Enabling Grids for E-science in Europe

www.eu-egee.org

Biomed Application Developer's Course
6th October 2004

Information Systems

Flavia Donno
Section Leader for
LCG Experiment Integration and Support
CERN IT

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

Contents

- Requirements of a Grid information and monitoring system
- The LCG Resource Information system
- Job Monitoring services
- Grid "health" monitoring

Requirements of a Grid Information & Monitoring Service

- Need information to know the Grid out there
  - information on grid resources and services
  - information on jobs

- Dynamic distributed environment
  - insertion and removal of information sources
  - haphazard LAN/WAN network connectivity
  - fine-grained access control (for accounting, jobs, privacy)

- The system must allow new types of information to be used

Current Situation

No dynamic, complete information system available today

- Resource information directory
  - MDS – *Monitoring and Discovery Service*
  - BDII – *Berkeley Database Information Index*
  - GLUE and Globus Schema
- Dynamic job information
  - R-GMA – *Relational Grid Monitoring Architecture*
- Probes
  - test job analysis

Globus MDS enhanced with BDIIs

- LCG-2 currently uses
  GT Monitoring and Discovery Service (MDS) architecture together with Berkley Database Information Indexes (BDII)

- The information system is built on *LDAP*
  Light-weight Directory Access Protocol
- A Schema describes the attributes and the types of the attributes associated with data objects
- Example: **GlueSiteInfo**

```
dataGridVersion: LCG-2_0_0
installationDate: 20040413110OZ
objectClass: SiteInfo
siteName: nikhef.nl
siteSecurityContact: grid-support-admin@nikhef.nl
sysAdminContact: grid-support-admin@nikhef.nl
userSupportContact: grid-support-admin@nikhef.nl
```

is not flexible!

