



Enabling Grids for
E-science in Europe

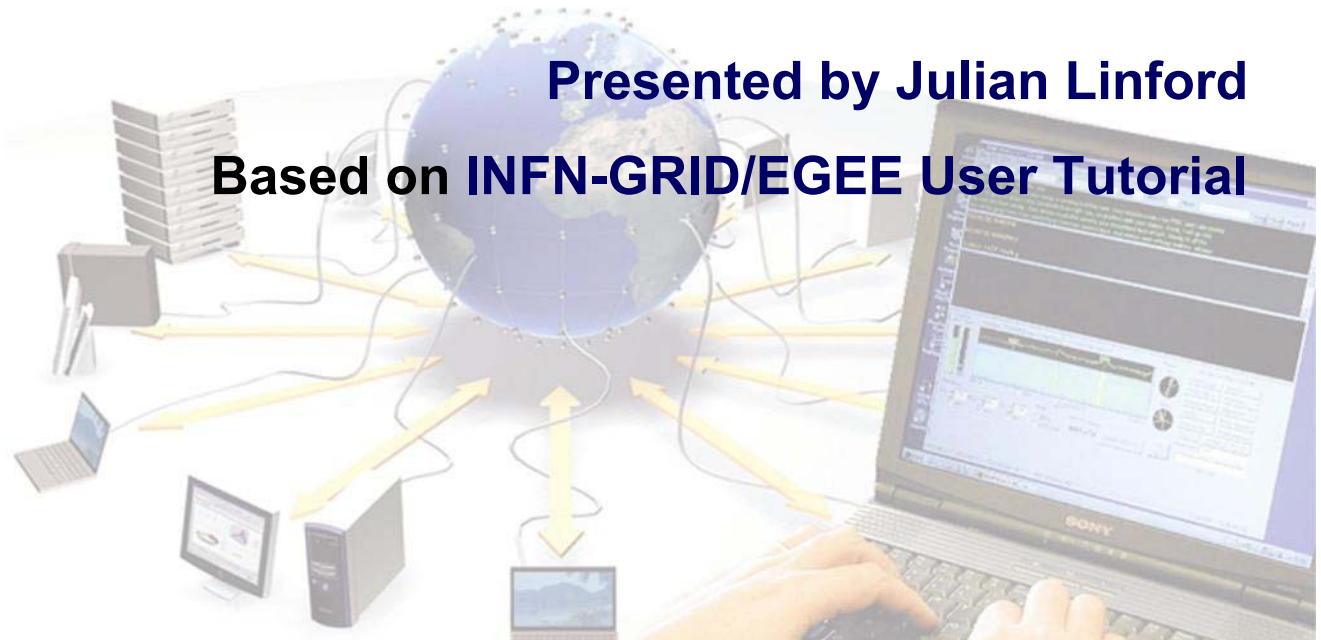
www.eu-egee.org

*ESRIN Grid Workshop Tutorial
Introduction to Grid Computing
Frascati, 3 February 2005*

Grid Information System

Presented by Julian Linford

Based on INFN-GRID/EGEE User Tutorial



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

↑ The Information System (IS) in EGEE/LCG

- ▶ Users, Components, Design, Infrastructure

↑ Information System tools

- ▶ User level
- ▶ Software manager level
- ▶ Site manager level

↑ A new era: R-GMA

↑ Summary

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 developer

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

The Information System Elements

MDS: Monitoring and Discovery Service

- ▶ Adopted from Globus
- ▶ It is the technology adopted by LCG/EGEE to manage the 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

