and resource sharing within and between many diverse VO's ?**



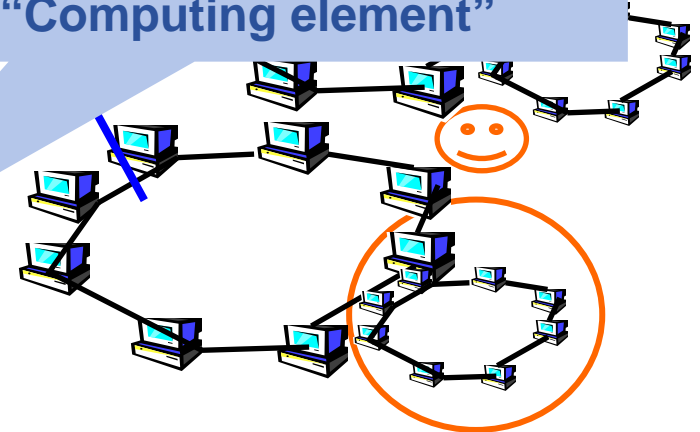
- **When using a PC or workstation you**
  - Login with a username and password (“Authentication”)
  - Use rights assigned to you (“Authorisation”)
  - Run jobs
  - Manage files: create them, read/write, list directories
- **When using a Grid you**
  - Login with digital credentials (“Authentication”)
  - Use rights assigned to you (“Authorisation”)
  - Run jobs
  - Manage files: create them, read/write, list directories
- **Middleware that masks**
  - Internet (not a bus) links your services
  - Providers in different admin domains

- **Grid middleware runs on each shared resource**
  - Data storage
  - (Usually) batch queues on pools of processors
- **Users join VO's**
- **Virtual organisation negotiates with sites to agree access to resources**
- **Distributed services (both people and middleware) enable the grid, allow single sign-on**



At each site that provides computation:

- Local resource management system
- (= batch queue)
  - Condor
  - PBS
  - Torque
  - ...
- EGEE term: queue is a "Computing element"



# Authorisation, Authentication (AA)

Users in many locations and organisations

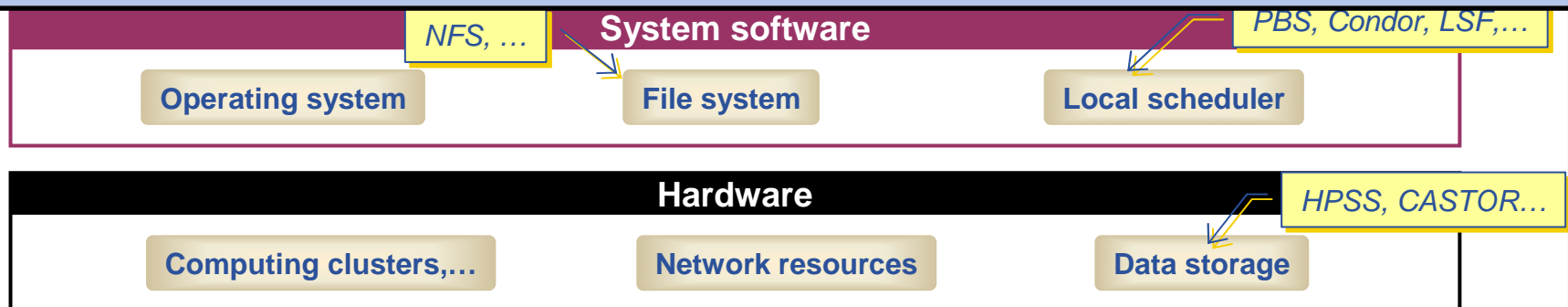


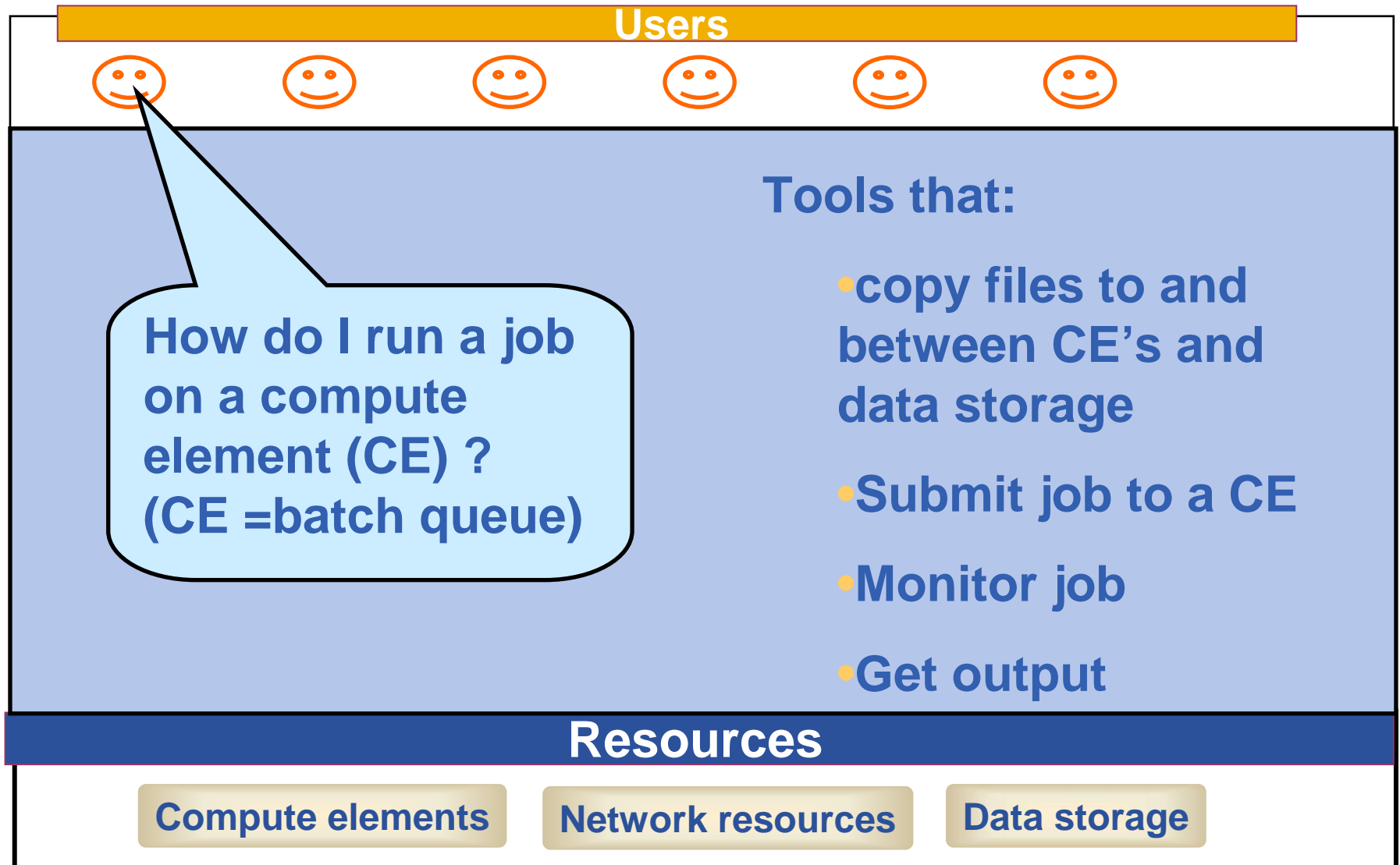
Access services (“user interface”) :  
logon, upload credentials, run m/w

## GRID SERVICES

Build on Grid Security Infrastructure

“Gate keeping”:  
map user’s credential to local user id / account





Users



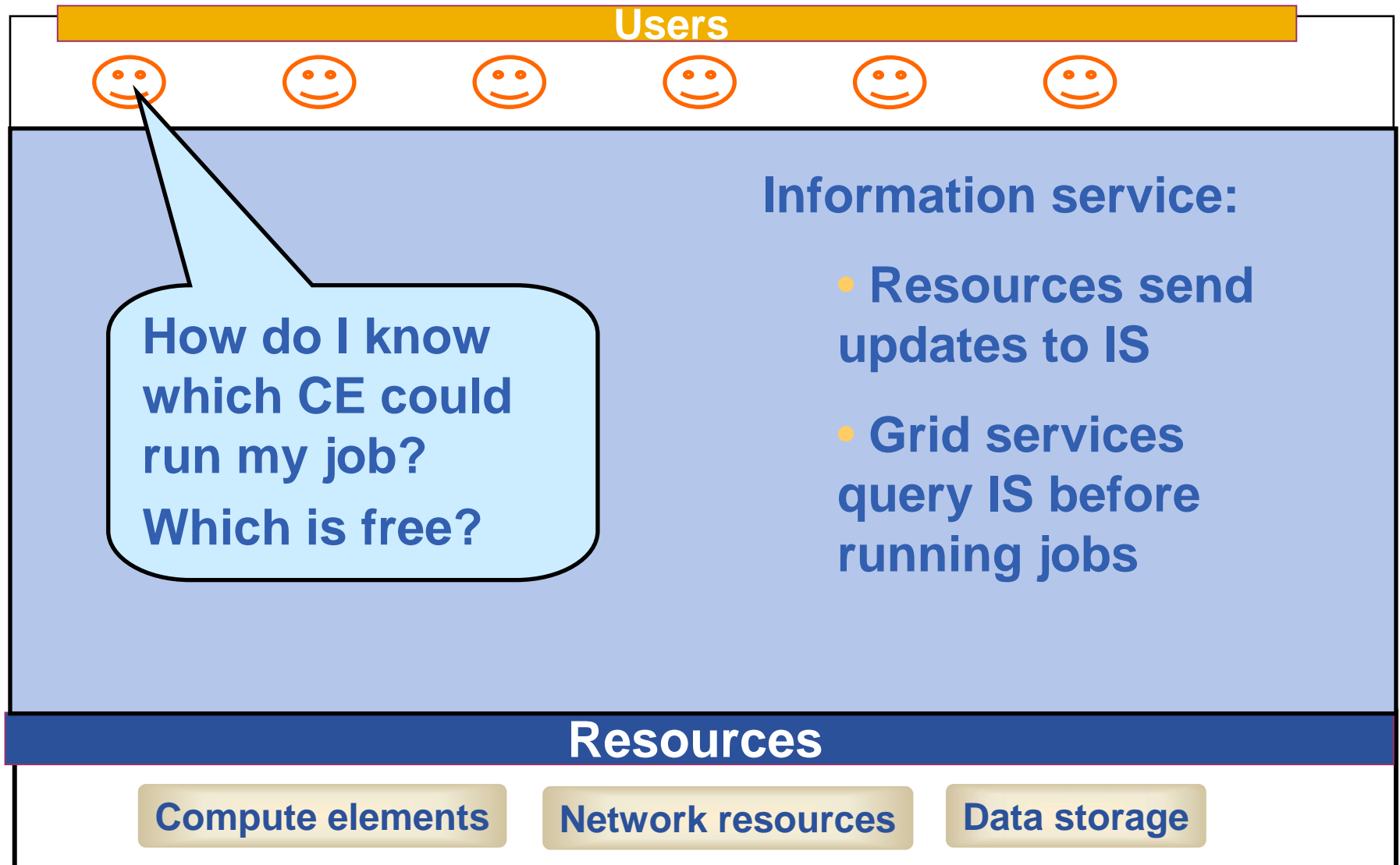
How do I run a job on a compute element (CE) ?  
(CE =batch queue)

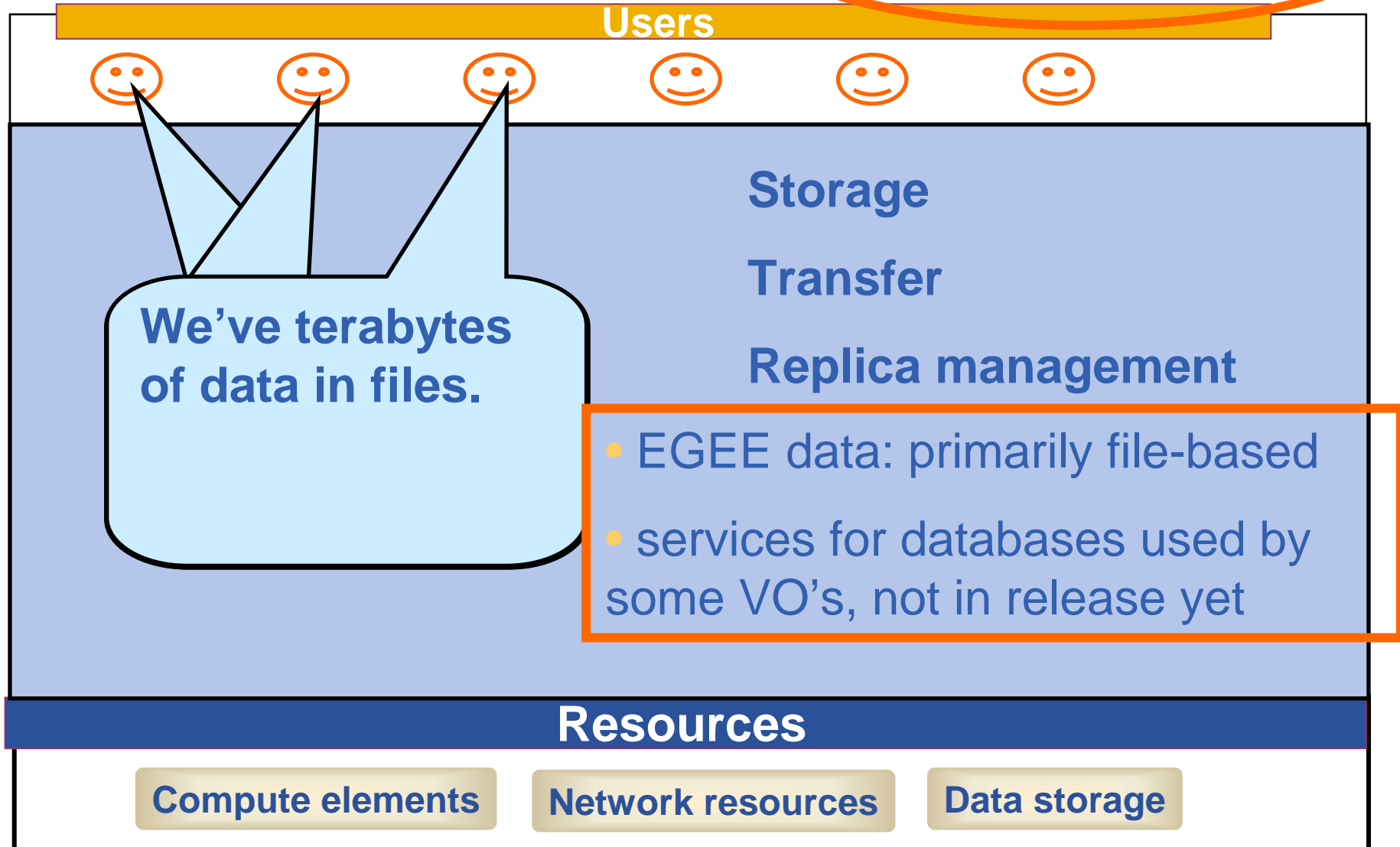
Tools that:

- copy files to and between CE's and data storage
- Submit job to a CE
- Monitor job
- Get output

Resources

- Compute elements
- Network resources
- Data storage





- **A software toolkit: a modular “bag of technologies”**
  - Made available under liberal open source license
- **Not turnkey solutions, but *building blocks* and *tools* for application developers and system integrators**
- **Tools built on Grid Security Infrastructure to include:**
  - Job submission: run a job on a specific remote compute element
  - Information services: So I know which computer to use
  - File transfer: so large data files can be transferred
    - GridFTP: supports multiple channels for one transfer
- **(Most) production grids are (currently) based on the Globus Toolkit release 2**
- **Globus Alliance: <http://www.globus.org/>**

- **GT2 Toolkit**
- **An example of the command line interface:**
  - Job submission – need to know name of a CE to use

```
globus-job-submit grid-data.rl.ac.uk/jobmanager-pbs /bin/hostname -f
```

```
https://grid-data.rl.ac.uk:64001/1415/1110129853/
```

```
globus-job-status https://grid-data.rl.ac.uk:64001/1415/1110129853/
```

```
DONE
```

```
globus-job-get-output https://grid-data.rl.ac.uk:64001/1415/1110129853/
```

```
grid-data12.rl.ac.uk
```

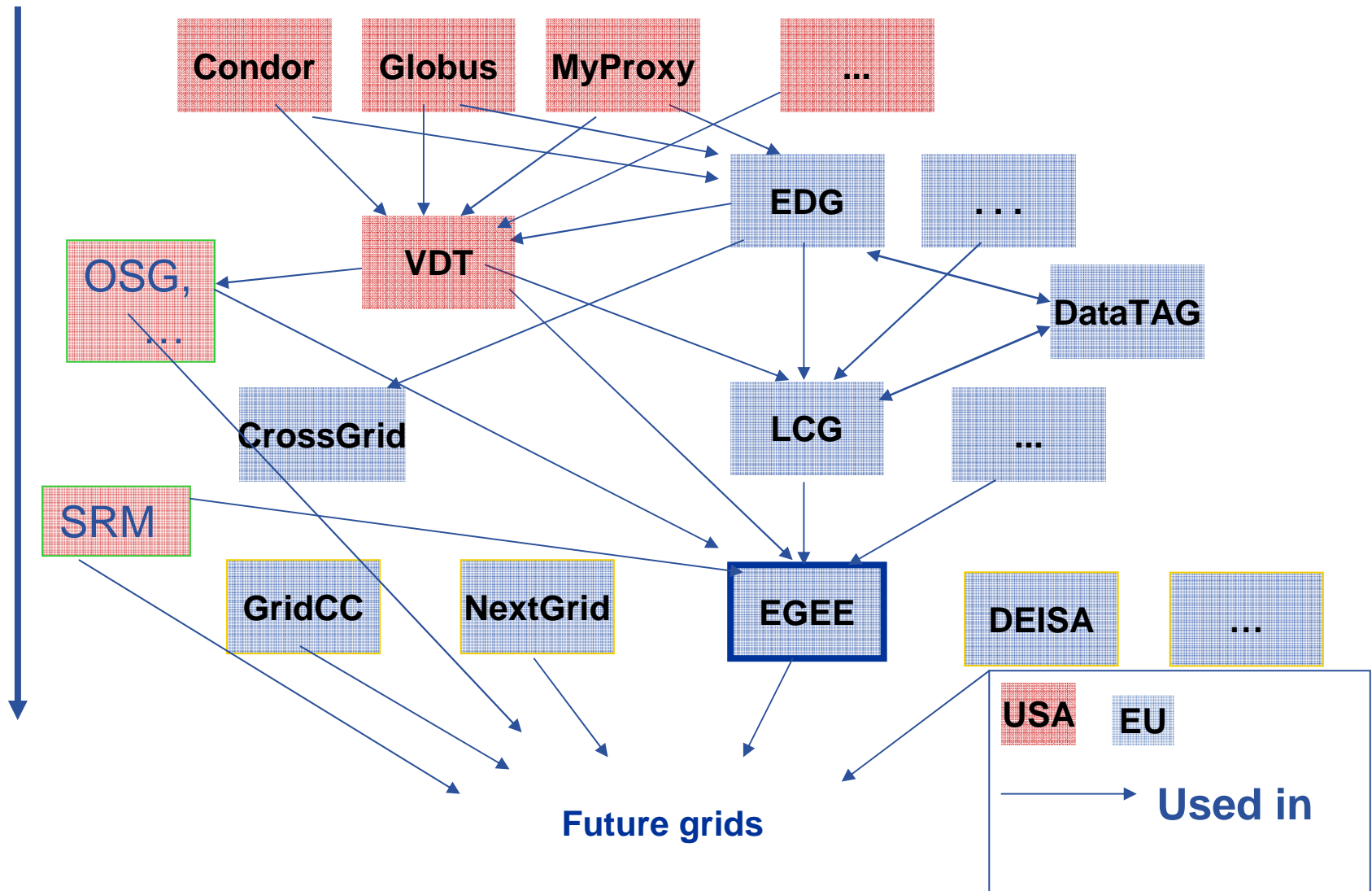


- **GT2: a toolkit – not a turnkey solution**
- **Need higher level tools including:**
  - **Job submission** to “a grid” not a CE
  - **Data management**
  - **Logging** who’s done what, statistics about jobs,...
  - **Monitoring** whats happening on the grid
- **So EGEE is built on more than GT2 !**

