Middleware components in EGEE

David Fergusson
NeSC Training team
dfmac@nesc.ac.uk

Acknowledgements

This presentation includes slides and information from many sources:

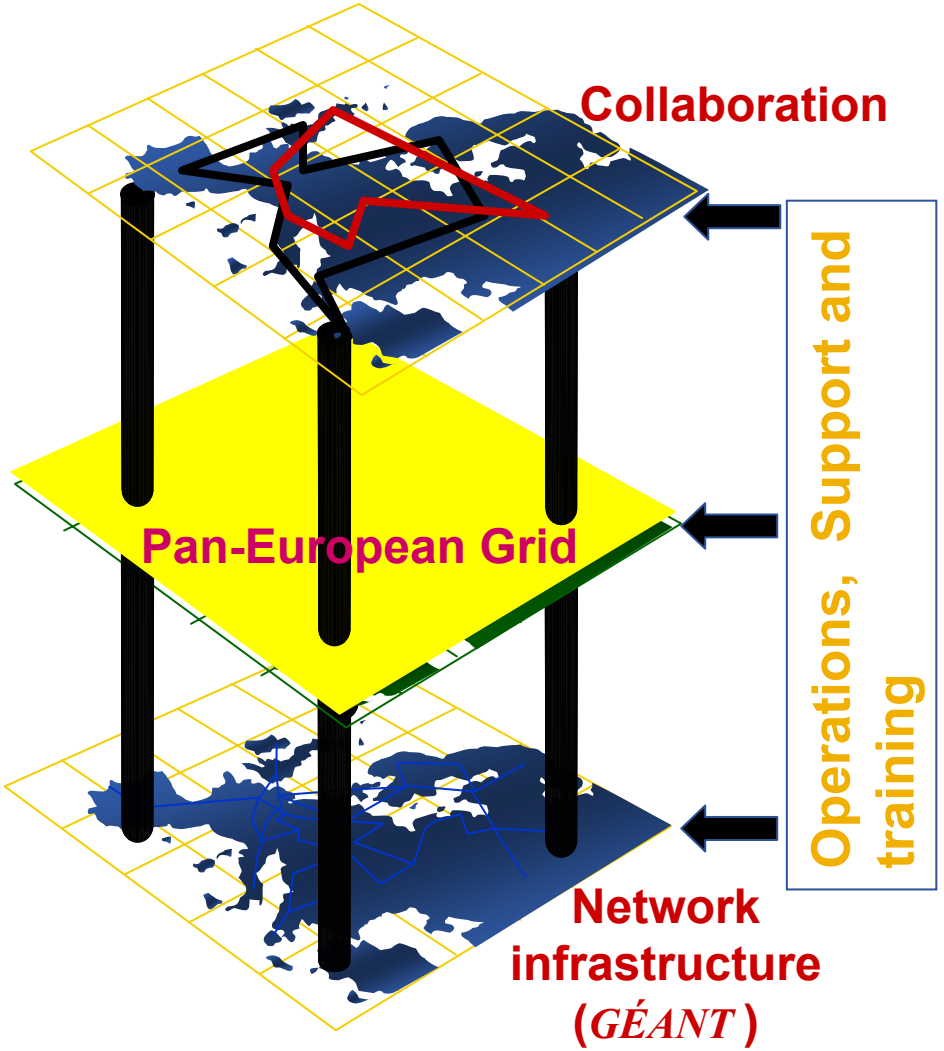
- Roberto Barbera (Slides on middleware are based on presentations given in Edinburgh, April 2004)
- Frederic Hemmer & Erwin Laure (JRA1)
- Other colleagues in EGEE
- The European DataGrid training team
- Authors of the LCG-2 User Guide v. 2.0 : Antonio Delgado Peris, Patricia Méndez Lorenzo, Flavia Donno, Andrea Sciabà, Simone Campana, Roberto Santinelli
  https://edms.cern.ch/file/454439//LCG-2-UserGuide.html
- Additional slides and preparation by Mike Mineter

Outline

- **Overview**
- **Major components**
  - Information services
  - Data management
- **Lifecycle of a job**
- **Summary**

Towards a European e-Infrastructure

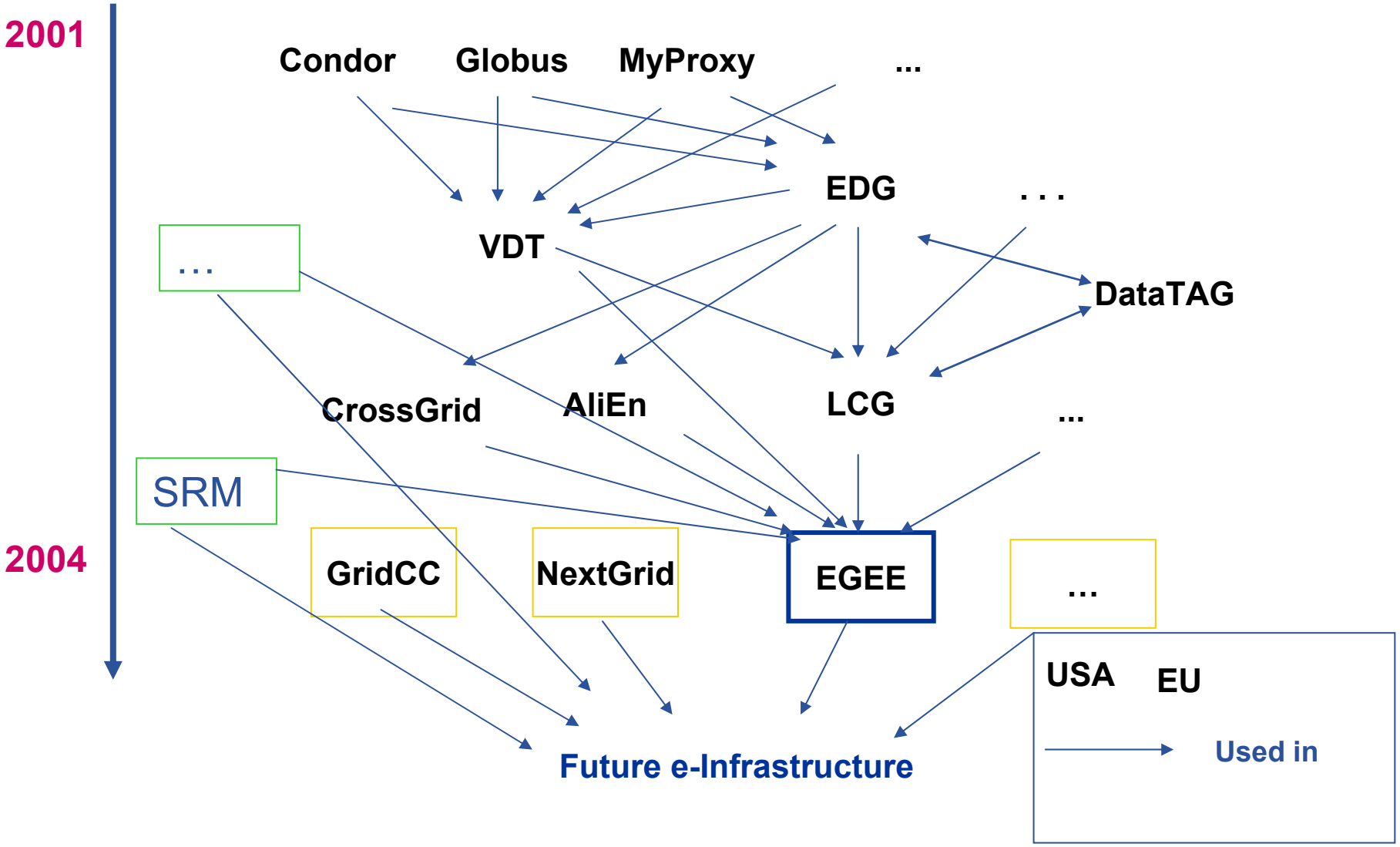
- **To underpin European science and technology in the service of society**
- **To link with and build on**
  - National, regional and international initiatives
  - Emerging technologies (e.g. fibre optic networks)
- **To foster international cooperation**
  - both in the creation and the use of the e-infrastructure



Advantages of grid for biomed

- **Local cluster always faster (speed of light)**
  - Is there always a local cluster? – hospitals
  - Cost of local cluster vs on-demand access
  - Adequate local cluster (capacity)?
- **Clinical collaboration**
- **Future data growth (Clinical decision support)**
  - Individual patient data :
    - Scans
    - Individual genetics
    - Family history
    - Proteomic profiles
- **A cluster is fine for prototyping but not adequate for fully distributed real world solutions**
  - eg. Pharma companies moving to leverage global systems

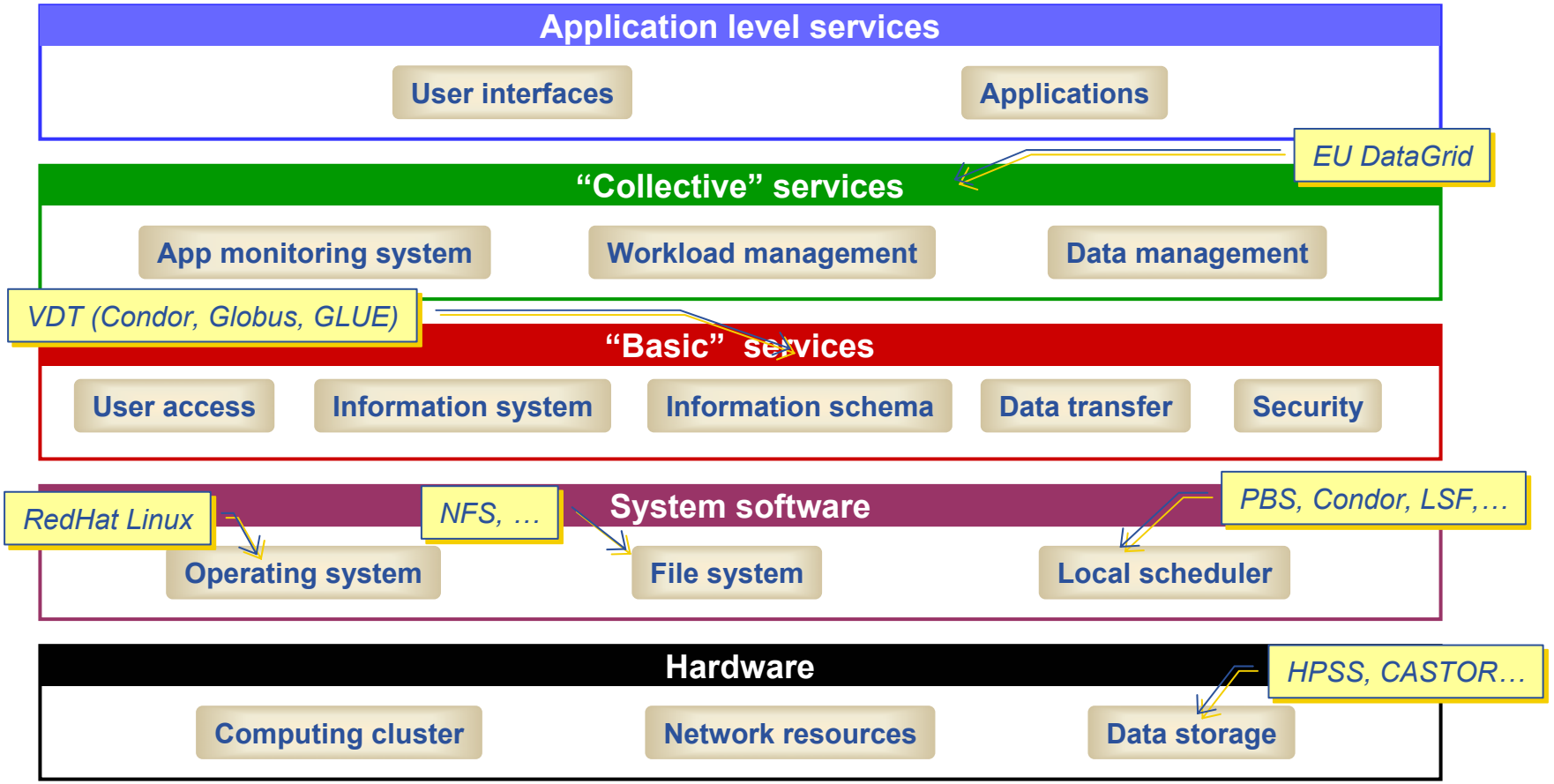
Part of the Grid "ecosystem"