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
```

lcg-infosites output
We will see it during the talk

- ✦ The **general IS architecture** has managed the information
- ✦ It has been provided by different **providers and servers**
- ✦ It follows the **Glue Schema**
- ✦ The **LDAP Protocol** has been used to access the information

General
Elements

Next slides

First
Block

- The LDAP Protocol: Generalities
- Its Data Model
- The EGEE/LCG Schema: The Glue Schema
- Current LDAP implementation

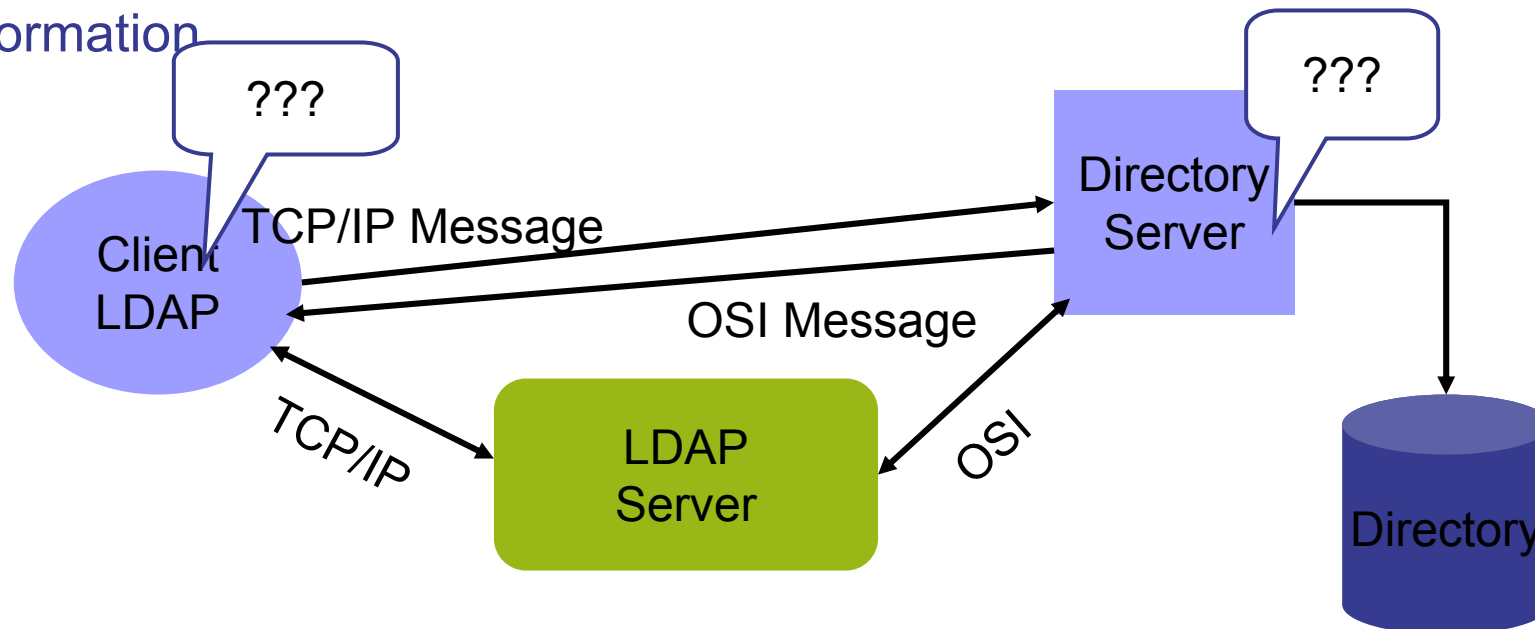
Second
Block

- The providers and servers
- Local GRILes, site GIISes and BDII