# Parts of the Grid “ecosystem”

2001

2004



- **“The Virtual Data Toolkit (VDT) is an ensemble of grid middleware that can be easily installed and configured. In our experience, installing grid software is challenging and time consuming. The goal of the VDT is to make it as easy as possible for users to deploy, maintain and use grid middleware.”**  
<http://www.cs.wisc.edu/vdt/>



**User Interface (UI):**

The place where users logon to the Grid



**Resource Broker (RB):**

Matches the user requirements with the available resources on the Grid



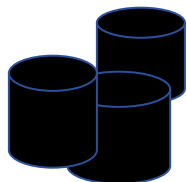
**Information System:**

Characteristics and status of CE and SE  
Uses "GLUE schema"



**Computing Element (CE):**

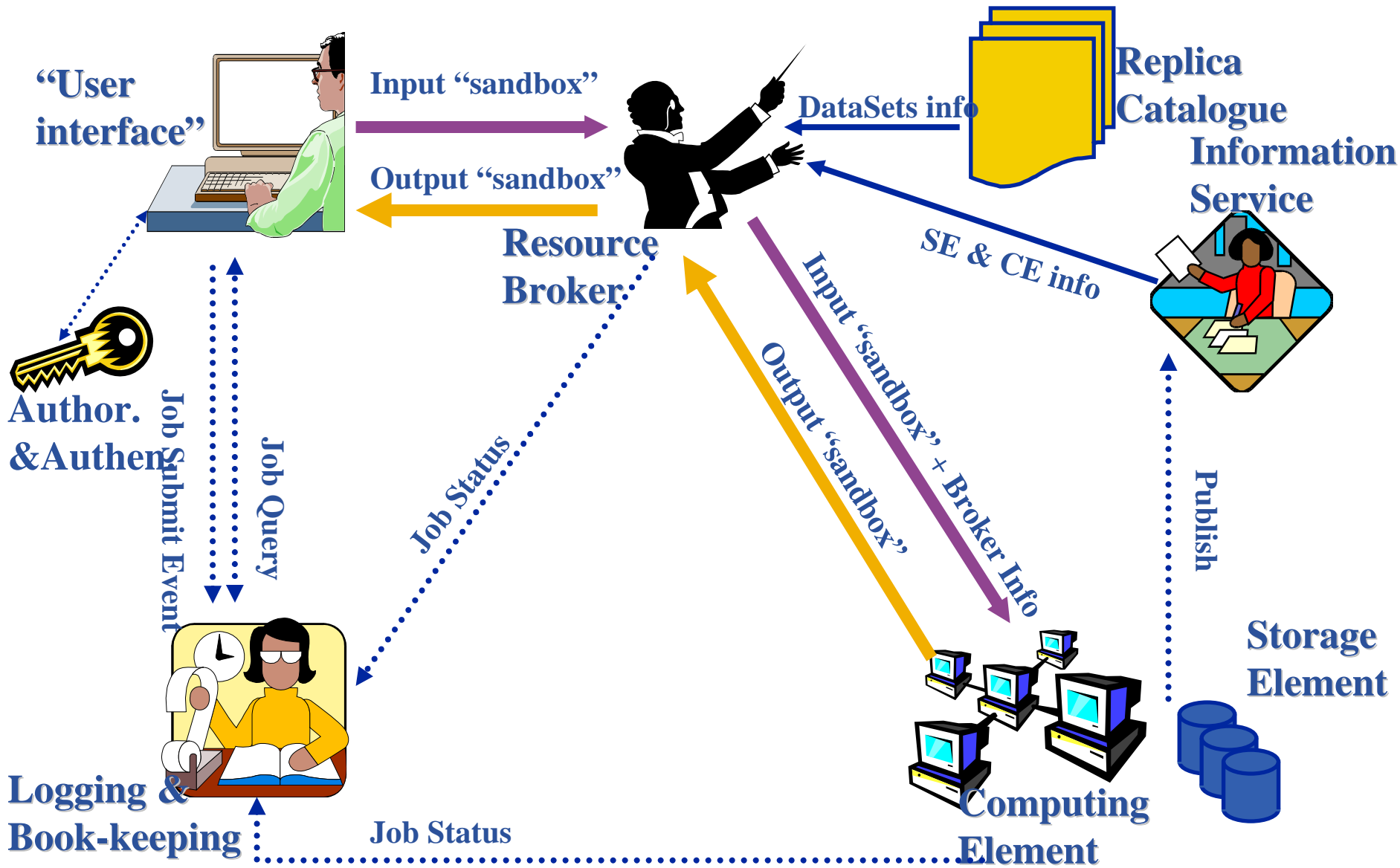
A batch queue on a site's computers where the user's job is executed



**Storage Element (SE):**

provides (large-scale) storage for files

# Current production middleware



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

## Example JDL file

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

# A closer look at the main EGEE grid services

## 1. Security, Authentication and Authorisation

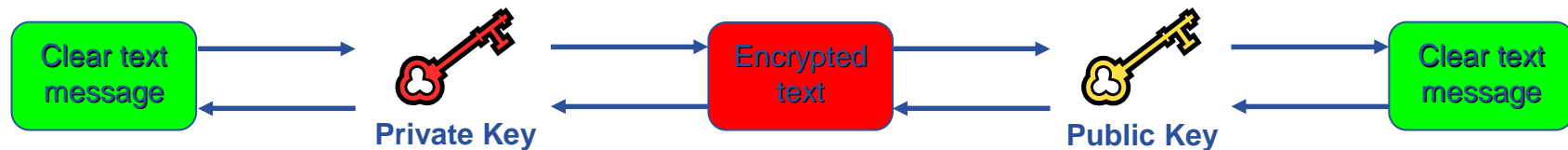
- **Principal**
  - An entity: a user, a program, or a machine
- **Credentials**
  - Some data providing a proof of identity
- **Authentication**
  - Verify the identity of a principal - how the user tells services who they are?
- **Authorization**
  - Map an entity to some set of privileges
- **Confidentiality**
  - Encrypt the message so that only the recipient can understand it
- **Integrity**
  - Ensure that the message has not been altered in the transmission
- **Non-repudiation**
  - Impossibility of denying the authenticity of a digital signature
- **Delegation**
  - Principal delegates authority to a service to act for them, in using another service



- **Authentication based on X.509 PKI infrastructure**
  - Trust between **Certificate Authorities** (CA) and sites, CAs and users is established (offline)
  - CAs issue (long lived) **certificates** identifying sites and individuals (much like a passport)
    - Commonly used in web browsers to authenticate to sites
  - In order to reduce vulnerability, on the Grid user identification is done by using (short lived) **proxies** of their certificates
- **Proxies can**
  - Be **delegated** to a service such that it can act on the user's behalf
  - Include **additional attributes** (like VO information via the VO Membership Service VOMS)
  - Be stored in an **external proxy store** (myProxy)
  - Be **renewed** (in case they are about to expire)

# Use of public and private keys

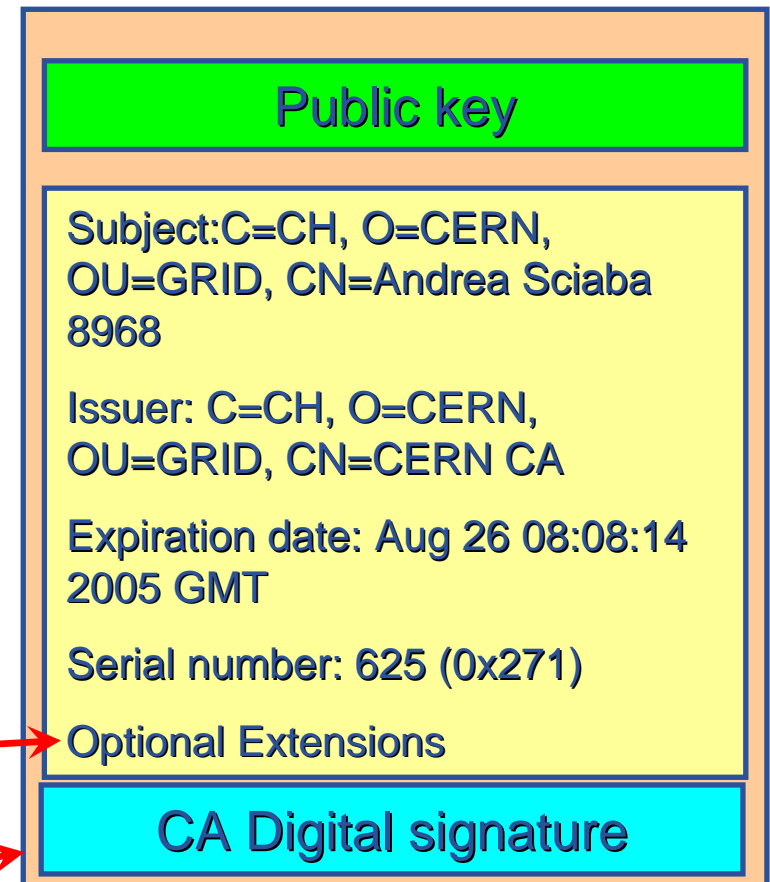
- Basis for authentication, integrity, confidentiality, non-repudiation
- Certificate: held in two parts
  - Public key + user identity + CA signature
  - Private key: only the owner (should) use this
- Asymmetric encryption



- Digital signatures
  - A hash derived from the message and encrypted with the signer's private key
  - Signature is checked by decrypting with the signer's public key
- Public key is trusted only because it is signed by a trusted third party (**Certification Authority**)
  - Effect: users and sites can trust each other's identity as expressed in certificates

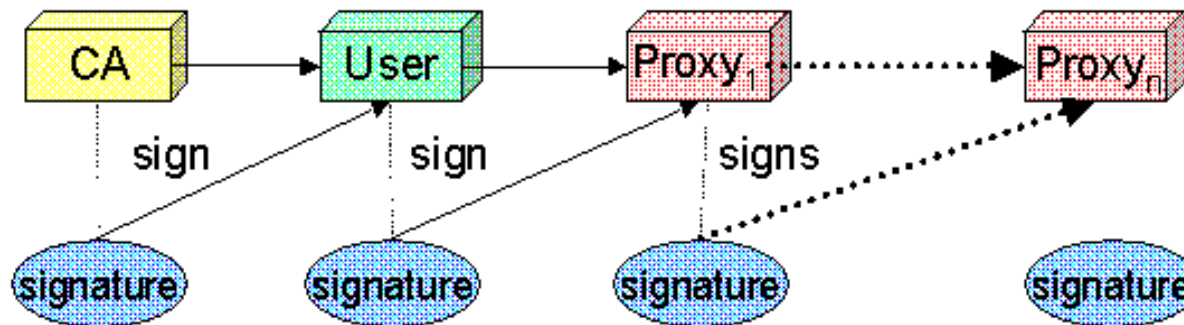
- An X.509 Certificate contains:

- owner's public key; →
- identity of the owner; →
- info on the CA; →
- time of validity; →
- Serial number; →
- Optional extensions →



- digital signature of the CA →

- *de facto* standard for Grid middleware
- Based on PKI
- To support....
  - Single sign-on: to a machine on which your certificate is held
  - Delegation: a service can act on behalf of a person
- ....GSI introduces **proxy certificates**
  - Short-lived certificates signed with the user's certificate or a proxy
  - Reduces security risk, enables delegation



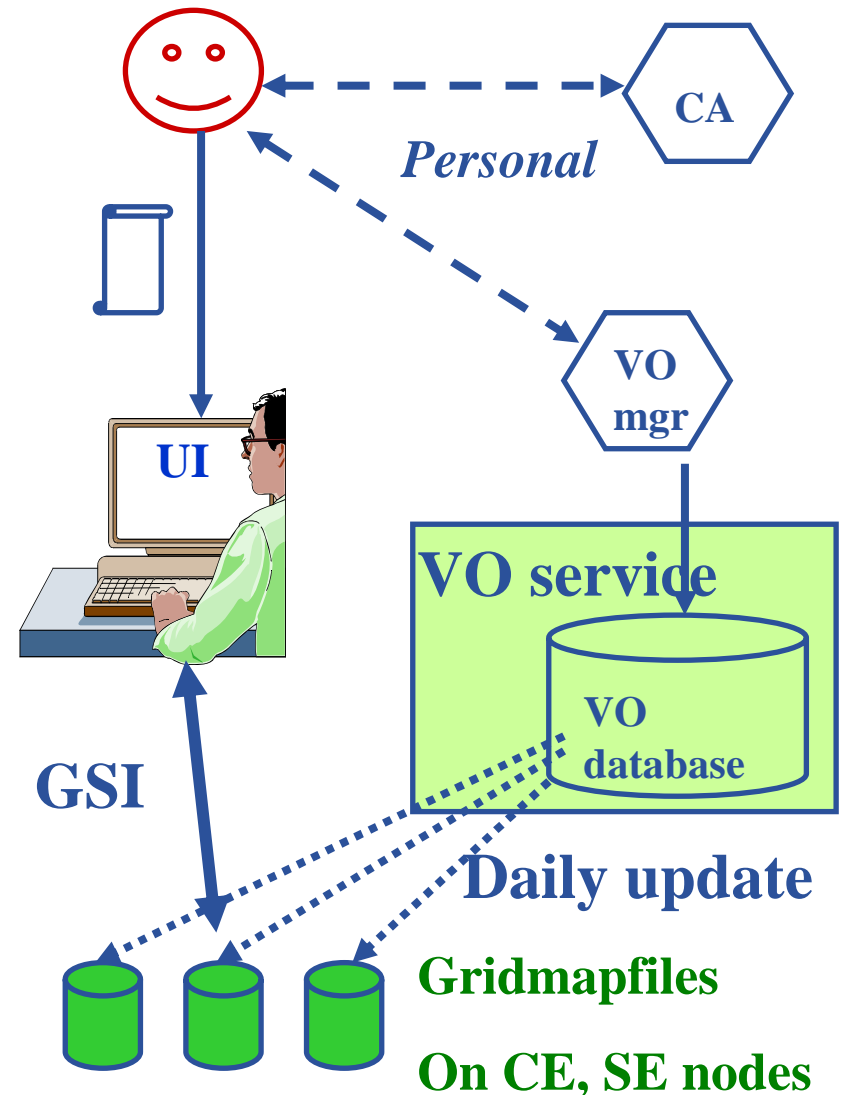
- **You may need:**
  - To interact with a grid from many machines
    - And you realise that you must NOT, EVER leave your certificate where anyone can find and use it....
  - To use a portal, and delegate to the portal the right to act on your behalf
    - The portal obtains then uses a proxy certificate for you
  - To run jobs that might last longer than the lifetime of a short-lived proxy
  
- **Solution: you can store a long-lived proxy in a “MyProxy server” and derive a proxy certificate when needed.**

- **Authentication**

- User obtains certificate from Certificate Authority
- Connects to UI by ssh
- Downloads certificate
- **Single logon** – to UI - create proxy
- then **Grid Security Infrastructure** uses proxies to identify users to other machines

- **Authorisation - currently**

- User joins Virtual Organisation
- VO negotiates access to Grid nodes and resources (CE, SE)
- Authorisation tested by CE, SE: **gridmapfile** maps user to local account



- “X 509 Digital certificate”
- **Certification Authorities (CAs)**
  - ~one per country; each builds network of “Registration Authorities” (RA) who issue certificates
- **CAs are mutually recognized – to enable international collaboration**
  - International Grid Trust Federation <http://www.gridpma.org/>
- **For Ireland: <https://www.cs.tcd.ie/grid-ireland/gi-ca/>**
- **CA issues certificates to**
  - Users: you get a Certificate and use it to access a grid
  - Sites providing resources
- **VITAL:**
  - **KEEP YOUR CERTIFICATE SECURE !!!!!**

## Before VOMS

- User is authorised as a member of a single VO
- All VO members have same rights
- Gridmapfiles are updated by VO management software: map the user's DN to a local account
- `grid-proxy-init`

