Enabling Grids for E-science in Europe

www.eu-egee.org

First Latin-American GRID Workshop
Mérida, Venezuela, 15-20 November 2004

Grid Information System: User Interface, Internals and APIs

Patricia Méndez Lorenzo

patricia.mendez@cern.ch

LCG Experiment Integration and Support
CERN IT/GD-EIS

EGEE is a project funded by the European Union under contract IST-2003-508833

Contents

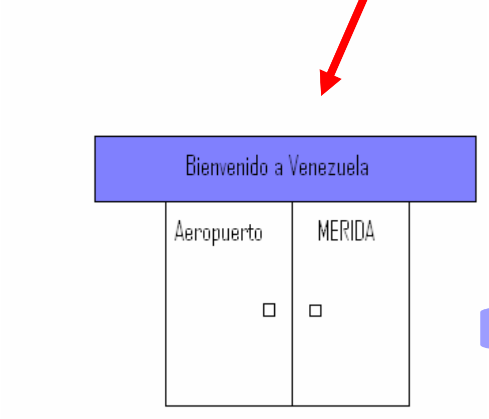
- Some examples of the Information System (IS)
- The IS in EGEE/LCG
  - ►Components, Design, Infrastructure
- Available tools for retrieving information
  - ► as a user or Grid software developer
  - ► as a site manager
- A new era: R-GMA
- A quick summary for the hands-on session

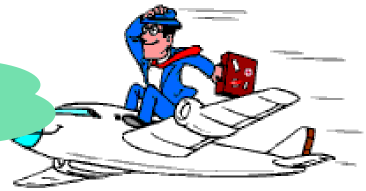
Information arriving in Venezuela

Arriving in a new country:

VENEZUELA

My entry door: The airport





I will need a hotel and a nice restaurant and I 'd like to visit a museum…

e needs
formation!!

These are Information Servers… I do not know how they get the info. I trust them

A esk

Interactive Screens, etc

Restaurants, hotels, taxis, buses… are

SERVICES

Information arriving in GRID

Arriving in a new country:

GRID

My entry door: The UI

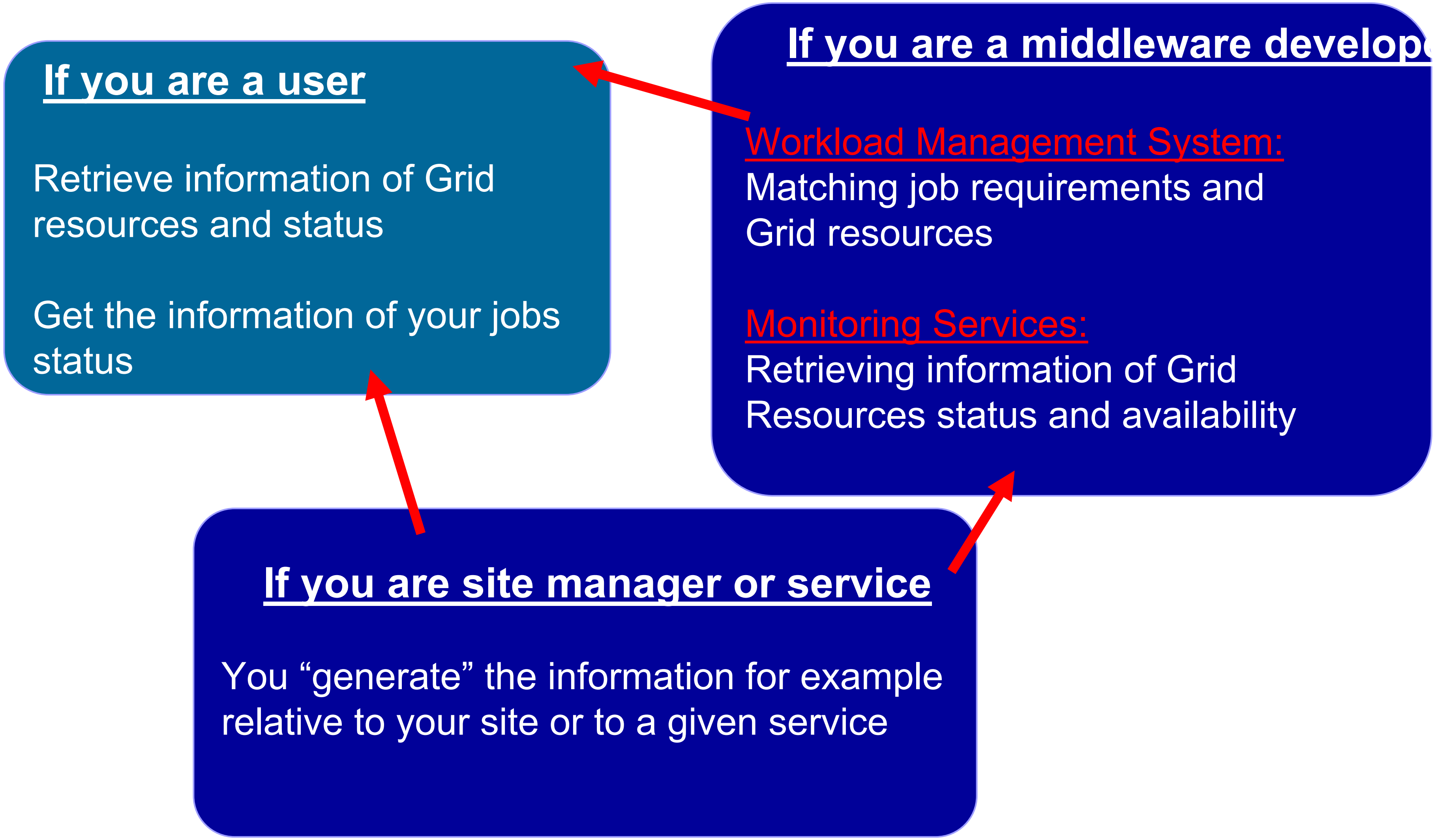
I will need a site with more than 30 CPUs and available space of 2TB and…

She needs information!!

I have several tools to get this information. It depends on what I want and how I want it. However this time it might be important how the information is retrieved

CEs, SEs, sites are SERVICES

Uses of the IS in EGEE/LCG



If you are a user

Retrieve information of Grid resources and status

Get the information of your jobs status

If you are a middleware develope

Workload Management System:
Matching job requirements and Grid resources

Monitoring Services:
Retrieving information of Grid Resources status and availability

If you are site manager or service

You “generate” the information for example relative to your site or to a given service

Elements behind the IS

```
******************************************************************************
These are the data for alice: (in terms of CPUs)
******************************************************************************
#CPU  Free   Total Jobs    Running   Waiting    Computing Element

----------------------------------------------------------------------------

52     51       0            0          0    ce.prd.hp.com:2119/jobmanager-lcgpbs-long

16     14       3            2          1    lcg06.sinp.msu.ru:2119/jobmanager-lcgpbs-long

[…………]

The total values are:
----------------------
10347  5565     2717        924       1793
```

She will use some EGEE/LCG tools and after few moments…

¤ **Something has managed this information: (General IS architecture)**

¤ **Something has provided it: (Providers, Servers)**

¤ **It is following a certain "schema": (GLUE Schema)**

¤ **And she has accessed it following a protocol: (Access Protocol: LDAP)**

The Information System Elements

MDS: Monitoring and Discovery Service

- ► Adopted from Globus
- ► It is the general architecture of EGEE/LCG to manage Grid information

General steps:

1st. At each site **providers** report static and dynamic service status to **servers**

2nd. A **central system** queries these servers and stores the retrieved information in a database

3rd. This information will be accessed through a given **access protocol**

4th. The central system provides the information in a **given schema**

MDS is the EGEE/LCG Information System

The LDAP Protocol: Generalities

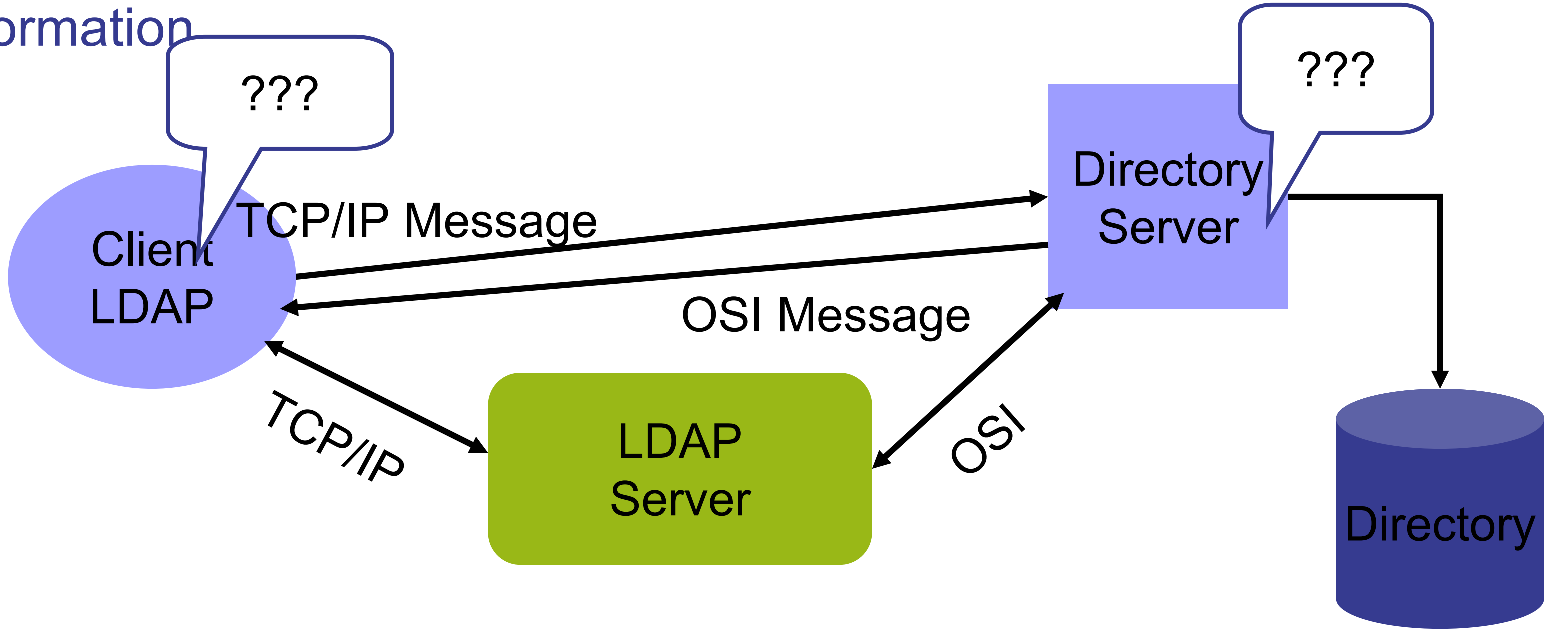
LDAP (Lightweight Directory Access Protocol)

√ It establishes the transport and format of the messages used by a client to access a directory

√ LDAP can be used as access protocol for a large number of databases

√ It provides a standard data model; the DIT (Data Information Tree)

√ It is the internal protocol used by the EGEE/LCG services to share information



The LDAP Protocol: The Data Inputs

► The LDAP information model is based on **entries**

► These are **attribute** collections defined by a unique and global DN (Distinguished Name)

► Information is organized in a tree-like structure. A special attribute, **objectclass**, can be defined for each entry. It defines the classes tree corresponding to this entry. This attribute can be used to filter entries containing that object class