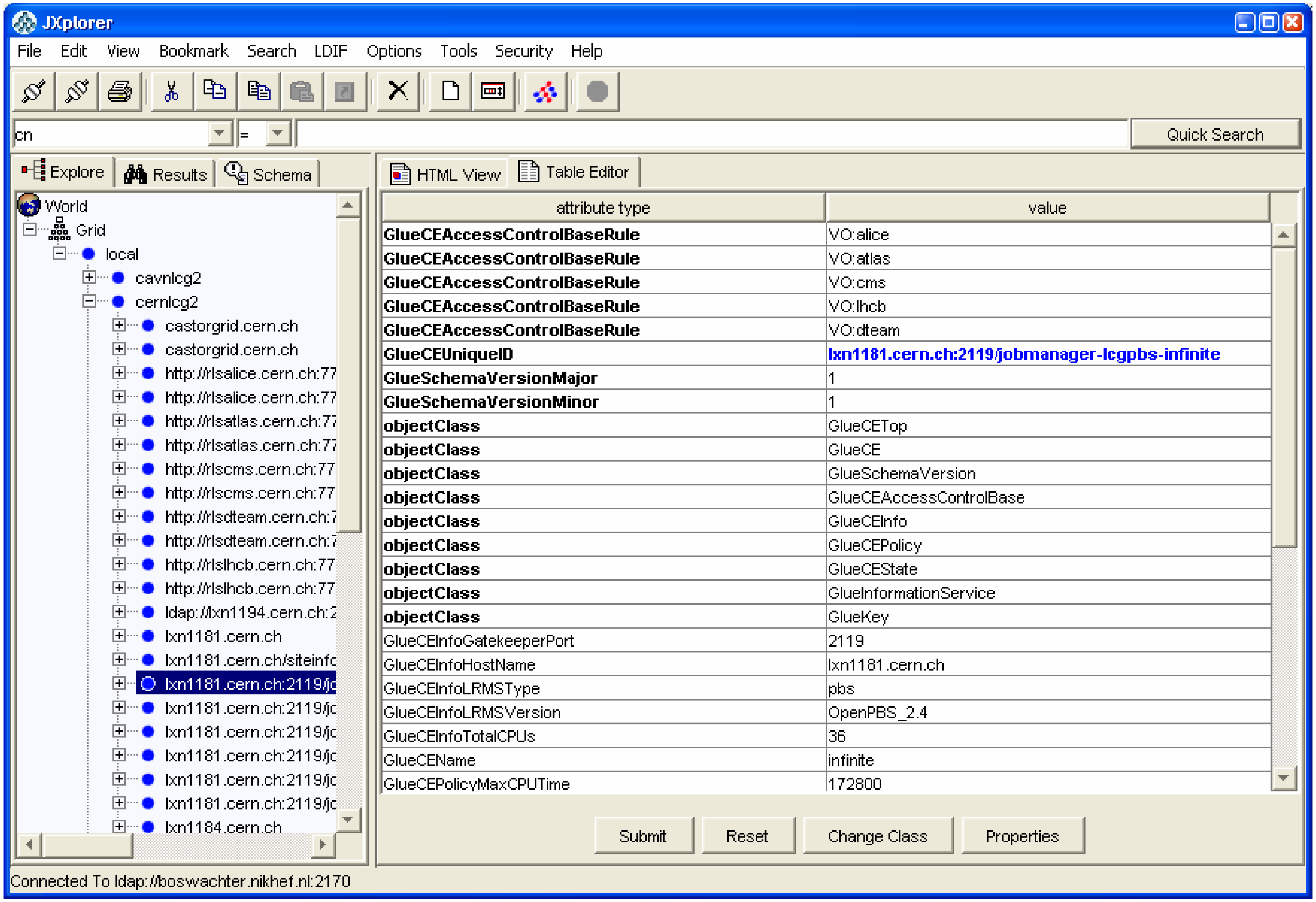
LDAP hierarchy

- Lightweight Directory Assess Protocol (LDAP) offers a hierarchical view of information

- The entries are arranged in a Directory Information Tree (DIT)

- Resources (computers, storage, …) each publish their part in this tree

An LDAP Hierarchy



- World
  - Grid
    - local
      - cavnlcg2
      - cernlcg2
        - castorgrid.cern.ch
        - castorgrid.cern.ch
        - http://rlsalice.cern.ch:77
        - http://rlsalice.cern.ch:77
        - http://rlsatlas.cern.ch:77
        - http://rlsatlas.cern.ch:77
        - http://rlscms.cern.ch:77
        - http://rlscms.cern.ch:77
        - http://rlsdteam.cern.ch:7
        - http://rlsdteam.cern.ch:7
        - http://rlslhcb.cern.ch:77
        - http://rlslhcb.cern.ch:77
        - ldap://lxn1194.cern.ch:2
        - lxn1181.cern.ch
        - lxn1181.cern.ch/siteinfc
        - lxn1181.cern.ch:2119/jc
        - lxn1181.cern.ch:2119/jc
        - lxn1181.cern.ch:2119/jc
        - lxn1181.cern.ch:2119/jc
        - lxn1181.cern.ch:2119/jc
        - lxn1181.cern.ch:2119/jc
        - lxn1184.cern.ch