Virtual Data Toolkit

- **http://www.cs.wisc.edu/vdt/**
- **Condor Group**
  - Condor/Condor-G
  - DAGMan
  - Fault Tolerant Shell
  - ClassAds
- **Globus Alliance**
  - Job submission (GRAM)
  - Information service (MDS)
  - Data transfer (GridFTP)
  - Replica Location (RLS)
- **EDG & LCG**
  - Make Gridmap
  - Certificate Revocation List Updater
  - GLUE Schema
- **ISI & UC**
  - Chimera & Pegasus
- **NCSA**
  - MyProxy
  - GSI OpenSSH
  - UberFTP
- **LBL**
  - PyGlobus
  - Netlogger
- **Caltech**
  - MonaLisa
- **VDT**
  - VDT System Profiler
  - Configuration software
- **Others**
  - KX509 (U. Mich.)

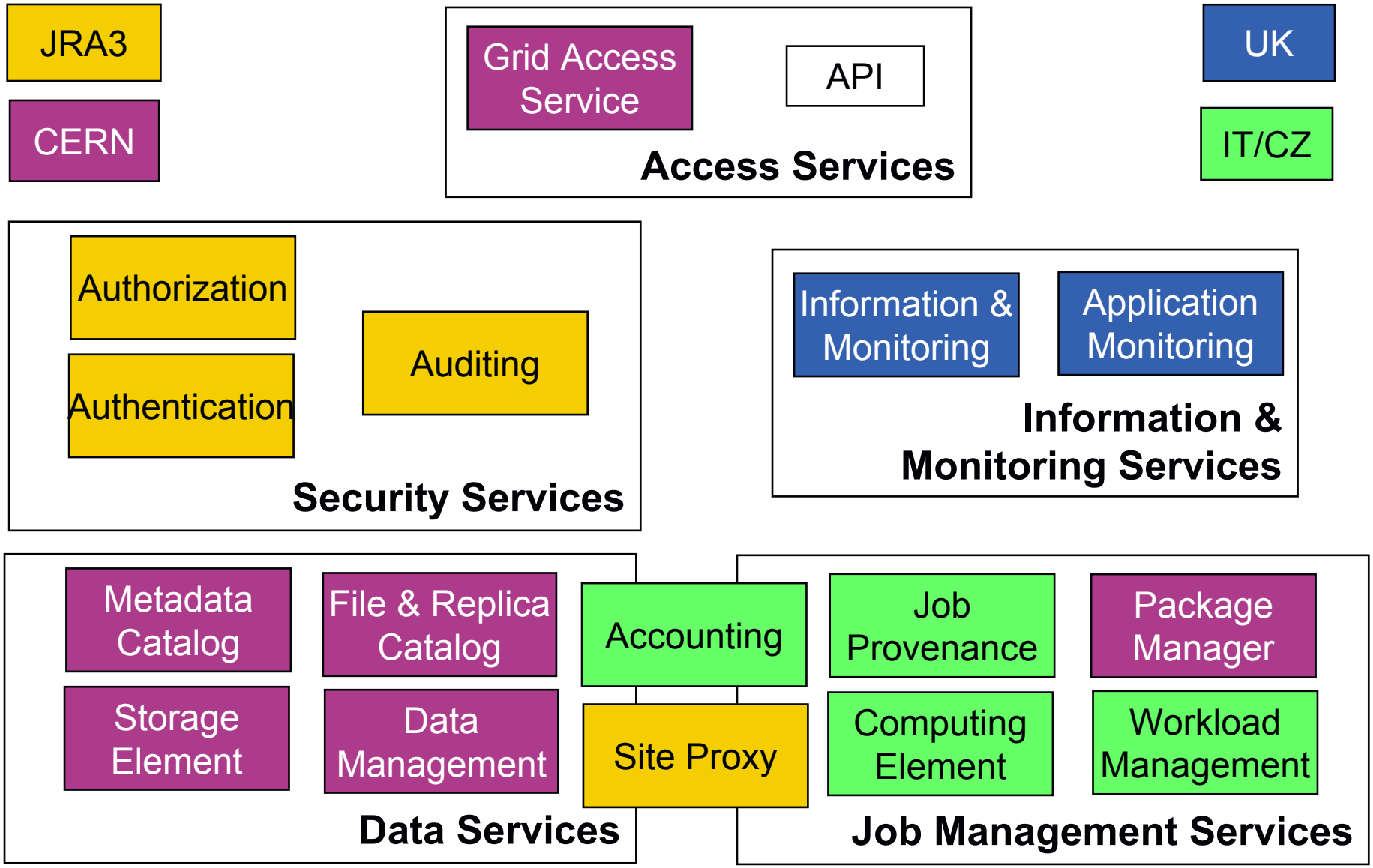
Current production mware: LCG-2



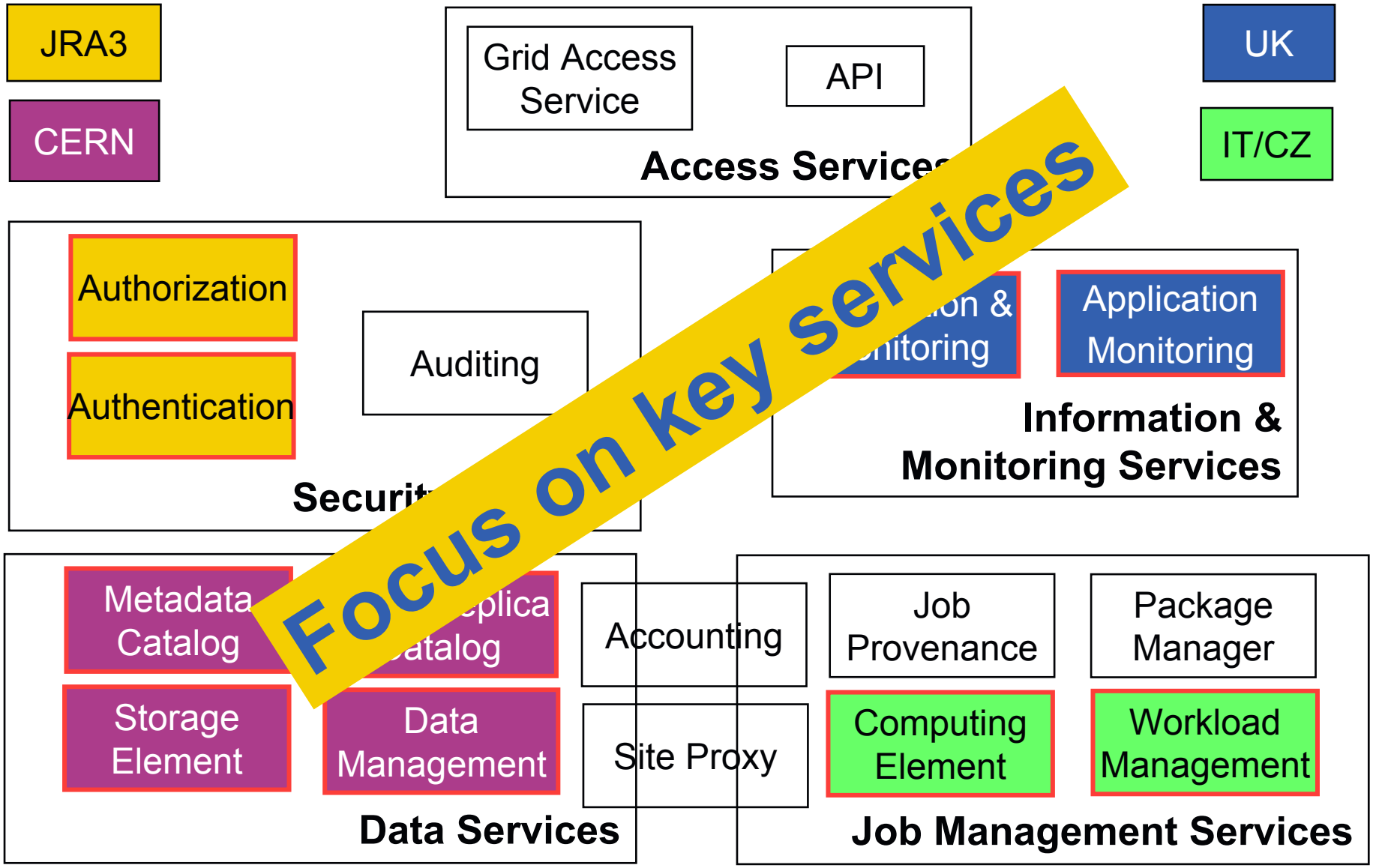
gLite

Web services based service architecture

gLite Services and Responsible Clusters



gLite Services for Release 1



gLite Services for Release 1
Software stack and origin (simplified)

- **Computing Element**
  - Gatekeeper (Globus)
  - Condor-C (Condor)
  - CE Monitor (EGEE)
  - Local batch system (PBS, LSF, Condor)
- **Workload Management**
  - WMS (EDG)
  - Logging and bookkeeping (EDG)
  - Condor-C (Condor)
- **Storage Element**
  - File Transfer/Placement (EGEE)
  - glite-I/O (AliEn)
  - GridFTP (Globus)
  - SRM: Castor (CERN), dCache (FNAL, DESY), other SRMs
- **Catalog**
  - File and Replica Catalog (EGEE)
  - Metadata Catalog (EGEE)
- **Information and Monitoring**
  - R-GMA (EDG)
- **Security**
  - VOMS (DataTAG, EDG)
  - GSI (Globus)
  - Authentication for C and Java based (web) services (EDG)