- Information transfer between these elements

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 (Directory Information Tree)
- √ It is the internal protocol used by the EGEE/LCG services to share information



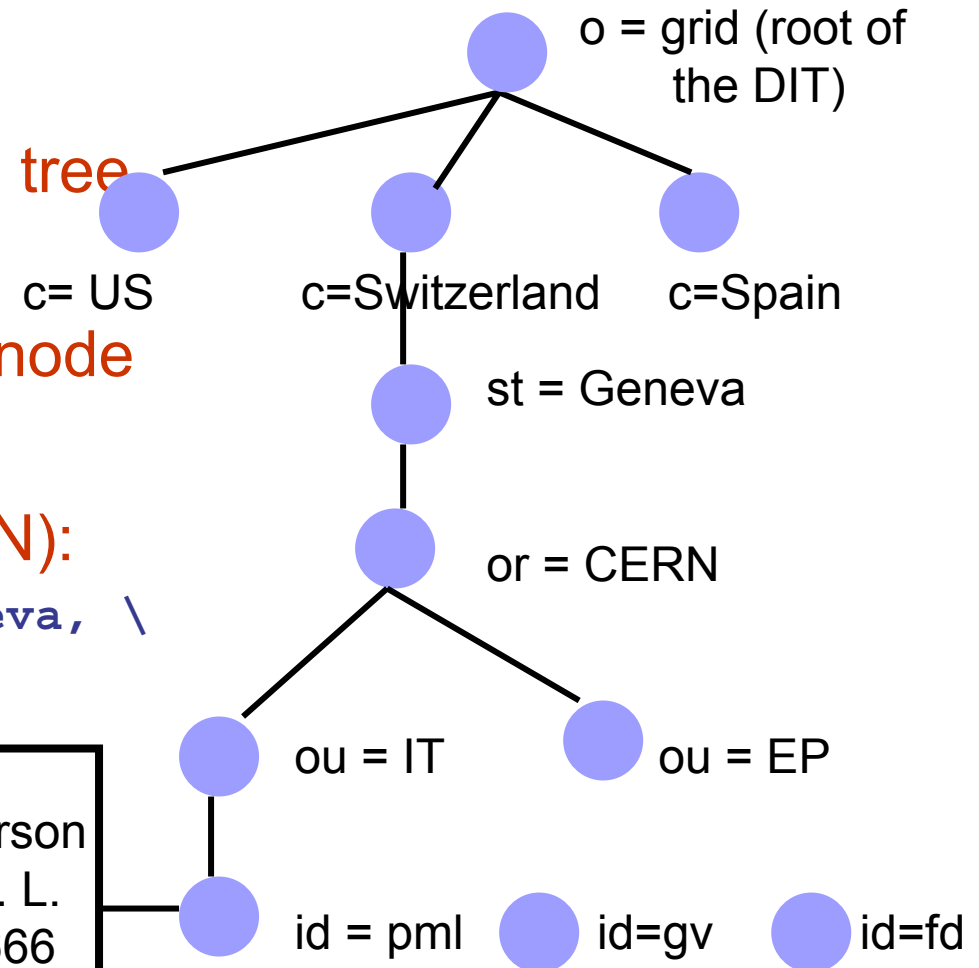
The LDAP Protocol: DIT

▶ LDAP structures data as a tree

▶ 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 LDAP Protocol: The Data Model

- ▶ 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)

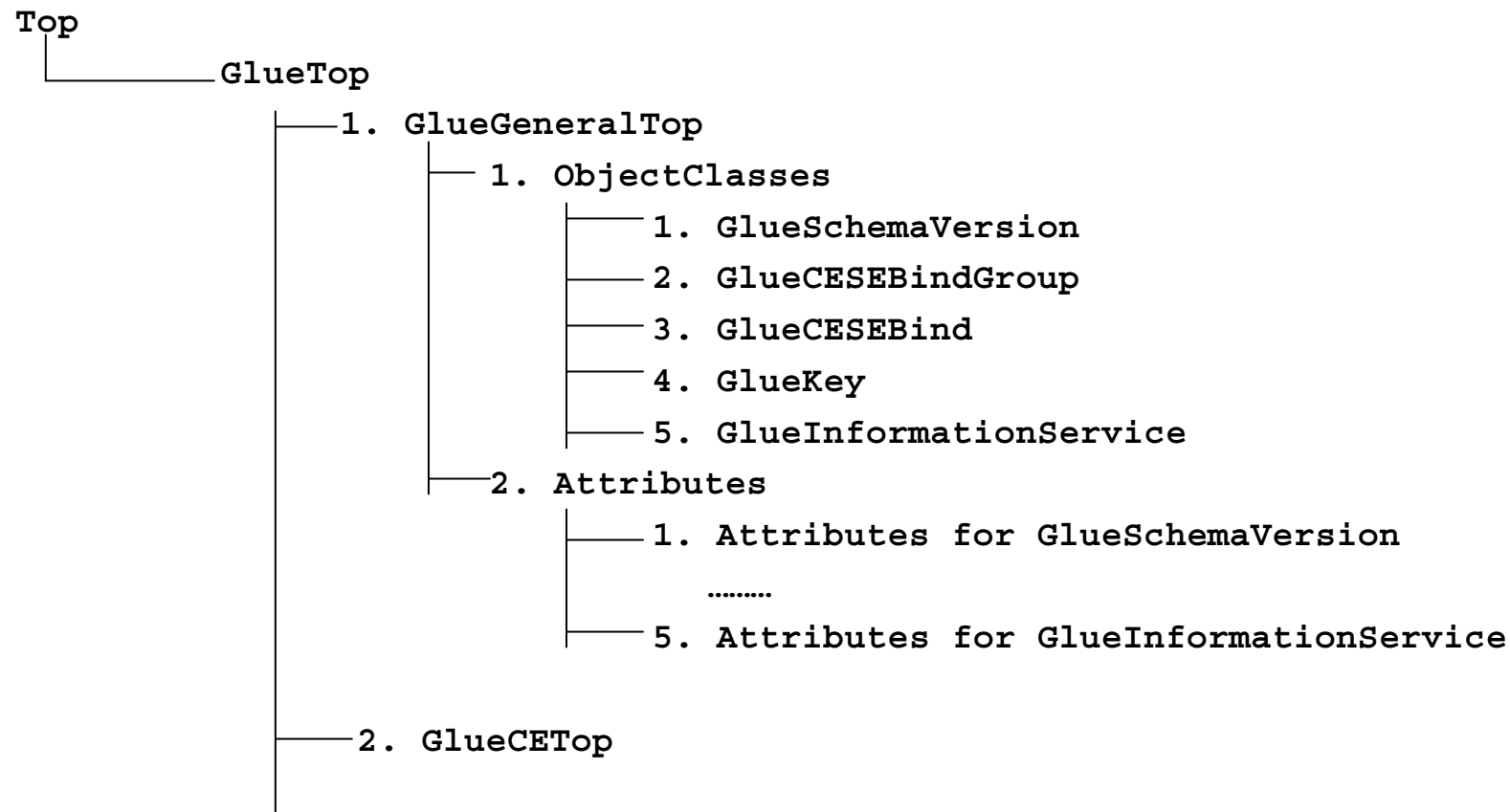
```
dn: <distinguished name>  
objectclass:<objectclassname>  
<attributetype>:<attributevalue>  
<attributetype>:<attributevalue>
```