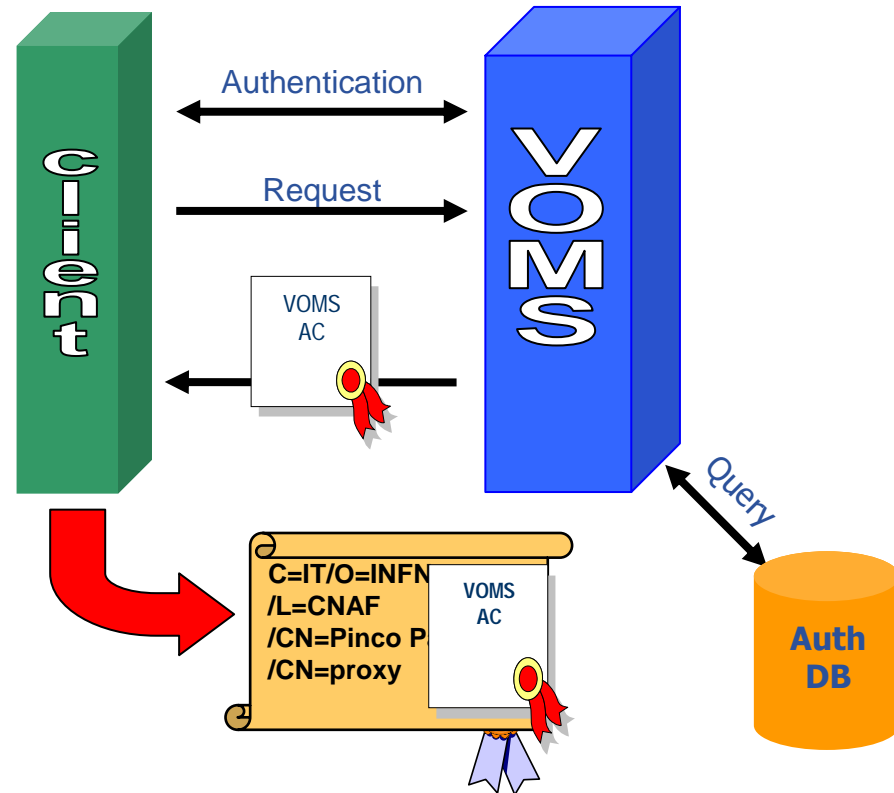
## VOMS

- User can be in multiple VOs
  - Aggregate rights
- VO can have groups
  - Different rights for each
    - Different groups of experimentalists
    - ...
  - Nested groups
- VO has roles
  - Assigned to specific purposes
    - E.g. system admin
    - When assume this role
- Proxy certificate carries the additional attributes
- `voms-proxy-init`

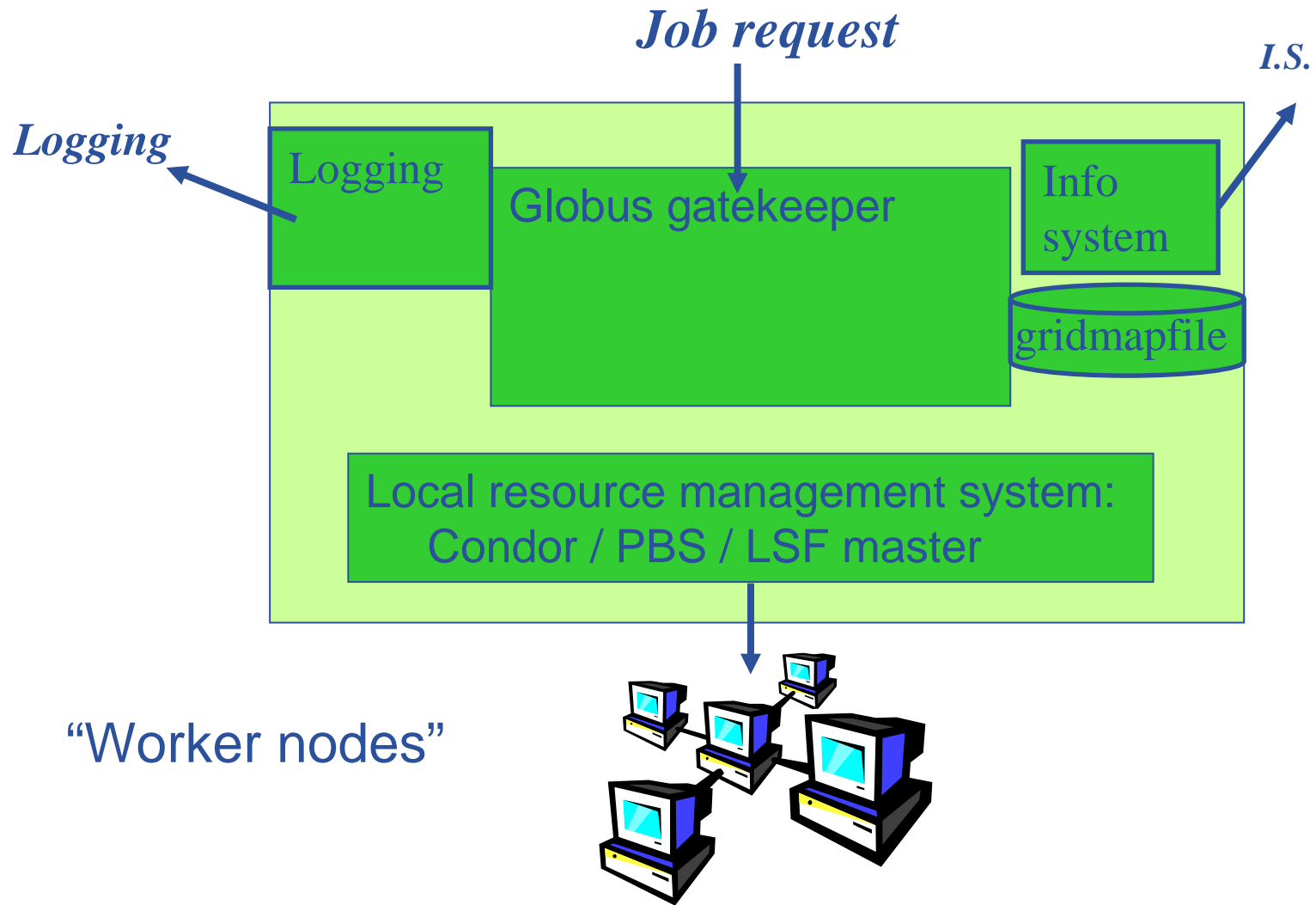
**VOMS – now in both the production (LCG) and pre-production (gLite) middleware**



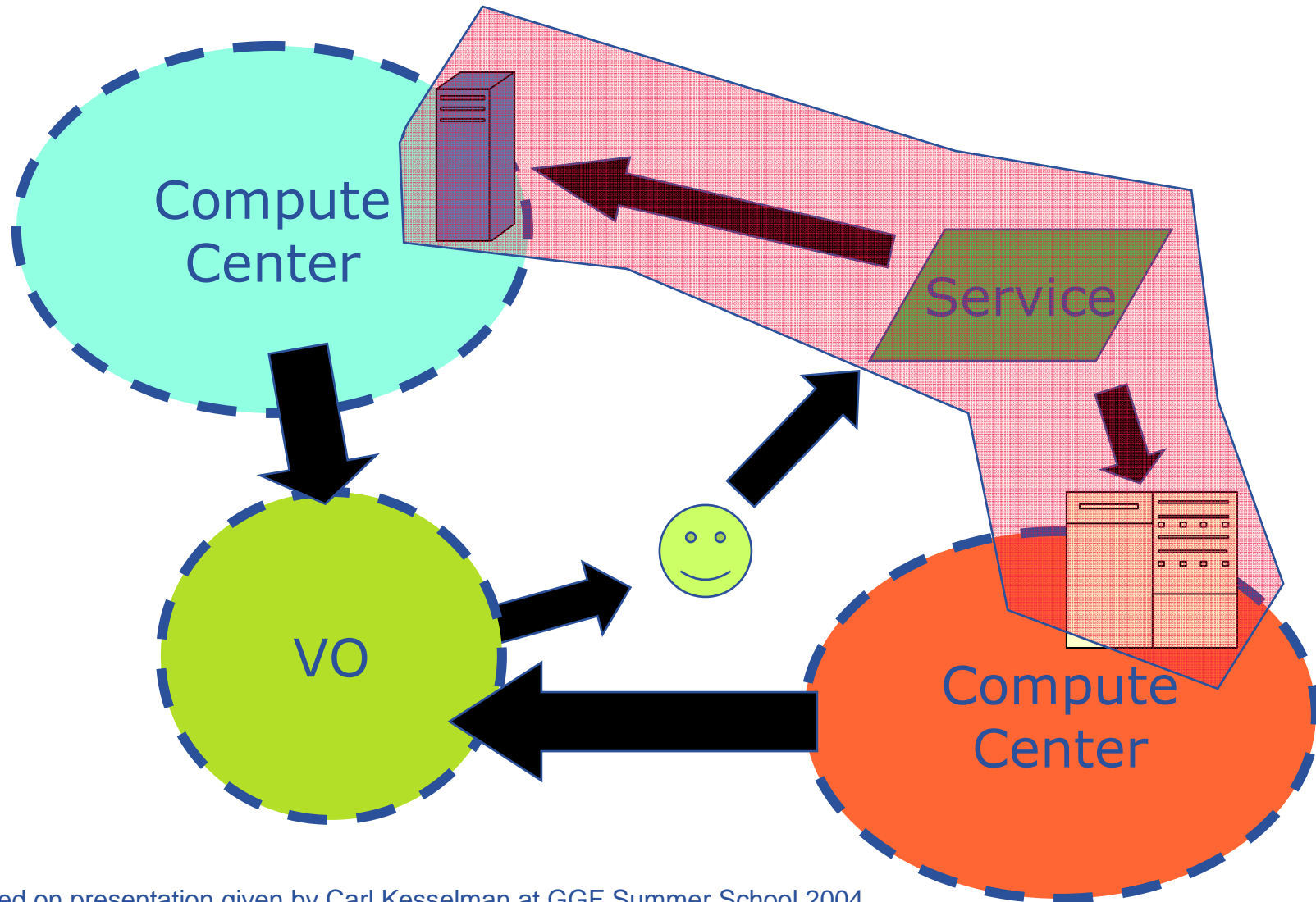
- Bare certificates are not enough for defining user capabilities on the Grid
- Users belong to VO's, to groups inside a VO and may have special roles
- VOMS provides a way to add attributes to a certificate proxy:
  - mutual authentication of client and server
  - VOMS produces a signed **Attribute Certificate (AC)**
  - the client produces a new proxy that contains the attributes
- The attributes are used to provide the user with additional capabilities according to the VO policies.



**VOMS – now in both the production (LCG) and pre-production (gLite) middleware**



# Use Delegation to Establish Dynamic Distributed System



slide based on presentation given by Carl Kesselman at GGF Summer School 2004

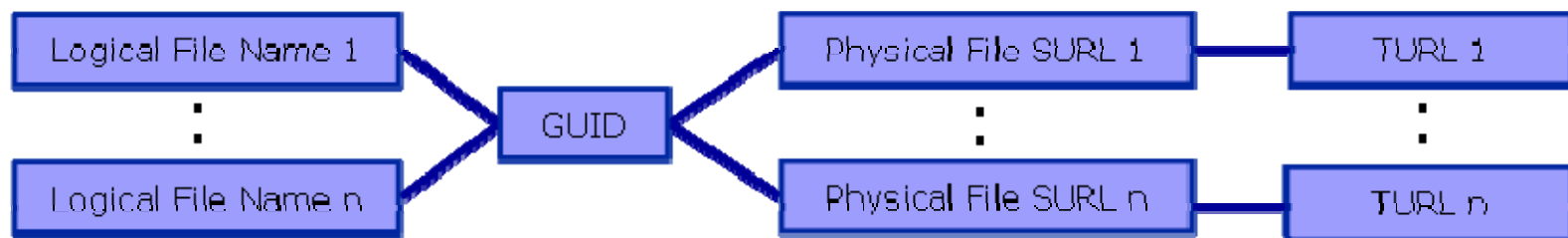
# A closer look at the main EGEE grid services

## 2. Data services

- **Files**
  - File Access Pattern:
    - Write once, read-many
  
- **3 service types for data**
  - Storage
  - Catalogs
  - Movement

- **Provides**
  - Storage for files
  - Transfer protocol (gsiFTP)
  - POSIX-like file access
    - Grid File Access Layer (**GFAL**)
      - *API interface*
      - *To read parts of files too big to copy*
- **Two types**
  - “Classic” SE
    - Massive storage system - disk or tape based
  - “SRM” SE
    - SE’s are virtualised by common interface: “SRMv1”
    - SRM = Storage Resource Manager
    - work in progress to migrate to SRMv2

- **Logical File Name (LFN)**
  - An alias created by a user to refer to some item of data, e.g. “lfn:cms/20030203/run2/track1”
- **Globally Unique Identifier (GUID)**
  - A non-human-readable unique identifier for an item of data, e.g. “guid:f81d4fae-7dec-11d0-a765-00a0c91e6bf6”
- **Site URL (SURL) (or Physical File Name (PFN) or Site FN)**
  - The location of an actual piece of data on a storage system, e.g. “srm://pcrd24.cern.ch/flatfiles/cms/output10\_1” (SRM)  
“sfn://lxshare0209.cern.ch/data/alice/ntuples.dat” (Classic SE)
- **Transport URL (TURL)**
  - Temporary locator of a replica + access protocol: understood by a SE, e.g. “rfio://lxshare0209.cern.ch//data/alice/ntuples.dat”



- **File catalogs in LCG:**
  - They keep track of the location of copies (replicas) of files
  - The DM tools and APIs and the WMS interact with them
- **EDG's Replica Location Service (RLS, "old!")**
  - Replica Metadata Catalog (**RMC**) + Local Replica Catalog (**LRC**)
  - Some performance problems
  - Currently on Grid-Ireland
- **New LCG File Catalog (LFC, "current!")**
  - In production
  - Better performance and scalability
  - Provides new features: security, hierarchical namespace, transactions...
- **LFC and lcg-utils commands exist to manage catalog and files on SE's**

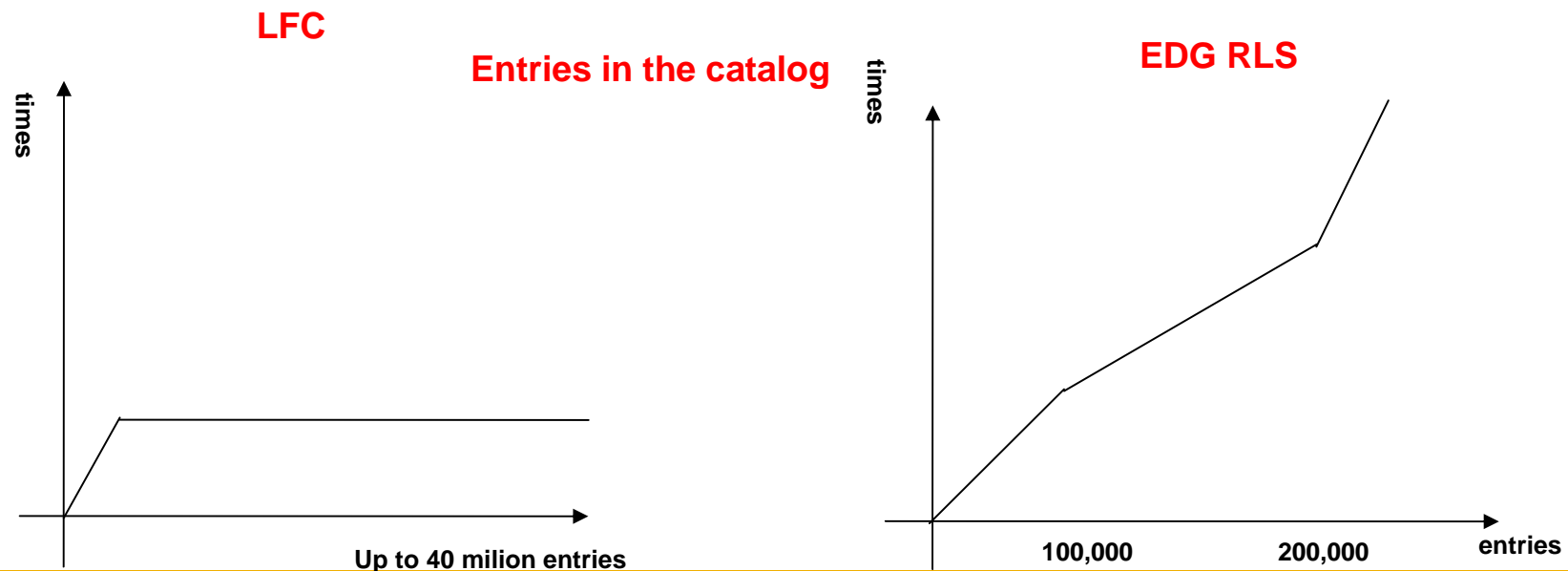


## • LFC

- No significant increase in operation times with a large number of entries – insert time or query rate
- Tested with up to 40 million entries
- Two previous catalogues combined into one

## • Predecessor from EDG (European DataGrid)

- Individual query time increased rapidly up to 100,000 entries
- Individual insert time for an individual increased beyond 200,000 entries.



If a site acts as a central catalog for several VOs, it can either have:

- One LFC server, with one DB account containing the entries of all the supported VOs. You should then create one directory per VO.
- Several LFC servers, having each a DB account containing the entries for a given VO.

Both scenarios have consequences on the handling of database backups

- **Minimum requirements (First scenario)**
  - 2Ghz processor with 1GB of memory (not a hard requirement)
  - Dual power supply
  - Mirrored system disk

The **L**CG **F**ile **C**atalog fixes the performance and scalability problems of EDG (European Data Grid) file catalogs.

## Provides

- Bulk operations.
- Cursors for large queries.
- Timeouts and retries for client operations.

## Added features :

- User exposed transaction API.
- Hierarchical namespace and namespace operations.
- Integrated GSI Authentication and Authorization.
- Access Control Lists (Unix Permissions and POSIX ACLs).
- Checksums.