Main Differences to LCG-2

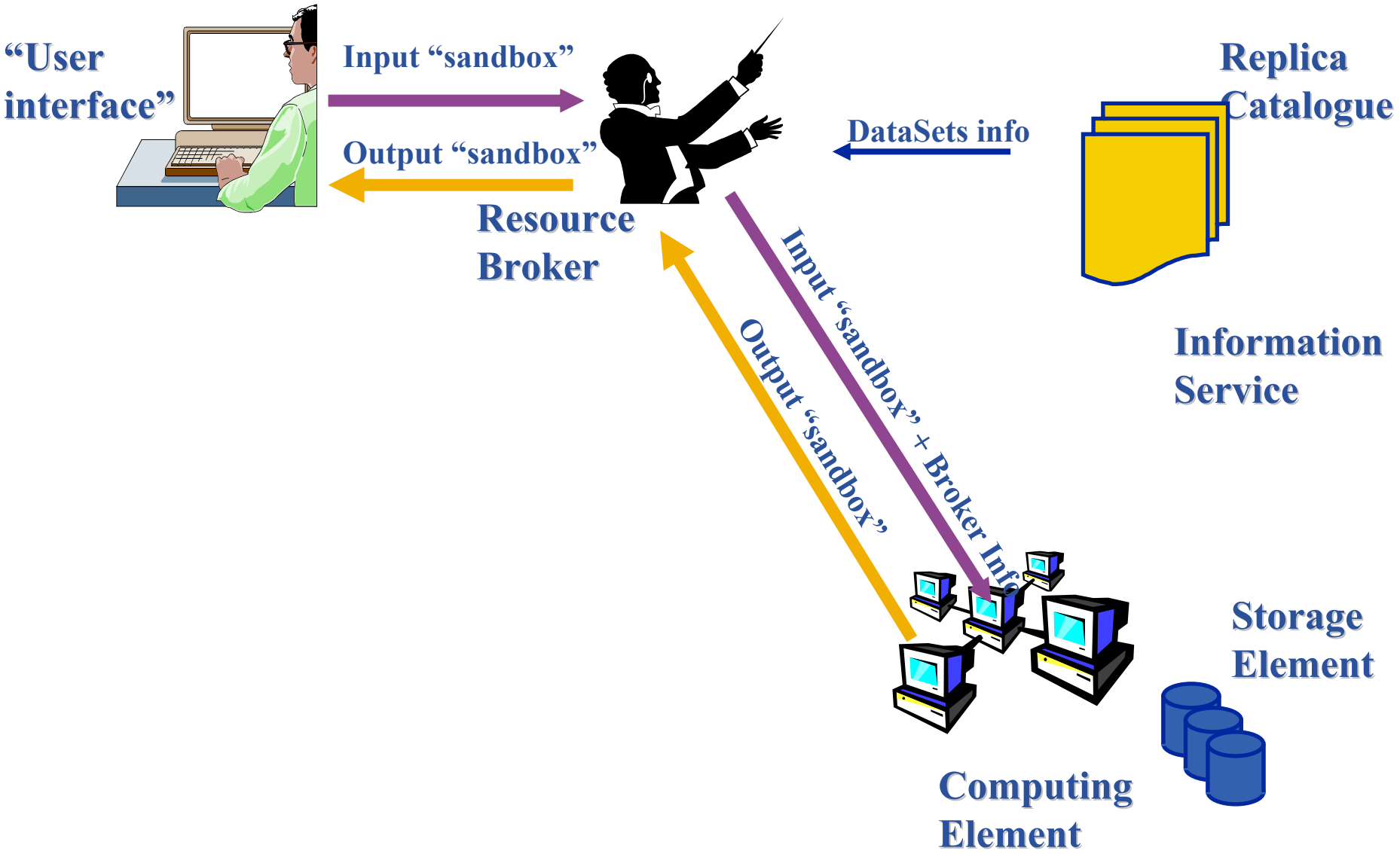
- **Workload Management System works in push and pull mode**
- **Computing Element moving towards a VO based scheduler guarding the jobs of the VO (reduces load on GRAM)**
- **Distributed and re-factored file & replica catalogs**
- **Secure catalogs (based on user DN; VOMS certificates being integrated)**
- **Scheduled data transfers**
- **SRM based storage**
- **Information Services:**
  R-GMA with improved API and registry replication
- **Prototypes of additional services**
  - Grid Access Service (GAS)
  - Package manager
  - DGAS based accounting system
  - Job provenance service
- **Move towards Web Services**

gLite
Lightweight Middleware for Grid Computing

Outline

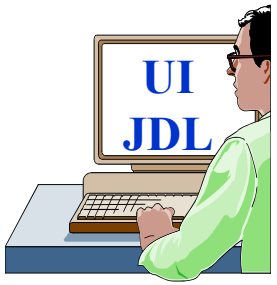
- **Overview**
- Major components
  - Information services
  - Data management
- **Lifecycle of a job**
- **Summary**

Major components



User Interface node

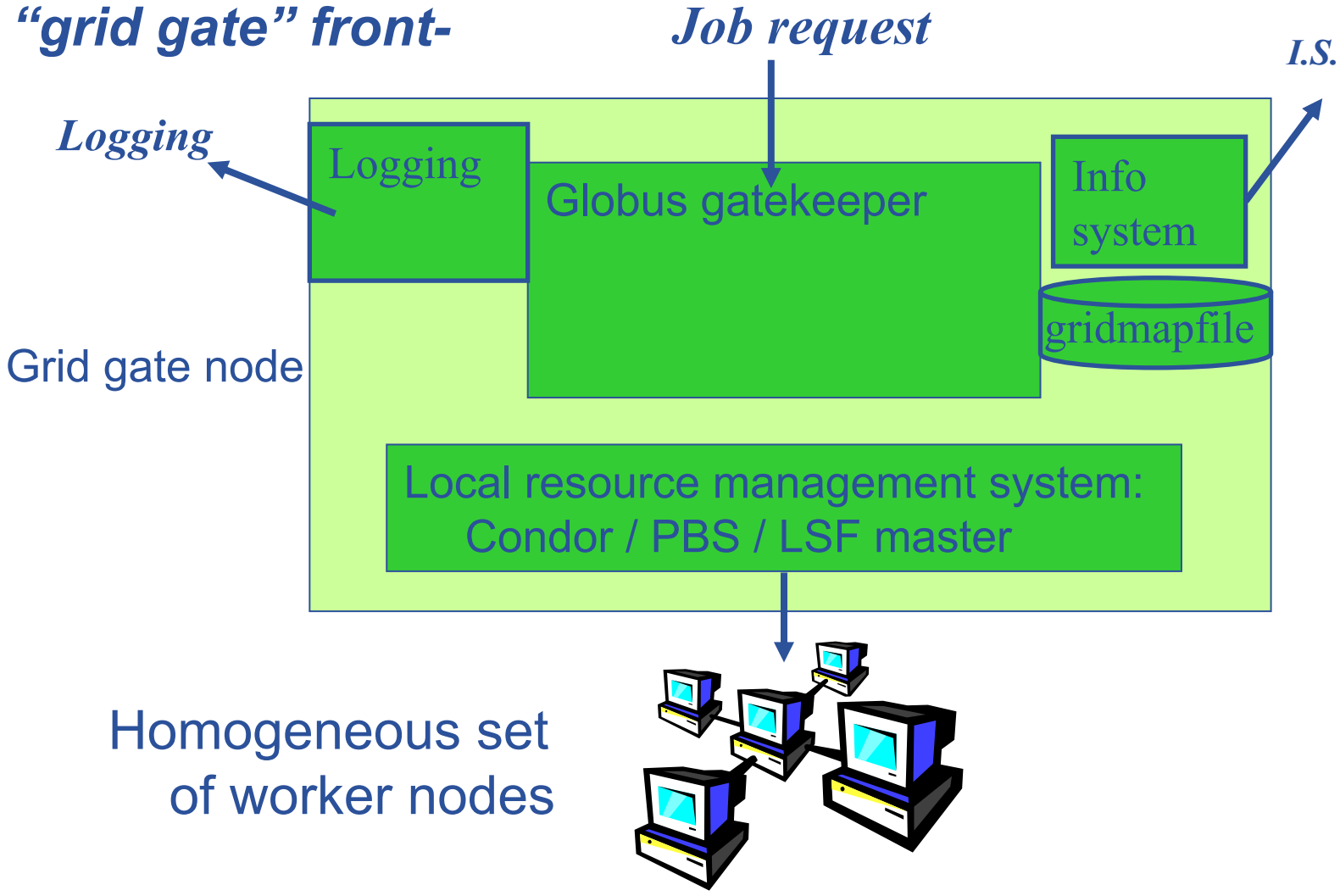
- **The user's interface to the Grid**
- **Command-line interface to**
  - Proxy server
  - Job operations
    - To submit a job
    - Monitor its status
    - Retrieve output
  - Data operations
    - Upload file to SE
    - Access file
    - …
  - Other grid services
- **Also C++ and Java APIs**



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

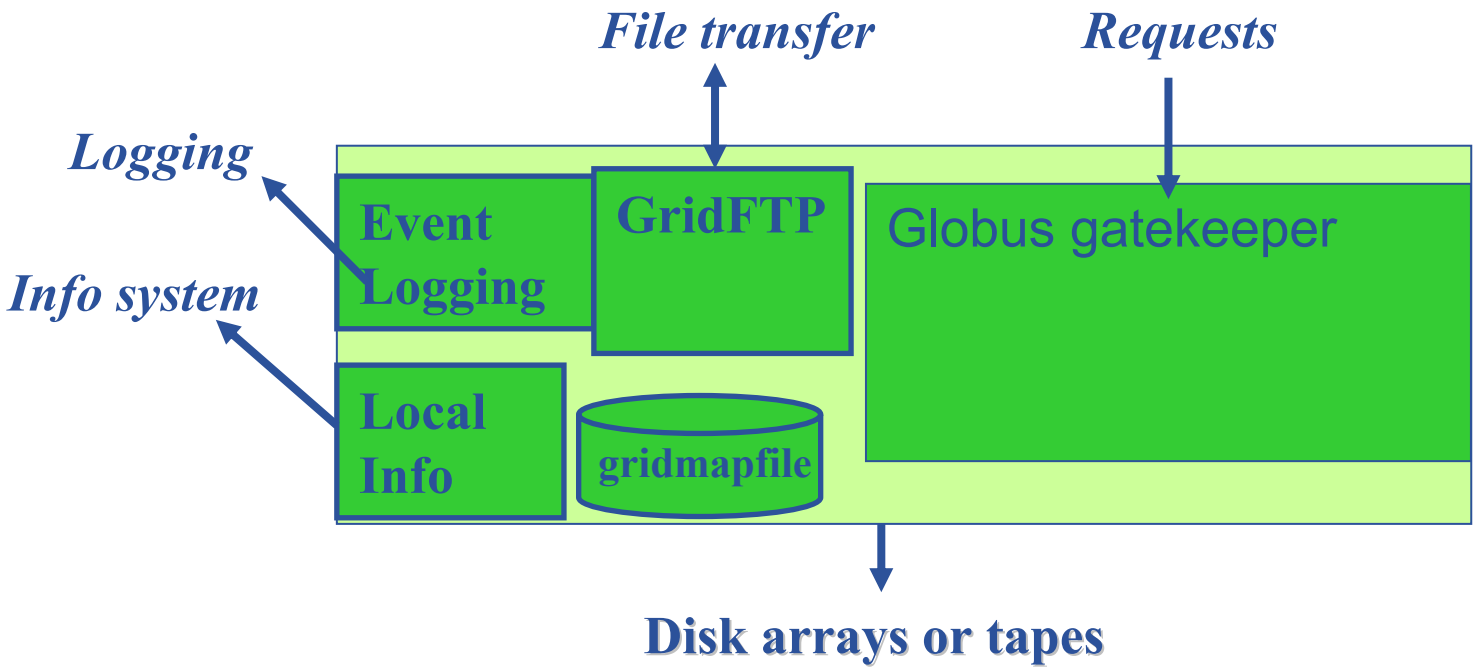
"Compute element"

A CE is a grid batch queue with a "grid gate" front-end:



Storage elements and files

- **Storage elements hold files: write once, read many**



Workload Management System (WMS)

- **Distributed scheduling**
  - multiple UI's where you submit your job
  - multiple RB's from where the job is sent to a CE
  - multiple CE's where the job can be put in a queuing system
- **Distributed resource management**
  - multiple information systems that monitor the state of the grid
  - Information from SE, CE, sites