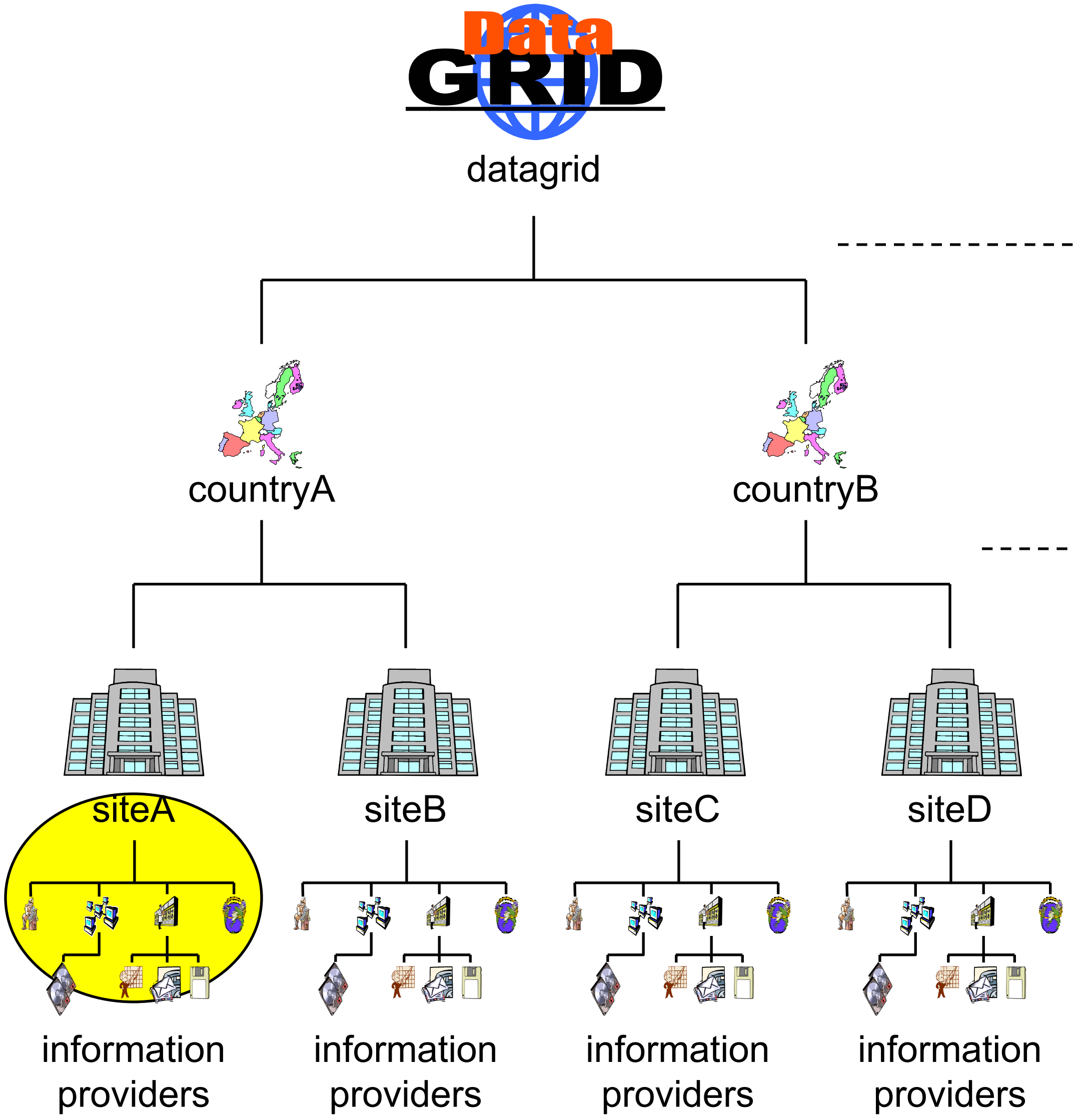
attribute type	value
GlueCEAccessControlBaseRule	VO:alice
GlueCEAccessControlBaseRule	VO:atlas
GlueCEAccessControlBaseRule	VO:cms
GlueCEAccessControlBaseRule	VO:lhcb
GlueCEAccessControlBaseRule	VO:dteam
GlueCEUniqueID	**lxn1181.cern.ch:2119/jobmanager-lcgpbs-infinite**
GlueSchemaVersionMajor	1
GlueSchemaVersionMinor	1
objectClass	GlueCETop
objectClass	GlueCE
objectClass	GlueSchemaVersion
objectClass	GlueCEAccessControlBase
objectClass	GlueCEInfo
objectClass	GlueCEPolicy
objectClass	GlueCEState
objectClass	GlueInformationService
objectClass	GlueKey
GlueCEInfoGatekeeperPort	2119
GlueCEInfoHostName	lxn1181.cern.ch
GlueCEInfoLRMSType	pbs
GlueCEInfoLRMSVersion	OpenPBS_2.4
GlueCEInfoTotalCPUs	36
GlueCEName	infinite
GlueCEPolicyMaxCPUTime	172800

Connected To ldap://boswachter.nikhef.nl:2170

MDS GRISs & GIISs

- Information providers are scripts that generate LDIF-formatted info.
  - Information is cached by the server to improve performance
- The MDS Grid Resource Information Service (GRIS) invokes the Information Providers as an OpenLDAP backend
- The GRIS soft-registers with an Index Server (GIIS) – queries to a GIIS get forwarded to the GRISes
- The GIIS can then act as a single point of contact for a number of resources
  - A GIIS may represent a site, country, virtual organization, etc.
- In turn a GIIS may register with another GIIS

EDG GRIS/GIIS Hierarchy



- There is a top level datagrid GIIS to which all of the country GIISs register

- Each country has a GIIS to which all of the site GIISs register

- Each Site has a Grid Information Index Server (GIIS) which acts as a single point of contact for all of the sites resources. The GRISs register with their site GIIS