► The information is imported and exported from and to the LDAP server by **LDIF files** (LDAP Data Interchange Format)

```
dn: <distinguished name>
objectclass:<objectclassname>
<attributetype>:<attributevalue>
<attributetype>:<attributevalue>

dn: <distinguished name>
objectclass:<objectclassname>
<attributetype>:<attributevalue>
<attributetype>:<attributevalue>
```

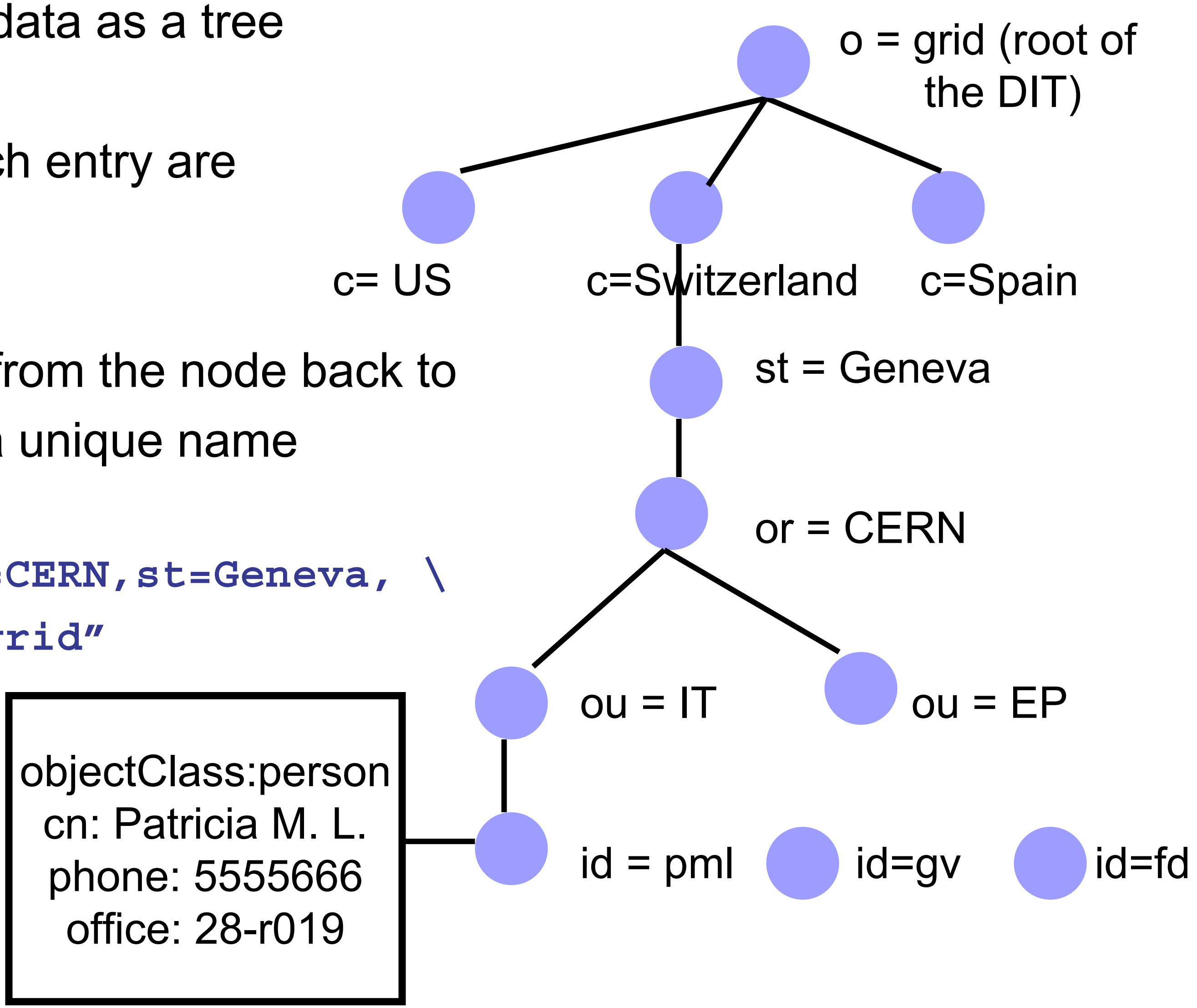
► Those fields delimited by <> can be defined by the application following a certain **schema**

►The schema describes the attributes and the types associated with the data objects

The LDAP Protocol: DIT

- ► LDAP structures data as a tree
- ► The values of each entry are uniquely named
- ► Following a path from the node back to the root of the DIT, a unique name is built (the DN):

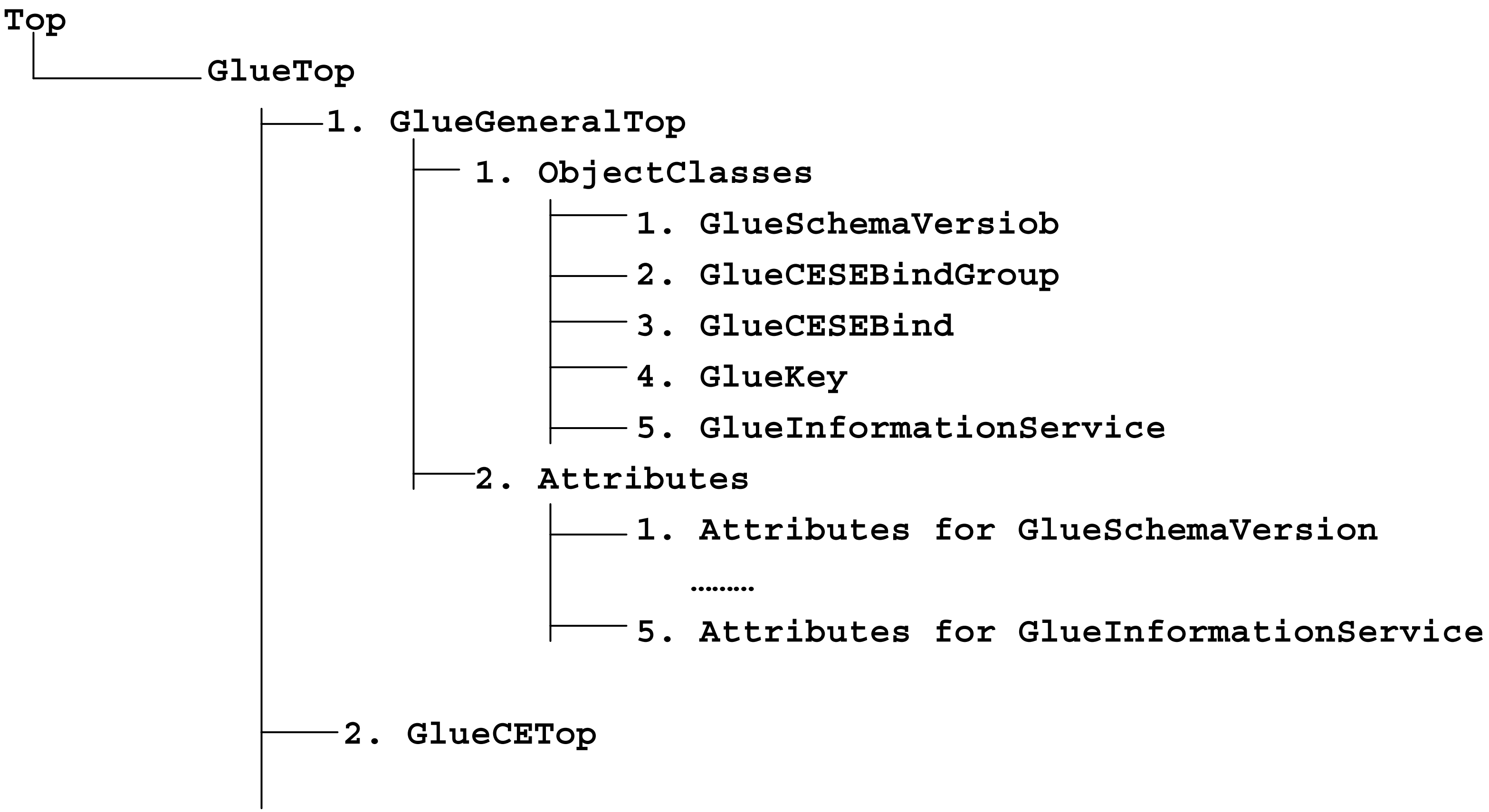
**“id=pml,ou=IT,or=CERN,st=Geneva, **
c=Switzerland,o=grid”



objectClass:person
cn: Patricia M. L.
phone: 5555666
office: 28-r019

The Glue Schema in EGEE/LCG: Design

- ♠ It describes the Grid resources information stored by the IS
- ♠ It follows the DIT hierarchical structure for objectclasses and attributes:



```
Top
 └──────── GlueTop
              ├──1. GlueGeneralTop
              │      ├── 1. ObjectClasses
              │      │      ├──── 1. GlueSchemaVersiob
              │      │      ├──── 2. GlueCESEBindGroup
              │      │      ├──── 3. GlueCESEBind
              │      │      ├──── 4. GlueKey
              │      │      └──── 5. GlueInformationService
              │      └───2. Attributes
              │             ├──── 1. Attributes for GlueSchemaVersion
              │             │          ………
              │             └──── 5. Attributes for GlueInformationService
              │
              ├───2. GlueCETop
              │
```

The Glue Schema in EGEE/LCG: Examples

1. Some General Attributes:

- ¤ Base class (`objectclass: GlueTop`): No attributes
- ¤ Schema Version Number (`objectclass: GlueSchemaVersion`)
  - `GlueSchemaVersionMajor`: Major Schema Version Number
  - `GlueSchemaVersionMinor:` Minor Schema Version Number

2. Attributes for the CE

- ¤ Base Class (`objectclass: GlueCETop`): No attributes
- ¤ CE (`objectclass: GlueCE`)
  - `GlueCEUniqueID`: Unique identifier for the CE

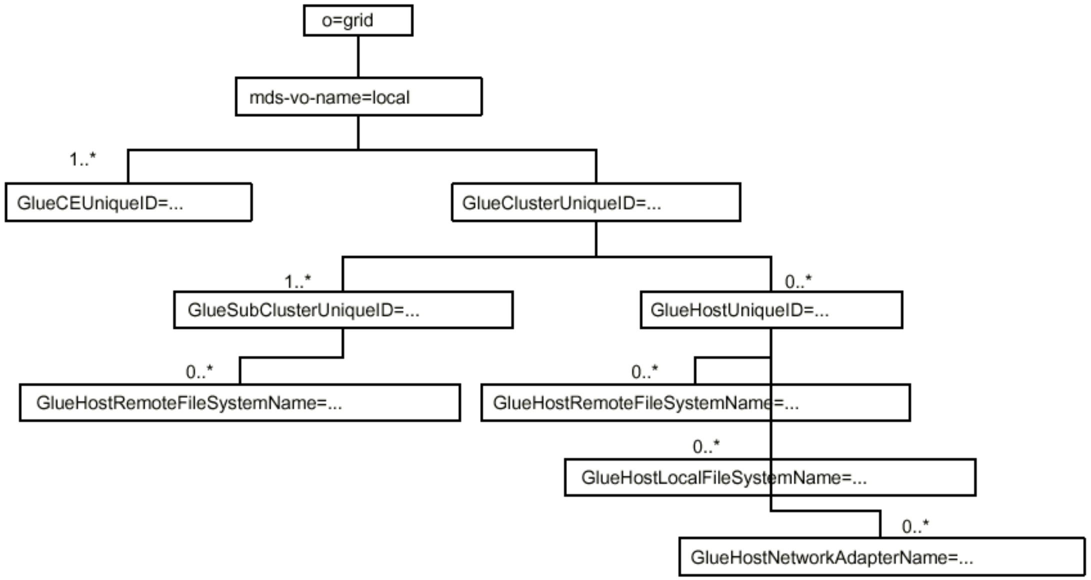
3. Attributes for the SE

- ¤ Base Class (`objectclass: GlueSETop`): No attributes
- ¤ Architecture (`objectclass: GlueSLArchitecture`)
  - `GlueSLArchitectureType`: type of storage hardware (disk, tape, etc)

4. Mixed Attributes

- ¤ Association between one CE and one or more SEs (`objectclass: GlueCESEBindGroup`)
  - `GlueCESEBindGroupCEUniqueID`: unique ID for the CE
  - `GlueCESEBindGroupSEUniqueID`: unique ID for the SE

The Glue Schema in EGEE/LCG: DIT



o=grid

mds-vo-name=local

1..* GlueCEUniqueID=...