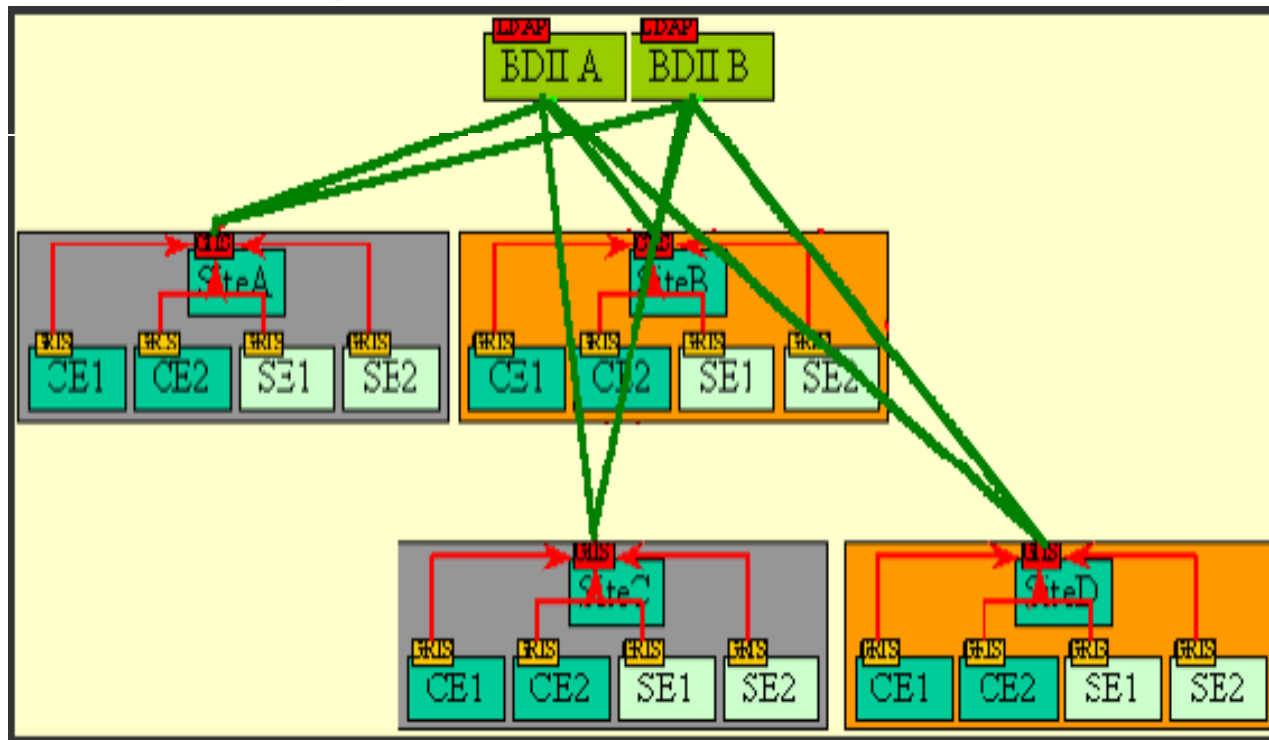
Supported database backends: **Oracle** and **MySQL**

**GFAL** integration and support to **lcg-\*** done by Grid Deployment group

# A closer look at the main EGEE grid services

## 3. Information services

- Users can interrogate BDII servers by 2 sets of commands
  - `lcg-infosites`
  - `lcg-info`



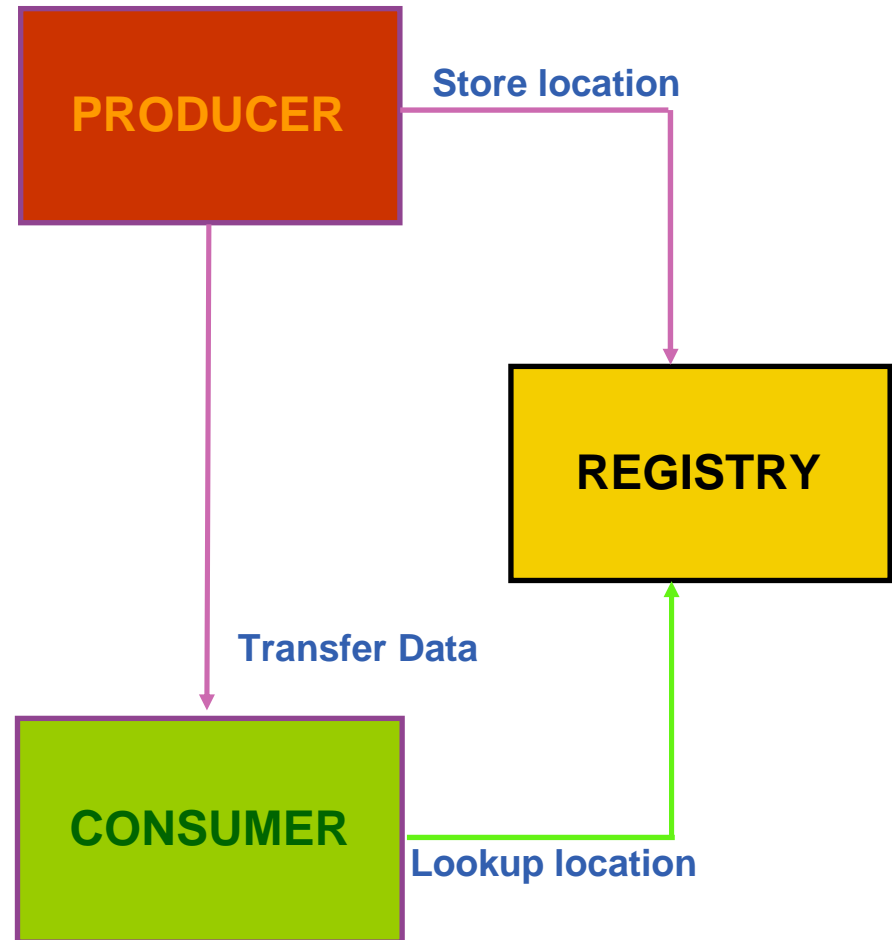
- LDAP (Lightweight Directory Access Protocol)
- Glue Schema.

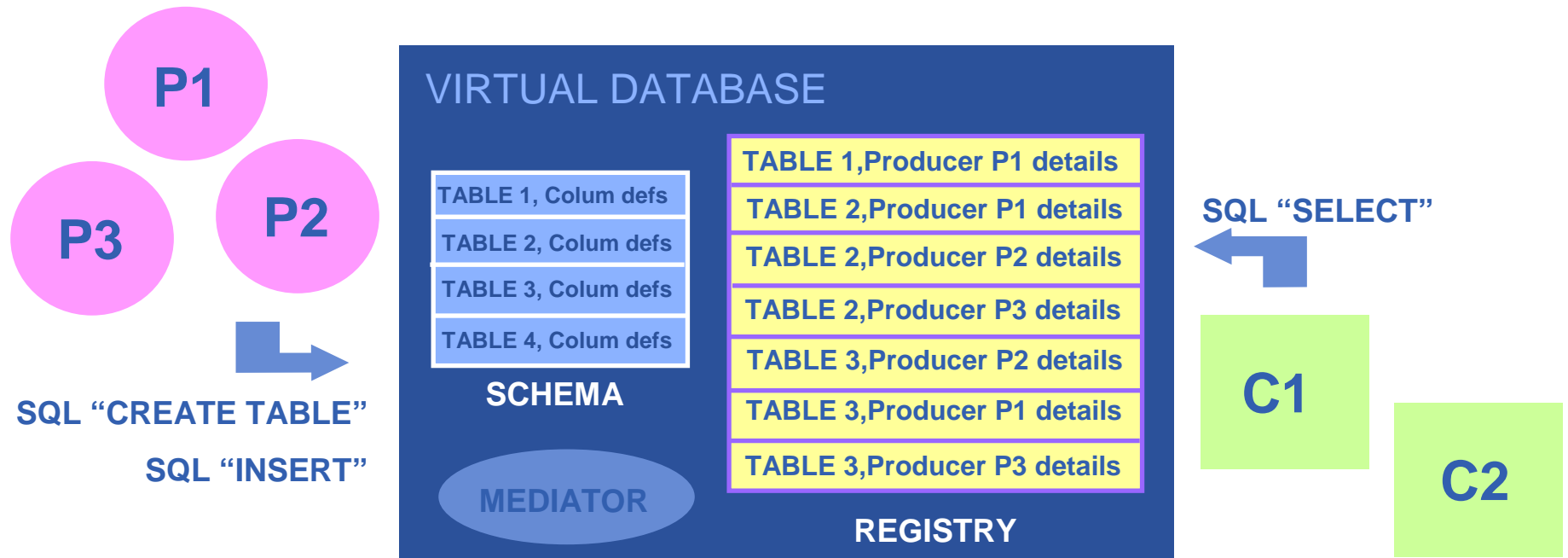
- **Relational Grid Monitoring Architecture (R-GMA)**
  - Developed as part of the EuropeanDataGrid Project (EDG)
  - Now as part of the EGEE project.
  - Based the Grid Monitoring Architecture (GMA)
  
- **Uses a relational data model.**
  - Data are viewed as a table.
  - Data structure defined by the columns.
  - Each entry is a row (tuple).
  - Queried using Structured Query Language (SQL).

name	ID	birth	Group
Tom	4	1977-08-20	HR

`SELECT * FROM people WHERE group='HR'`

- The Producer stores its location (URL) in the Registry.
- The Consumer looks up producer URLs in the Registry.
- The Consumer contacts the Producer to get all the data or the Consumer can listen to the Producer for new data.





There is no central repository!!! There is only a "Virtual Database".

Schema is a list of table definitions: additional tables/schema can be defined by applications

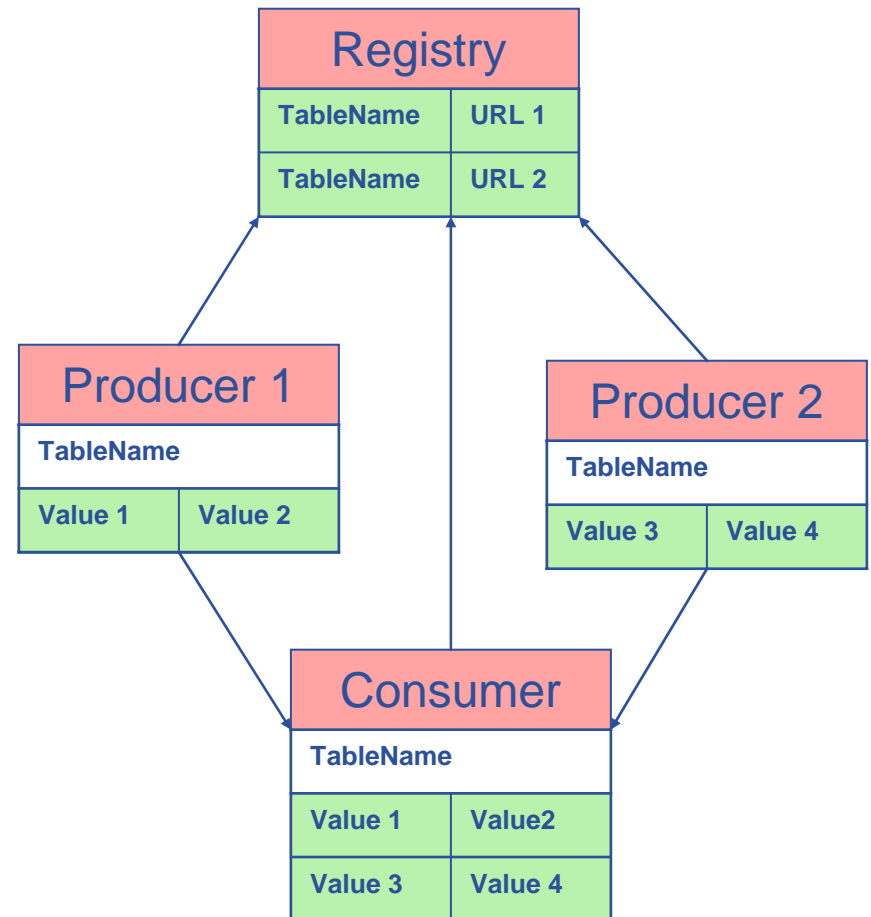
Registry is a list of data producers with all its details.

Producers publish data – from sites, from applications

Consumer read data published.



- The Consumer will get all the URLs that could satisfy the query.
- The Consumer will connect to all the Producers.
- Producers that can satisfy the query will send the tuples to the Consumer.
- The Consumer will merge these tuples to form one result set.



- **Real-time monitor**

- <http://www.hep.ph.ic.ac.uk/e-science/projects/demo/index.html>

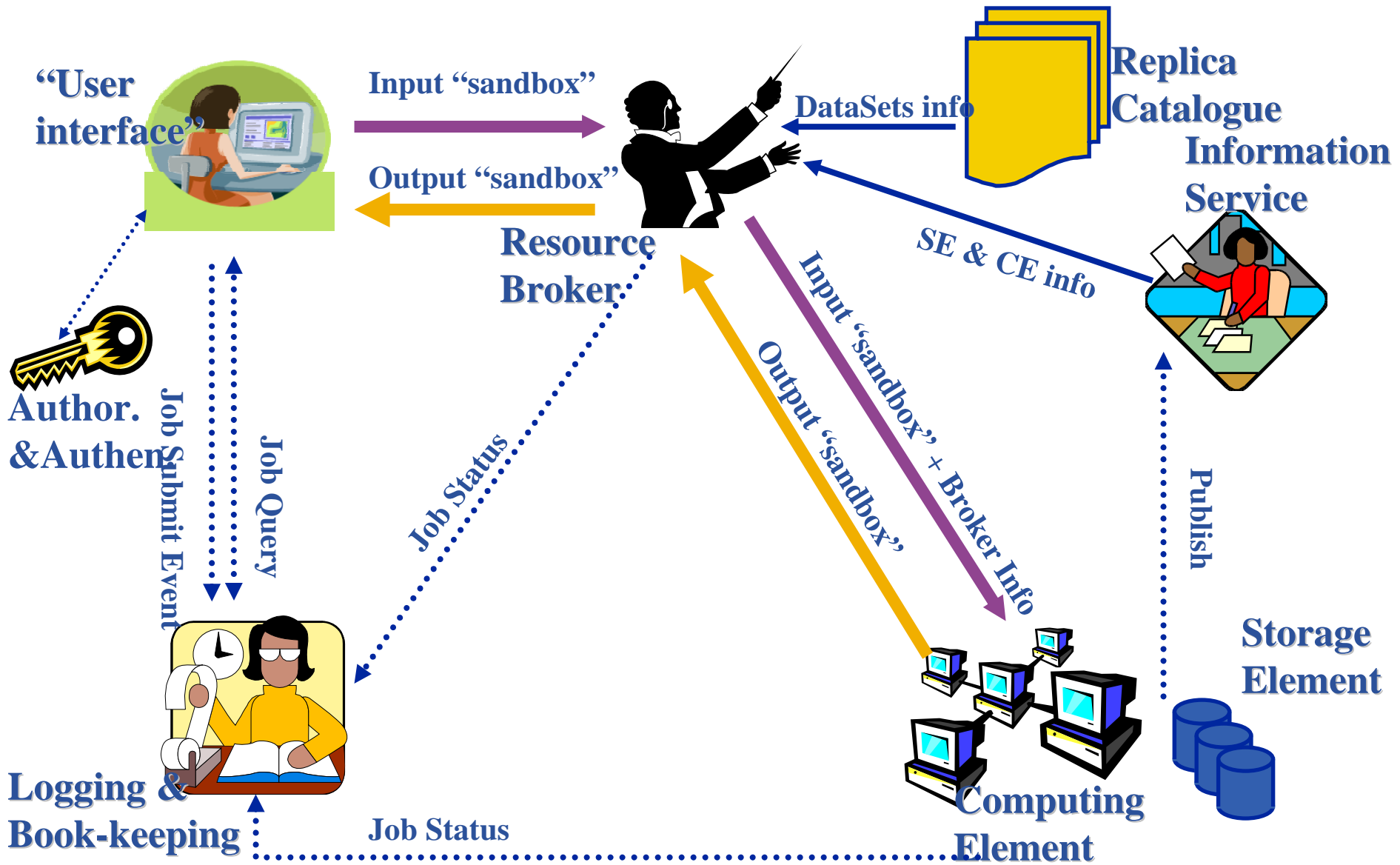
- **Current status**

- <http://goc.grid-support.ac.uk/gridsite/monitoring/>

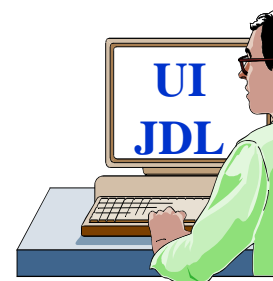
# A closer look at the main EGEE grid services

## 4. Job submission

# Current production middleware



- **The user's interface to the Grid**
- **Command-line interface to**
  - Proxy server
  - Job operations
    - To submit a job
    - Monitor its status
    - Retrieve output
  - Data operations
    - Upload file to SE
    - Create replica
    - Discover replicas
  - Other grid services
- **Also C++ and Java APIs**



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

- 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:testbed0-00019";
DataAccessProtocol = "gridftp";
Requirements = other.Architecture=="INTEL" && \
               other.OpSys=="LINUX" && other.FreeCpus >=4;
Rank = "other.GlueHostBenchmarkSF00";
```

- 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/gridre
OutputSandbox = {"stderr.log", "stdout.log"};
InputData = "lfn: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**

- 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

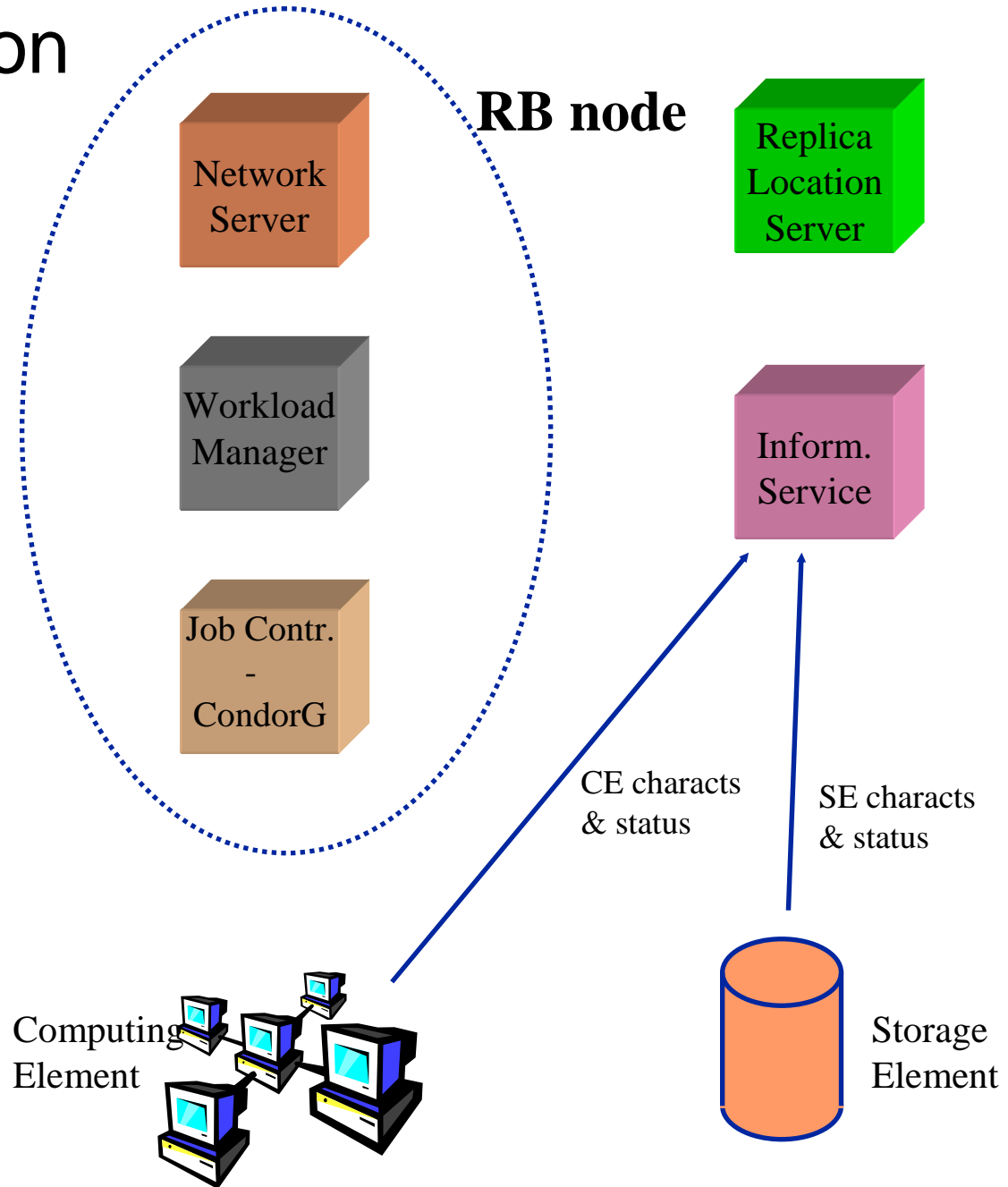
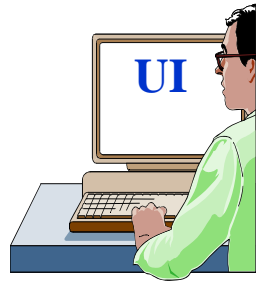
```