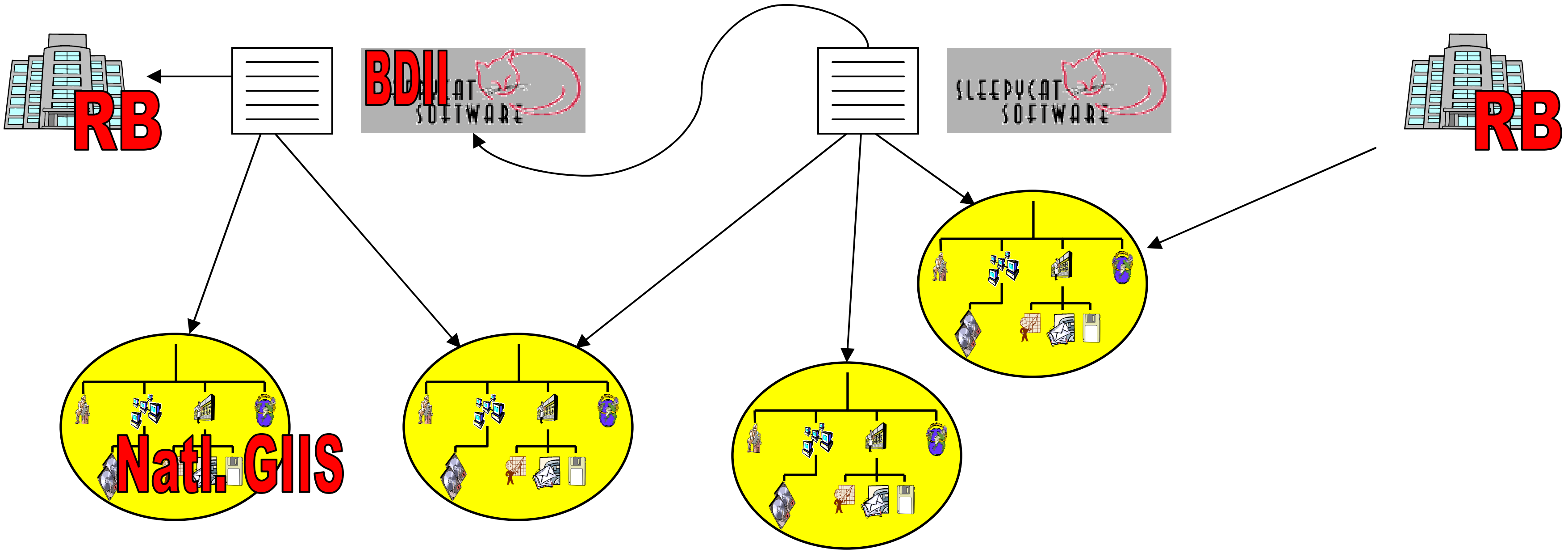
- Information providers publish information to a local LDAP server known as a Grid Resource Information Server (GRIS)

Adding stability and speed: BDII

- The GRIS/GIIS system can answer 1query/15min

  LDAP designed for static, slow changing information

- Cache information statically in DBM files (BDII)
- Cache is transparent: same OpenLDAP, same DIT layout
- Script queries set of GIISs periodically and stores in DBM
- GIISs with amnesia are ignored