GlueClusterUniqueID=...

1..* GlueSubClusterUniqueID=...

0..* GlueHostUniqueID=...

0..* GlueHostRemoteFileSystemName=...

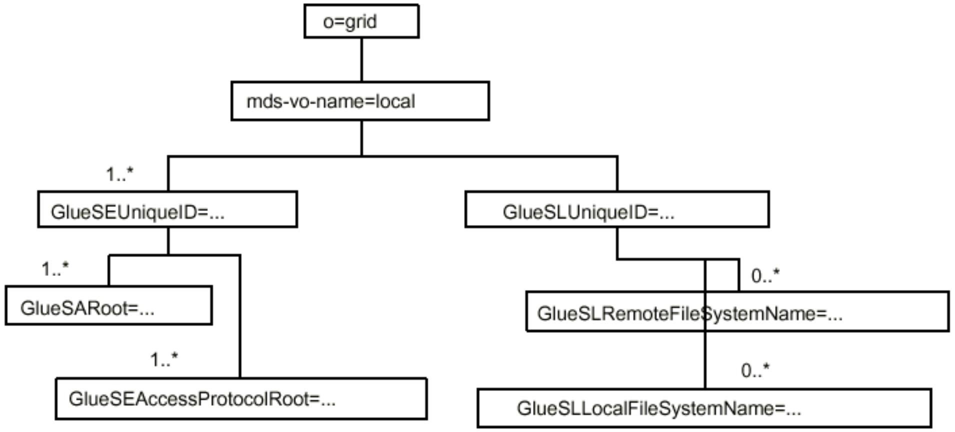
0..* GlueHostRemoteFileSystemName=...

0..* GlueHostLocalFileSystemName=...

0..* GlueHostNetworkAdapterName=...

DIT for the Computer Resources

DIT for the Storage Resources



o=grid

mds-vo-name=local

1..* GlueSEUniqueID=...

GlueSLUniqueID=...

1..* GlueSARoot=...

1..* GlueSEAccessProtocolRoot=...

0..* GlueSLRemoteFileSystemName=...

0..* GlueSLLocalFileSystemName=...

How to handle the Information in an LDAP server

¤ OpenLDAP is an open source implementation of LDAP protocol

¤ It provides CLI and C/C++ APIs to search, add, remove, modify entries in the directory. Synchronous and asynchronous operations are allowed

¤ APIs description:
`http://www.openldap.org/software/man.cgi?query=ldap`

¤ All these APIs have correspondent CLIs already included in the distribution

→ ldapadd

→ ldapdelete

→ ldapmodify

→ ldapsearch

(Make a *"man"* to these commands to get more information)

¤ OpenLDAP includes also:

→ JLDAP: LDAP class libraries for Java

→ JDBC: LDAP-Java JDBC-LDAP Bridge Driver

The use of the command lines in LDAP

♠ ldapsearch

Read port of the BDII

`% ldapsearch \`	
`-x \`	Simple authentication
`-H ldap://grid017.ct.infn.it:2170 \`	Uniform resource identifier
`-b 'mds-vo-name=local,o=grid' \`	Base DN for search
`'(objectclass=GlueSE)' \`	Filter
`GlueSEUniqueID \`	Attributes to be returned

(Make “man ldapsearch” to retrieve the whole set of options)

The ldapsearch Implementation in EGEE/LCG

Some wrappers of ldapsearch exist in LCG middleware, but they are not directly exposed to users

→ Part of the internal WMS software

→ Part of the Monitoring tools

ldapsearch test

Test the following commands on your PC:

```
% ldapsearch -x -LLL -h grid017.ct.infn.it -p
  2170 -b “o=grid”
```

```
% ldapsearch -x -LLL -h grid017.ct.infn.it -p
  2170 -b “o=grid” ‘(objectclass=GlueSE)’
  GlueSEName GlueSEPort
```

Change objectclasses, attributes...

The use of the command lines in LDAP

♠ ldapadd, ldapmodify, ldapdelete

Write port of the BDII

```
% ldapadd \
  -x \
  -H ldap://grid017.ct.infn.it:2171 \
  -D ‘mds-vo-name=local,o=grid’ \
  -f <your-file>
```

Simple authentication

Uniform resource identifier

DN binddn to bind to the directory

File containing your new entry

ldapadd, ldapmodify and ldapdelete in LCG-2

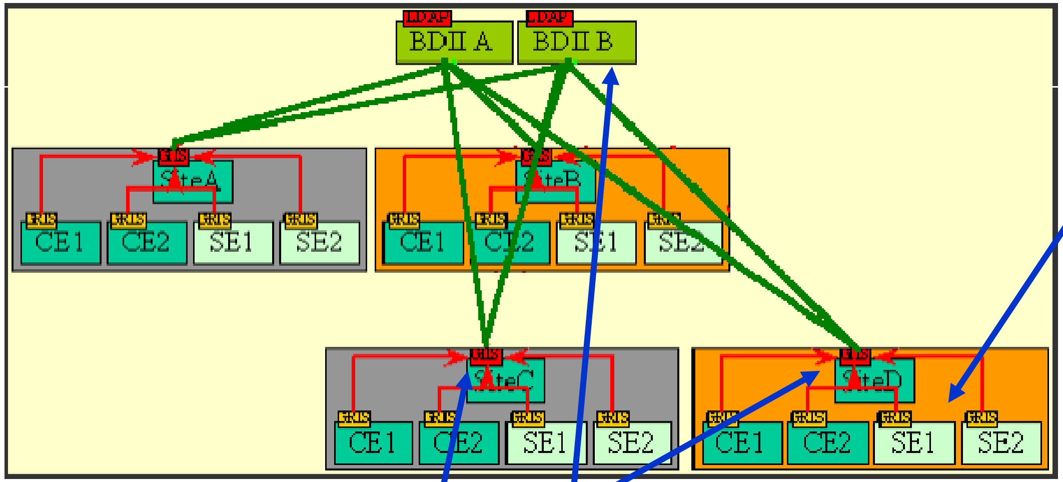
- LCG does NOT allow the use of these commands to create or modify information
- Several tools have been developed to include information in the servers

→ They are not based on LDAP

→ The query tools of LDAP can however retrieve this information

Collectors and Providers

GRISs, GIISs and BDII



- **Local GRISs** run on CEs and SEs at each site and report dynamic and static information regarding the status and availability of the services

```
% ldapsearch -x -h
<hostname> -p 2135 -b “mds-
vo-name=local,o=grid”
```

- At each site, a **site GIIS** collects the information of all resources given by the GRISs

```
% ldapsearch -x -h <hostname> -p 2135 -b “mds-vo-name=<name>,o=grid”
```

- Each site can run a **BDII**. It collects the information coming from the GIISs and collects it in a data base

```
% ldapsearch -x -h <hostname> -p 2170 -b “o=grid”
```

The BDII

Enabling Grids for E-science in Europe

This is the information server directly invoked by users and services

√ Because only those sites listed in the BDII really exist (it registers site GIISs)

√ Because it provides information to the RB (to find resources)

√ Because it is needed by the data management tools. The "**`lcg-utils`**" tools use it (see the Data Management talk)

√ Fundamental service to allow for stability (seen many times during the Alice DC for example). It is possible to define a hierarchy of Information Systems.

√ Because it can be configured by each VO following its needs using global production configuration file distributed by CERN via AFS.

```
/afs/cern.ch/project/gd/www/gis/lcg2-bdii/<alice>
```

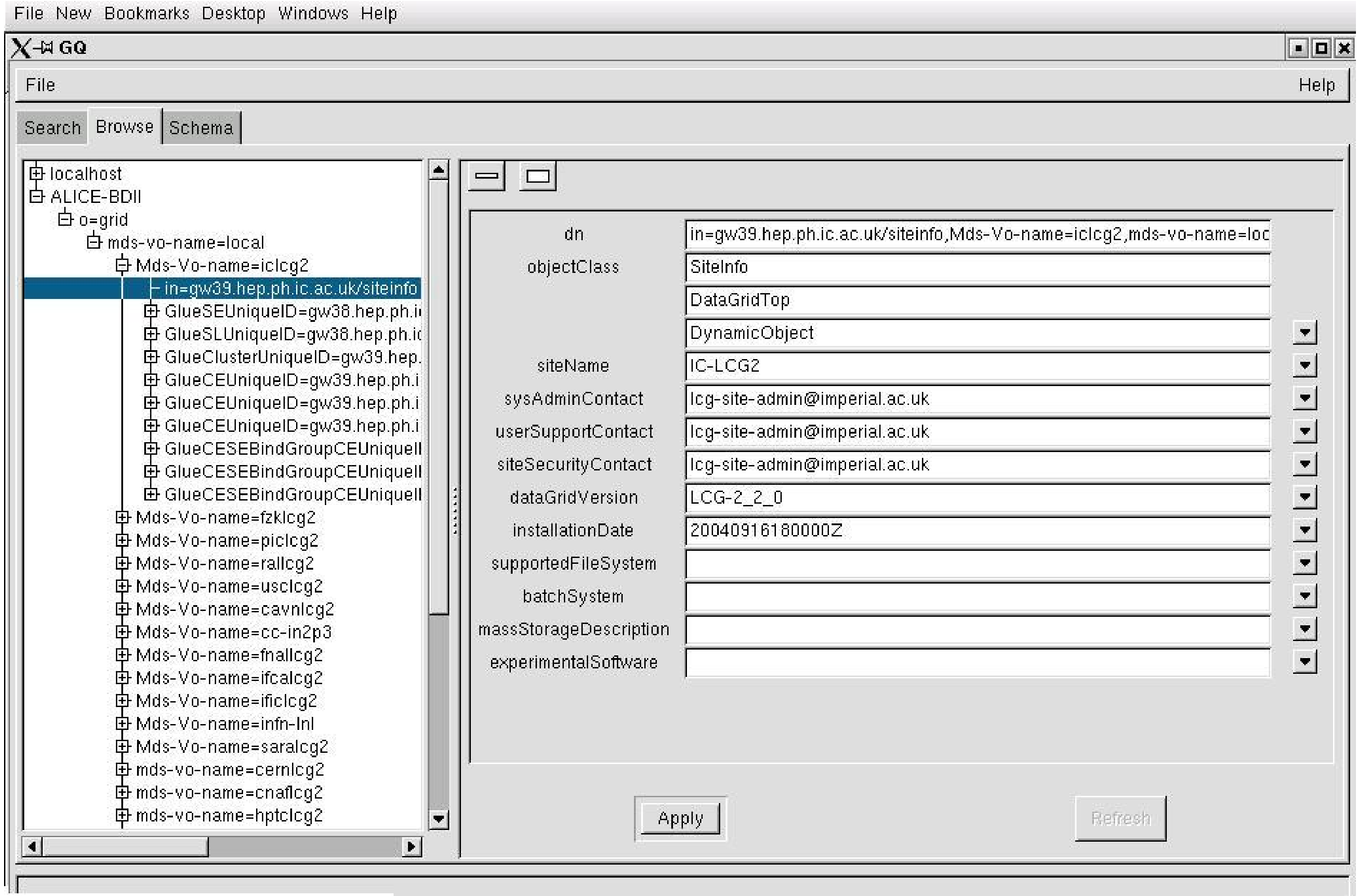
→ The VOs members and the LCG group have access to these files