Executable = "gridTest";
StdError = "stderr.log";
StdOutput = "stdout.log";
InputSandbox = {"/home/joda/test/...st"};
OutputSandbox = {"stderr.log", "stdout.log"};
InputData = "lfn:testbed0-0001";
DataAccessProtocol = "gridftp";
Requirements = other.Architecture=="INTEL" && \
               other.OpSys=="LINUX" && other.FreeCpus >=4;
Rank = "other.GlueHostBenchmarkSF00";
    
```

Uses BDII Information System



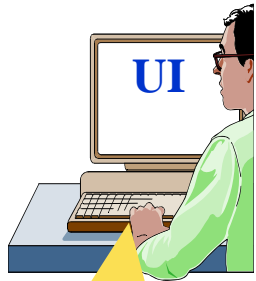
# Job submission



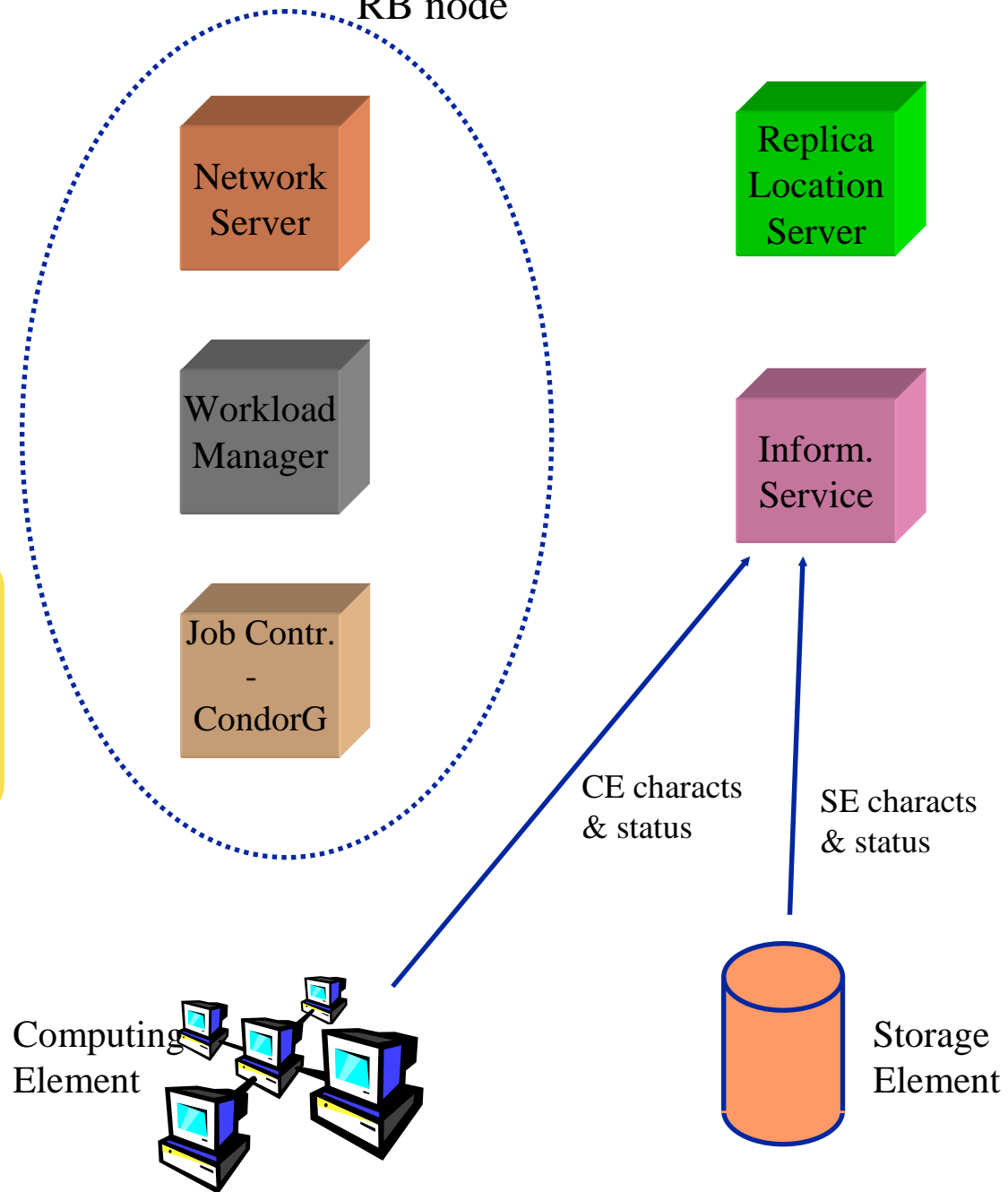
Job Status

submitted

RB node



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



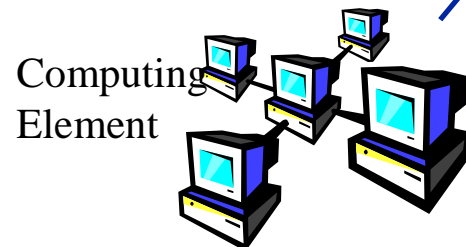
```
edg-job-submit myjob.jdl
```