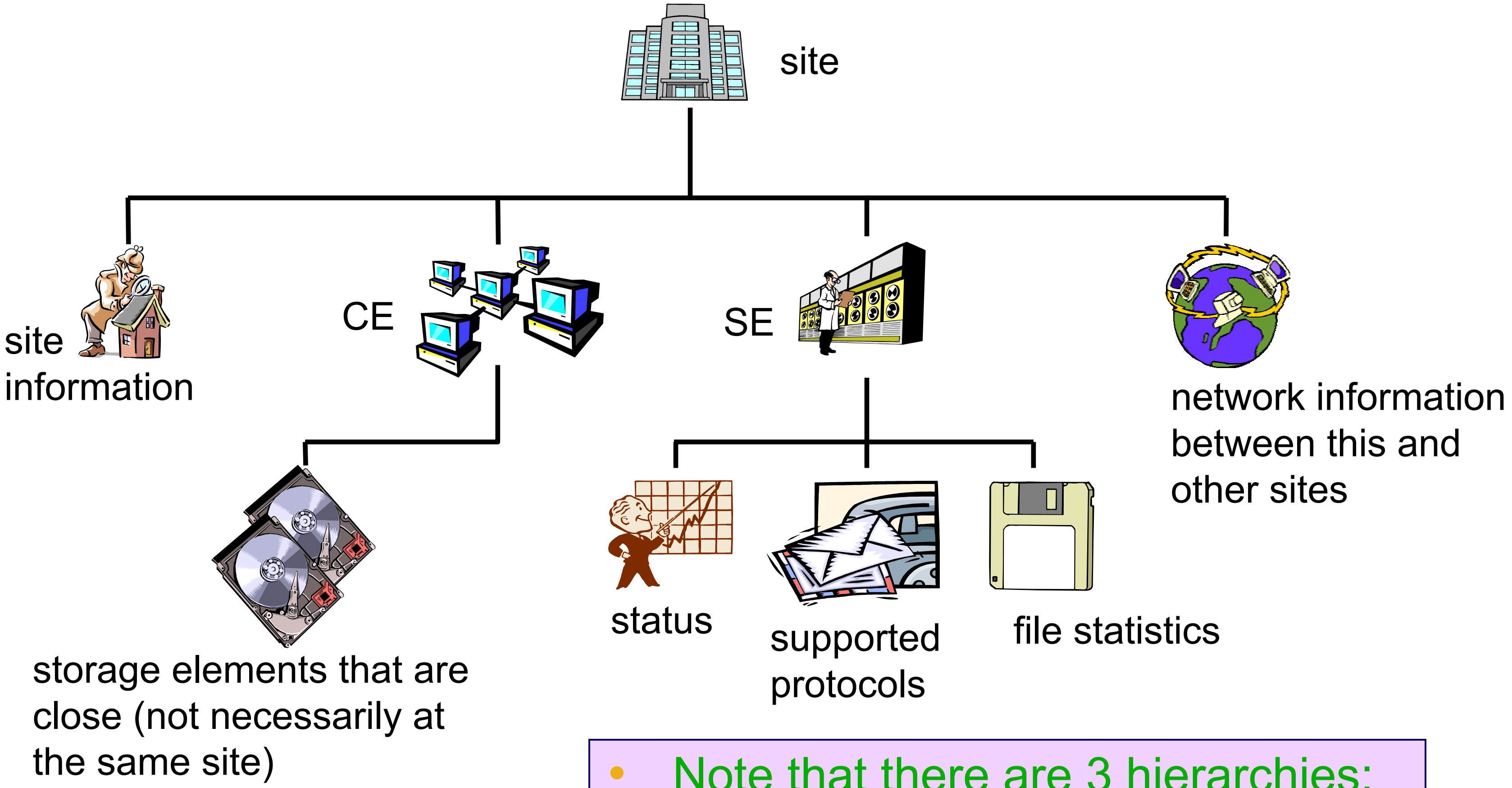


EDG Information Providers

Enabling Grids for E-science in Europe

- The EDG have produced information providers:
  - Site information
  - The Computing Element
  - The Storage Element
  - Network Monitoring
- Publication according to predefined GLUE schema
- All of the information is dynamic,
  they have a time stamp and a time to live
  (used by the cache mechanism) associated with them

EDG Information Providers & the Directory Information Tree


site
site information
CE
SE
network information between this and other sites
storage elements that are close (not necessarily at the same site)
status
supported protocols
file statistics

- Note that there are 3 hierarchies:
  - The GIIS/GRIS structure
  - The DIT
  - The BDII linkage

Querying the Information & Monitoring Service

- Queries can be posed to the current Information and Monitoring Service using LDAP search commands:

```
ldapsearch\
 -x\                                     “simple” authentication
 -H ldap://boswachter.nikhef.nl:2170\    uniform resource identifier
 -b 'Mds-Vo-name=local,o=grid\           base distinguished name for search
 'objectclass=StorageElment‘\            filter
 seId SEsize \                           attributes to be returned
 -s base|one|sub                         scope of the search specifying just
                                              the base object, one-level or the
                                              complete subtree
```

Resource Brokering

- The RB uses the MDS/BDII information for brokering
- Key information:
  - GlueCEApplicationRuntimeEnvironment tags
  - TotalCPUs, FreeCPUs
  - EstimatedTraversalTime (ETT)
  - Network Cost

- With each RB, a local BDII is deployed
- can index additional local resources

- Information requirements from JDL are to be met

The LDAP APIs

- C and C++ APIs available from OpenLDAP (contrib/ldapcpp)
- Allow for synchronous and asynchronous operations, add, remove, query entries
- API description can be found:

  http://www.openldap.org/software/man.cgi?query=ldap
- Also available from the OpenLDAP Project:
  - **JLDAP - LDAP Class Libraries for Java**
    ***contributed by* Novell**
  - **JDBC-LDAP - Java JDBC - LDAP Bridge Driver**
    ***contributed by* Octet String**
- Wrappers exists in LCG middleware,
  however they are not directly exposed to users.