Resource Broker nodes

- **Run the Workload Management System**
  - To accept job submissions
  - Dispatch jobs to appropriate Compute Element (CE)
  - Allow users
    - To get information about their status
    - To retrieve their output
- **A configuration file on each UI node determines which RB node(s) will be used**
- **When a user submits a job, JDL options are to:**
  - Specify CE
  - Allow RB to choose CE (using optional tags to define requirements)
  - Specify SE (then RB finds “nearest” appropriate CE, after interrogating Replica Location Service)

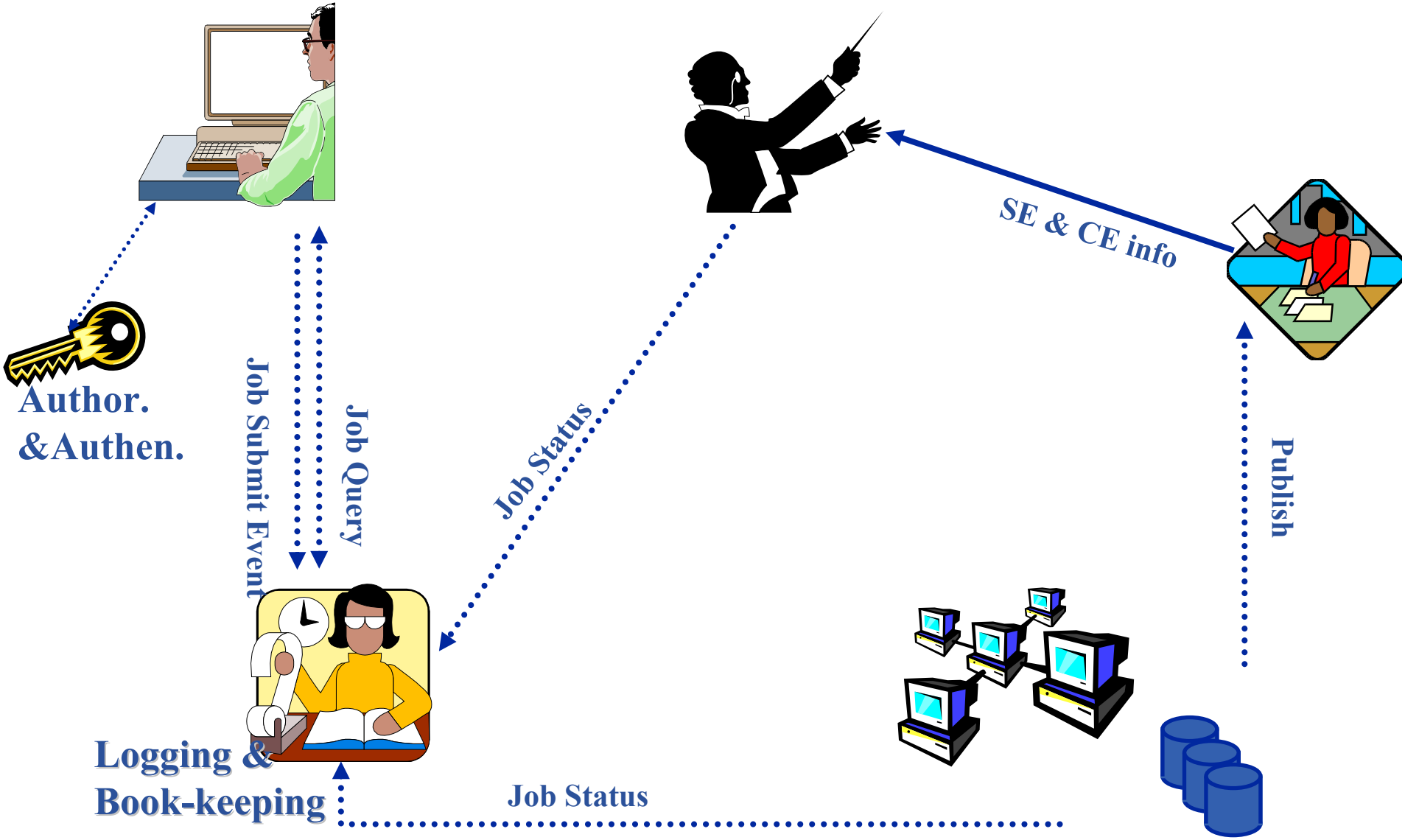
Outline

- Overview
- Major components
  - Information services
  - Data management
- Lifecycle of a job
- Summary

Logging and Book-keeping

- Who did what when??
- What's happening to my job?
- Usually runs on Resource Broker node
- See LCG-2 user guide for a bit more on this

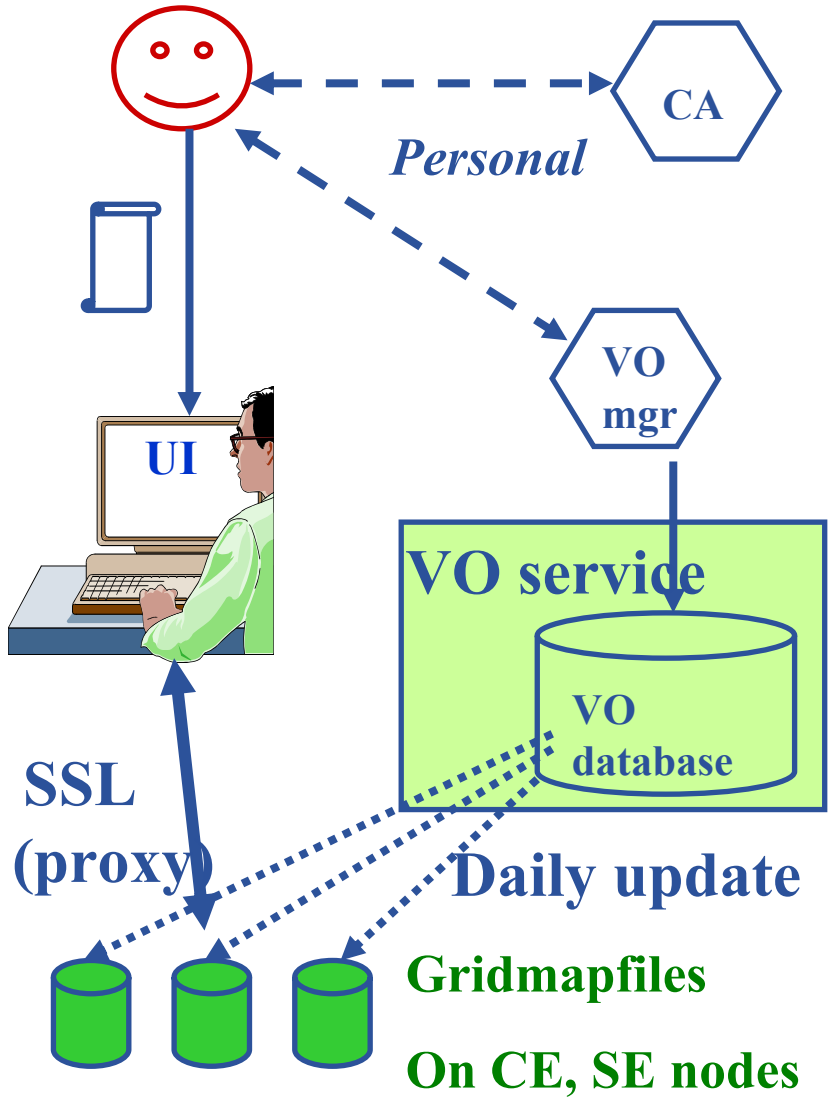
Accounting


SE & CE info
Author.
&Authen.
Job Submit Event
Job Query
Job Status
Publish
Logging &
Book-keeping
Job Status

Authentication, Authorisation

- **Authentication**
  - User obtains certificate from CA
  - Connects to UI by ssh
  - Downloads certificate
  - Invokes Proxy server
  - **Single logon** – to UI - then **Secure Socket Layer with proxy identifies user to other nodes**
- **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



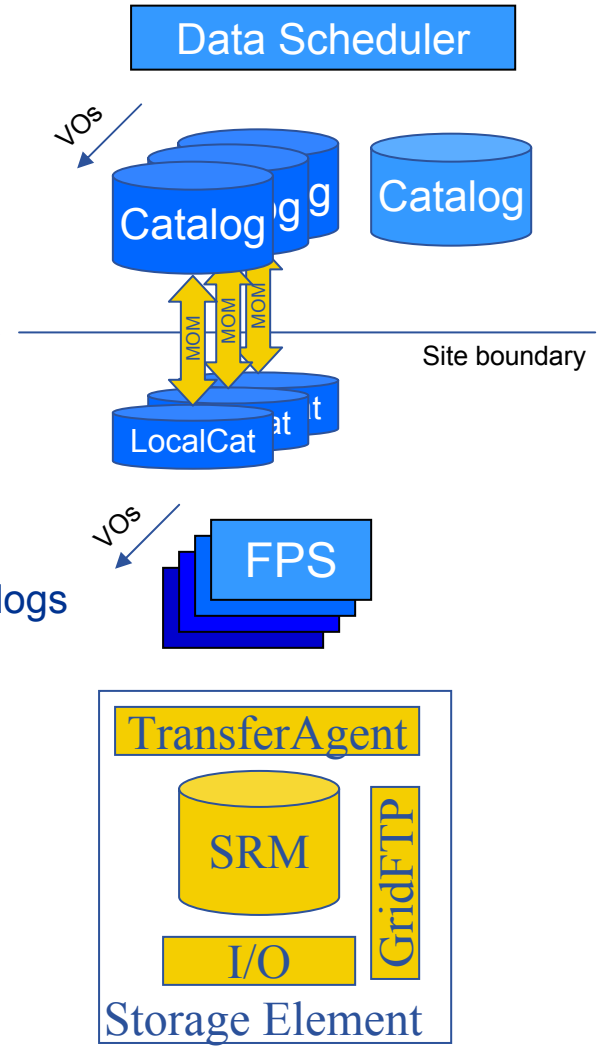
Outline

- **Overview**
- **Major components**
  - Information services
  - **Data management**
- **Lifecycle of a job**
- **Summary**

Data Management Services

- **Efficient and reliable data storage, movement, and retrieval on the infrastructure**

- **Storage Element**
  - Reliable file storage (SRM based storage systems)
  - Posix-like file access (gLite I/O)
  - Transfer (gridFTP)
- **File and Replica Catalog**
  - Resolves logical filenames (LFN) to physical location of files (URL understood by SRM) and storage elements
  - Hierarchical File system like view in LFN space
  - Single catalog or distributed catalog (*under development*) deployment possibilities
- **File Transfer and Placement Service**
  - Reliable file transfer and transactional interactions with catalogs
- **Data Scheduler**
  - Scheduled data transfer in the same spirit as jobs are being scheduled taking into account e.g. network characteristics (collaboration with JRA4)
  - *Under development*
- **Metadata Catalog**
  - Limited metadata can be attached to the File and Replica Catalog
  - Interface to application specific catalogs have been defined

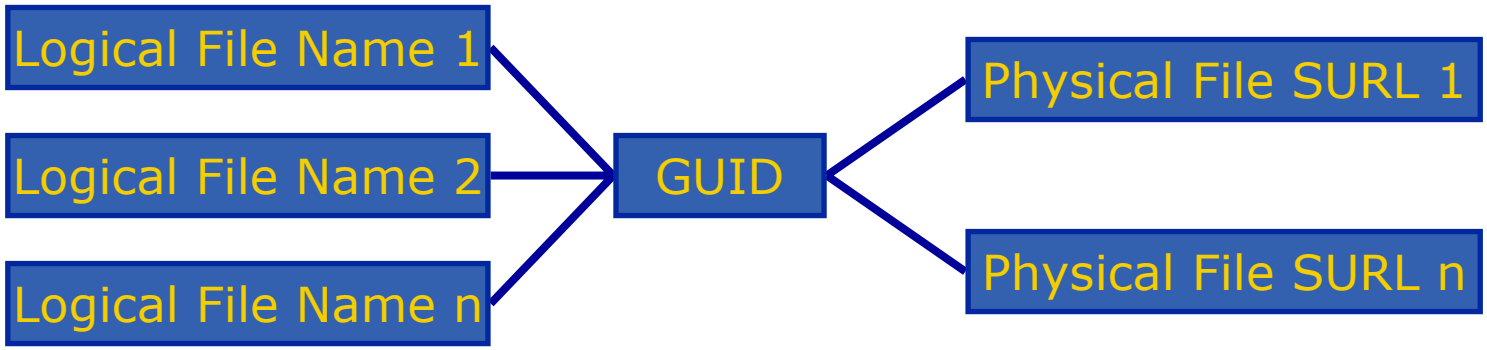


Data management

- **User data: generally file-oriented (some RDBMS exceptions exist)**
- **Small files: On UI; passed to/from CE via sandbox**
- **Large files: require SE**
  - Replica files on different SEs
    - Fault tolerance
    - Performance:
      - *run job on CE "close" to data*
      - *share load on SE*
  - Replica Catalogue - what replicas exist for a file?
  - Replica Location Service - where are they?