```
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 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:



Some examples of the Glue Schema (I)

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 for the CE information (`objectclass: GlueCETop`) : No attributes
- ⌘ CE (`objectclass: GlueCE`)
 - `GlueCEUniqueID`: unique identifier for the CE
 - `GlueCEName`: human-readable name of the service
- ⌘ CE Status (`objectclass: GlueCEState`)
 - `GlueCEStateRunningJobs`: number of running jobs
 - `GlueCEStateWaitingJobs`: number of jobs not running
 - `GlueCEStateTotalJobs`: total number of jobs (running + waiting)
 - `GlueCEStateStatus`: queue status: queueing (jobs accepted but not running), production (jobs accepted and run), closed (neither accepted nor run), draining (jobs not accepted but those already queued are running)
 - `GlueCEStateWorstResponseTime`: worst possible time between the submission of the job and the start of its execution

Some examples of the Glue Schema (II)

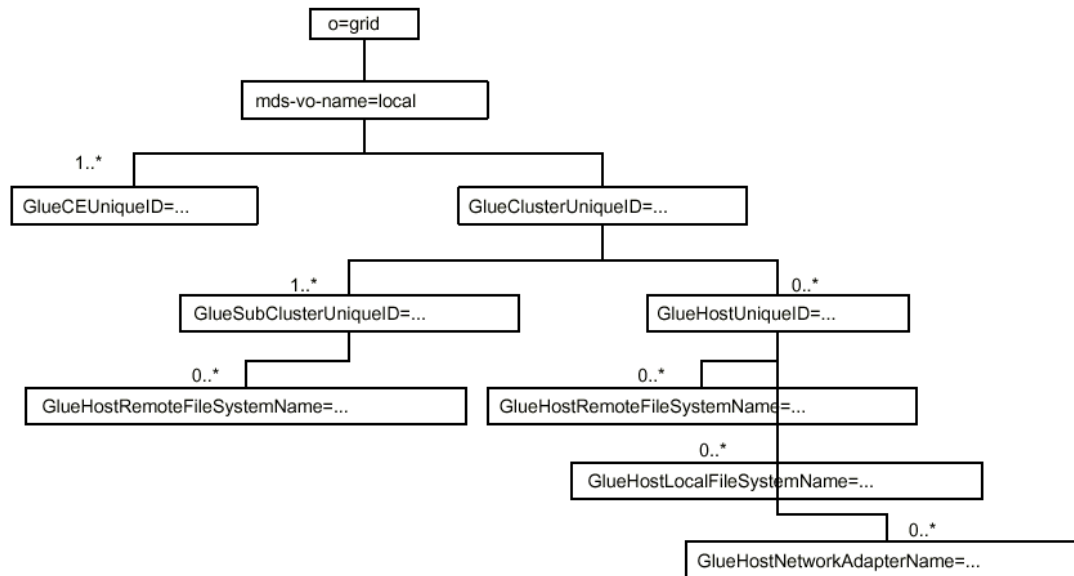
3. Attributes for the SE

- ⌘ Base Class (`objectclass: GlueSETop`) : No attributes
- ⌘ Architecture (`objectclass: GlueSLArchitecture`)
 - `GlueSLArchitectureType`: type of storage hardware (disk, tape, etc)
- ⌘ Storage Service Access Protocol (`objectclass: GlueSEAccessProtocol`)
 - `GlueSEAccessProtocolType`: protocol type to access or transfer files
 - `GlueSEAccessProtocolPort`: port number for the protocol
 - `GlueSEAccessProtocolVersion`: protocol version
 - `GlueSEAccessProtocolAccessTime`: time to access a file using this protocol

4. Mixed Attributes

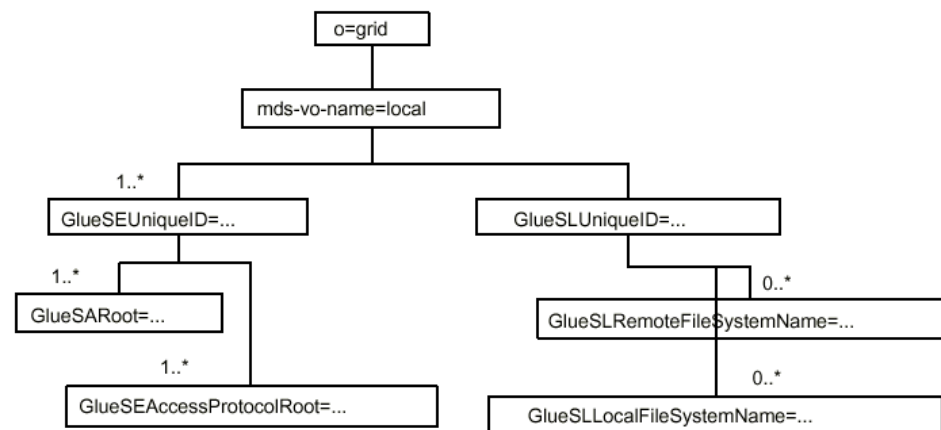
- ⌘ Association between one CE and one or more SEs (`objectclass: GlueCESEBindGroup`)
 - `GlueCESEBindGroupCEUniqueID`: unique ID for the CE
 - `GlueCESEBindGroupSEUniqueID`: unique ID for the SE

The Glue Schema in EGEE/LCG: DIT



**DIT for the
Computer
Resources**

**DIT for the
Storage
Resources**



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
 - Idapadd
 - Idapdelete
 - Idapmodify
 - Idapsearch(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

♠ **Idapsearch**

```
% idapsearch \
```

```
-x \
```

```
-H ldap://grid017.ct.infn.it:2170
```

```
-b 'mds-vo-name=local,o=grid' \
```

```
'(objectclass=GlueSE)' \
```

```
GlueSEUniqueID \
```

Read port of the BDII



Simple authentication

\ Uniform resource identifier

Base DN for search

Filter

Attributes to be returned

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

The Idapsearch Implementation in EGEE/LCG

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

→ Part of the internal WMS software

→ Part of the Monitoring tools

Idapsearch example in LCG

```
dn:GlueServiceURI=http://rlscert01.cern.ch:7777/cms/v.2.2/edg-local-replica-ca  
  talog/services/edg-local-replica-catalog,mds-vo-name=local,o=grid  
objectclass: GlueTop  
objectClass: GlueService  
objectClass: GlueSchemaVersion  
GlueServiceURI: http://rlscert01.cern.ch:7777/cms/v2.2/edg-local-replica/catal  
  og/services/edg-local-replica-catalog  
GlueServiceType: edg-local-replica-catalog  
GlueServicePrimaryOwnerName: LCG  
GlueServicePrimaryOwnerContact: mailto:hep-project-grid-cern-testbed-managers@  
  cern.ch  
GlueServiceHostingOrganization: CERN  
GlueServiceMajorversion: 1  
GlueServiceMinorVersion: 0  
GlueServicePatchVersion: 1  
GlueServiceAccessControlRule: cms  
GlueServiceInformationServiceURL: MDS2GRID:ldap://adc0002.cern.ch:2170/mds-vo-  
  name=local,mds-vo-name=local,o=grid  
GlueServiceStatus: running  
GlueSchemaVersionMajor: 1  
GlueSchemaVersionMinor: 1
```