The LCG-2 C++ Info LDAP APIs

- C++ APIs available from LCG EIS
- API description still not available

  You can check the source code in CVS (TAG: v1_1_4):

  http://isscvs.cern.ch:8180/cgi-bin/cvsweb.cgi/lcg-info-api/ldap/?cvsroot=lcgware
- The APIs are included in LCG-2_2_0
- Only query functionality available for the moment
- Some work in progress to provide plug-ins and technology independent APIs. Check CHEP2004:

  http://indico.cern.ch/contributionDisplay.py?contribId=114&sessionId=23&confId=0

The LCG-2 C++ Info APIs

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

```
#include <dlfcn.h>
#include <stdio.h>
#include <iostream>
#include <strstream>
#include <string>
#include <vector>
#include <iterator>

#include “lcg-info-api-ldap/InfoFromLDAP.h"
#include “lcg-info-api-ldap/AllInfoLDAP.h"

#include "stdlib.h"
#include "ltdl.h"

using namespace std;
using namespace LcgInfo;

int main( int argc, char* argv[] )
{

bool errors = false;
string filter, attribute;
vector<string> attributes;

[…]
```

The LCG-2 C++ Info APIs

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

void *InfoFromLDAP = dlopen(lib_loc,RTLD_LAZY); I
f(!InfoFromLDAP)
{
   cout<<"Cannot load library: "<<dlerror() <<endl;
  return 1;
}
create_t* create_infoldap = (create_t*)dlsym(InfoFromLDAP,"create");
destroy_t* destroy_infoldap =
(destroy_t*)dlsym(InfoFromLDAP,"destroy");

if (!create_infoldap || !destroy_infoldap)
{
  cout<<"Cannot load symbols: "<<dlerror()<<endl;
  return 1;
}
AllInfoLDAP *ldapinfo = create_infoldap();
ConfigBuffer *conf = new ConfigBuffer("/opt/lcg/etc/lcginfo.conf");

ldapinfo->setConfig(*conf);

std::vector< vector<std::string> > myvec2;
std::vector< vector<std::string> >::iterator iter;

myvec2 = ldapinfo->query(filter,attributes);
for (iter=myvec2.begin();iter!=myvec2.end();iter++)
{
   std::cout << *iter << std::endl;
}
destroy_infoldap(ldapinfo);
dlclose(InfoFromLDAP);
#endif
```

```
% cat /opt/lcg/etc/lgcinfo.conf

Host = lxb0705.cern.ch
Port = 2170
Timeout = 30
Base_dn = "mds-vo-name=local,o=grid"
```

Dinamically loadable library

The LCG-2 C++ Info APIs

```
% cat compile_info_api

#!/bin/sh
CC= /opt/gcc-3.2.2/bin/gcc
LCG_LOCATION=/opt/lcg
GLOBUS_LOCATION=/opt/globus
GLOBUS_FLAVOR=gcc32dbgpthr
$CC -I${LCG_LOCATION}/include \
        -I${GLOBUS_LOCATION}/include/${GLOBUS_FLAVOR} ${1}.c \
                                        -L${GLOBUS_LOCATION}/lib \
                                        -lldap _${GLOBUS_FLAVOR} -o ${1}


% ./compile_info_api lcg-is-search
```

Compiling and Linking

experiment integration and support

HANDS ON

The LCG-2 Future Info APIs