```
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;
```

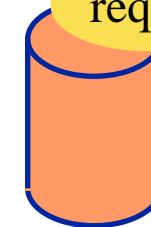
Job  
Statu  
s

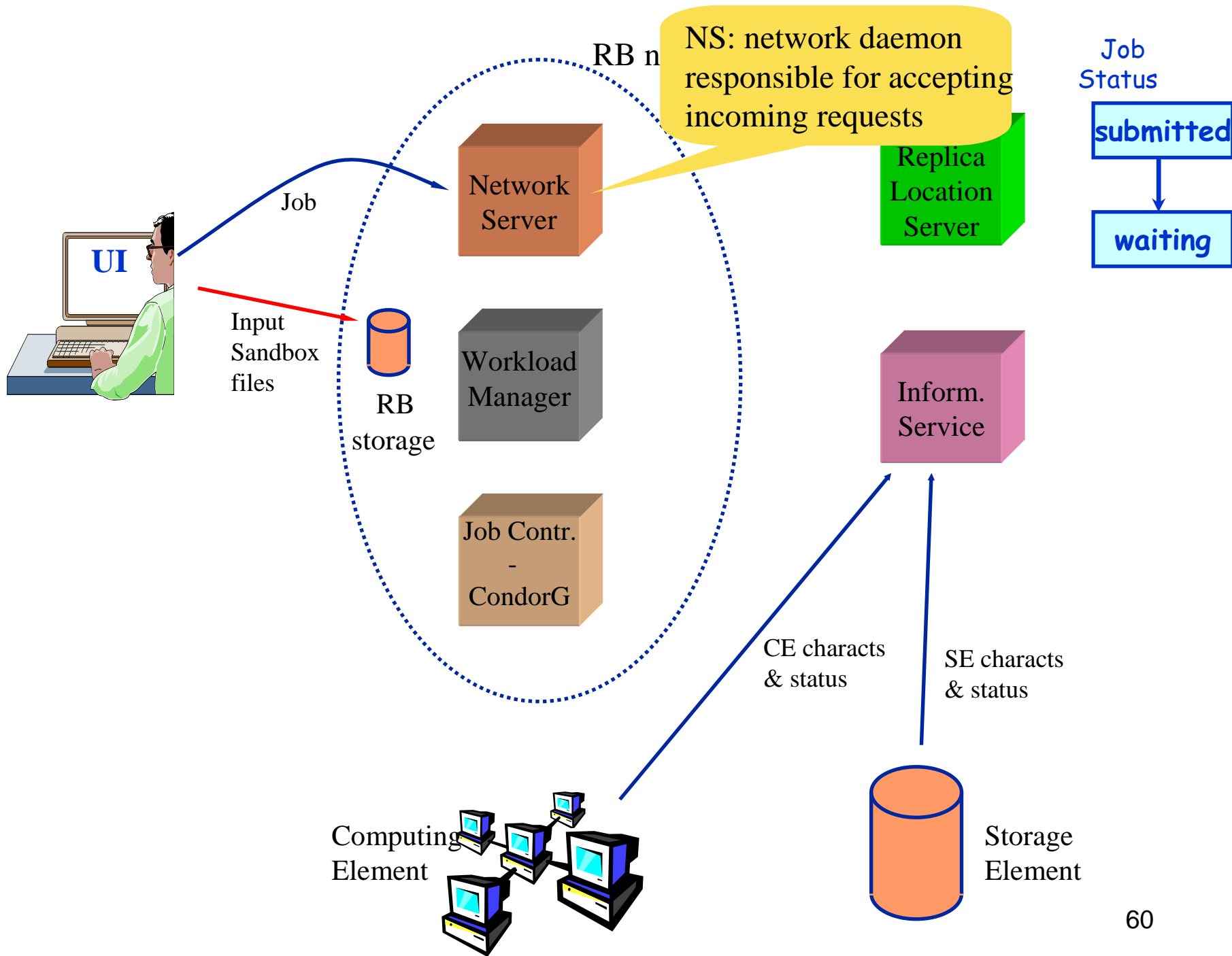
submitted

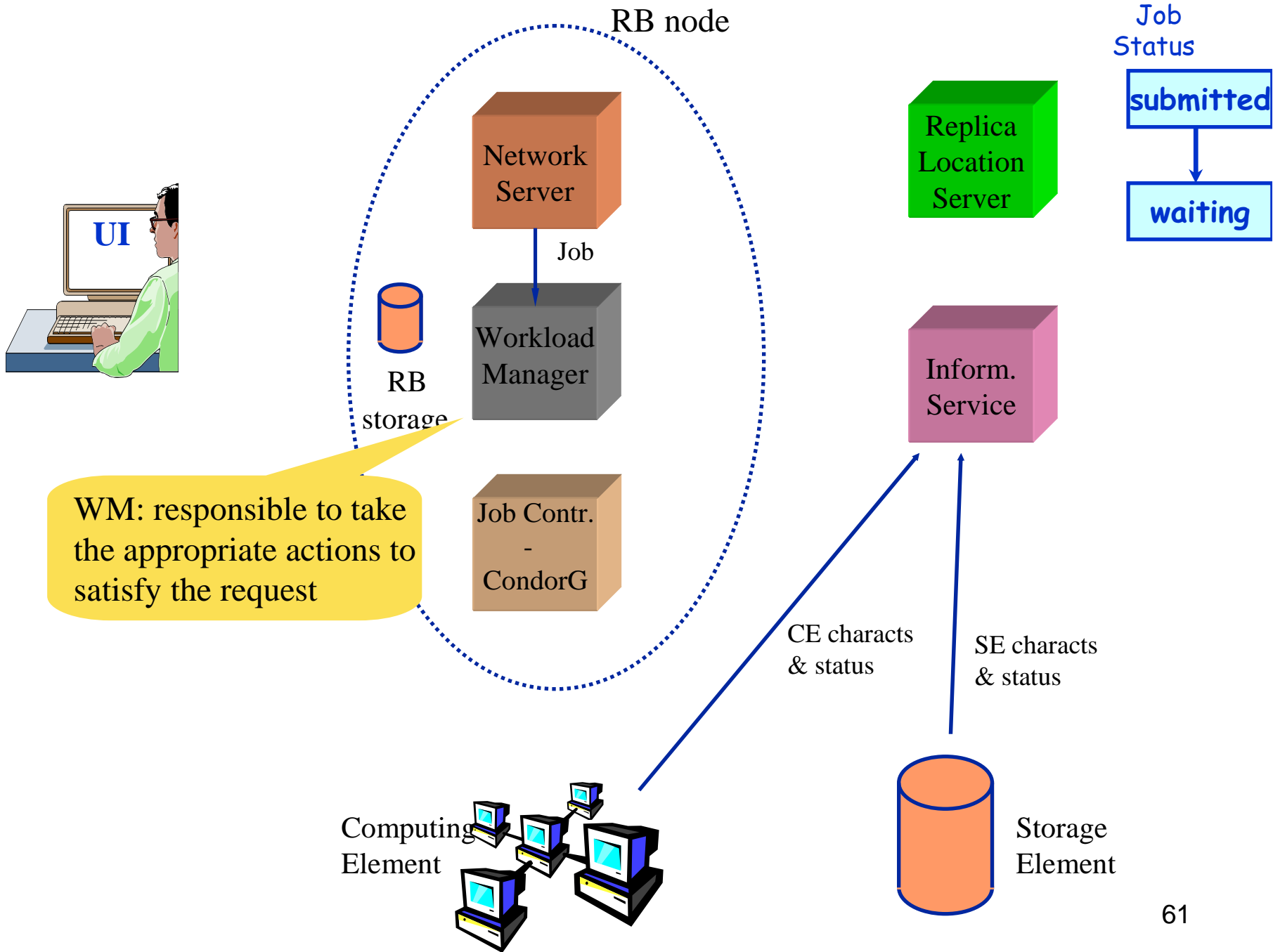


CE characts  
& status

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

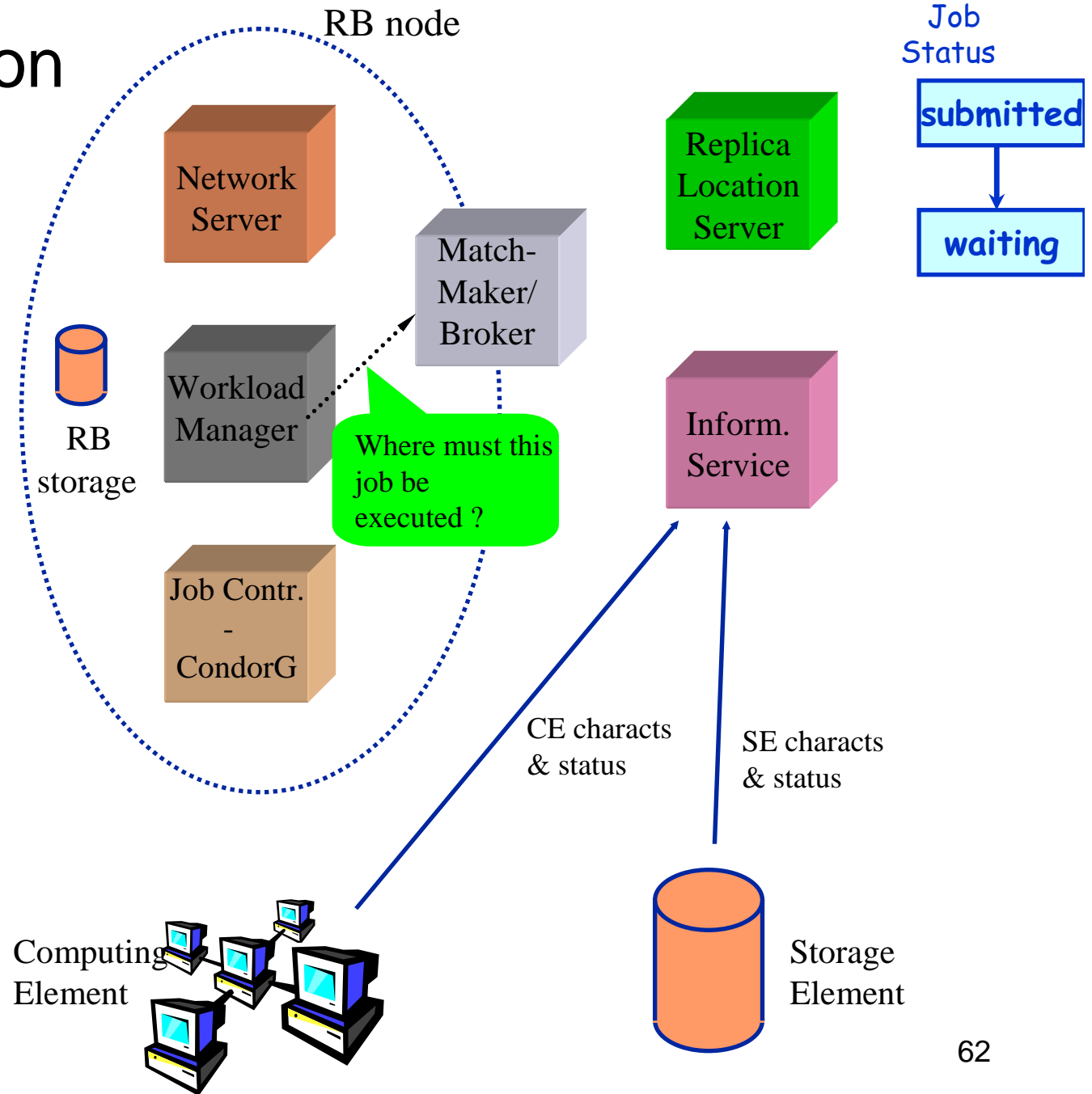
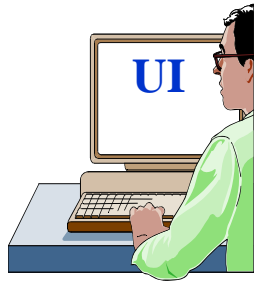




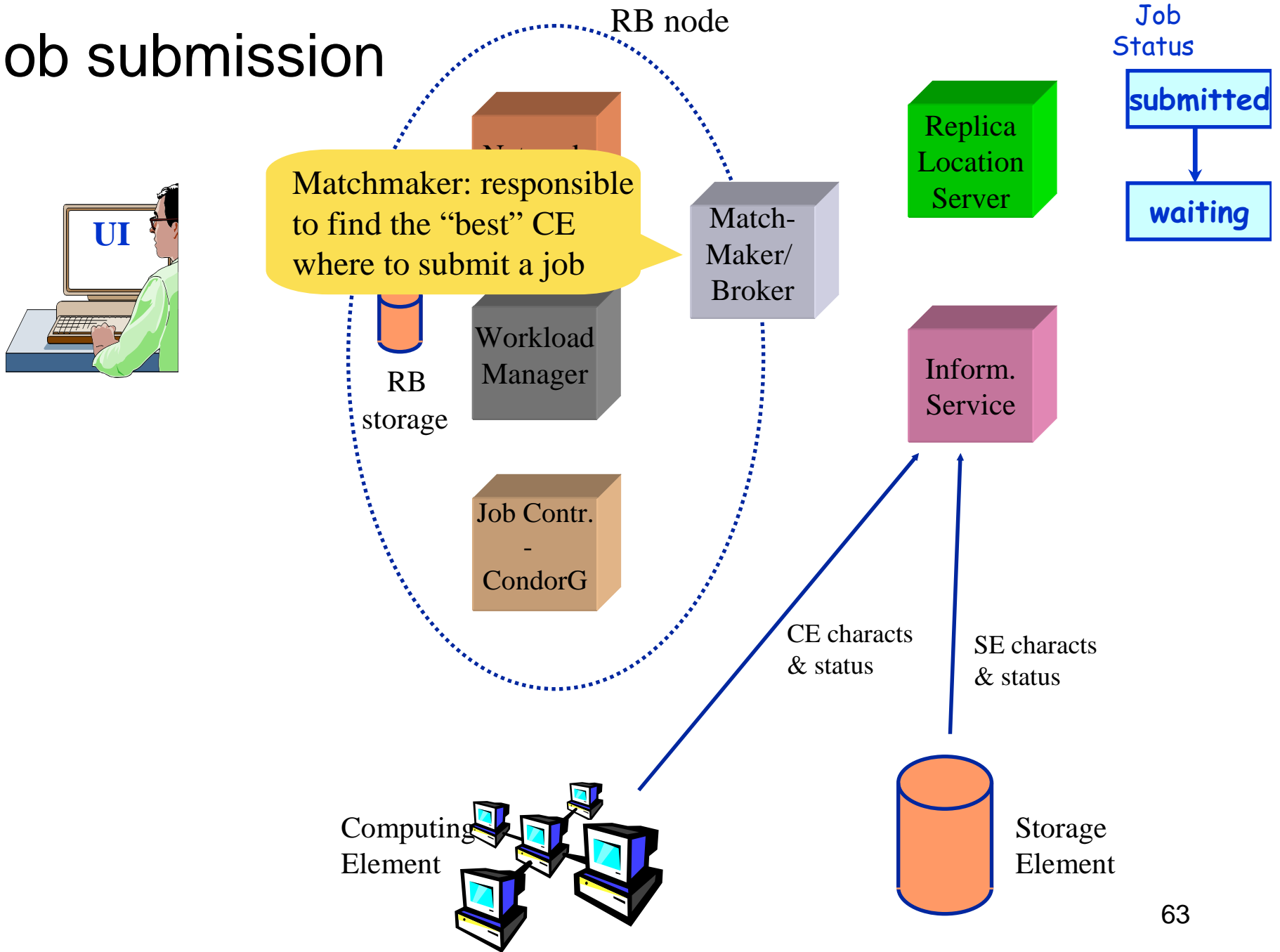


WM: responsible to take the appropriate actions to satisfy the request

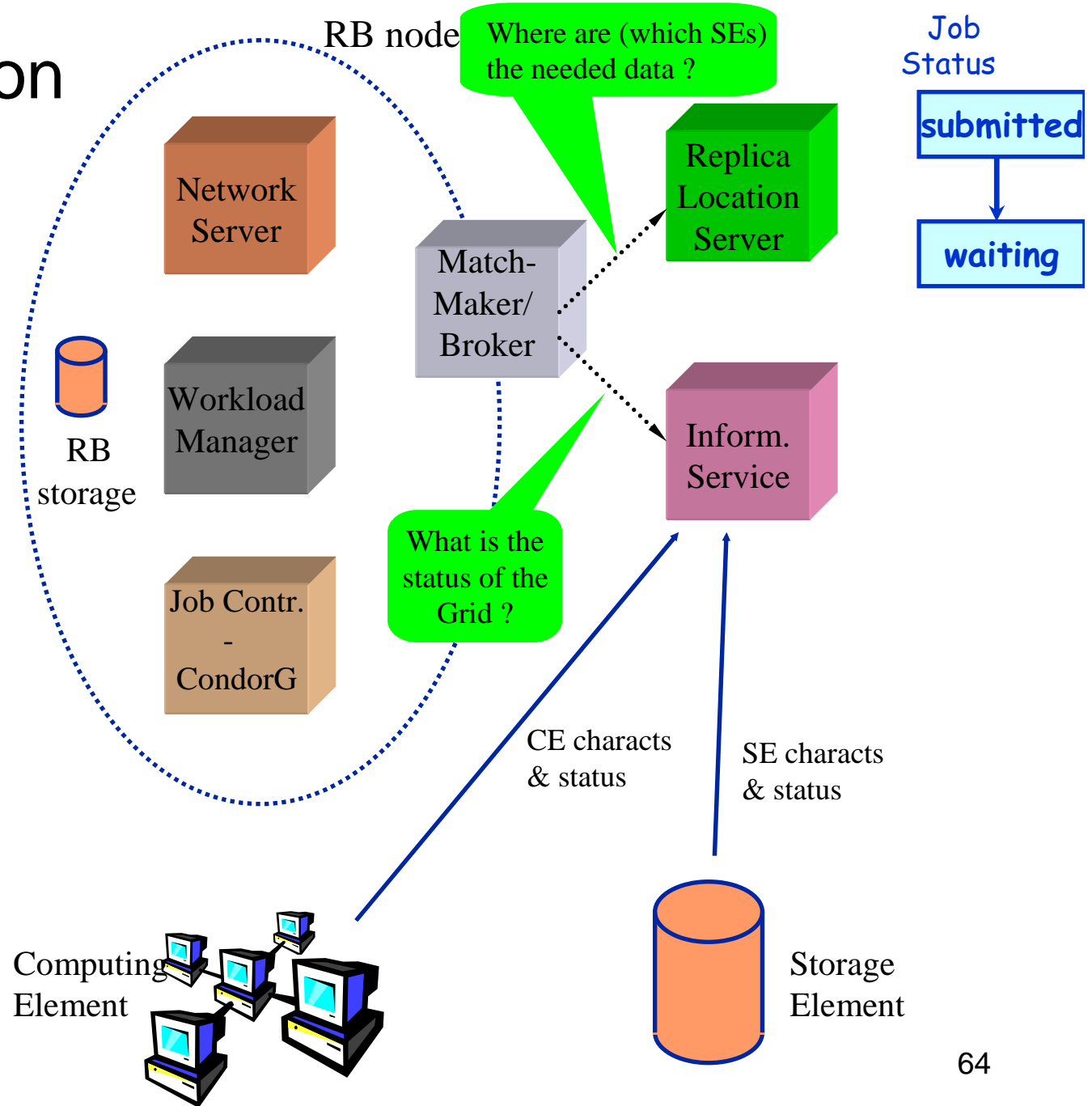
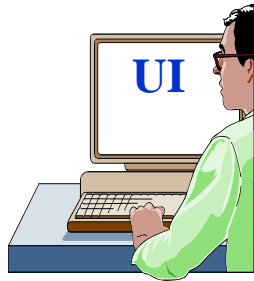
# Job submission



# Job submission

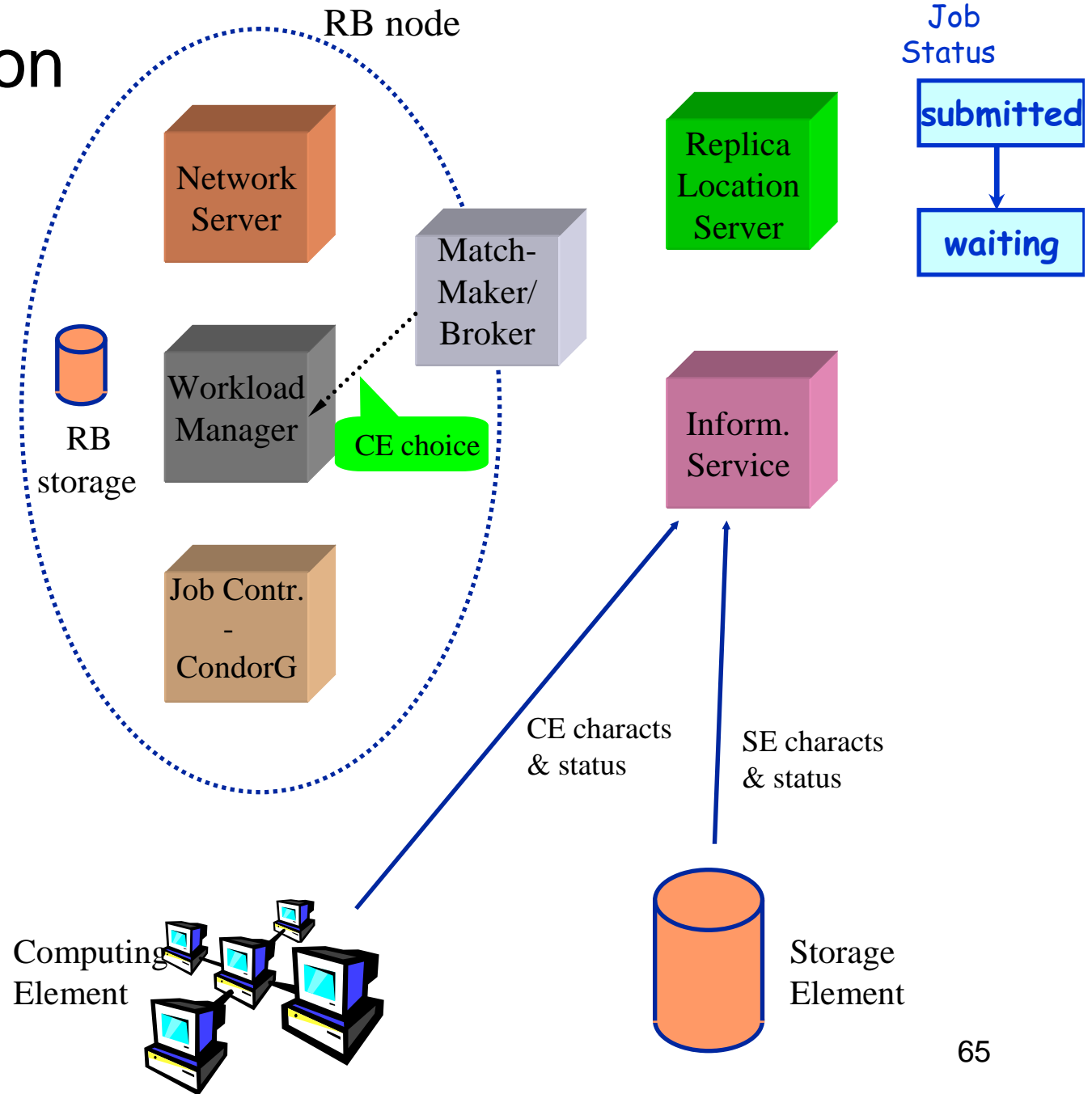
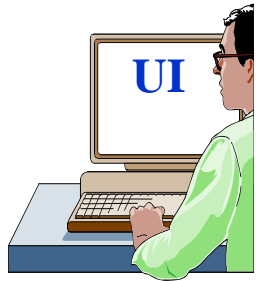


# Job submission

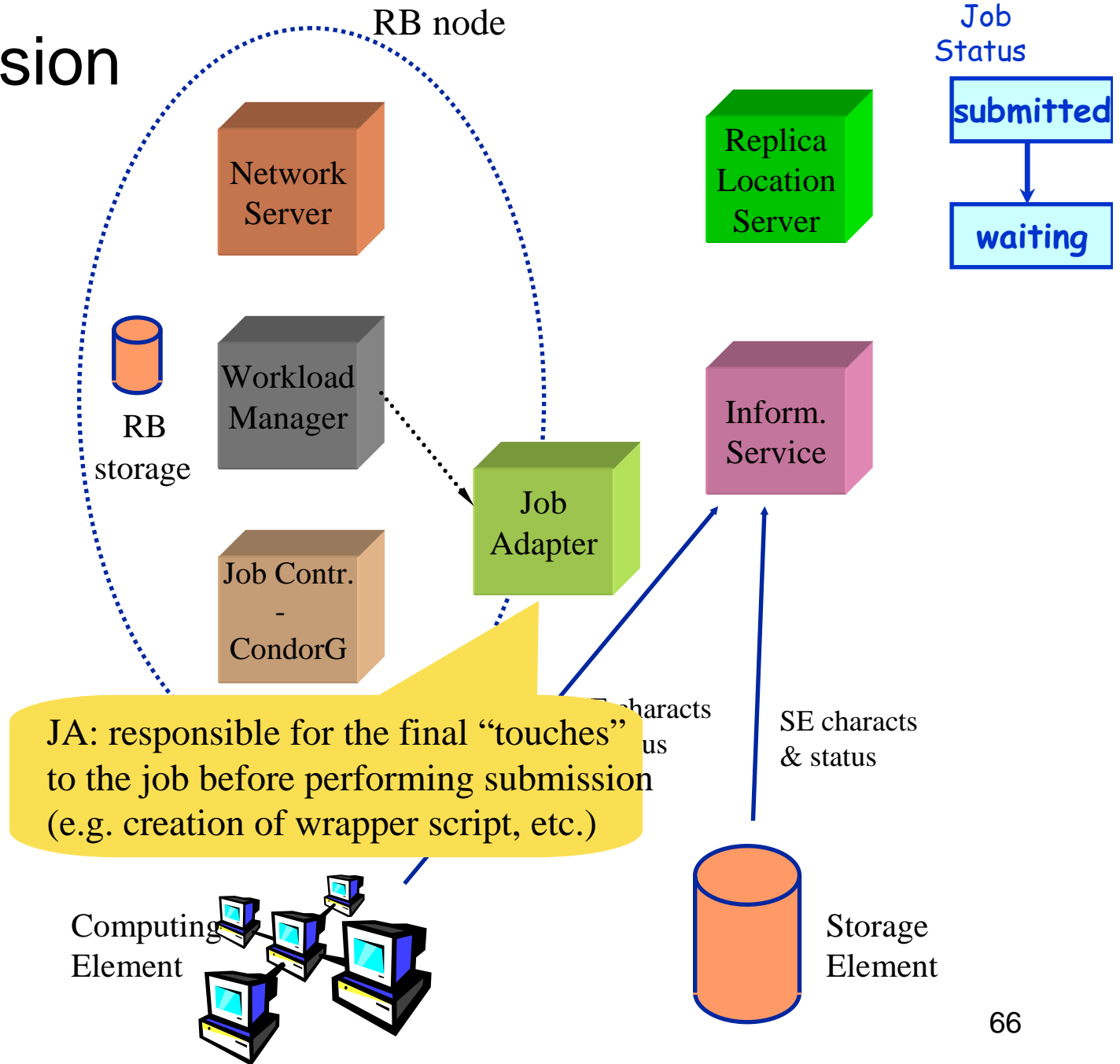
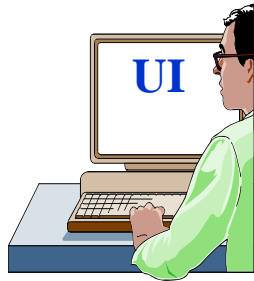




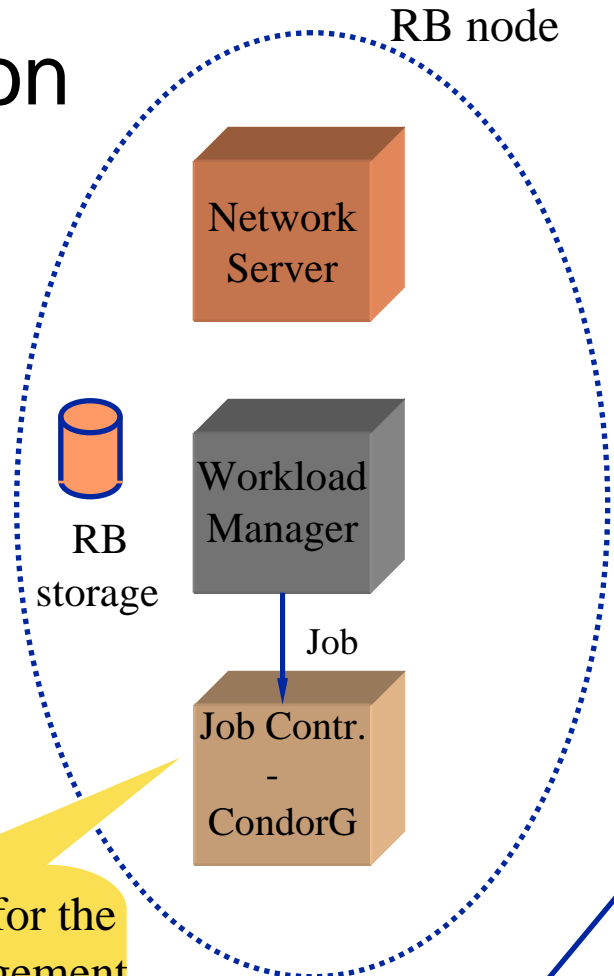
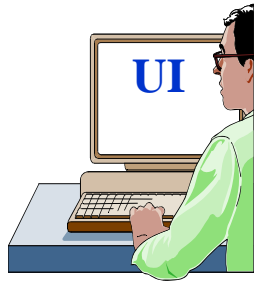
# Job submission



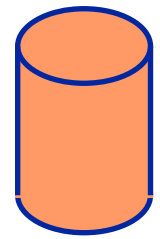
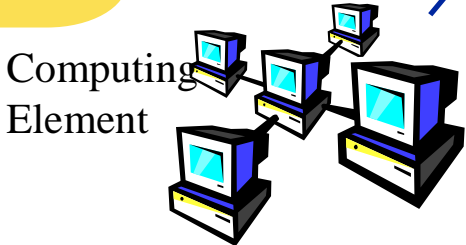
# Job submission



# Job submission



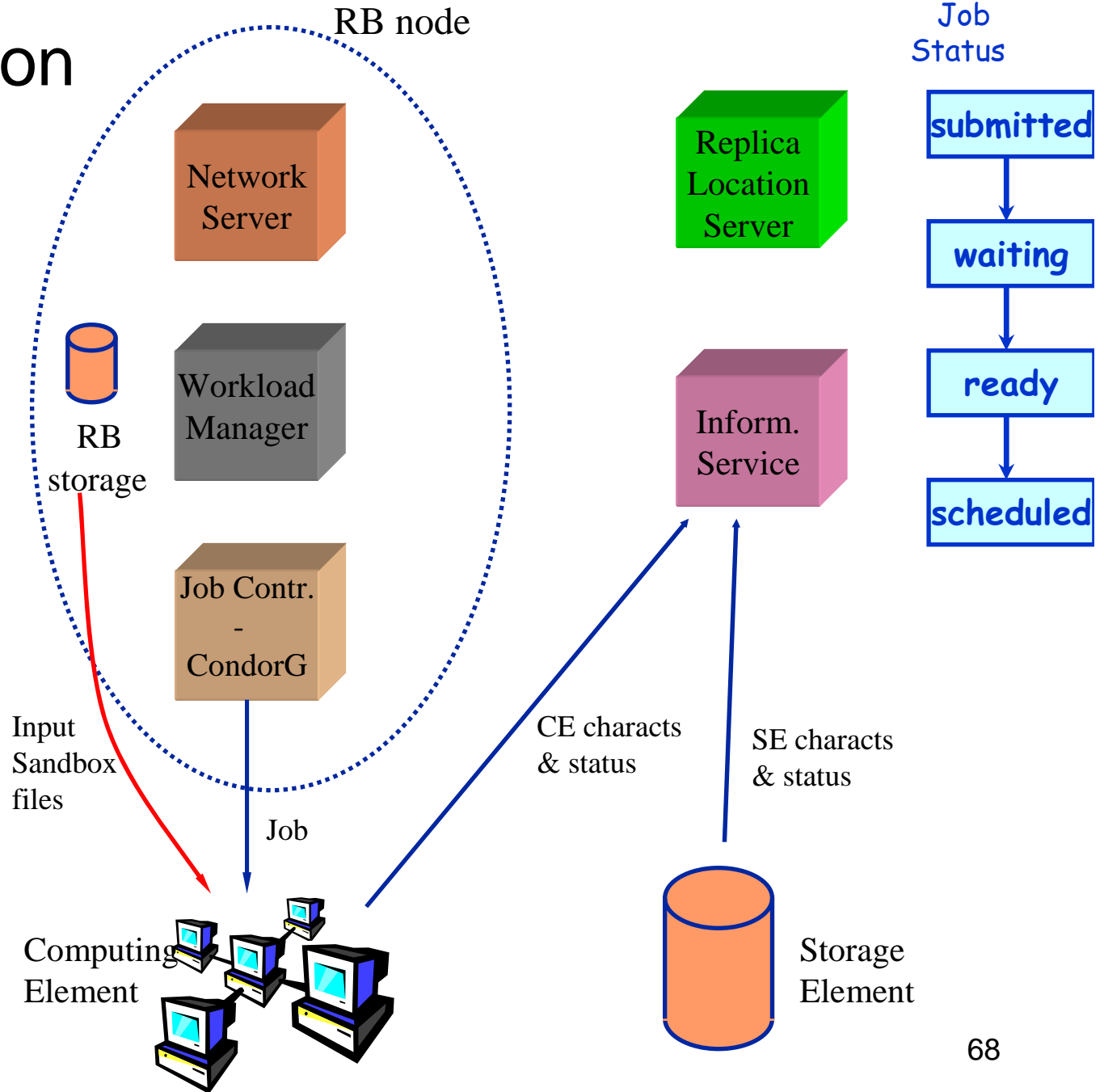
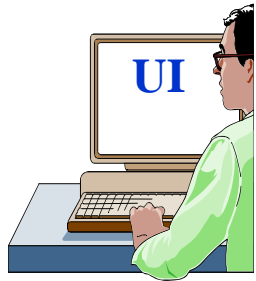
JC: responsible for the actual job management operations (done via CondorG)

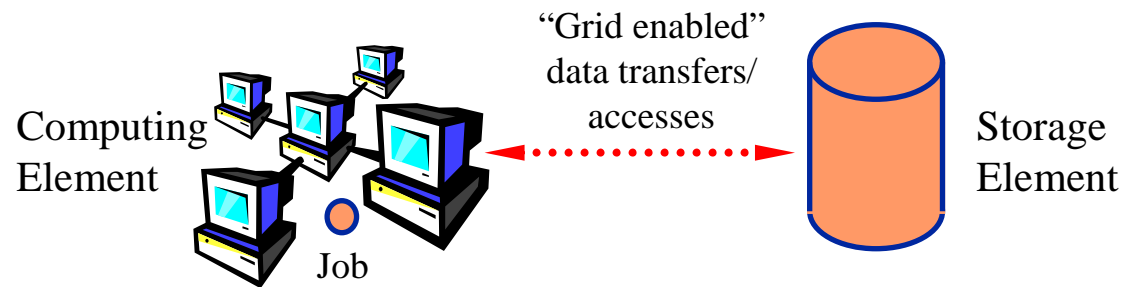
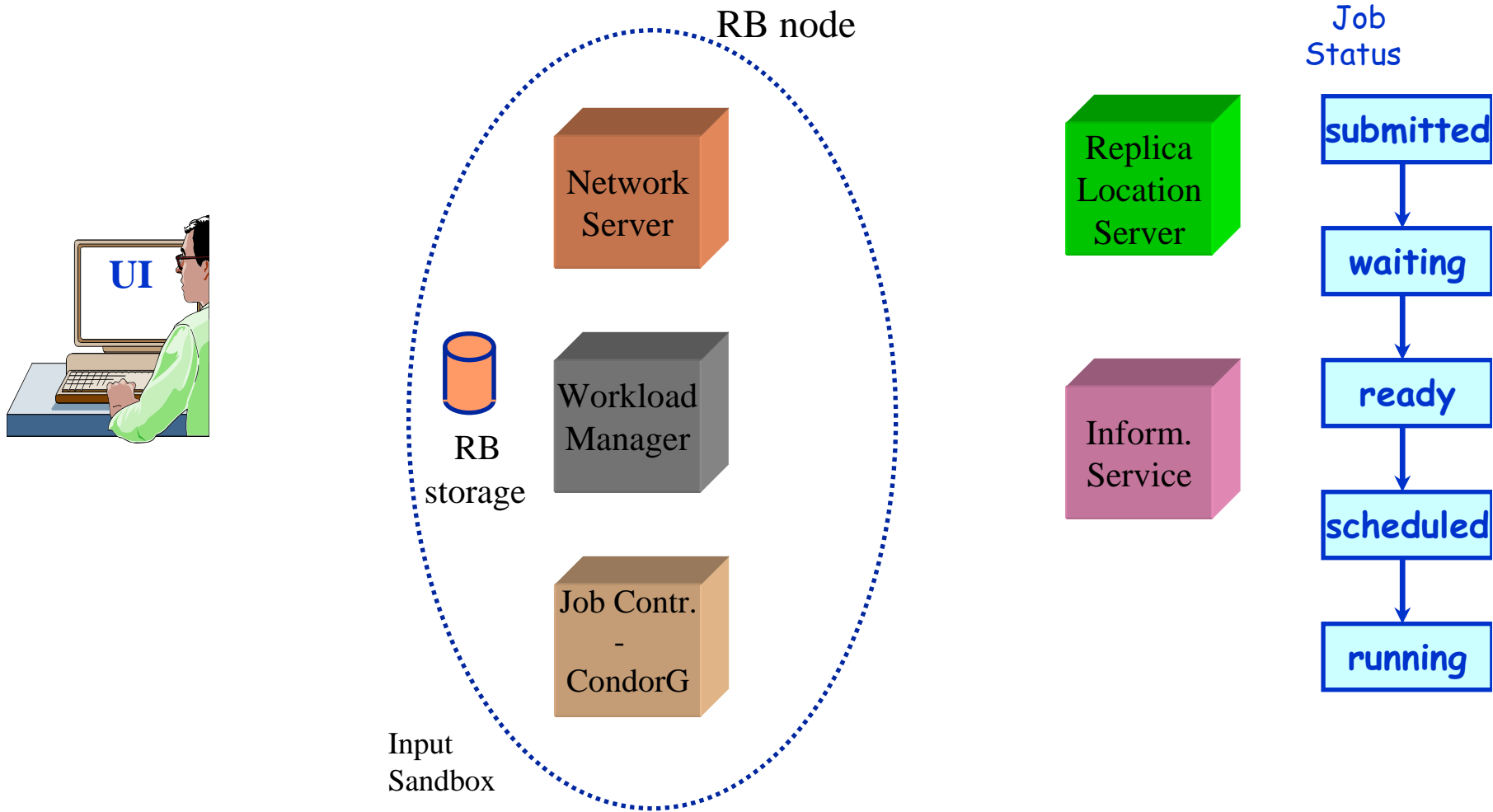


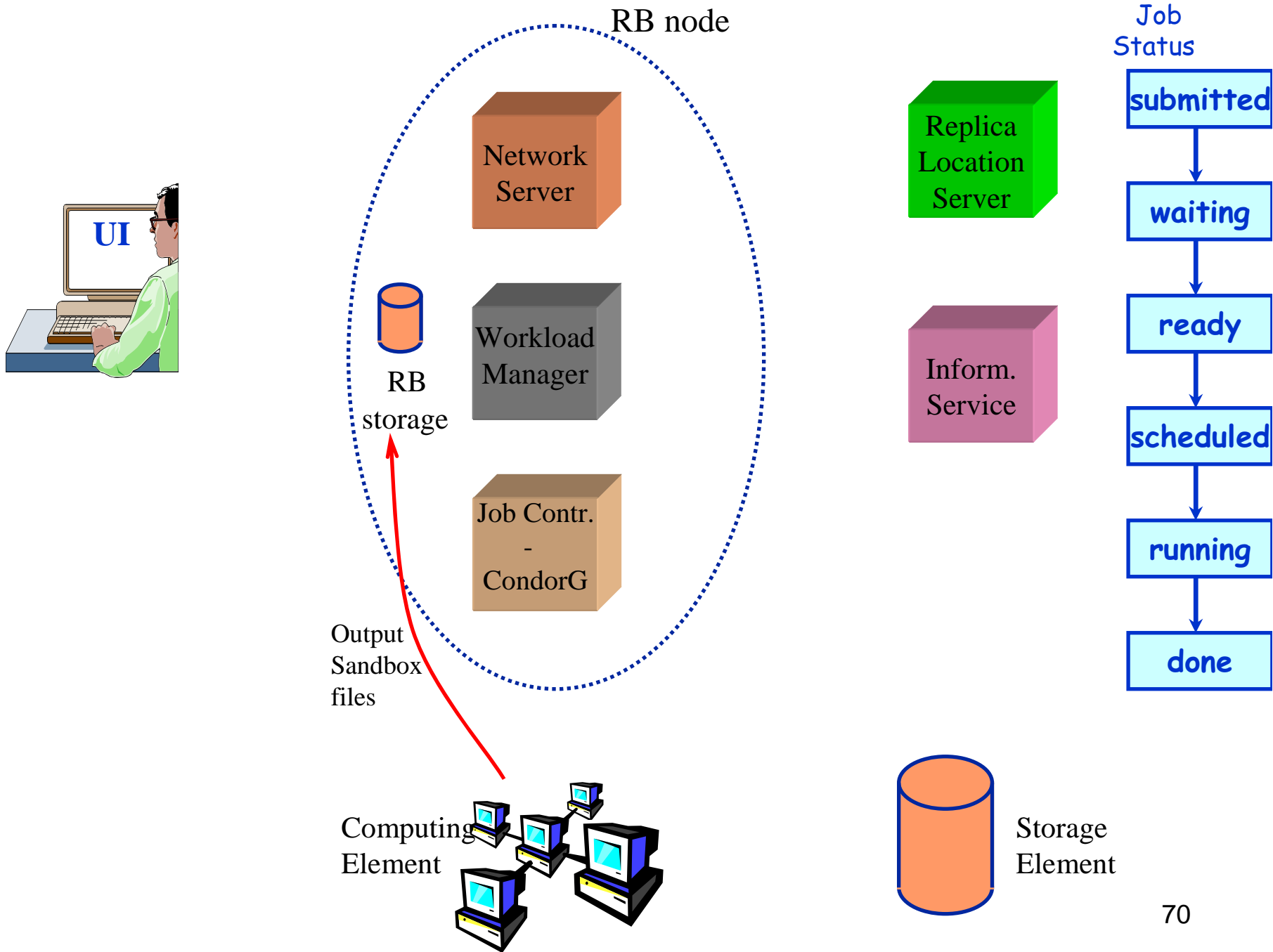
CE characts & status

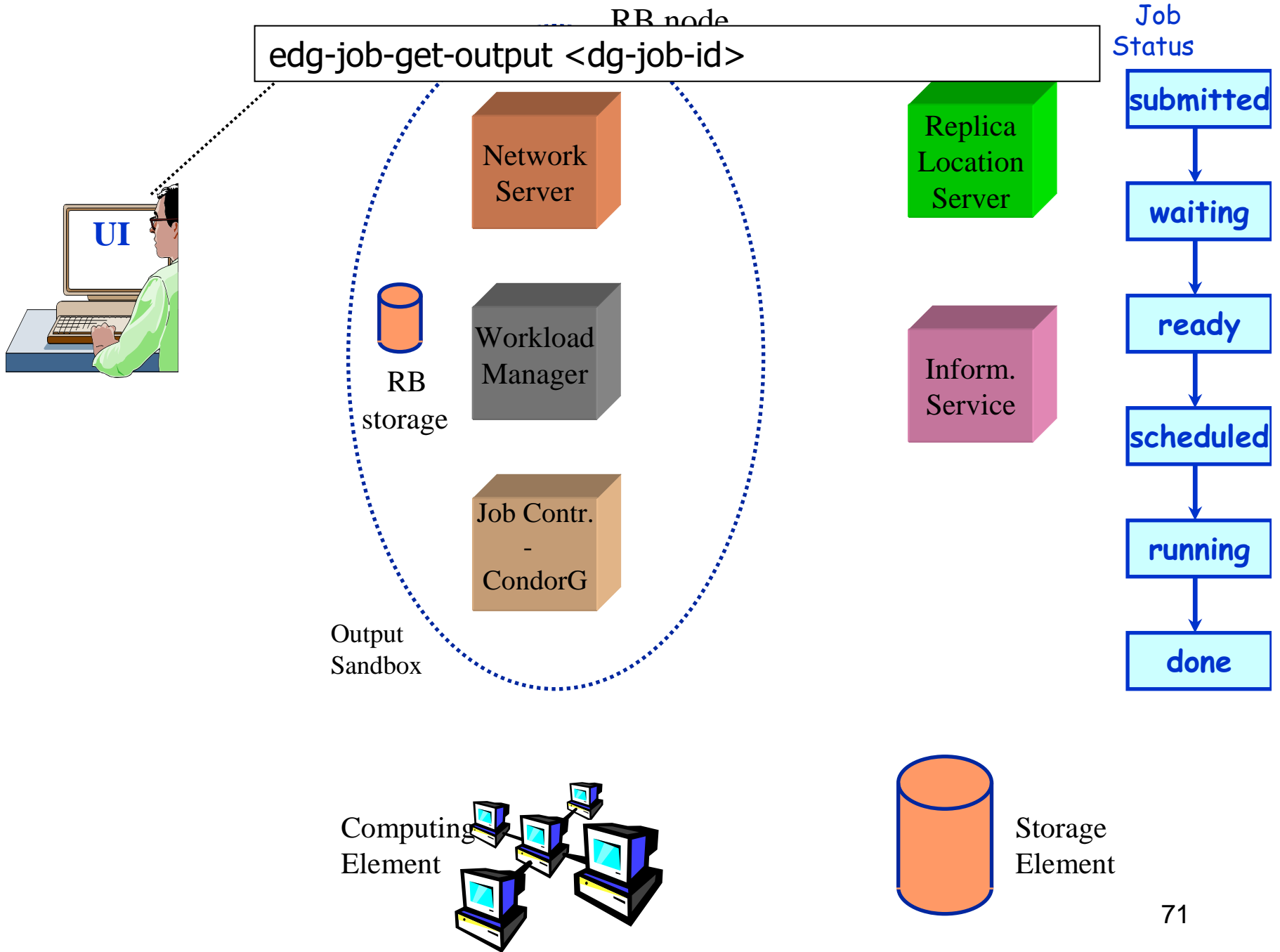
SE characts & status

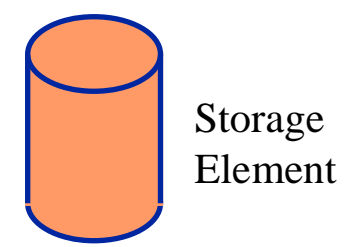