The use of the command lines in LDAP

♠ **Idapadd, Idapmodify, Idapdelete**

Write port of the BDI

```
% 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

Idapadd, Idapmodify and Idapdelete 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

Components of the IS: GRISs, GIISs and BDII

Abbreviations:

BDII: Berkeley DataBase
Information Index

GIIS: Grid Index Information
Server

GRIS: Grid Resource
Information Server

Each site
can run
a **BDII**. It

collects the information
coming from the GIISs
`% ldapsearch -x -h <hostname>
-p 2170 -b "o=grid"`

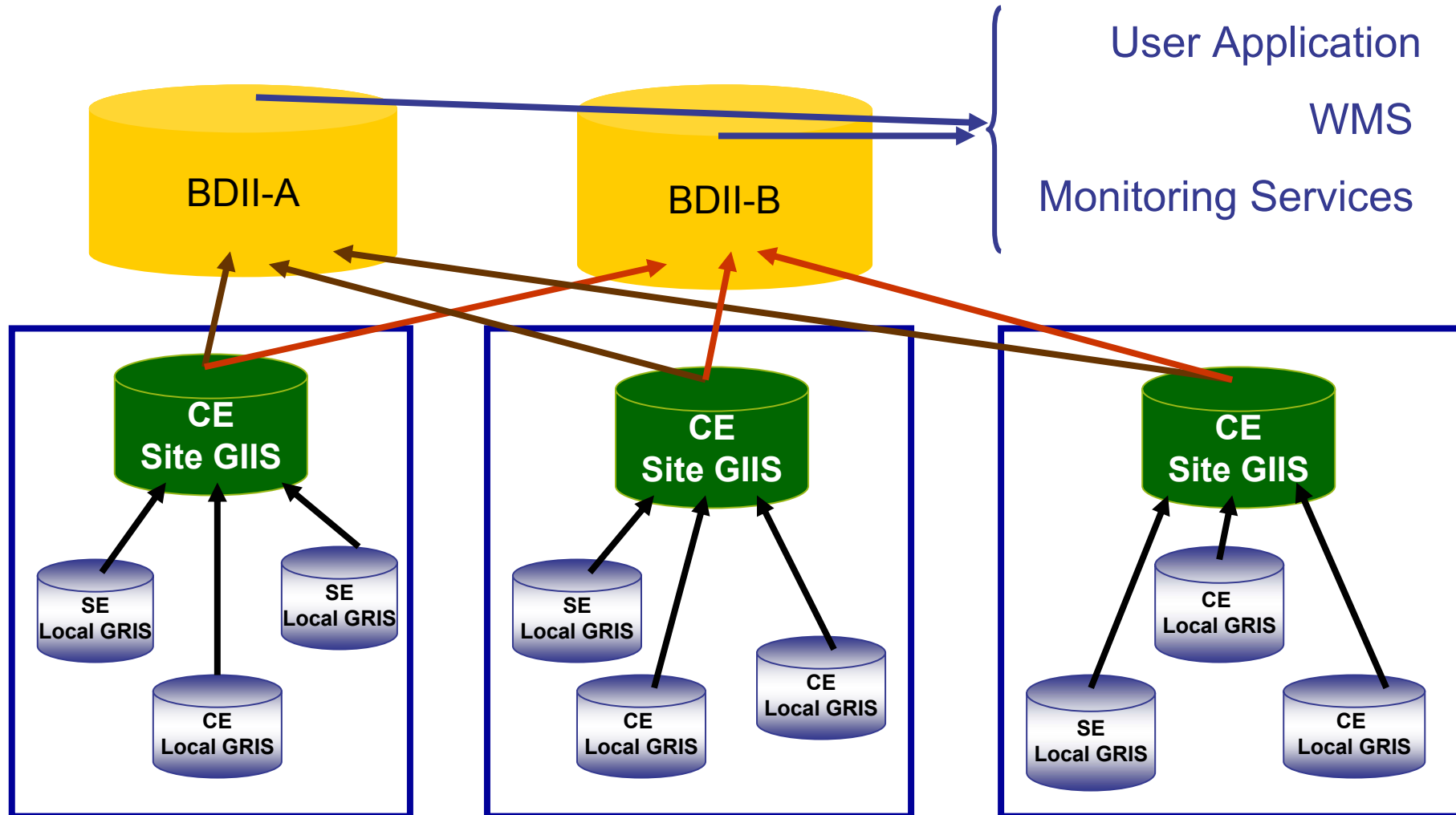
At each site, a **site GIIS** collects the
information
given by the GRISs

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

Local GRISes run on CEs and SEs at each site and report
dynamic and static information

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

GRISs, GISs and BDII connections



The BDII

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

Information Transfer between the IS Components

▶ Lower level: GRIS

- Scripts and configuration files generate Idif 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 GRISes

▶ High level: BDII

- Publish the information of the site GIISes 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)

Next slides: Tools based on LDAP

User
privileges

- **lcg-is-search**: C++ executable
- **lcg-infosites**: Perl script. The next “edg-rm pi”
- Implementation in the experiment software

Software
installation
privileges

- **lcg-ManageVOTag**: Software tags publication

Site admin
privileges

- **lcg-user-configuration**: Information generation

Information System Tools: User Level

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



experiment integration and support

☺ 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 (.so shared and .a static): `/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`

lcg-is-search

Usage:

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

CLI Application: [lcg-infosites](#)

- This is a script which invokes lcg-is-search
- Already deployed in LCG-2 in the last release (version 2.2.3)
- 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



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: Examples (I)

```
> 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          5000000000000         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: Examples (II)