→ Each VO decides where jobs should be executed independently of the rest of the Grid

The BDII

Enabling Grids for E-science in Europe

DIT of the Alice BDII for Production



```
⊞ localhost
⊟ ALICE-BDII
  ⊟ o=grid
    ⊟ mds-vo-name=local
      ⊟ Mds-Vo-name=iclcg2
        ├ in=gw39.hep.ph.ic.ac.uk/siteinfo
        ⊞ GlueSEUniqueID=gw38.hep.ph.i
        ⊞ GlueSLUniqueID=gw38.hep.ph.i
        ⊞ GlueClusterUniqueID=gw39.hep.
        ⊞ GlueCEUniqueID=gw39.hep.ph.i
        ⊞ GlueCEUniqueID=gw39.hep.ph.i
        ⊞ GlueCEUniqueID=gw39.hep.ph.i
        ⊞ GlueCESEBindGroupCEUniqueI
        ⊞ GlueCESEBindGroupCEUniqueI
        ⊞ GlueCESEBindGroupCEUniqueI
      ⊞ Mds-Vo-name=fzklcg2
      ⊞ Mds-Vo-name=piclcg2
      ⊞ Mds-Vo-name=rallcg2
      ⊞ Mds-Vo-name=usclcg2
      ⊞ Mds-Vo-name=cavnlcg2
      ⊞ Mds-Vo-name=cc-in2p3
      ⊞ Mds-Vo-name=fnallcg2
      ⊞ Mds-Vo-name=ifcalcg2
      ⊞ Mds-Vo-name=ificlcg2
      ⊞ Mds-Vo-name=infn-lnl
      ⊞ Mds-Vo-name=saralcg2
      ⊞ mds-vo-name=cernlcg2
      ⊞ mds-vo-name=cnaflcg2
      ⊞ mds-vo-name=hptclcg2
```

dn	in=gw39.hep.ph.ic.ac.uk/siteinfo,Mds-Vo-name=iclcg2,mds-vo-name=loc
objectClass	SiteInfo
	DataGridTop
	DynamicObject
siteName	IC-LCG2
sysAdminContact	lcg-site-admin@imperial.ac.uk
userSupportContact	lcg-site-admin@imperial.ac.uk
siteSecurityContact	lcg-site-admin@imperial.ac.uk
dataGridVersion	LCG-2_2_0
installationDate	20040916180000Z
supportedFileSystem	
batchSystem	
massStorageDescription	
experimentalSoftware	

globus-mds: top responsible service

► Lower level: GRIS

- Scripts and configuration files generate ldif files containing the information (for example, general information of the nodes)

- Other tools responsible of the dynamic information (for example, available and/or used space into a SE) – the so called information providers

- globus-mds runs such tools every few seconds. The system merges the dynamic information with the static one and register it to the local cache.

► Medium level: local GIIS

- Same procedure taking the information from the registered GRISs

► High level: BDII

- Publish the information of the site GIISs making a refresh every 2 minutes

► An example: the Resource Broker

- This is a Grid service and publishes its information and status to the information system as described above (it is a server)
- However it uses a BDII for matchmaking purposes (it is a client)

Information System Tools

1. You are a user with no privileges

- Using LDAP you cannot generate but just retrieve information (ldapsearch)
- Some C++ APIs and scripts have been developed to make this job easier

♠ **lcg-is-search**

LDAP C++ API included in LCG-2 to retrieve information

☺ Why the need for this tool?

1. API allows users to interrogate the IS from any application or services
2. Better way of presenting the information (no way with ldapsearch)

☺ Which kind of tools are installed? (rpm: lcg-info-api-ldap-1.1-1.4 included in Gilda testbed)

1. A library: `/opt/lcg/lib/liblcg-info-api-ldap`
2. Headers: `/opt/lcg/include/lcg-info-api-ldap/`
3. Several handy executables: `lcg-is-search,` `lcg-infosites, …`

☺ Where do I find it?

WNs and UIs in `/opt/lcg/bin`

This will be tested during the hands-on session

Information System Tools

```
> lcg-is-search –h <host> -f objectclass=<your_request> -a \
'<your_attributes>'
```

lcg-is-search basic Code

```
#include<dlfcn.h>
#include<iostream> (<strtream>)
#include<vector> (<iterator>, <string>)

#include <lcg-info-api-ldap/InfoFromLDAP.h>
#include<lcg-info-api-ldap/AllInfoLDAP.h>

int main (int argc, char*argv[]){
char *hosttest;
char* first_ptr;
char* last_ptr;
hosttest = getenv("LCG_GFAL_INFOSYS");
first_ptr = hosttest;
last_ptr = strchr(hostest,":");
*last_ptr = '\0';
++last_ptr;
std::string host(first_ptr);
typedef enum {_param_,_host_,_port_,_filter_,_attr_} arg_t;
arg_t expected = _param_;
```

to include DLOPEN

including classes of the package

If not specified, the host will be taken from LCG_GFAL_INFOSYS

including external arguments

Information System Tools

lcg-is-search basic Code (cont.)

```
for (int i = 1; i<argc: i++){
   string token;
   bool read_token = true;
   istream* in = new istrstream(argv[i]);
   switch (expected){
   case _port_: (*in) >> port; expected = _param_;break;
   case _host_: (*in) >> host; expected = _host_;break;
   case _filter_: (*in) >> filter; expected = _filter_;break;
   case _attr_: (*in) >> attribute;
   if (attribute[0] !=‘-’){
      attributes.push_back(attribute.c_str());
      break;
   }
   else {
      token = attribute.c_str();
      read_token = false;
   }
     default:
       if(read_token) (*in) >> token;
       if(token ==“-p”) expected = _port_;
       else if(token == “-h”) expected = _host_;
       else if(token == “-f”) expected = _filter_;
       else if(token == “-a”) expected = _attr_;
```

Part of the code
To include external arguments

Information System Tools

lcg-is-search basic Code (cont.)

```
  else {
     cout<<"invalid parameter" <<token<<endl;
   }
 }
 delete in;

#ifndef __WINDOWS__;
char* lib_loc = "liblcg-info-api-ldap.so";

void *InfoFromLDAP = dlopen(lib_loc,RTLD_LAZY);

create_t* create_infoldap = (create_t*)dlsym(InfoFromLDAP,"create");
destroy_t* destroy_infoldap = (destroy_t*)dlsym(InfoFromLDAP,"destroy");

AllInfoFromLDAP *ldapinfo = create_infoldap();
ldapinfo-> query(host,filter,attributes,port);

destroy_infoldap(ldapinfo);
dlclose(InfoFromLDAP);
#endif
}
```

including the library to load

loading dynamically the library

instantiating the class

calling its method

destroying the pointer

Information System Tools

InfoFromLDAP.h

```
#include “AllInfoLDAP.h”
#include “LDAPConnection.h”

class InfoFromLDAP: public AllInfoLDAP{

Public:
   InfoFromLDAP();
   ~InfoFromLDAP();

   virtual void query(string, string, vector<string>, int);
Private:
   LDAPConnection* connection;
}
```

one of the LDAP wrappers

price to pay to include the dlopen package

the method of the class

pointer to connect the server

AllInfoLDAP.h

```
class AllInfoLDAP{
public:
virtual void query(string, string, vector<string>,int) = 0;
};
typedef AllInfoLDAP* create_t();
typedef void destroy_t(AllInfoLDAP*);
```

mandatory because of dlopen

Information System Tools

Enabling Grids for E-science in Europe

InfoFromLDAP.cpp

```
#include “LDAPQuery.h”
#include “LDAPSynchConnection.h”
#include “LDAPForwardIterator.h”
#include “InfoFromLDAP.h”

InfoFromLDAP::InfoFromLDAP(){};
InfoFromLDAP::~InfoFromLDAP(){};

void InfoFromLDAP::query(string host, string filter, vector<string> attributes,
     int port){
string information_index = “o=grid”;
int timeout = 30;

connection = new LDAPSynchConnection (information_index,host,port,timeout);
copy (attributes.begin(),attributes.end(),ostream_iterator<string>(cout,” ”));

LDAPQuery query(connection,filter,attributes);

Connection -> open();
Query.execute();

LDAPForwardIterator ldap_it(query.tuples());
```

wrappers of LDAP

instantiating a synch connection

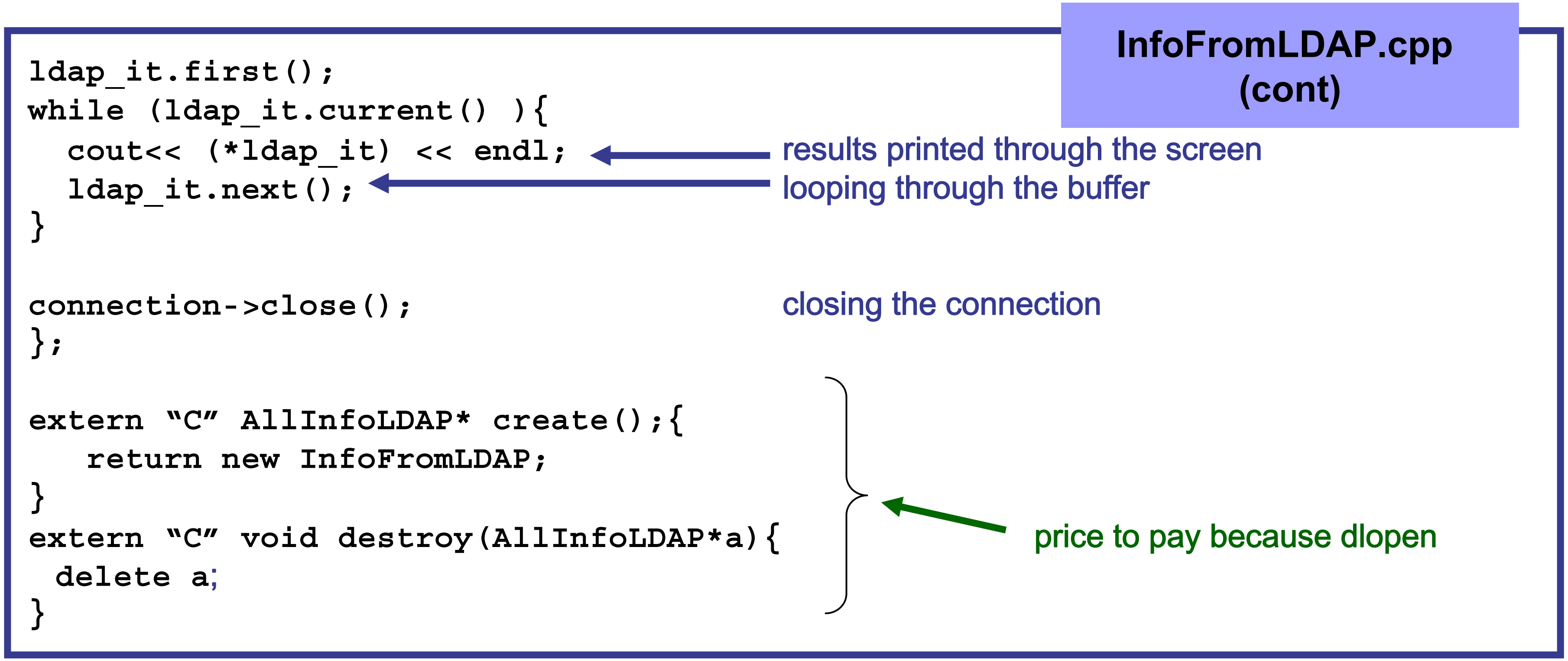
informing of the query to perform

connecting with the server

executing the query

iterating the IS buffer

Information System Tools



InfoFromLDAP.cpp (cont)

```
ldap_it.first();
while (ldap_it.current() ){
  cout<< (*ldap_it) << endl;
  ldap_it.next();
}

connection->close();
};

extern “C” AllInfoLDAP* create();{
   return new InfoFromLDAP;
}
extern “C” void destroy(AllInfoLDAP*a){
  delete a;
}
```

results printed through the screen

looping through the buffer

closing the connection

price to pay because dlopen

It seems dlopen is quite difficult to use (additional classes and code) but has fundamental advantages

dlopen

Enabling Grids for E-science in Europe

In some situations it can be very useful to load a certain library at r me

►In many cases you want your code to support multiple te

plug-ins

►You want to make your code independent

No way to instantiate the class in your code

- **The Solution:**

Plug-ins usage: load the libra cally at runtime only when needed.