Naming Conventions

- Logical File Name (LFN)
  - An alias created by a user to refer to some item of data e.g. “lfn:cms/20030203/run2/track1”
- Site URL (SURL) (or Physical File Name (PFN))
  - The location of an actual piece of data on a storage system e.g. “srm://pcrd24.cern.ch/flatfiles/cms/output10_1”
- Globally Unique Identifier (GUID)
  - A non-human readable unique identifier for an item of data e.g. “guid:f81d4fae-7dec-11d0-a765-00a0c91e6bf6”

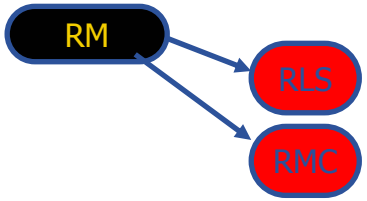


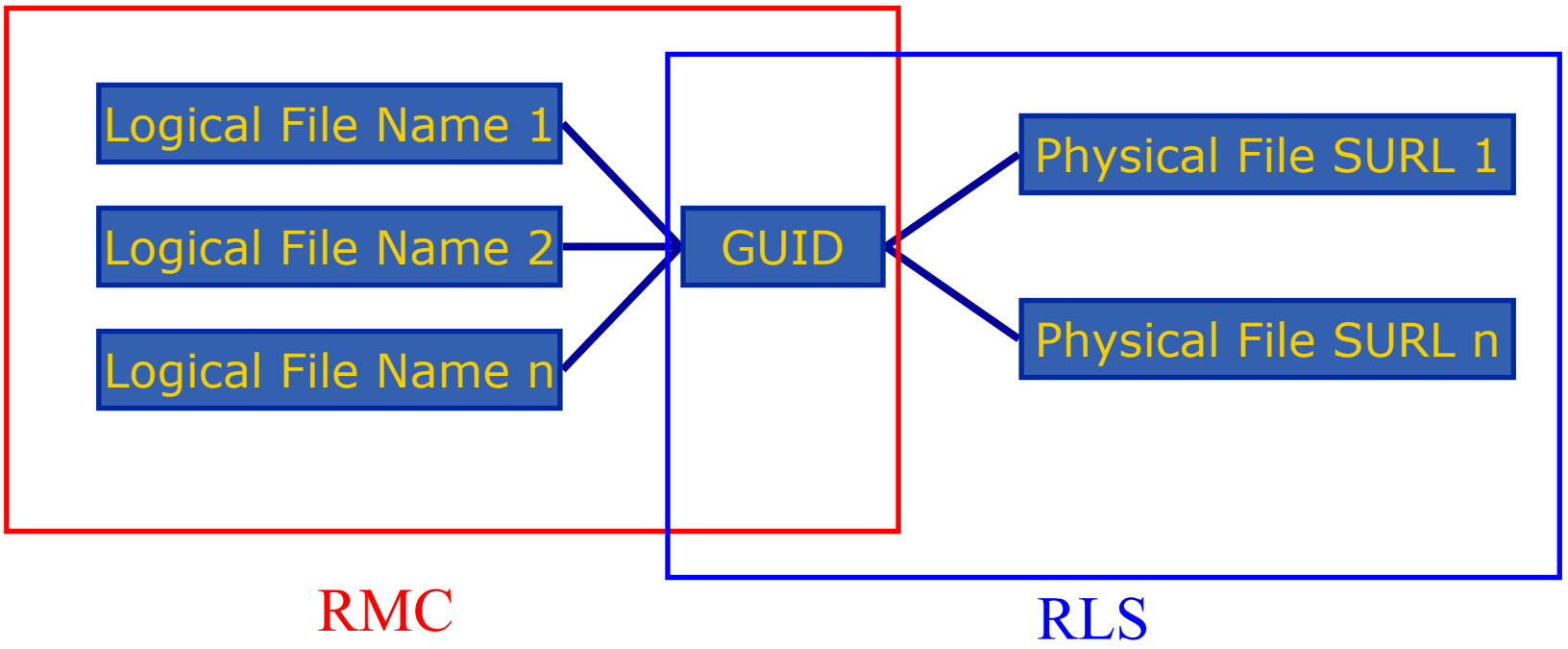
File Catalogs in LCG

- **File catalogs in LCG:**
  - They keep track of the location of copies (replicas) of Grid files
  - The DM tools and APIs and the WMS interact with them
- **EDG's Replica Location Service (RLS)**
  - Catalogs in use in LCG-2
  - Replica Metadata Catalog (RMC) + Local Replica Catalog (LRC)
  - Some performance problems detected during Data Challenges
- **New LCG File Catalog (LFC)**
  - In production in next LCG release; deployment in January 2005
  - Coexistence with RLS; migration tools provided
  - Accessible by defining: $LCG_CATALOG_TYPE=lfc and $LFC_HOST
  - Better performance and scalability
  - Provides new features: security, hierarchical namespace, transactions...

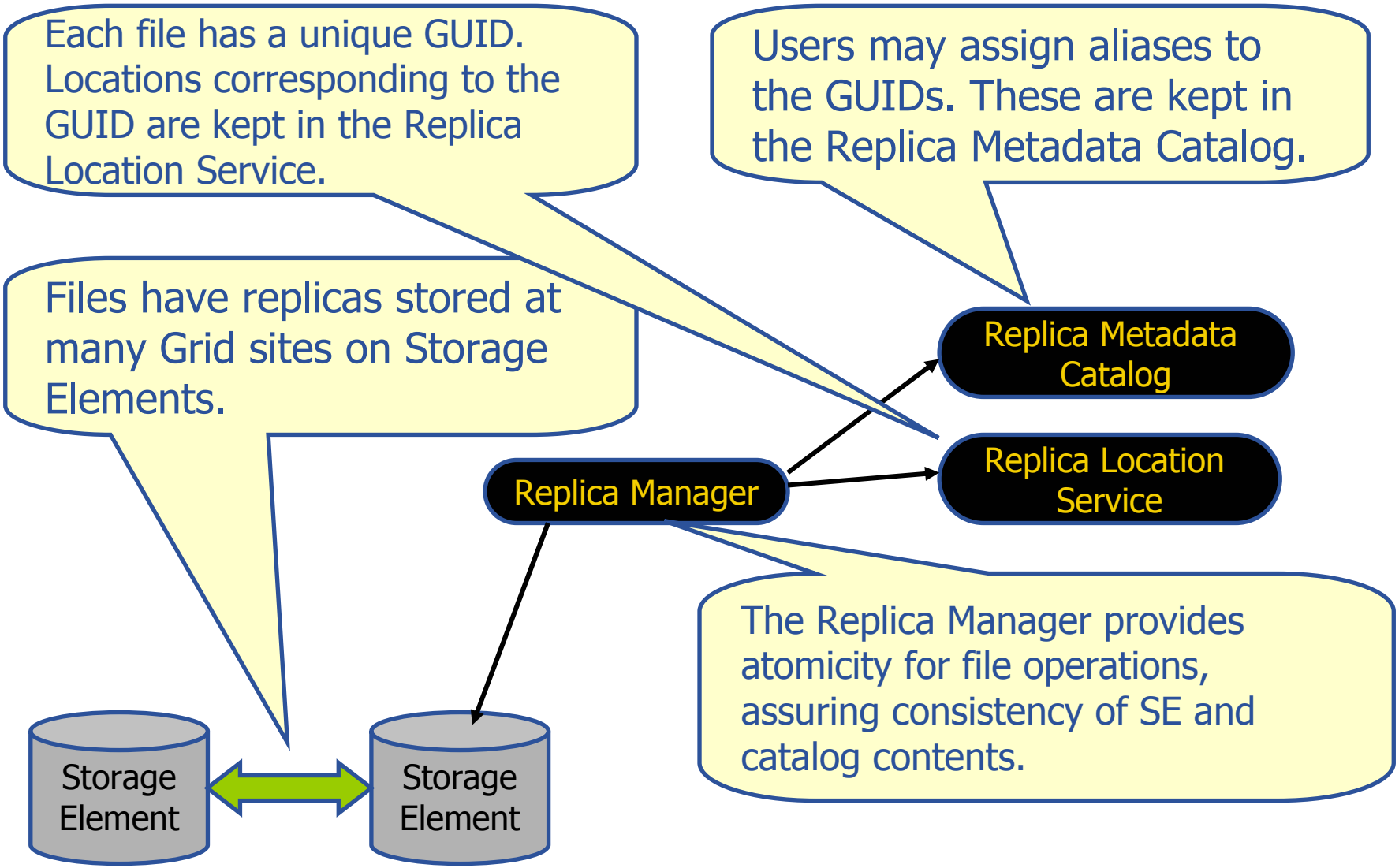
Replica Metadata Catalog (RMC) Replica Location Service (RLS)