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

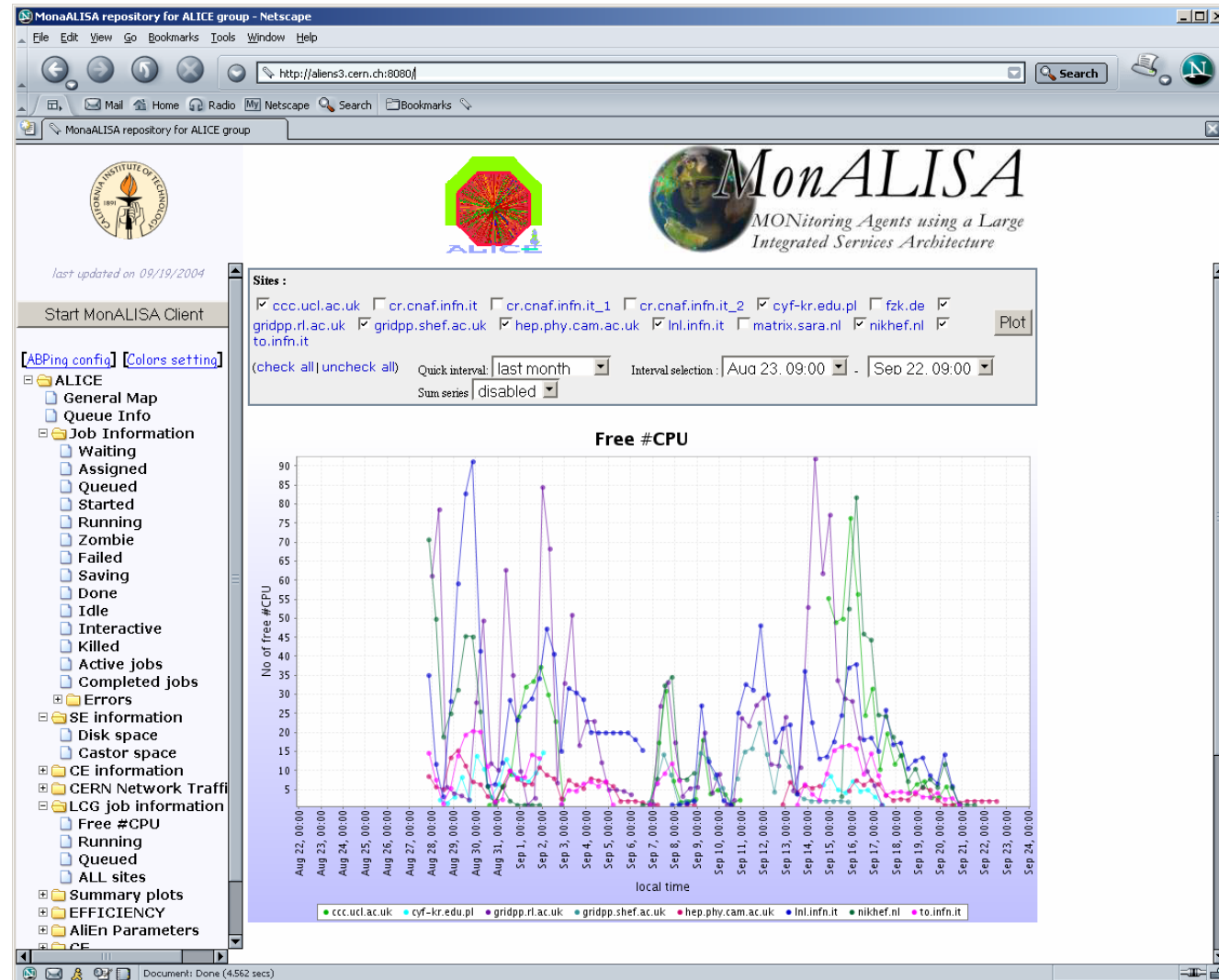
```
*****  
These are the data for alice: (in terms of CPUs)  
*****  
#CPU Free Total Jobs Running Waiting Computing Element  
-----  
52 51 0 0 0 ce.prd.hp.com:2119/jobmanager-lcgpbs-long  
16 14 3 2 1 lcg06.sinp.msu.ru:2119/jobmanager-lcgpbs-long  
[.....]  
The total values are:  
-----  
10347 5565 2717 924 1793
```

lcg-infosites: How does it work?

- `lcg-is-search` uses some `ldapsearch` wrappers designed by LCG to:
 - ▣ Define the connections and basic relative operations (open and close the connections, checks whether it is established, etc)
 - ▣ Query the database for the required information
 - ▣ Iterate through all the Info buffer to retrieve the information
- An additional class (**InfoFromLDAP**) has been created to handle the LDAP wrappers in just one method include all the above operations in just one method:
- Then `lcg-is-search` implements this class
- `lcg-infosites` implements the `lcg-is-search` executable and prints the outputs following the experiments requirements

lcg-infosites

Implementation
in MonALISA:
The monitoring
service of the
Alice
Experiment



Information System Tools: sgm Level

2. You have application software administrator privileges You can publish application specific information

1. You can install the software of your VO



experiment integration and support

♠ 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)

♠ 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

lcg-ManageVOTag

♠ 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 use 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

lcg-ManageVOTag: How does it work?

- ▶ 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: Administrator Level

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)
```

lcg-user-configuration: How does it work?

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

→ Under /opt/lcg/var a **GlueService.Idif\$\$** has just been created. It has already a Idif 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>
```

lcg-user-configuration: How does it work? (cont.)

- α 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

- α 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

- α 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 protocol?