```
#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 result of the query

Written in SQL

The configuration file is read

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

Enabling Grids for
E-science in Europe

R-GMA: Monitoring Job Information

Relational - Grid Monitoring Architecture

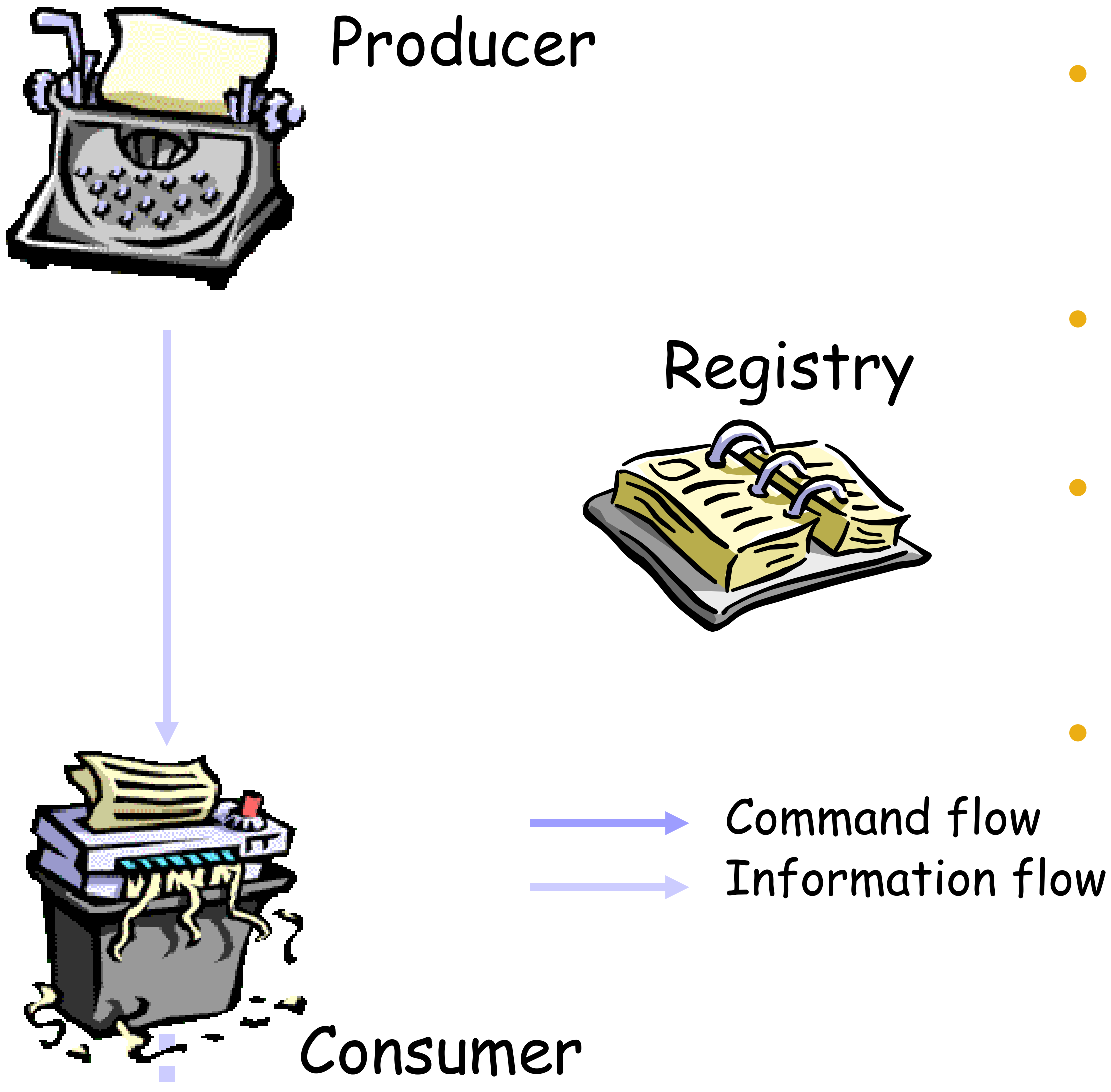
the power of SQL to the Grid Information System

R-GMA: Relational - Grid Monitoring Architecture

- LDAP does not allow queries over different objects
  - I.e. you can only query based on attributes of an object (no “Joins”)
- MDS is not designed for applications to publish their own data
  - It has relatively static descriptions of the data being published – the schema.

- R-GMA is a relational implementation of the Grid Monitoring Architecture (GMA) of the GGF
  - The relational model is very flexible and allows complex queries which make use of information in multiple objects
  - R-GMA provides a means for anyone to publish any information on the Grid – can also do the job of the current MDS
  - It is highly dynamic – with new Producers of information being noticed by existing Consumers

R-GMA
The Consumer Producer Model



- Use the Grid Monitoring Architecture from Global Grid Forum
- A relational implementation
- Applied to both information and monitoring
- Creates impression that you have one RDBMS per Virtual Organization

Relational Approach

- Not a general distributed RDBMS system, but a way to use the relational model in a distributed environment.