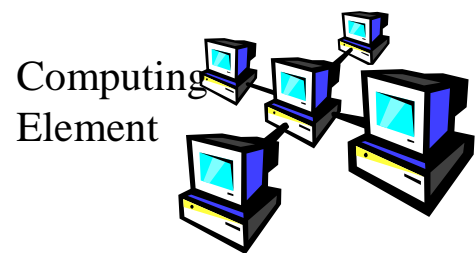
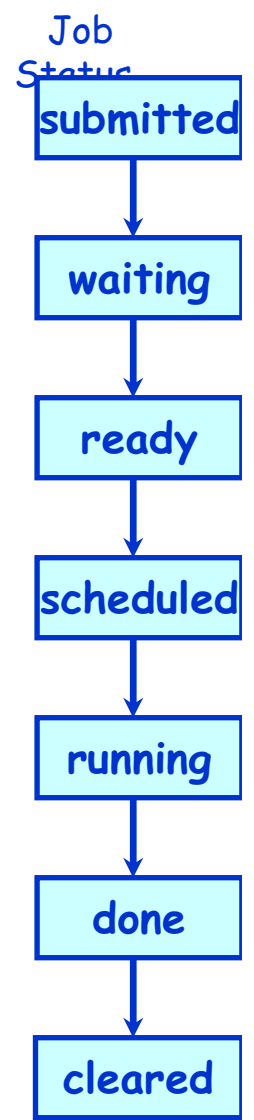
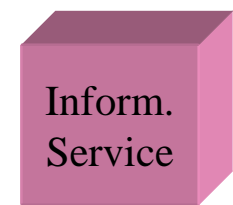
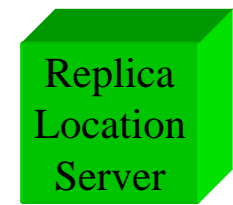
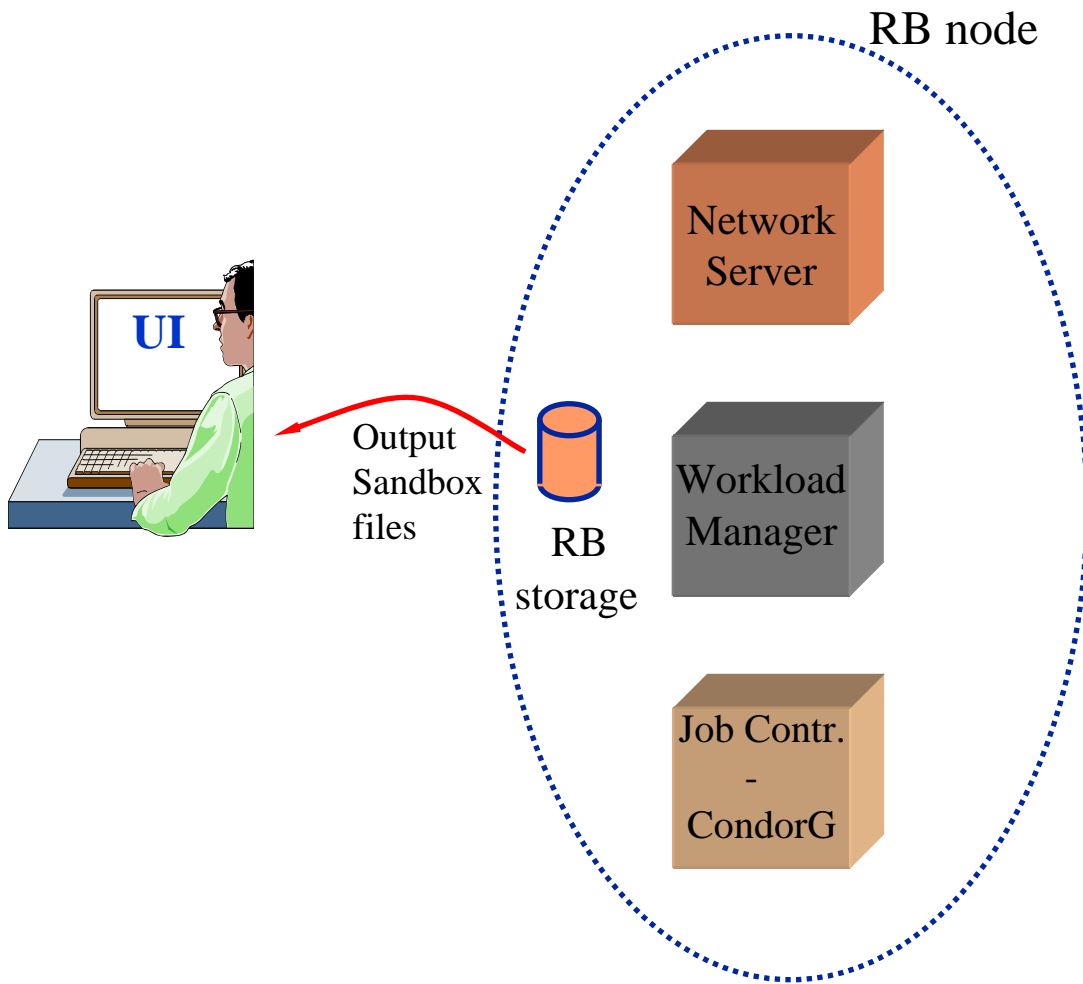
# Job submission





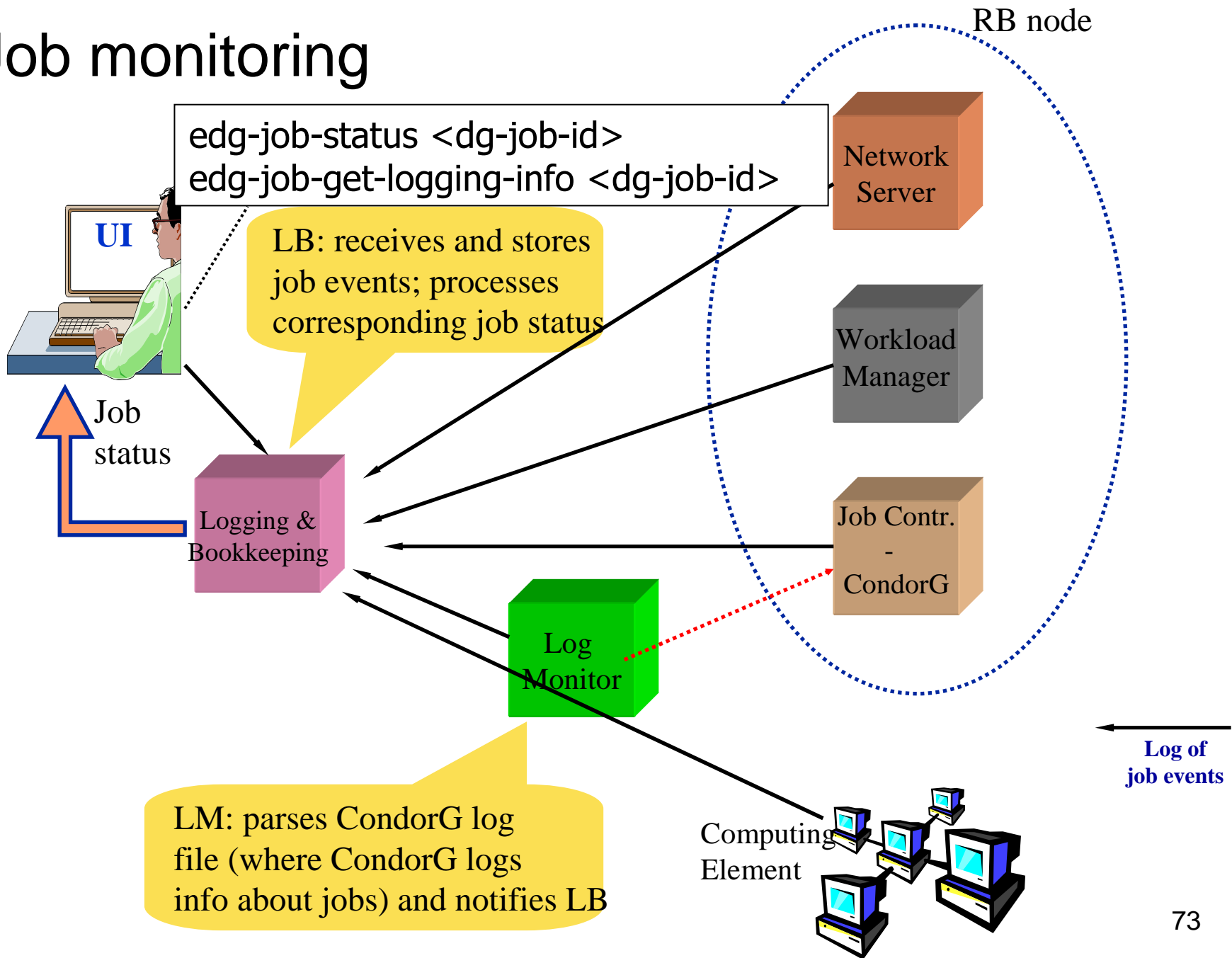






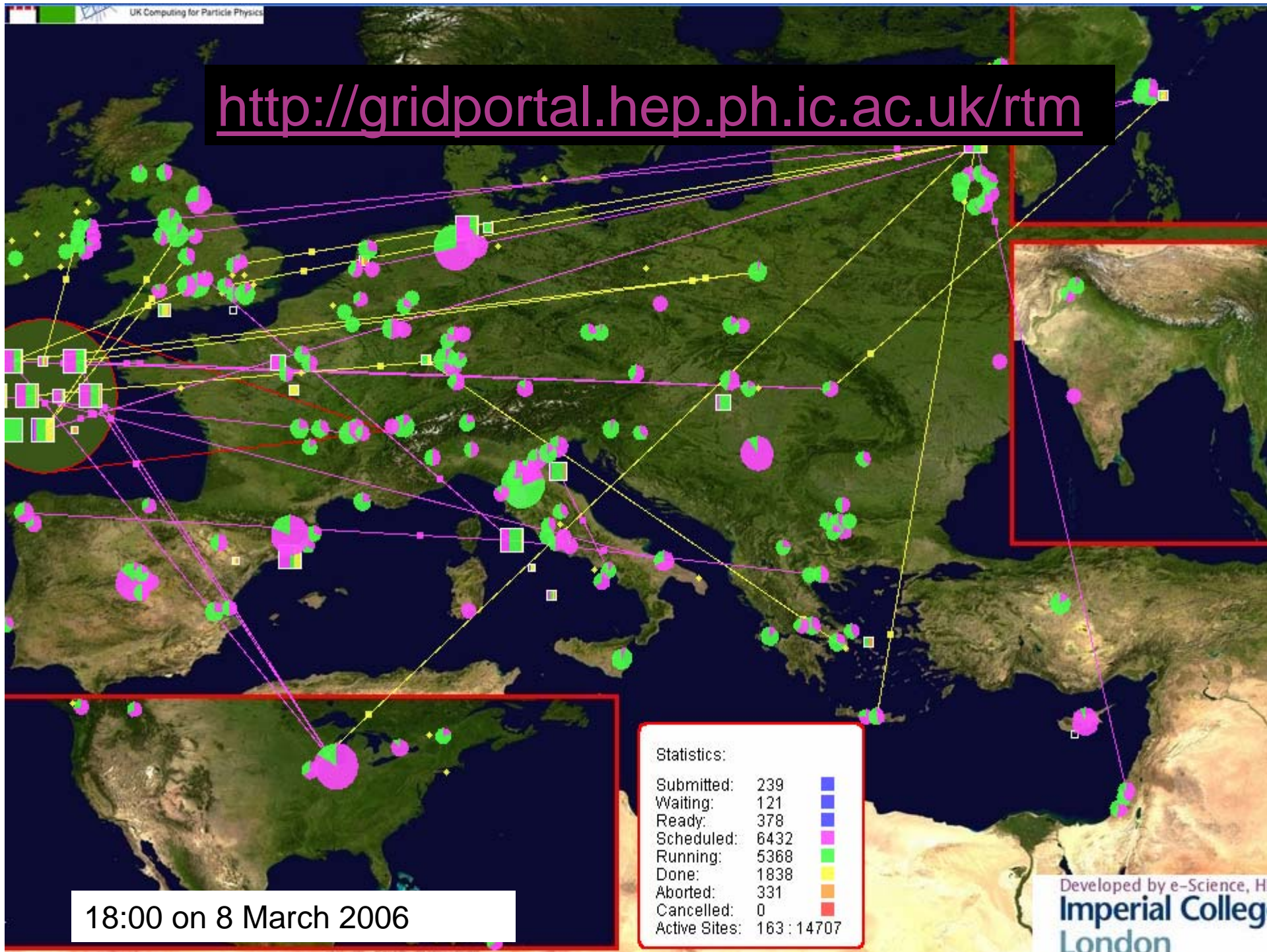


# Job monitoring



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 <i>reason</i>

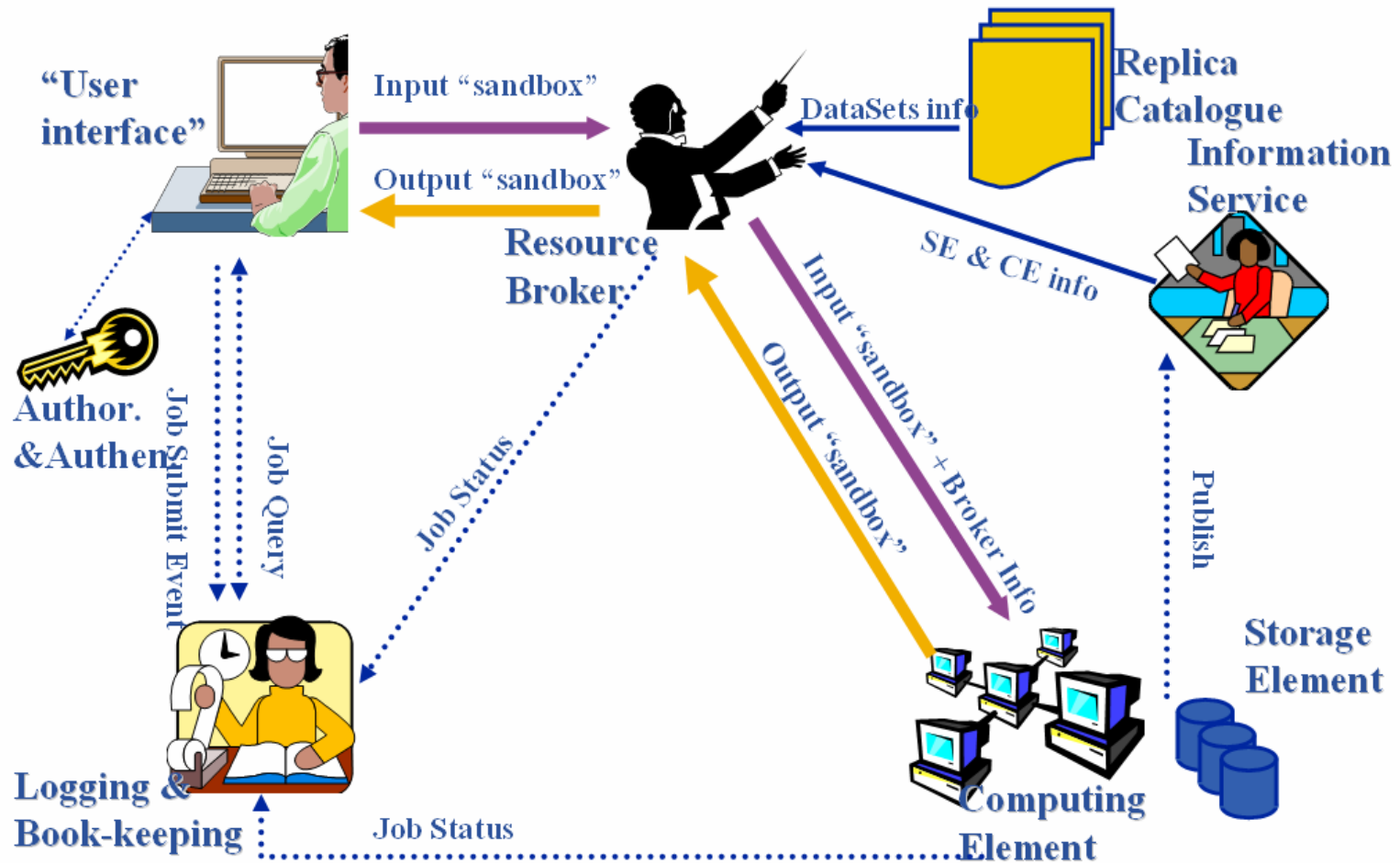
<http://gridportal.hep.ph.ic.ac.uk/rtm>



18:00 on 8 March 2006

- **From the rich grid ecosystem emerged the EGEE production middleware**
  - **Built on tools for**
    - Authorisation and authentication
    - Job submission (direct to a Computing Element)
    - File transfer
  - **...with higher level services**
    - Job submission to “a grid” (via resource broker)
    - Data management
    - Information Systems
  - **..and upon these can be built toolkits and services for new application communities**
    - Workflow
    - Portals ...
- **Authorisation and authentication underpin the middleware**
  - resource-sharing across organisations, without centralised control

- Now its time to begin to use the current production middleware...



- **EGEE** [www.eu-egee.org](http://www.eu-egee.org)
- **EGEE: 1<sup>st</sup> user Forum**  
<http://egee-intranet.web.cern.ch/egee-intranet/User-Forum>
- **LCG** <http://lcg.web.cern.ch/LCG/>