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

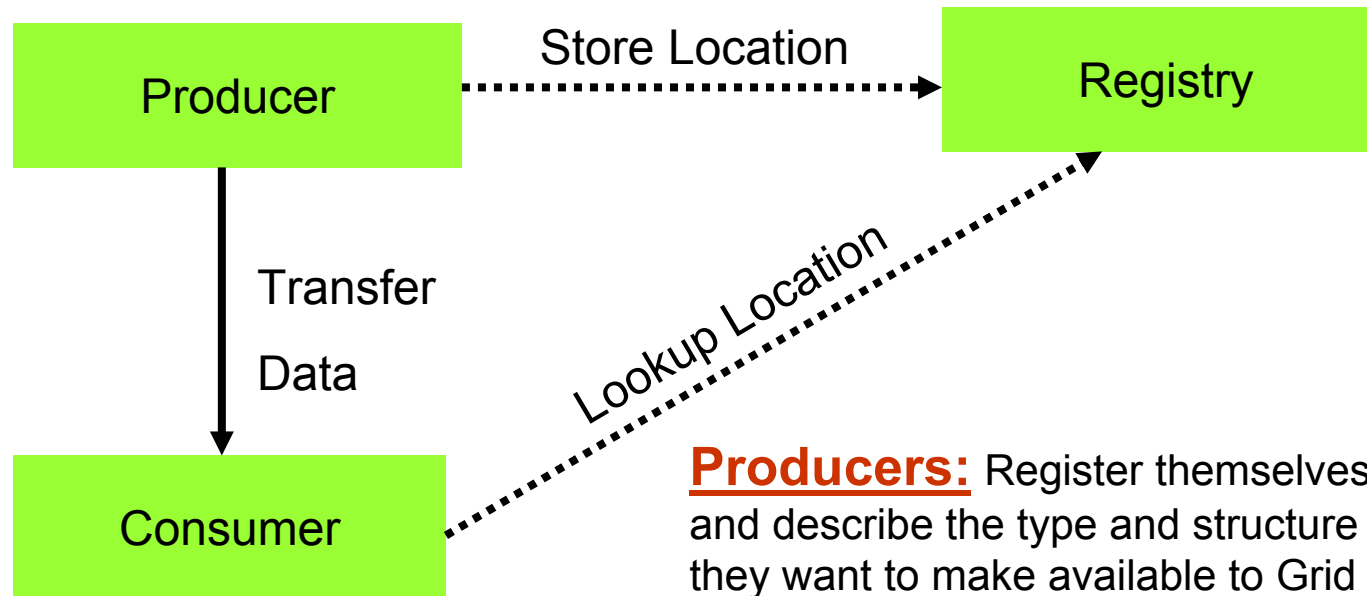
GMA (Grid Monitoring Architecture)

- 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 Architecture

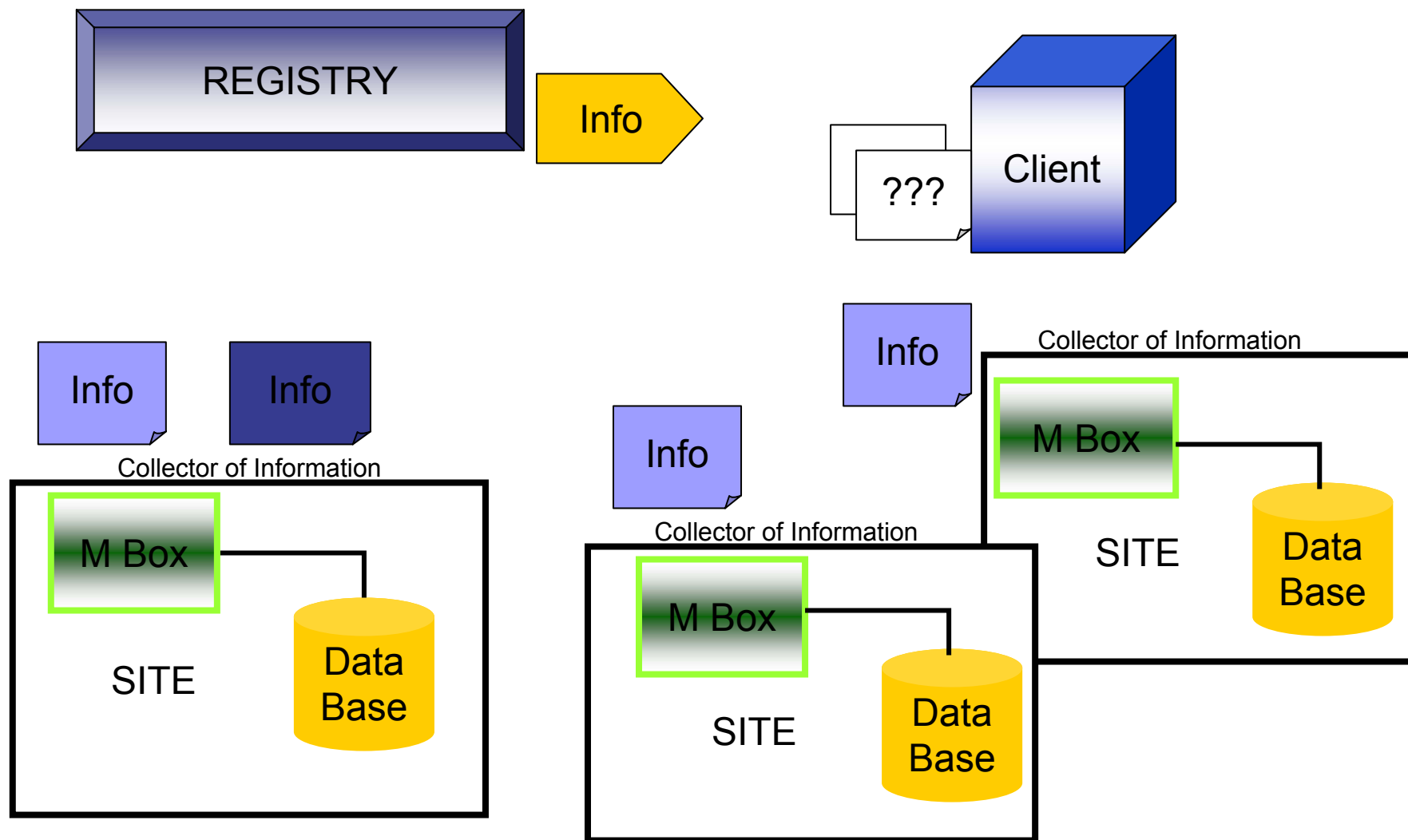


Producers: Register themselves with the Registry and describe the type and structure of the information they want to make available to Grid