- **But….**

It is quite easy to do in in C but not so easy in C++:

→ Because of name mangling

→ Because you have to expose the symbols of the whole class in C++

- **Solution:**
- Extern “C” (for the name mangling)
- Polymorphism (for the classes)

dlopen in our code

♠ In our code we want to load a class into the main (`lcg-is-search`); `InfoFromLDAP` to use its method query

♠ We cannot use “**new**” to instantiate the class

Solution:

1. We define a base class: `AllInfoLDAP.h` (pure virtual) and `InfoFromLDAP` will be derived from it (called module)

AllInfoLDAP.h

```
// the types of the class factory
typedef AllInfoLDAP* create_t();
typedef void destroy_t (AllInfoLDAP*)
```

2. Inside the module two helper functions (class factory functions) will be defined as extern “C”

InfoFromLDAP.cpp

```
// the class factories
extern “C” AllInfoLDAP* create() {result new InfoFromLDAP;}
extern “C” void destroy(AllInfoLDAP* a) {delete a;}
```

dlopen in our code

Finally in the code **`lcg_is_search.cpp`**

```
void *InfoFromLDAP = dlopen(“your lib”,RTLD_LAZY);

create_t* create_infoldap = (create_t*)   \
       dlsym(InfoFromLDAP,”create”);
destroy_t* destroy_infoldap = (destroy_t*)   \
       dlsym(InfoFromLDAP,”destroy”);

AllInfoLDAP* ldapinfo = create_infoldap();
ldapinfo ->query;

destroy_infoldap(ldapinfo);
dlclose(InfoFromLDAP)
```

implementation

Seems similar to new…

Using the method

Seems similar to delete…

lcg-is-search tests

You can try the same queries you made with ldapsearch:

```
% /opt/lcg/bin/lcg-is-search –f objectclass=GlueSE –a GlueSEName GlueSEPort
```

¤ You do not have additional information you did not ask for (the DNs)

¤ The lines are not cut at the end

Compare with ldapsearch

```
lcg-is-search -f objectclass=GlueTop -a‘(& (GlueServiceType=edg-local-replica-
   catalog) (GlueServiceAccessControlRule))’ GlueServiceAccessPointURL
```

First of all you do not care about hosts or ports. Just in the case you want an specific host, otherwise lcg-is-search looks at the one in default

```
ldapsearch -h grid017.ct.infn.it -p 2170 -x -LLL -b “o=grid”
   ‘(objectclass=GlueTop)’ ‘(& (GlueServiceType=edg-local-replica-catalog)
   (GlueServiceAccessControlRule))’ GlueServiceAccessPointURL
```

¤ You do not ask for the DN

¤ The lines are cut at the end of the buffer. It’s very difficult to wrap this information into your code

Implementation of lcg_is_search in LCG-2

♠ lcg-infosites

- This is a script which invokes lcg-is-search
- Already deployed in LCG-2 in the last release
- It is intended to be the most complete information retriever for the user:
  - √ Once he arrives at the Grid (on UIs)
  - √ To be used by the user applications (on WNs)
- Several versions of this script have been included in the software packages of ATLAS and the monitoring services of Alice (MonAlisa)
- You do not need a proxy

This will be tested during the hands-on session

lcg-infosites

```
> lcg-infosites --vo <your_vo> feature --is <your_bdii>
```

- It’s mandatory to include the vo and the feature
- The –is option means the BDII you want to query. If not supplied, the BDII defined into the LCG_GFAL_INFOSYS will be interrogated

Features and descriptions:

closeSE	Names of the CEs where the user’s VO is allowed to run together with their corresponding closest SEs
ce	Number of CPUs, running and waiting jobs and names of the CEs
se	SEs names together with the available and used space
lrc (rmc)	Name of the lrc (rmc) for the user’s VO
all	It groups all the features just described
help	Description of the script

lcg-infosites

```
> lcg-infosites --vo alice se --is lxb2006.cern.ch
```

```
************************************************
These are the data for alice: (in terms of SE)
************************************************
Avail Space (Kb)      Used Space (Kb)             SEs
-----------------------------------------------------------------
33948480              2024792                 se.prd.hp.com
506234244             62466684                teras.sara.nl
1576747008            3439903232              gridkap02.fzk.de
1000000000000         500000000000            castorgrid.cern.ch
304813432             133280412               gw38.hep.ph.ic.ac.uk
651617160             205343480               mu2.matrix.sara.nl
1000000000000         1000000000              lcgads01.gridpp.rl.ac.uk
415789676             242584960               cclcgseli01.in2p3.fr
264925500             271929024               se-a.ccc.ucl.ac.uk
668247380             5573396                 seitep.itep.ru
766258312             681359036               t2-se-02.lnl.infn.it
660325800             1162928716              tbn17.nikhef.nl
1000000000000         1000000000000           castorftp.cnaf.infn.it
14031532              58352476                lcgse01.gridpp.rl.ac.uk
1113085032            1034242456              zeus03.cyf-kr.edu.pl
                     [… … … … …]
```

lcg-infosites

```
*******************************************************************************
These are the data for alice: (in terms of CPUs)
*******************************************************************************
#CPU  Free   Total Jobs    Running   Waiting    Computing Element

-----------------------------------------------------------------------------

52     51       0            0          0    ce.prd.hp.com:2119/jobmanager-lcgpbs-long

16     14       3            2          1    lcg06.sinp.msu.ru:2119/jobmanager-lcgpbs-long

[…………]

The total values are:
----------------------
10347  5565     2717        924       1793
```

She will use some EGEE/LCG tools and after few moments…

She was using lcg-infosites with option ce

lcg-infosites test

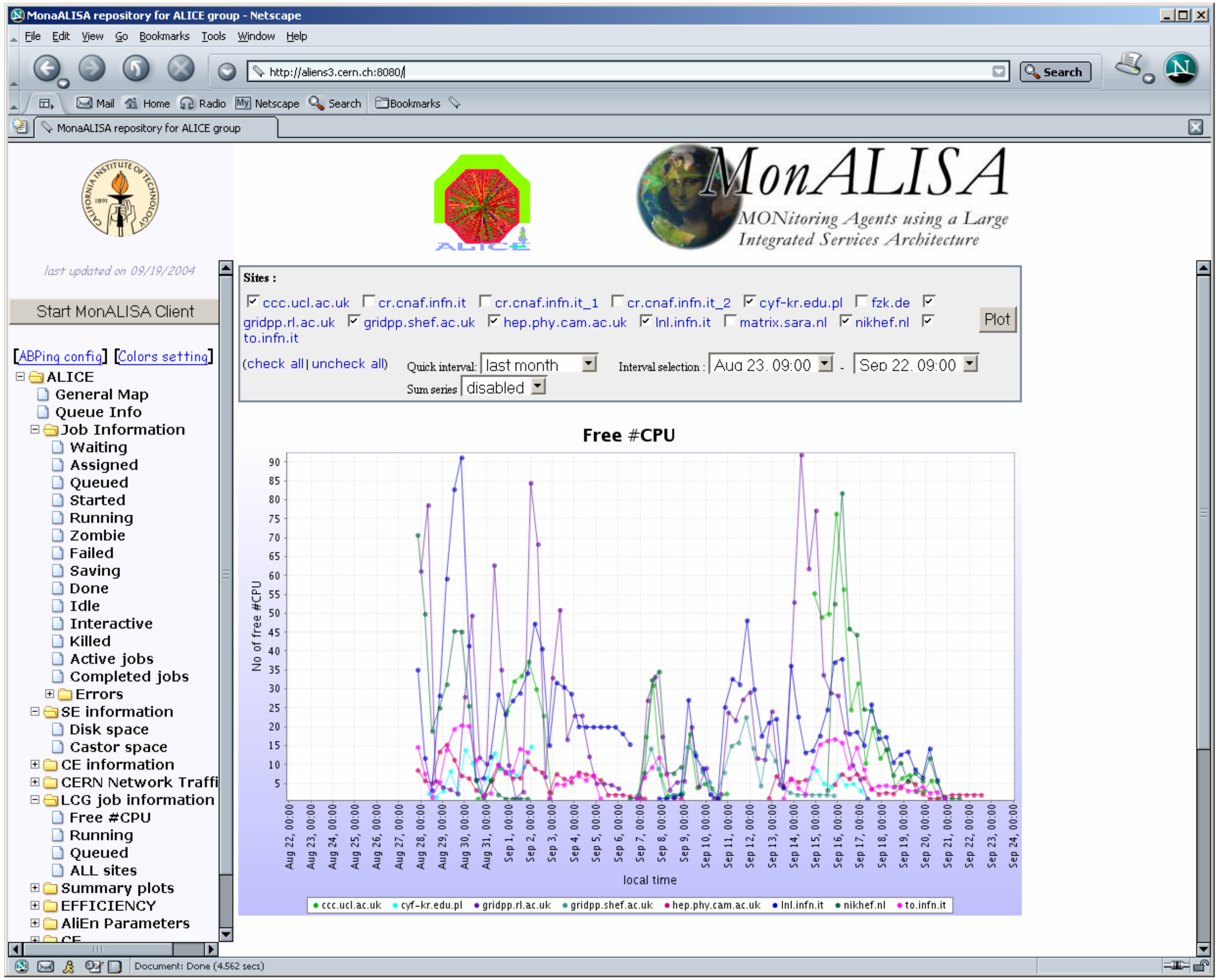
Test some lcg-infosites features:

```
%lcg-infosites –vo gilda ce

%lcg-infosites –vo gilda se

%lcg-infosites –vo gilda all

%lcg-infosites –vo gilda lrc

%lcg-infosites –vo gilda rmc
```

lcg-infosites

Implementation in MonALISA:

The monitoring service of the Alice experiment



http://aliens3.cern.ch:8080/

Sites :

☑ ccc.ucl.ac.uk ☐ cr.cnaf.infn.it ☐ cr.cnaf.infn.it_1 ☐ cr.cnaf.infn.it_2 ☑ cyf-kr.edu.pl ☐ fzk.de ☑ gridpp.rl.ac.uk ☑ gridpp.shef.ac.uk ☑ hep.phy.cam.ac.uk ☑ lnl.infn.it ☐ matrix.sara.nl ☑ nikhef.nl ☑ to.infn.it

(check all | uncheck all)

Quick interval: last month

Interval selection : Aug 23. 09:00 - Sep 22. 09:00

Sum series disabled

Information System Tools

2. You have application software administrator privilege: You can publish application specific information

1. You can install the software of your VO

♠ Through special Grid tools, an application software administrator can submit Grid requests for software installation and validation

♠ Once the software has been installed and validated, a tag specifying the software version can be published in the information system to announce software availability at a site

2. You can publish a software tag corresponding to the software you have installed

♠ Via the script: `lcg-ManageVOTag` (UIs and WNs)

**This will be tested duri
the hands-on sessio**

♠ The tag version is given as an argument to the script

♠ In case the user installs his software with his own tools, lcg-ManageVOtag can be independently used to publish the tag

Information System Tools

♠ lcg-ManageVOTag

```
lcg-ManageVOTag -host <CE_host> -vo <your_vo> --feature -tag \
<your_tag>
```

Features:

√ **add** → It allows to join one or more tags each time (sgm privileges mandatory)

√ **remove** → any tag can be deleted (sgm privileges mandatory)

√ **list** → all tags included by the sgm can be visualized (all users from any VO can used this feature)

It's mandatory the tag follows the `VO-<voname>-<your-information>` syntax

```
> lcg-ManageVOTag -host lxb0706.cern.ch -vo dteam --add -tag VO-dteam-SFW1

lcg-ManageVOTag: VO-dteam-SFW1 submitted for addition by dteam to
GlueApplicationSoftwareRunTimeEnvironment
```

Glue Schema attribute which will be filled with the software tag

Information System Tools

But… what is happening behind?