- **RMC:**
  - Stores LFN-GUID mappings
- **RLS:**
  - Stores GUID-SURL mappings



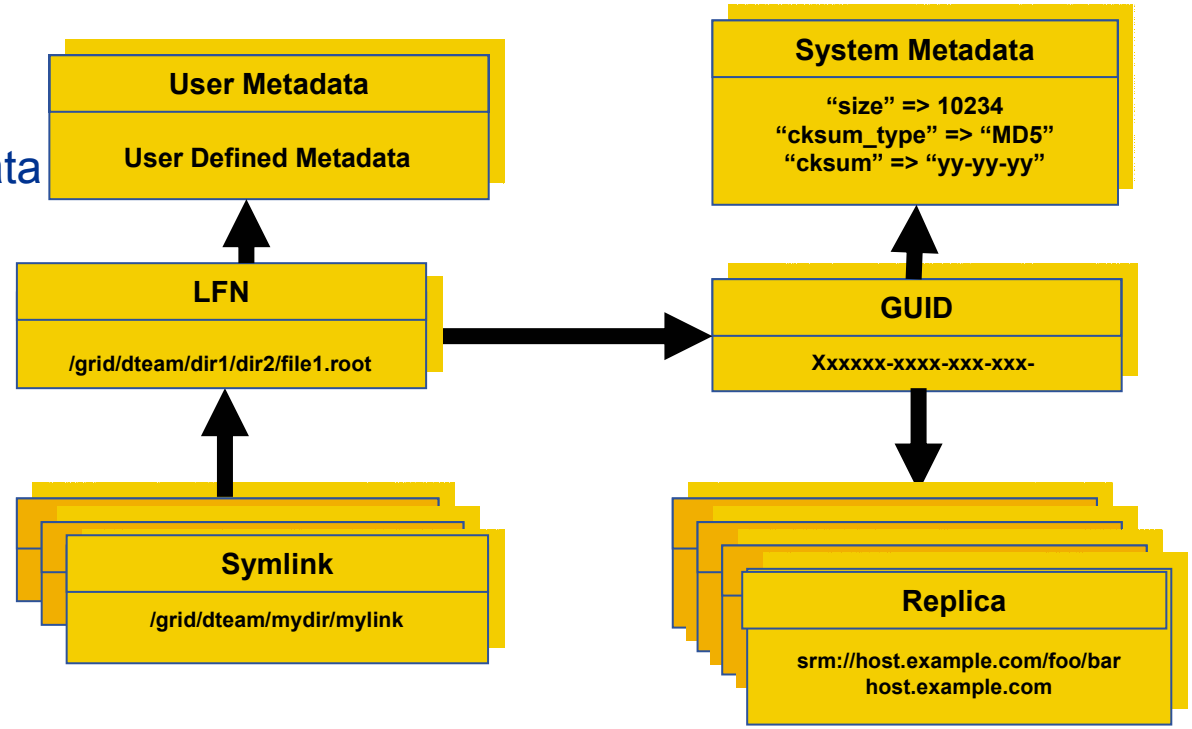


Data Replication Services: Basic Functionality



File Catalogs: The LFC

- **One single catalog**
- **LFN acts as main key in the database. It has:**
  - Symbolic links to it (additional LFNs)
  - Unique Identifier (GUID)
  - System metadata
  - Information on replicas
  - One field of user metadata



File Catalogs: The LFC (II)

- **Fixes performance and scalability problems seen in EDG Catalogs**
  - Cursors for large queries
  - Timeouts and retries from the client
- **Provides more features than the EDG Catalogs**
  - User exposed transaction API (+ auto rollback on failure)
  - Hierarchical namespace and namespace operations (for LFNs)
  - Integrated GSI Authentication + Authorization
  - Access Control Lists (Unix Permissions and POSIX ACLs)
  - Checksums
- **Interaction with other components**
  - Supports Oracle and MySQL database backends
  - Integration with GFAL and lcg_util APIs complete
  - New specific API provided
- **New features will be added (requests welcome!)**
  - ROOT Integration in progress
  - POOL Integration will be provided soon
  - VOMS will be integrated

Outline

- **Overview**
- **Major components**
  - Information services
  - Data management
- Lifecycle of a job
- **Summary**

Job Management Services

- **Efficient and reliable scheduling of computational tasks on the available infrastructure**
- **Started with LCG-2 Workload Management System (WMS)**
  - Inherited from EDG
  - Support partitioned jobs and jobs with dependencies
  - Support for different replica catalogs for data based scheduling
  - Modification of internal structure of WMS
    - Task queue: queue of pending submission requests
    - Information supermarket: repository of information on resources
    - Better reliability, better performance, better interoperability, support push and pull mode
  - *Under development*
    - Web Services interface supporting bulk submission (after V1.0)
      - *Bulk submission supported now by use of DAGs*
    - Distributed superscheduling (interaction among multiple WMSs)

Job Management Services II

- **Computing Element (CE)**
  - Service representing a computing resource
    - CE moving towards a VO based local scheduler
    - Incorporates new technologies provided by Condor and Globus
  - Web service interfaces
    - job management requests (run, cancel, suspend, resume, …)
      - *Still under development*
    - Policy based notifications on changes of the CE (pull model)

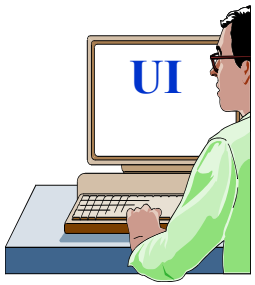
- **Job Provenance**
  - Keeps track of definition of submitted jobs, execution conditions and job life cycle for a long time *under development*

- **Grid Accounting (DGAS)**
  - Accumulates Grid accounting information about the usage of Grid resources by users / groups (e.g. VOs) for billing and scheduling policies. *Under development*

- **VOMS**
  - Virtual Organization Membership Service

- **Advanced Reservation service *under development***

- **Assured backward compatibility (to facilitate migration from current SA1 infrastructure)**



RB node

Network Server

Workload Manager

Job Contr.

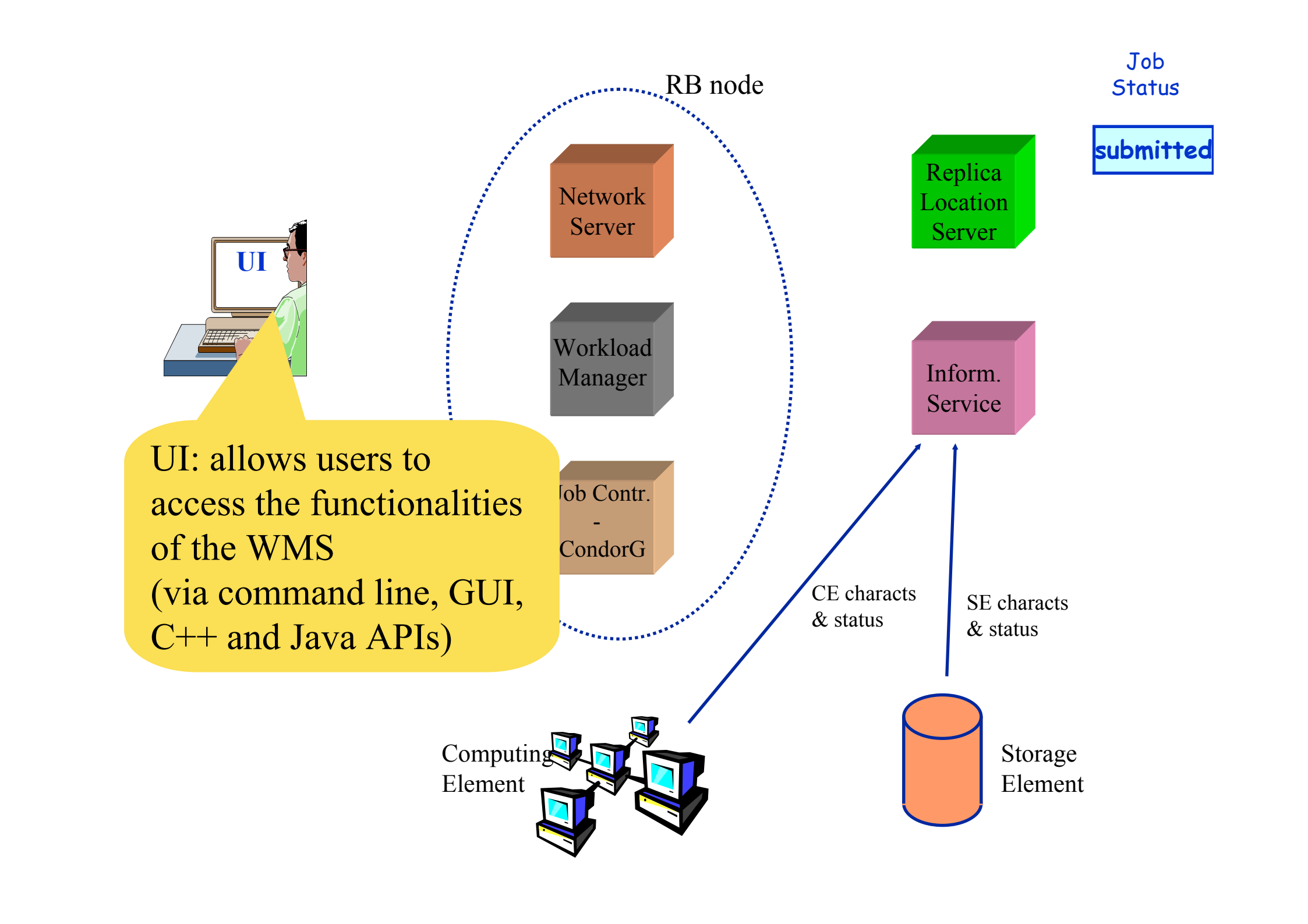
Replica Location Server

Inform. Service

Characts. & status

Computing Element

Storage Element


RB node
Job Status
submitted
Network Server
Replica Location Server
Workload Manager
Inform. Service
Job Contr. - CondorG
UI
UI: allows users to access the functionalities of the WMS (via command line, GUI, C++ and Java APIs)
CE characts & status
SE characts & status
Computing Element
Storage Element