- Producers announce: SQL “CREATE TABLE”
  publish: SQL “INSERT”

- Consumers collect: SQL “SELECT”

- The mediator is a component within the Consumer which locates one or more Producers and combines the information as necessary

- Information Catalogue collects pointers to producers

Examples from R-GMA

- Recently set up in LCG-2/EGEE

```
$ 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
```

- For D0 monitoring on EDG:
  3 interlinked job monitoring tables for the Dzero reconstruction

DZero Job Monitoring Tables

Job start table “d0jst4”, written by the job script:

```
rgma> describe d0jst4

Table: d0jst4

+-------------+-------------+------------+-------------+--------------+..-----------------+
| jihash      | jobID       | start_time | site        | command      |.. MeasurementTime |
+-------------+-------------+------------+-------------+--------------+..-----------------+
| VARCHAR(22) | VARCHAR(64) | INT        | VARCHAR(64) | VARCHAR(255) |.. TIME            |
+-------------+-------------+------------+-------------+--------------+..-----------------+
1 Rows in set
```

Combining with Job end table d0jen4 and submission table:

```
$ edg-rgma

rgma> history select
d0jst4.jihash, d0jen4.out_lfn, d0jen4.success_code, d0jen4.end_time-d0jst4.start_time
from d0jst4, d0jen4 where d0jst4.jihash=d0jen4.jihash;

+------------------------+----------------------------------------------------------+----------------------------------+-----------------------------------+
| jihash                 | out_lfn                                                  | success_code                     | d0jen4.end_time-d0jst4.start_time |
+------------------------+----------------------------------------------------------+----------------------------------+-----------------------------------+
| IEECvr4iYcBcsh4jpzuvHA | lfn:reco_all_001.raw_p13.06.01_000.20031218000347.tar.gz | Job completed OK                 | 610                               |
| QcQFuEY6tVvzcsXjvSusFg | lfn:reco_all_039.raw_p13.06.01_000.20040323182717.tar.gz | Job completed OK                 | 1581                              |
…
| PzSKKPipRBGU0WkIJTyl5A | lfn:reco_all_040.raw_p13.06.01_000.20040323235432.tar.gz | Job completed OK                 | 1363                              |
| FURWimGW0Qo+zFu/EyTmzw | lfn:reco_all_042.raw_p13.06.01_000.20040324001803.tar.gz | Job completed OK                 | 1376                              |
+------------------------+----------------------------------------------------------+----------------------------------+-----------------------------------+
```

R-GMA Browser

- Information in R-GMA can easily be browsed via the browser servlet.
- http://lcgic02.gridpp.rl.ac.uk:8080/R-GMA/BrowserServlet
- The browser shows the schema, what producers are registered and allows simple queries to be done.

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

R-GMA APIs example usage

Enabling Grids for E-science in Europe

```
#include "Consumer.hh"
#include "ResultSet.hh"

[…]

int main( int argc, char* argv[] )
{

char buff[1024];
std::ifstream sqlFile(file,std::ios::in);
if (sqlFile.bad())
{
   std::cout<<"ERROR: Error opening file for read"<<std::endl;
}
std::ostringstream os;
while(!sqlFile.getline(buff, sizeof(buff)).eof())
{
   os << buff << ' ';
}
sqlFile.close();
std::cout << os.str() << std::endl;

[…]
```

Read query from file

R-GMA APIs example usage

```
//Connect to the server:
edg::info::Consumer myConsumer(os.str(), edg::info::Consumer::LATEST);

//We pass the query ;LATEST means the latest query

edg::info::TimeInterval Timeout(60); // The definition of the timeout

myConsumer.start(Timeout); // Here we start executing the Consumer's query using a time limit.

while(myConsumer.isExecuting())
{
  sleep(2);
}

// isExecuting() Return all the available pieces of information.

if(myConsumer.hasAborted())
{
   std::cout<<"Consumer query timed-out\n"<<std::endl;
}

// hasAborted() Returns the execution status
edg::info::ResultSet resultSet = myConsumer.popIfPossible();
// popIfPossible() Return up to the next maxCount tuples of information
std::cout<<"ResultSet:\n"<<resultSet.toString().c_str()<<std::endl;

myConsumer.close(); //closes the connection
}
```

Asynchronous query

Summary

- Two main Information System technologies are used in LCG-2: one LDAP based from Globus and one developed by the European DataGrid Project, R-GMA
- The **GLUE** schema is used to describe Grid resource related information
- A coherent technology independent set of APIs is under way
- LDAP C and C++ APIs are available from OpenLDAP
- R-GMA C, C++, and Java APIs are available and documented