► The first time this command is used from the UI or the WN, globus-url-copy will be used to create a `/opt/edg/var/info/<VO>/<VO>.list` file including the first tag(s) you include

►The rest of the times the file will just the file will not be recreated and will just hold the new tags

►The `edg-ce-all` (info producer into the CE) will read the file and publish the info, setting the `GlueApplicationSoftwareRunTimeEnvironment` attribute value to the tags included in these files

Just interrogate the BDII or the GIIS:

```
ldapsearch –h lxb0705.cern.ch –p 2170 –x –b "o=grid" –LLL
objectclass=GlueSubCluster GlueApplicationSoftwareRunTimeEnvironment
```

```
dn: GlueSubClusterUniqueID=lxb0706.cern.ch,GlueClusterUniqueID=lxb0706.cern.ch

, Mds-Vo-name=eis,mds-vo-name=local,o=grid

GlueHostApplicationSoftwareRunTimeenvironment: VO-dteam-SFW1
```

Information System Tools

3. You have administrator privileges: You can produce the information

☺ Now you can create easily static information via a interactive script included in the SEs and CEs:

`/opt/lcg/libexec/lcg-user-configuration`

```
**********************************************************************
DESCRIPTION
This script is intended to provide the user with a tool able to include
attribute values related to the GlueService. This script is interactive
and the required values will be passed by you through the screen.
WARNING: ALL VALUES ARE MANDATORY. Some fields must be integer values.
These are announced
**********************************************************************
Asking now for the values of the attributes:
Introduce the GlueServiceURI
(your value)
Introduce the GlueServiceType
(your value)
```

This will be tested during the hands-on session

Information System Tools

Just wait maximal 2 minutes to refresh the BDII. Your entry is there

But… what has happened behind?

→ Under /opt/lcg/var a **GlueService.ldif$$** has just been created. It has already a ldif syntax and contains your new entry

```
dn: GlueServiceURI=<your_value>,Mds-Vo-name=local,o=grid
objectClass: GlueService
objectClass: GlueSchemaVersion
GlueServiceURI: <your_value>
GlueServiceAccessPointURL: <your_value>
GlueServiceType: <your_value>
GlueServicePrimaryOwnerName: <your_value>
GlueServicePrimaryOwnerContact: <your_value>
GlueServicePrimaryHostingOrganization: <your_value>
GlueServiceMajorVersion: <your_value>
GlueServiceMinorVersion: <your_value>
GlueServiceAccessControlRule: <your_value>
GlueServiceInformationServiceURL: <your_value>
GlueServiceStatus: <your_value>
GlueSchemaVersionMajor: <your_value>
GlueSchemaVersionMinor: <your_value>
```

Information System Tools

- The file /opt/lcg/var/lcg-info-generic-user.conf has been modified to include just one line:

```
provider_script=/opt/lcg/libexec/lcg-info-user -file
/opt/lcg/var/GlueService.ldif$$
```

Cat $file

- The system script /opt/lcg/sbin/lcg-info-generic-config runs the new file lcg-info-generic-user.conf. This will include the new configuration
- The system script /opt/lcg/libexec/lcg-info-wrapper will run too

```
#!/bin/sh
/opt/lcg/libexec/lcg-info-generic /opt/lcg/var/lcg-info-generic-user.conf
/opt/lcg/libexec/lcg-info-user -file /opt/lcg/var/GlueService.ldif$$
```

New line

Always there

R-GMA: New System

Why a new system?

Disadvantages of the old system:

¤ LDAP does not allow to query information from different entries

¤ MDS is not flexible enough to allow for dynamic publication of data from user applications

Advantages of the new system:

¤ R-GMA is quite flexible and allows cross queries between different entries

¤ Anyone can introduce new information in the system in a very easy way

¤ It is quite dynamic with new Producers of information being notified by existing Consumers

R-GMA: Characteristics

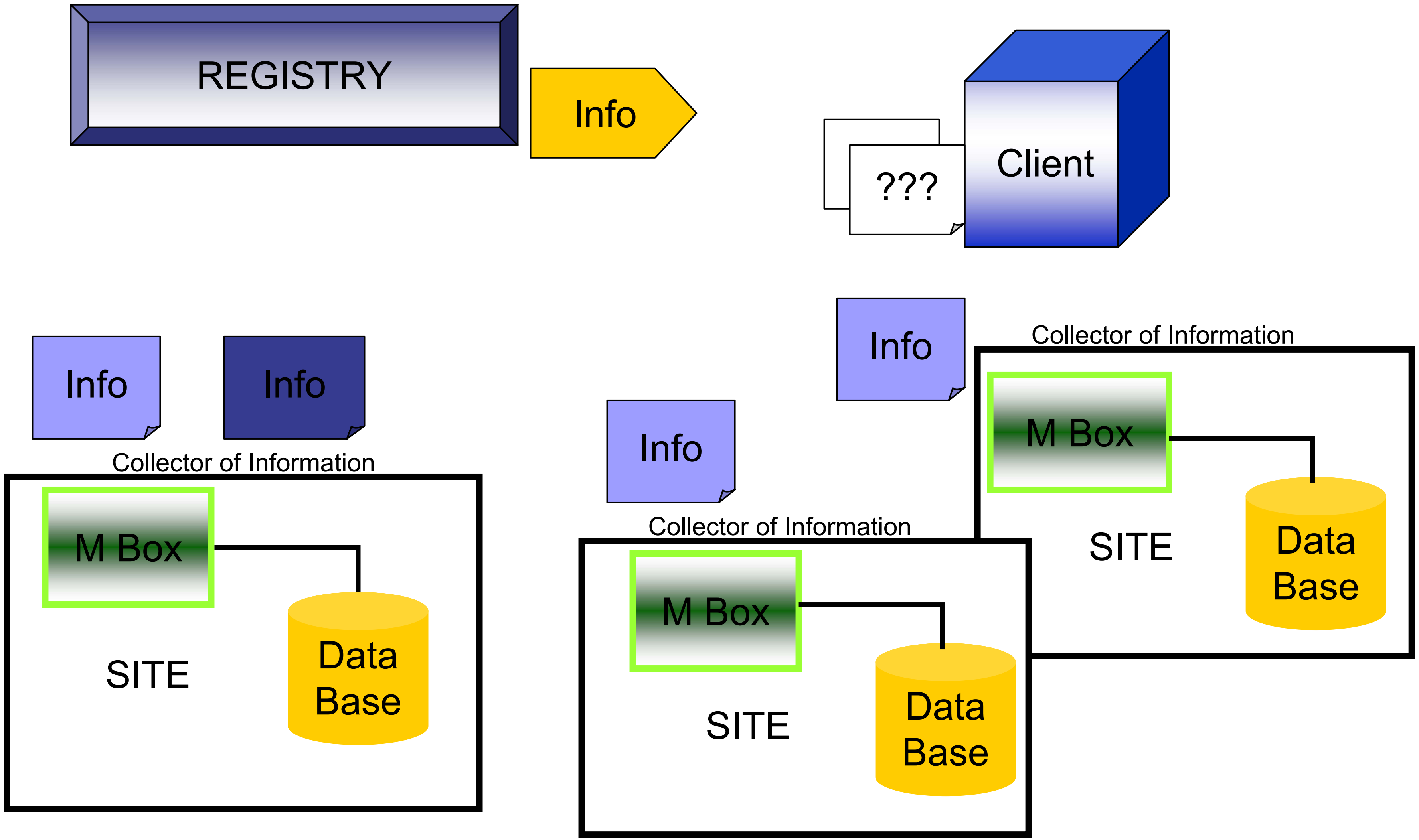
GMA (Grid Monitoring Architecture)

- From GGF (Global Grid Forum)
- Very simple; it does not define:
  - → Data model
  - → Data transfer mechanism
  - → Registry implementation

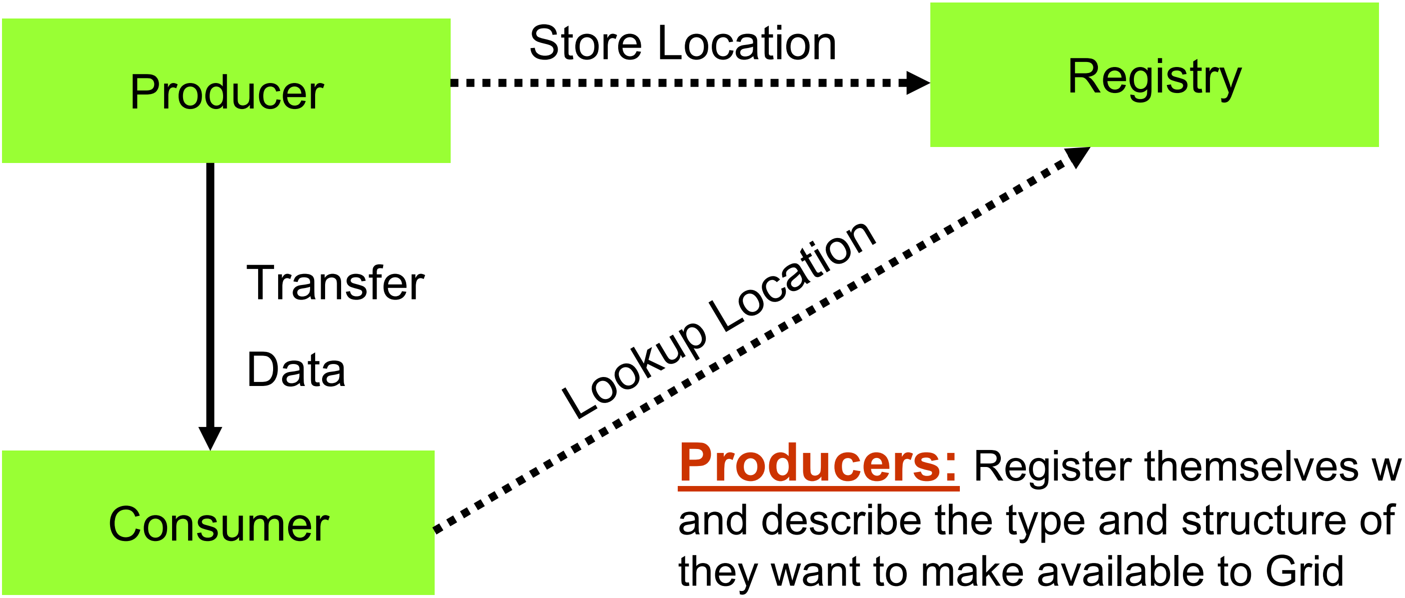
R-GMA (Relational GMA): Relational implementation