edg-job-submit myjob.jdl

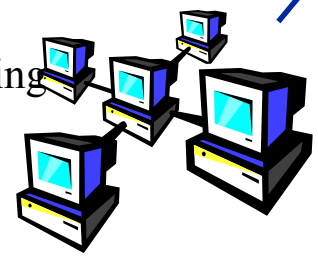
Myjob.jdl

```
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

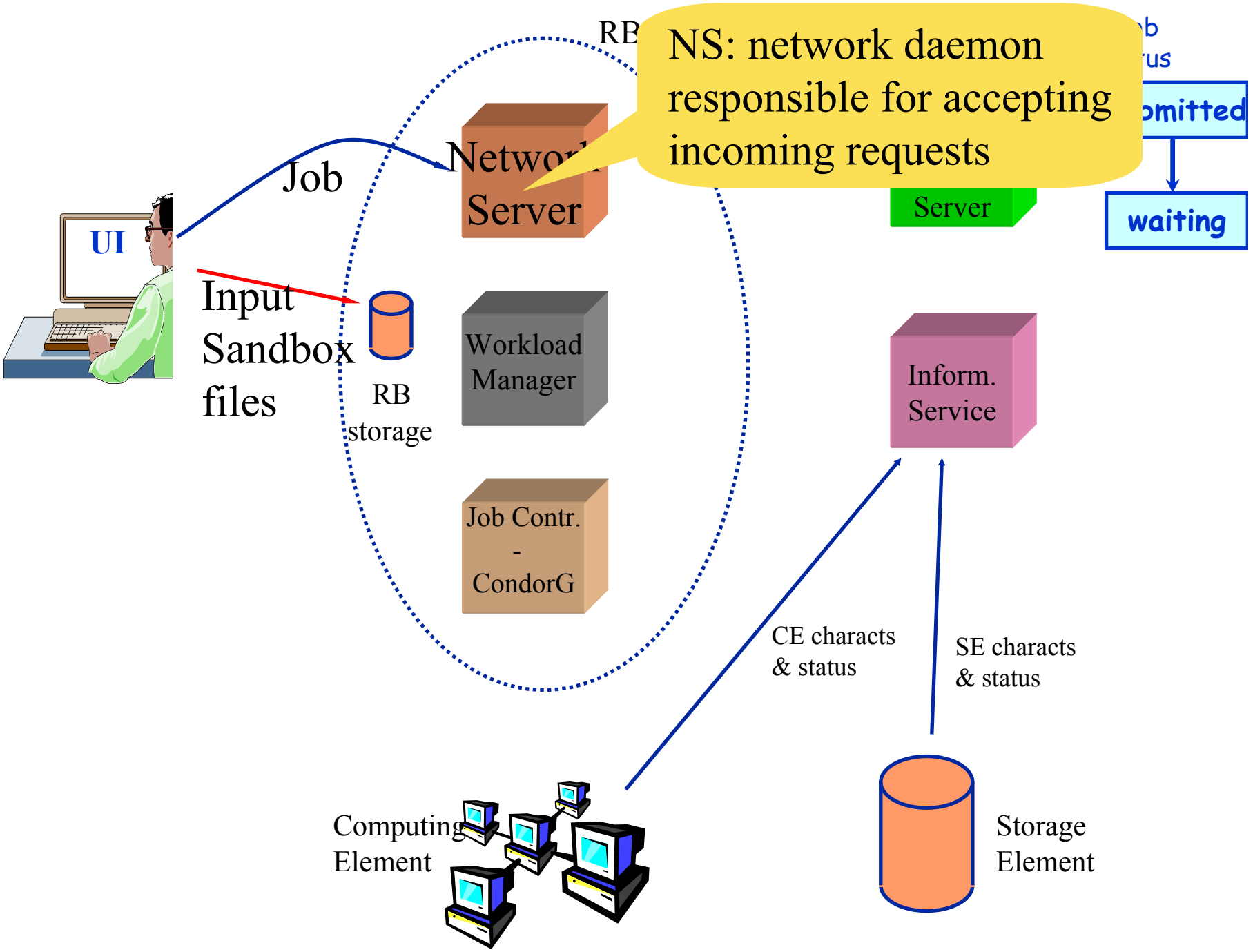
submitted

SE characts & status

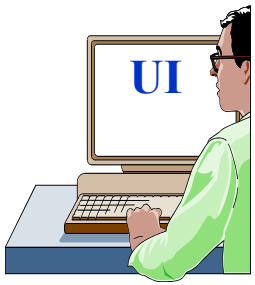


Computing Element

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


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

Job submission



RB node

Network Server

Job

RB storage

Workload manager

Job Contr. - CondorG

WM: acts to satisfy the request

Replica Location Server

Job Status

submitted

waiting

Inform. Service

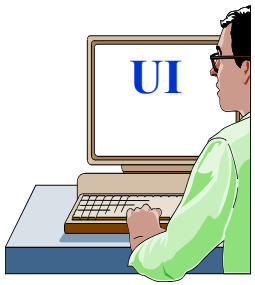
CE characts & status

SE characts & status

Computing Element

Storage Element

Job submission



RB node

Network Server

Match-Maker/ Broker

Replica Location Server

RB storage

Workload Manager

Where must this job be executed ?

Job Contr. - CondorG

CE characts & status

SE characts & status

Computing Element

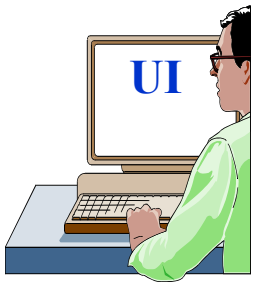
Storage Element

Job Status

submitted

waiting

Job submission



Where are (which SEs) the needed data ?

Network Server

Replica Location Server

Match-Maker/ Broker

RB storage

Workload Manager

Inform. Service

Job Cc - Cond

What is the status of the Grid ?

E characts & status

SE characts & status

Computing Element

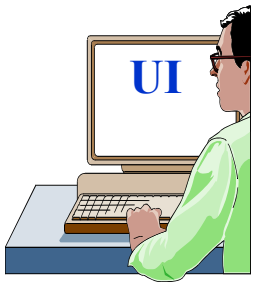
Storage Element

Job Status

submitted

waiting

Job submission



RB node

Network Server

Match-Maker/ Broker

Replica Location Server

Job Status

submitted

waiting

RB storage

Workload Manager

CE choice

Inform. Service

Job Contr. - CondorG

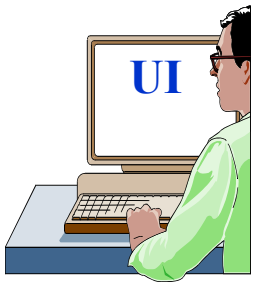
CE characts & status

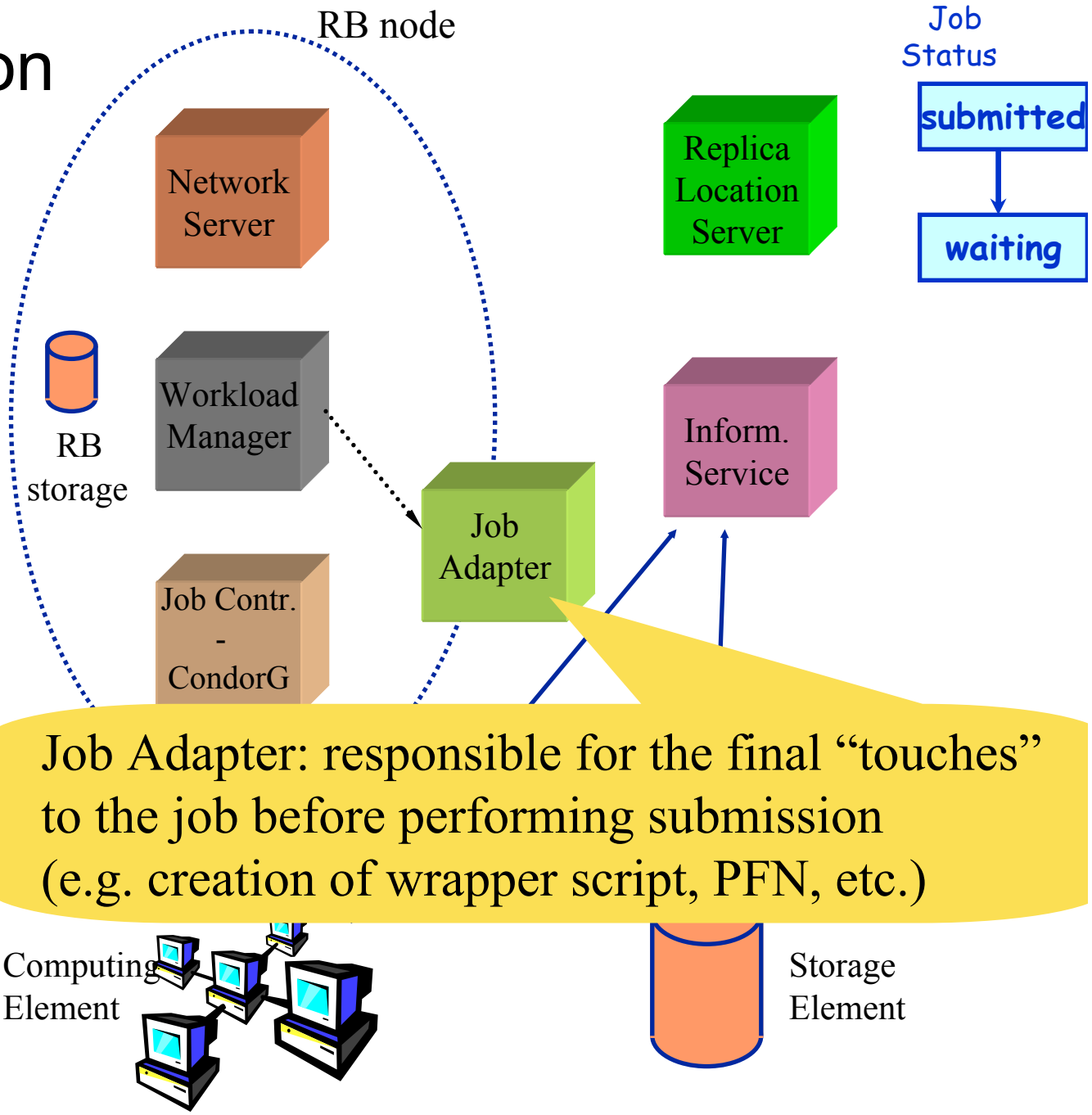
SE characts & status

Computing Element

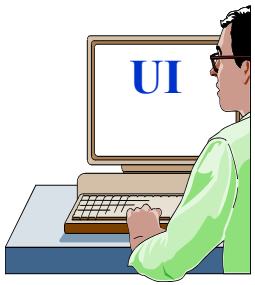
Storage Element

Job submission





Job submission



RB node

Network Server

RB storage

Workload Manager

Job

Job Contr.

Replica Location Server

Inform. Service

Job Status

submitted

waiting

ready

CE characts & status

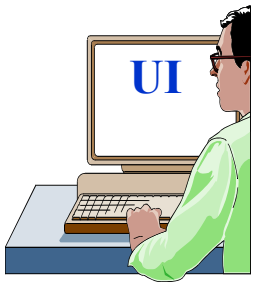
SE characts & status

Storage Element

Element

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

Job submission



RB node

Network Server

Workload Manager

RB storage

Job Contr. - CondorG

Job

Replica Location Server

Inform. Service

CE characts & status

SE characts & status

Computing Element

Storage Element

Job Status

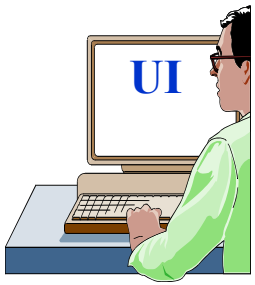
submitted

waiting

ready

scheduled

Job submission



RB node

Network Server

Workload Manager

Job Contr. - CondorG

RB storage

Replica Location Server

Inform. Service

Job Status

submitted

waiting

ready

scheduled

running

Input Sandbox files

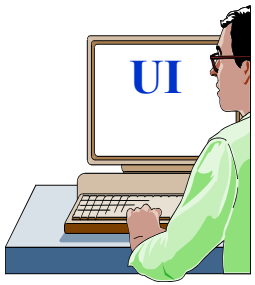
Computing Element

Job

"Grid enabled" data transfers/ accesses

Storage Element

Job submission



RB node

Network Server

Workload Manager

Job Contr. - CondorG

RB storage

Output Sandbox files

Computing Element

Replica Location Server

Inform. Service

Storage Element

Job Status

submitted

waiting

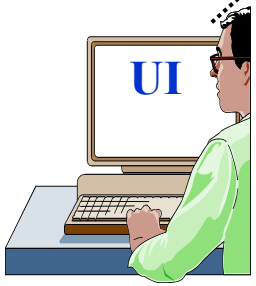
ready

scheduled

running

done

Job submis



RB node

edg-job-get-output <dg-job-id>

Network Server

Workload Manager

RB storage

Job Contr. - CondorG

Replica Location Server

Inform. Service

Job Status

submitted

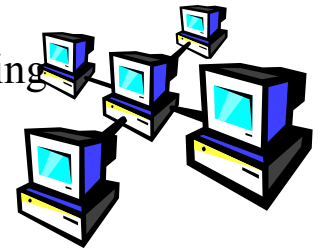
waiting

ready

scheduled

running

done



Computing Element

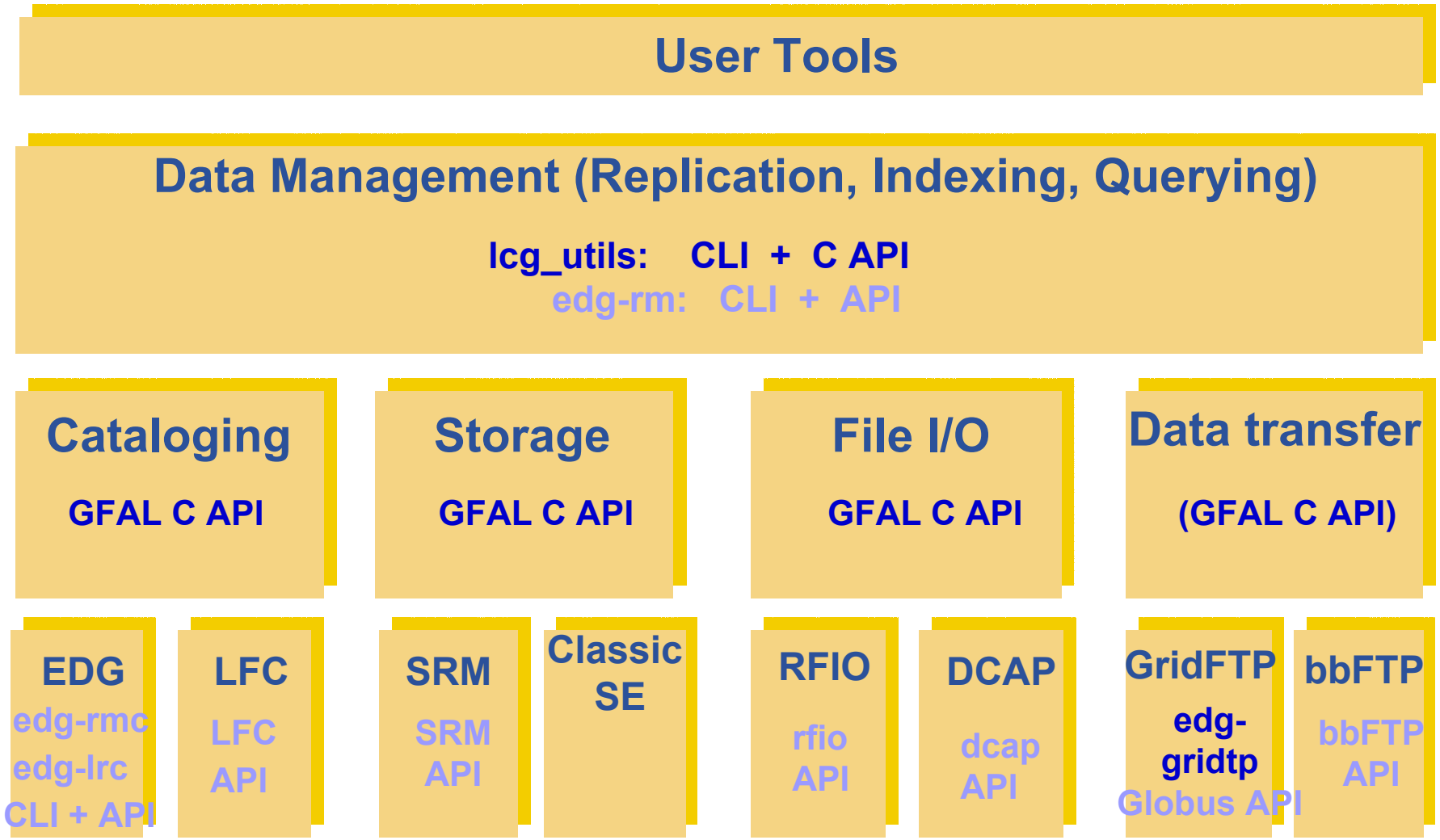
Storage Element

Command-line: Job submission

- edg-job-submit [–r *<res_id>]* [-c *<config file>]* [-vo *<VO>]* [-o *<output file>]* *<job.jdl>*
  - -r the job is submitted directly to the computing element identified by *<res_id>*
  - -c the configuration file *<config file>* is pointed by the UI instead of the standard configuration file
  - -vo the Virtual Organization (if user is not happy with the one specified in the UI configuration file)
  - -o the generated edg_jobId is written in the *<output file>*
    - Useful for other commands, e.g.:
    - **edg-job-status –i** *<input file>* (or edg_jobId)

      -i the status information about edg_jobId contained in the <input file> *are displayed*

DM CLIs & APIs overview



DM CLIs & APIs: LFC API

Low level methods (many POSIX-like):

lfc_access
lfc_aborttrans
lfc_addreplica
lfc_apiinit
lfc_chclass
lfc_chdir
lfc_chmod
lfc_chown
lfc_closedir
lfc_creat
lfc_delcomment
lfc_delete
lfc_deleteclass
lfc_delreplica
lfc_endtrans
lfc_enterclass
lfc_errmsg
lfc_getacl
lfc_getcomment
lfc_getcwd
lfc_getpath
lfc_lchown
lfc_listclass
lfc_listlinks
lfc_listreplica
lfc_lstat
lfc_mkdir
lfc_modifyclass
lfc_opendir
lfc_queryclass
lfc_readdir
lfc_readlink
lfc_rename
lfc_rewind
lfc_rmdir
lfc_selectsrvr
lfc_setacl
lfc_setatime
lfc_setcomment
lfc_seterrbuf
lfc_setfsize
lfc_starttrans
lfc_stat
lfc_symlink
lfc_umask
lfc_undelete
lfc_unlink
lfc_utime
send2lfc

lcg_utils: Data transfer & Storage

lcg-cp	**Copies a Grid file to a local destination**
lcg-cr	**Copies a file to a SE and registers the file in the LRC**
lcg-del	**Deletes one file (either one replica or all replicas)**
lcg-infosites	**Gives information about resources on the Grid**
lcg-rep	**Copies a file from SE to SE and registers it in the LRC**

---

lcg-gt	**Gets the TURL for a given SURL and transfer protocol**
lcg-sd	**Sets file status to “Done” in a specified request**

lcg_utils API

- **lcg_utils API:**
  - High-level data management C API
  - Same functionality as lcg_util command line tools
- **Single shared library**
  - liblcg_util.so (+ libgfal.so)
- **Single header file**
  - lcg_util.h

lcg_utils: Data Transfer and Storage