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: Design



R-GMA tools: Browser

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

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

R-GMA Browser

All tables

EDG Info Providers

Network Monitoring

CMS

EDG Info Providers

GlueCE

GlueCEAccessControlBaseRule

GlueCESEBind

GlueCluster

GlueHostRemoteFileSystem

GlueSA

GlueSAAccessControlBaseRule

GlueSE

GlueSEAccessProtocol

GlueSEAccessProtocolSupportedSecurity

GlueSL

GlueSubCluster

GlueSubClusterSoftwareRunTimeEnvironment

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://gpgrg06.gridpp.rl.ac.uk:8080/R-GMA/LatestProducerServlet ConnectionId:301164355

Continuous Producer

producerServlet:http://gpgrg06.gridpp.rl.ac.uk:8080/R-GMA/StreamProducerServlet ConnectionId:291549138

producerServlet:http://gpgrg06.gridpp.rl.ac.uk:8080/R-GMA/StreamProducerServlet ConnectionId:291549226

Query

Data GRID WP3

edg-rgma: Virtual Database

- Recently set up in LCG-2/EGEE
- You can use it 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
```

LCG Tools from R-GMA

♣ InfoFromRGMA: Parallel development to [InfoFromLDAP](#)

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

♣ InfoToRGMA: **You have the power, You create the information**

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



△ Both APIs implement the following classes included in the R-GMA package:

▣ **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

This tools have not been deployed and are used for testing the possible R-GMA features

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?

↑ Solution:

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



General Features of the Interface

Characteristics:

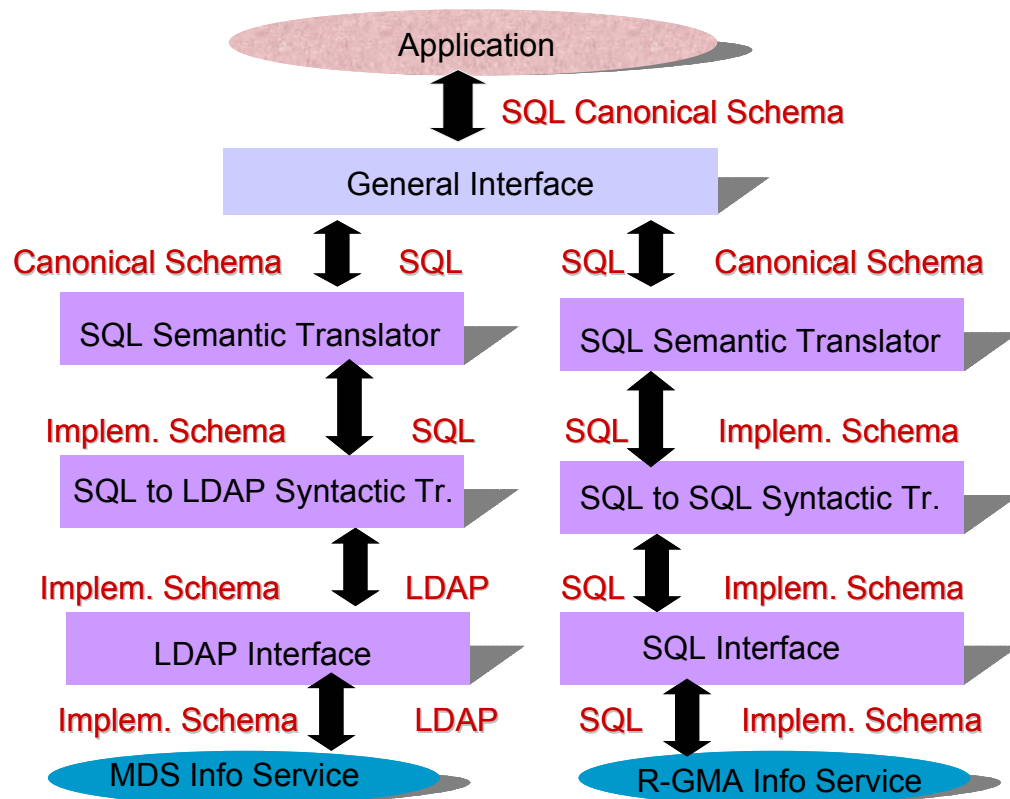
1. The User Applications see just one interface
2. The query language and data model are included
4. The query and schema are syntactically and semantically translated internally in a transparent manner

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

General schema of the API



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
  
```

Summary

- Two main Information System technologies are used in LCG-2
 - **LDAP**: based on Globus
 - **R-GMA**: developed by the European DataGrid project
- Both technologies provide a data model:
 - **DIT**: In the case of LDAP
 - **SQL**: In the case of R-GMA
- The **GLUE** schema is used to describe Grid resource related information in both cases. Both technologies have implemented it depending on their data models
- Different tools to retry and produce information have been developed in LCG-2 based on both technologies. These APIs are available in C, C++ and Java.
- User tools (mostly Perl scripts) based on these APIs are already deployed and are being used to retry information (based right now in LDAP)
- These tools and APIs will be explained and tested during the hands-on session