- Powerful data model and query language
- All data modeled as tables
- SQL as query language. It can express most queries in one expression
- You have a Relational DB for each VO

R-GMA: Design


REGISTRY
Info
???
Client
Info
Info
Collector of Information
M Box
SITE
Data Base
Info
Info
Collector of Information
M Box
SITE
Data Base
Collector of Information
M Box
SITE
Data Base

R-GMA Architecture



Producers: Register themselves with the Registry and describe the type and structure of the information they want to make available to Grid

Consumers: Query the Registry to find out the information available and locate Producers which provide such information. They can connect directly the Producers

Registry: General collector, its arrow line represents the main flow of data

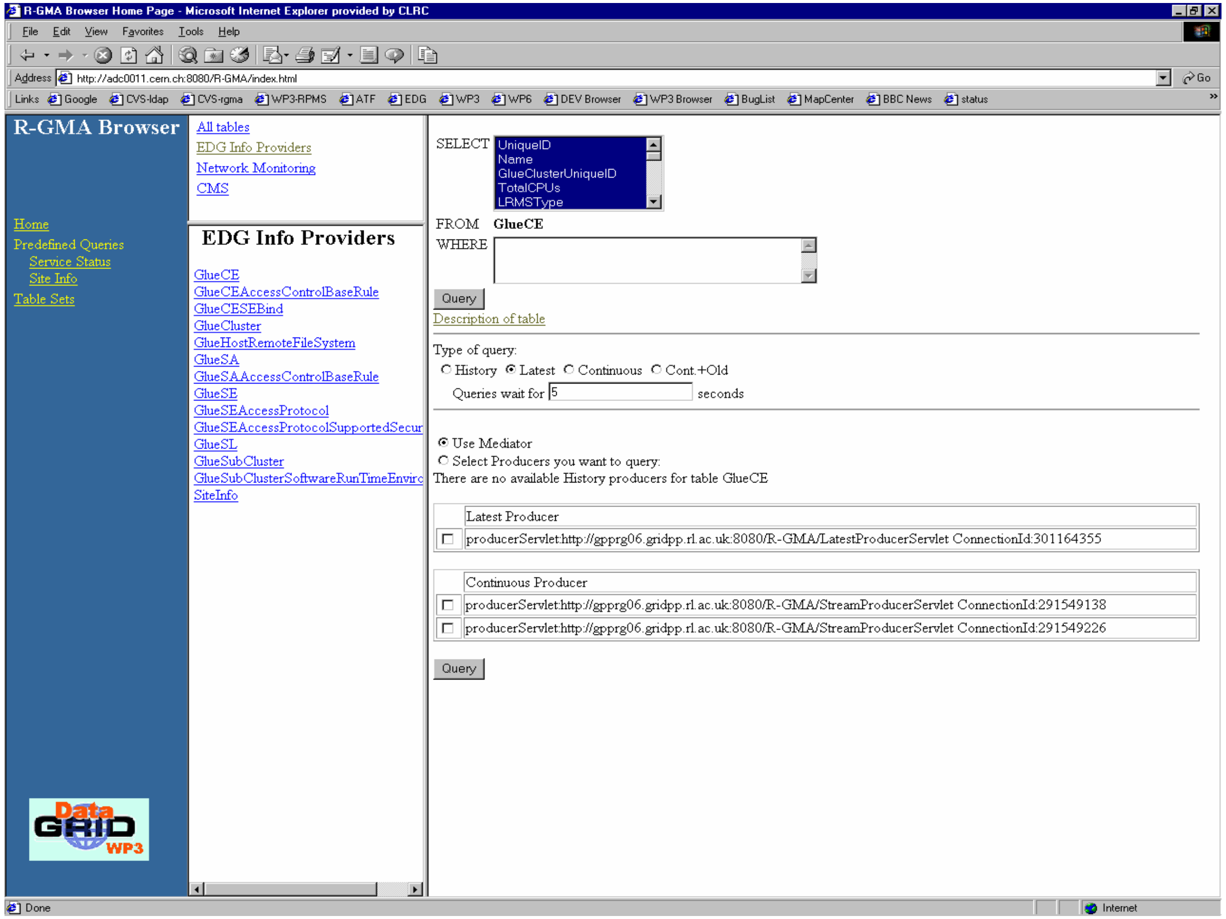
R-GMA tools: Browser

Enabling Grids for E-science in Europe

The user can retrieve the R-GMA information via the browser servlet

http://lcgic01.gridpp.rl.ac.uk:8080/R-GMA/index.html

It shows the schema, the registered producers and allows simple queries


R-GMA Browser Home Page - Microsoft Internet Explorer provided by CLRC
File Edit View Favorites Tools Help
Address
Go
Links Google CVS-ldap CVS-rgma WP3-RPMS ATF EDG WP3 WP6 DEV Browser WP3 Browser BugList MapCenter BBC News status
Data GRID WP3
Done
Internet

Address: http://adc0011.cern.ch:8080/R-GMA/index.html

R-GMA Browser

Home
Predefined Queries
  Service Status
  Site Info
Table Sets

All tables
EDG Info Providers
Network Monitoring
CMS

EDG Info Providers

GlueCE
GlueCEAccessControlBaseRule
GlueCESEBind
GlueCluster
GlueHostRemoteFileSystem
GlueSA
GlueSAAccessControlBaseRule
GlueSE
GlueSEAccessProtocol
GlueSEAccessProtocolSupportedSecur
GlueSL
GlueSubCluster
GlueSubClusterSoftwareRunTimeEnvirc
SiteInfo

SELECT UniqueID, Name, GlueClusterUniqueID, TotalCPUs, LRMSType

FROM **GlueCE**

WHERE

Query

Description of table

Type of query:

○ History ● Latest ○ Continuous ○ Cont.+Old

Queries wait for 5 seconds

● Use Mediator
○ Select Producers you want to query:

There are no available History producers for table GlueCE

	Latest Producer
□	producerServlet:http://gpprg06.gridpp.rl.ac.uk:8080/R-GMA/LatestProducerServlet ConnectionId:301164355

	Continuous Producer
□	producerServlet:http://gpprg06.gridpp.rl.ac.uk:8080/R-GMA/StreamProducerServlet ConnectionId:291549138
□	producerServlet:http://gpprg06.gridpp.rl.ac.uk:8080/R-GMA/StreamProducerServlet ConnectionId:291549226

Query

R-GMA: APIs

General R-GMA documentation can be found in:

http://hepunx.rl.ac.uk/edg/wp3/

R-GMA APIs are available in C, C++, and Java

Quite complete APIs. They are described in:

http://hepunx.rl.ac.uk/edg/wp3/documentation/doc/api/c/index.html

http://hepunx.rl.ac.uk/edg/wp3/documentation/doc/api/cpp/index.html

http://hepunx.rl.ac.uk/edg/wp3/documentation/doc/api/java/index.html

edg-rgma: Virtual Database

- Recently set up in LCG-2/EGEE
- You can make with some of the APIs to produce or retrieve information
- Make `edg-rgma -c help` to retrieve more information

```
$ edg-rgma

rgma> latest select sitename,sysAdminContact from SiteInfo;

+------------------------+---------------------------------------------------+
| sitename               | sysAdminContact                                   |
+------------------------+---------------------------------------------------+
| IC-LCG2                | b.macevoy@imperial.ac.uk                          |
| LCGCERTTB4             | Piera.Bettini@cern.ch                             |
| Uni-Wuppertal          | lcg-admin@physik.uni-wuppertal.de                 |
| RAL-LCG2               | lcg-support@gridpp.rl.ac.uk                       |
| nikhef.nl              | grid-support-admin@nikhef.nl                      |
+-----------------------+---------------------------------------------------+
5 Rows in set
```

R-GMA: Classes

⌂ The headers are visible in your UI under:

`/opt/edg/include/info`

⌂ Those directly used in this tutorial are:

- **Consumer.hh**
  - ¤ Executes a SQL query to return tuples to the user
  - ¤ Able to find the producers of information
- **ResultSet.hh**
  - ¤ Handle the results strings
- **StreamProducer.hh**
  - ¤ Register a table when it is created and subsequently to publish information

LCG APIS from R-GMA

- **InfoFromRGMA:** Parallel development to `InfoFromLDAP`

```
> lcg-is-search-rgma  <your_file>
```

InfoFromRGMA.h

```
#include "AllInfoRGMA.h"

class InfoFromRGMA: public AllInfo{
public:
InfoFromRGMA();
~InfoFromRGMA();
virtual void query(char*);
}
```

This will be tested during the hands-on session

LCG APIs from R-GMA

InfoFromRGMA.cpp

```
#include “Consumer.hh”
#include “ResultSet.hh”
#include “InfoFromRGMA.h”

void InfoFromRGMA::query(char* file){

char buff[1024];
std::ifstream sqlFile(file,std::ios::in);
std::ostringstream os;
while (!sqlFile.getline(buff,sizeof(buff)).eof() ){
os << buff << ‘ ’;
}
sqlFile.close();

edg::info::Consumer myConsumer(os.str(),edg::info::Consumer::
             CONTINUOUS);
```

reading the file

Constructing a consumer

LCG APIs from R-GMA

InfoFromRGMA.cpp

```
edg::info::TimeInterval Timeout(60);
myConsumer.start(Timeout);
while(myConsumer.isExecuting()){
sleep(2);
}
edg::info::ResultSet resultSet = myConsumer.popIfPossible();
std::cout<<ResultSet:\n”<<resultSet.toString().c_str()<<std::endl;
myConsumer.close();
};

extern “C” AllInfoRGMA create(){ return new InfoFromRGMA;}

extern “C” void destroy(AllInfoRGMA* a){ delete a;}
```

initiate streaming with each Producer

getting results and printing them by screen

dlopen

LCG APIS from R-GMA

♠ InfoToRGMA:

You have the power, You create the information

```
> lcg-is-search-rgma  <your_file>
```

InfoToRGMA.h

```
#include “AllInfoRGMA.h”

class InfoToRGMA: public AllInfo{
public:
InfoToRGMA();
~InfoToRGMA();
virtual void add(char*);
}
```

This will be tested during the hands-on session

LCG APIS from R-GMA

In this package a configuration file should be included with the following data:

1. The name of the table where your info is included
2. Your information

Example of Configuration File

```
theTABLE = userTable

theREQUEST = INSERT INTO userTable (userID, aString,
anInt, MeasurementDate,MeasurementTime) VALUES
(‘test’,’producertest’,5.18,32,’2004-10-19’,’18:59:00’)
```

This will be tested during the hands-on session

LCG APIs from R-GMA

InfoToRGMA.cpp

```
#include “StreamProducer.hh”
#include “ConfigBuffer.hh”
#include “InfoToRGMA.h”

void InfoFromRGMA::add(char* file){

string thefile = file;
configBuffer *theconfigfile = new ConfigBuffer(thefile);
std::string table = theconfigfile->get_attribute_value(“theTABLE”);
std::string request = theconfigfile->get_attribute_value(“theREQUEST”);

edg::info::StreamProducer myProducer;
myProducer.declareTable(table,””);
myProducer.setTerminationInterval(edg::info::TimeInterval(1200));
myProducer.setMinRetentionPeriod(edg::info::TimeInterval(600));
myProducer.insert(request);
```

The future in LCG-2

LDAP can be considered the past in LCG

A new protocol has been deployed based on web services: **R-GMA**

Problem:

- Each protocol has its own schema, its own technology
- Users and developers have to adapt their software and applications to the new protocols

Questions:

- What to do with the already existing tools?
- What to do in the future to if a new protocol is arriving?