```
int lcg_cp (char *src_file, char *dest_file, char *vo, int nbstreams, char
    * conf_file, int insecure, int insecure);

int lcg_cr (char *src_file, char *dest_file, char *guid, char *lfn, char *vo,
    char *relative_path, int nbstreams, char *conf_file, int insecure, int
    verbose, char *actual_guid);

int lcg_del (char *file, int aflag, char *se, char *vo, char *conf_file, int
    insecure, int verbose);

int lcg_rep (char *src_file, char *dest_file, char *vo, char
    *relative_path, int nbstreams, char *conf_file, int insecure, int
    verbose);

int lcg_gt (char *surl, char *protocol, char **turl, int *regid, int *fileid,
    char **token);

int lcg_sd (char *surl, int regid, int fileid, char *token, int oflag);
```

lcg_utils: Catalog interaction

```
int lcg_aa (char *lfn, char *guid, char *vo, char *insecure, int verbose);

int lcg_la (char *file, char *vo, char *conf_file, int insecure, char ***lfns);

int lcg_lg (char *lfn_or_surl, char *vo, char *conf_file, int insecure, char
    *guid);

int lcg_lr (char *file, char *vo, char *conf_file, int insecure, char ***pfns);

int lcg_ra (char *lfn, char *guid, char *vo, char *conf_file, int insecure);

int lcg_rf (char *surl, char *guid, char *lfn, char *vo, char *conf_file, int
    insecure, int verbose, char *actual_guid);

int lcg_uf (char *surl, char *guid, char *vo, char *conf_file, int insecure);
```

Available APIs

```
#include <iostream>
#include <stdlib.h>
#include <string.h>
#include <string>
#include <stdio.h>
#include <errno.h>

// lcg_util is a C library. Since we write C++ code here, we need to
// use extern C
//
extern "C"
{
#include <lcg_util.h>
}
using namespace std;
/*********************************************************************************/
/* The folling example code shows you how you can use the lcg_util API for */
/* replica management. We expect that you modify parts of this code in      */
/* to make it work in your environment. This is particularly indicated      */
/* by ACTION, i.e. your action is required.                                 */
/*********************************************************************************/
int main ()
{
cout << "Data Management API Example " << endl;
char *vo = "cms"; // ACTION: fill in your correct VO here: gilda !
cout << "-----------------------------------------------" << endl;
```

Available APIs

```
// Copy a local file to the Storage Element and register it in RLS
//
char *localFile = "file:/tmp/test-file"; // ACTION: create a testfile
char *destSE = "lxb0707.cern.ch"; // ACTION: fill in a specific SE char

*actualGuid = (char*) malloc(50);
int verbose = 2; // we use verbosity level 2
int nbstreams = 8; // we use 8 parallel streams to transfer a file

lcg_cr(localFile, destSE, NULL,
        NULL, vo, NULL, nbstreams,
       NULL, 0, verbose, actualGuid);

if (errno)
{
  perror("Error in copyAndRegister:");
 return -1;
} else {
   cout << "We registered the file with GUID: " << actualGuid << endl;
}
cout << "-----------------------------------------------" << endl;
```

Available APIs

```
// Call the listReplicas (lcg_lr) method and print the returned URLs
//
// The actualGuid does not contain the prefix "guid:". We add it here and
// then use the new guid as a parameter to list replicas
//
std::string guid = "guid:";
guid.insert(5,actualGuid);
char ***pfns = (char***) malloc(200);

lcg_lr((char*) guid.c_str(), vo, NULL, 0, pfns);

if(errno)
{
   perror("Error in listReplicas:");
  free(pfns);
  return -1;
} else {
  cout << "PFN = " << **pfns << endl;
}

free(pfns);

cout << "-----------------------------------------------" << endl;
```

Available APIs

```
// Delete the replica again
//
int rc = lcg_del((char*) guid.c_str(), 1, destSE, vo, NULL, 0, verbose);

if(rc != 0)
{
   perror("Error in delete:");
  return -1;
} else {
  cout << "Delete OK" << endl;
}

return 0;

}
```

Bibliography

- **General LCG-2 information**
  - EGEE Homepage
    http://public.eu-egee.org/
  - EGEE's NA3: User Training and Induction
    http://www.egee.nesc.ac.uk/
  - LCG Homepage
    http://lcg.web.cern.ch/LCG/
  - LCG-2 User Guide
    https://edms.cern.ch/file/454439//LCG-2-UserGuide.html
  - GILDA
    http://gilda.ct.infn.it/
  - GENIUS (GIDA web portal)
    http://grid-tutor.ct.infn.it/

Bibliography

- **Information on Data Management middleware**
  - LCG-2 User Guide (chapters 3rd and 6th)
    https://edms.cern.ch/file/454439//LCG-2-UserGuide.html
  - Evolution of LCG-2 Data Management. J-P Baud, James Casey.
    http://indico.cern.ch/contributionDisplay.py?contribId=278&sessionId=7&confId=0
  - Globus 2.4
    http://www.globus.org/gt2.4/
  - GridFTP
    http://www.globus.org/datagrid/gridftp.html
  - bbFTP
    http://doc.in2p3.fr/bbftp/

- **Information on Storage Elements**
  - SRM:
    http://sdm.lbl.gov/srm-wg/
  - CASTOR:
    http://castor.web.cern.ch/castor/
  - dCache:
    http://www.dcache.org/

Bibliography

- **Information on LCG tools and APIs**
  - Manpages (in UI)
    - lcg_utils: lcg-* (commands), lcg_* (C functions)
    - GFAL: gfal_* (the rest of the commands will be added)
  - Header files (in $LCG_LOCATION/include)
    - lcg_util.h, gfal_api.h
  - CVS developement (sources for LCG commands)

    http://isscvs.cern.ch:8180/cgi-bin/cvsweb.cgi/?hidenonreadable=1&f=u&logsort=date&sortby=file&hideattic=1&cvsroot=lcgware&path=
- **Information on other tools and APIs**
  - EDG CLIs and APIs

    http://edg-wp2.web.cern.ch/edg-wp2/replication/documentation.html
  - RFIO

    http://doc.in2p3.fr/doc/public/products/rfio/rfio.html (In French!)
  - dcap

    http://www.dcache.org/manuals/libdcap.shtml
  - Globus

    http://www-unix.globus.org/api/c/ , ...globus_ftp_client/html , ...globus_ftp_control/html
  - Article on Globus usage (callbacks, etc)

    http://www-106.ibm.com/developerworks/grid/library/gr-cglobus/