- **LCG User Guide**  
<https://edms.cern.ch/file/454439//LCG-2-UserGuide.pdf>
- **User Scenario**  
<https://edms.cern.ch/file/498081//UserScenario2.pdf>
- **JDL Attributes**  
[http://server11.infn.it/workload-grid/docs/DataGrid-01-TEN-0142-0\\_2.pdf](http://server11.infn.it/workload-grid/docs/DataGrid-01-TEN-0142-0_2.pdf)  
<https://edms.cern.ch/document/590869/1>
- **Global Grid Forum** <http://www.gridforum.org/>
- **Globus Alliance** <http://www.globus.org/>
- **VDT** <http://www.cs.wisc.edu/vdt/>

- VOMS on EGEE: User Guide available at <http://glite.web.cern.ch/glite/documentation/default.asp>
- VOMS
  - Available at <http://infnforge.cnaf.infn.it/voms/>
  - Alfieri, Cecchini, Ciaschini, Spataro, dell'Agnello, Fronher, Lorentey, From gridmap-file to VOMS: managing Authorization in a Grid environment
  - Vincenzo Ciaschini, A VOMS Attribute Certificate Profile for Authorization
- GSI
  - Available at [www.globus.org](http://www.globus.org)
  - A Security Architecture for Computational Grids. I. Foster, C. Kesselman, G. Tsudik, S. Tuecke. *Proc. 5th ACM Conference on Computer and Communications Security Conference*, pp. 83-92, 1998.
  - A National-Scale Authentication Infrastructure. R. Butler, D. Engert, I. Foster, C. Kesselman, S. Tuecke, J. Volmer, V. Welch. *IEEE Computer*, 33(12):60-66, 2000.
- RFC
  - S.Farrell, R.Housley, An internet Attribute Certificate Profile for Authorization, RFC 3281