EIS
experiment integration and support

Solution:

A new interface able to globalize all protocols with just one schema and just one query language

General Features of the Interface

Characteristics:

1. The User Applications see just one interface

2. The query language and data model are included

4. The query and schema are syntactically and semantically translated internally in a transparent manner

User Requirements:

1. Perform the query via SQL

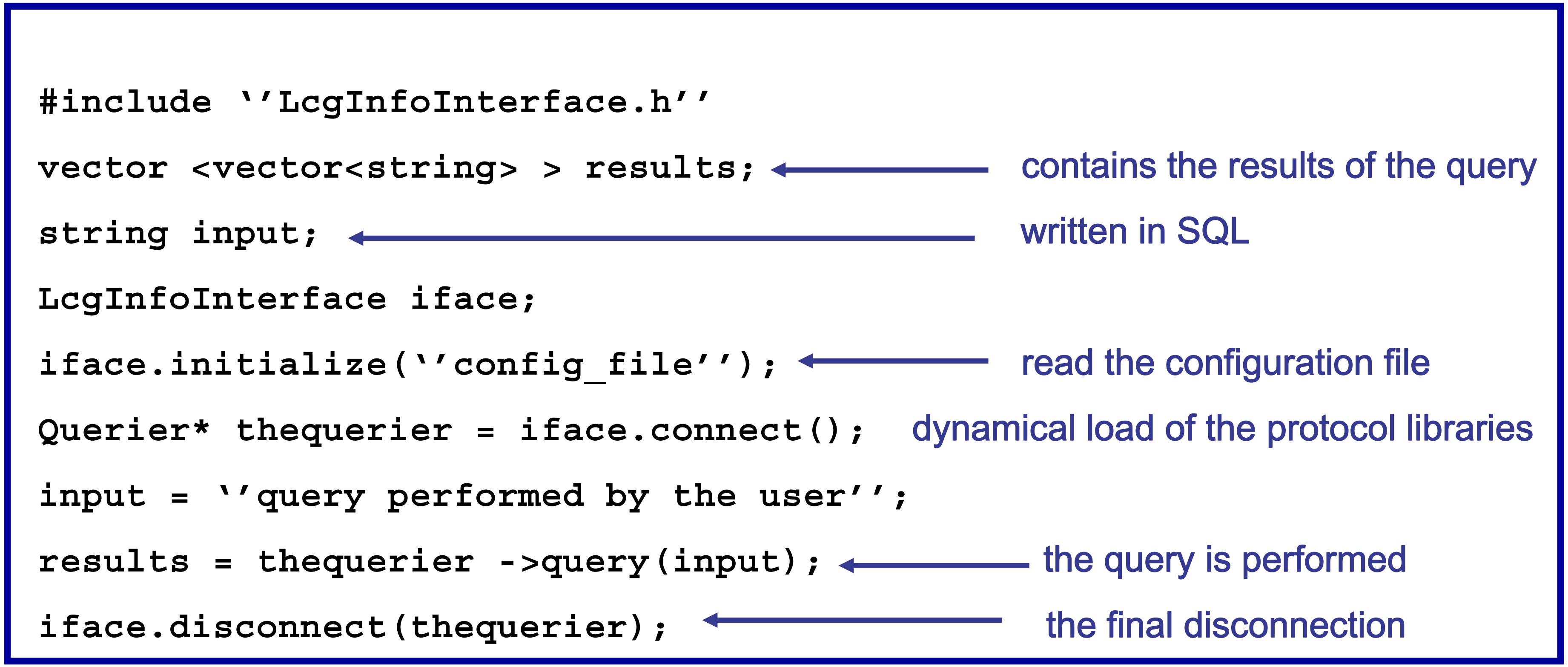
2. Configuration file to include the protocol and additional parameters mandatory for each protocol

3. Use the canonical schema

General Interface Tool

Enabling Grids for E-science in Europe

Your user Application can look like as:



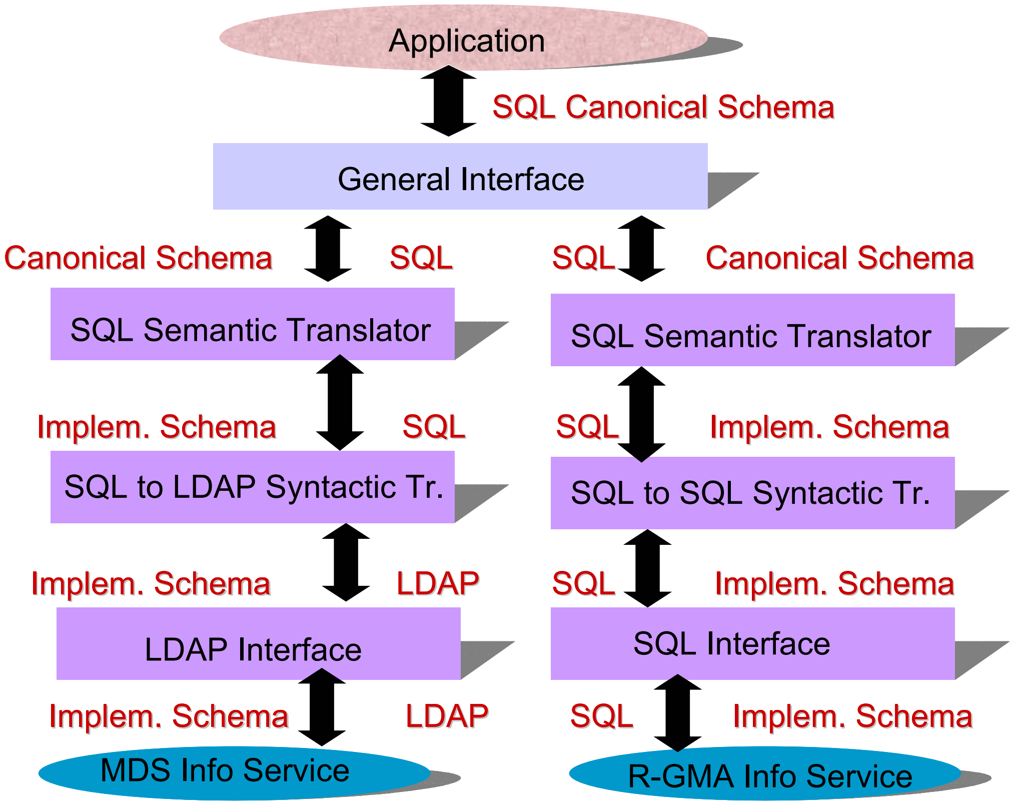
```
#include ‘’LcgInfoInterface.h’’
vector <vector<string> > results;
string input;
LcgInfoInterface iface;
iface.initialize(‘’config_file’’);
Querier* thequerier = iface.connect();
input = ‘’query performed by the user’’;
results = thequerier ->query(input);
iface.disconnect(thequerier);
```

contains the results of the query
written in SQL
read the configuration file
dynamical load of the protocol libraries
the query is performed
the final disconnection

http://grid-deployment.web.cern.ch/grid-deployment/eis/docs/LcgInfoInterface/namespaces.html
http://grid-deployment.web.cern.ch/grid-deployment/eis/docs/LcgInfoInterface/LcgInfoInterface_refman.pdf

General Interface Tool

General schema of the API



Application

SQL Canonical Schema

General Interface

Canonical Schema

SQL

SQL

Canonical Schema

SQL Semantic Translator

SQL Semantic Translator

Implem. Schema

SQL

SQL

Implem. Schema

SQL to LDAP Syntactic Tr.

SQL to SQL Syntactic Tr.

Implem. Schema

LDAP

SQL

Implem. Schema

LDAP Interface

SQL Interface

Implem. Schema

LDAP

SQL

Implem. Schema

MDS Info Service

R-GMA Info Service

Some examples

```
SELECT StorageServiceUniqueID
ComputingElementUniqueID FROM Glue.Bind

lxb0707.cern.ch
lxb0706.cern.ch:2119/jobmanager-pbs-long

lxb0710.cern.ch
lxb0706.cern.ch:2119/jobmanager-pbs-long

lxb0707.cern.ch
lxb0706.cern.ch:2119/jobmanager-pbs-short

lxb0710.cern.ch
lxb0706.cern.ch:2119/jobmanager-pbs-short

castorgridtest.cern.ch
lxb0706.cern.ch:2119/jobmanager-pbs-long

oplapro12.cern.ch
lxb0706.cern.ch:2119/jobmanager-pbs-long
```

and now... Let's have fun!!!

Hands-on session

The Hands-on session includes two types of exercises:

1. Those which will be just shown because they require sgm (lcg-ManageVOTag) or root (lcg-user-configuration) privileges

2. The following needs your work
   - ◙ C++ APIs and Perl scripts
   - ◙ You just have to work with the C++ APIs and we provide you with the needed Makefile, libraries and headers
   - ◙ **Just concentrate on the C++ applications**
   - ◙ The Perl scripts are lcg-utilities which use the C++ APIs. Use them to get familiar with the lcg-utilities

3. General Remarks:
   - ¤ Work in couples, it will be easier
   - ¤ Do not hesitate to ask questions and have a look at the solutions each time you get stack
   - ¤ Ask your tutors in case of problems

Generalities for all APIs

Where to find the sources?

Headers: `/opt/lcg/include`
Libraries: `/opt/lcg/lib`
Executables: `/opt/lcg/bin`
At your home directory you have already installed : `IS_exercises/ldap`
`/rgma`

1. Makefiles:

```
ldap/Makefiles/Makefile_search_ldap
ldap/Makefiles/Makefile_general_ldap
rgma/Makefiles/Makefile_search_rgma
rgma/Makefiles/Makefile_add_rgma
```

2.Templates:

```
ldap/Templates/Template_search_ldap.cpp
ldap/Templates/Template_general_ldap.cpp
rgma/Templates/Template_search_rgma.cpp
rgma/Templates/Template_add_rgma.cpp
```

3. Solutions:

```
ldap/Solutions/Solution_search_ldap.cpp
ldap/Solutions/Solution_general_ldap.cpp
rgma/Solutions/Solution_search_rgma.cpp
rgma/Solutions/Solution_add_rgma.cpp
```

LDAP Exercises

1. Use the liblcg-info-api-ldap library

We propose you to generate a main program which:

¤ Write an application that requires the following arguments: host, port, filter, attribute(s). It loads dynamically this library (dlopen). Then it invokes the “query” method of InfoFromLDAP.h and prints out the resultof the user query on the screen.

¤ The query method definition is in InfoFromLDAP.h

¤ Have a look at the lcg-infosites script. It uses the executable generated by the solution provided.

¤ If you have time copy this script (placed in /opt/lcg/bin) in a local area and replace the lcg-is-search executable with your executable and try to run it

¤ Compare results with those obtained from lcg-infosites

LDAP Exercises

3. Use the liblcg-info-search-rgma library

Try to generate a main program which:

1. Passes a file including the query (written in SQL)
2. Makes a dynamic load of this library (dlopen)
3. Invokes the general query method of the InfoFromRGMA.h

Remarks:

¤ The Solutions directory includes some examples of SQL queries

LDAP Exercises

3. Use the liblcg-info-add-rgma library

You will have to generate a main program which:

1. Passes a file including the request (written in SQL)
2. Makes a dynamical download of this library (dlopen)
3. Invokes the general query method of the InfoToRGMA.h

Remarks:

¤ The Solutions directory includes some examples of SQL queries