Additional tools for EGEE application developers

Three days were not enough for everything…

Gergely Sipos
MTA SZTAKI
Budapest

EGEE User Course
10-12. Dec, 2007
Kuala Lumpur, Malaysia

www.eu-egee.org
www.glite.org

Overview

- **Programming APIs for gLite services**
  - WMProxy (~WMS API)
  - SEE-GRID File Management API (~LFC&LCG API)
  - GFAL API → you already know…
- **Additional services in gLite**
  - AMGA
  - R-GMA
- **RESPECT – Initiative to collect useful tools that work:**
  - GridWay
  - GANGA
  - P-GRADE → you already know...

WMProxy

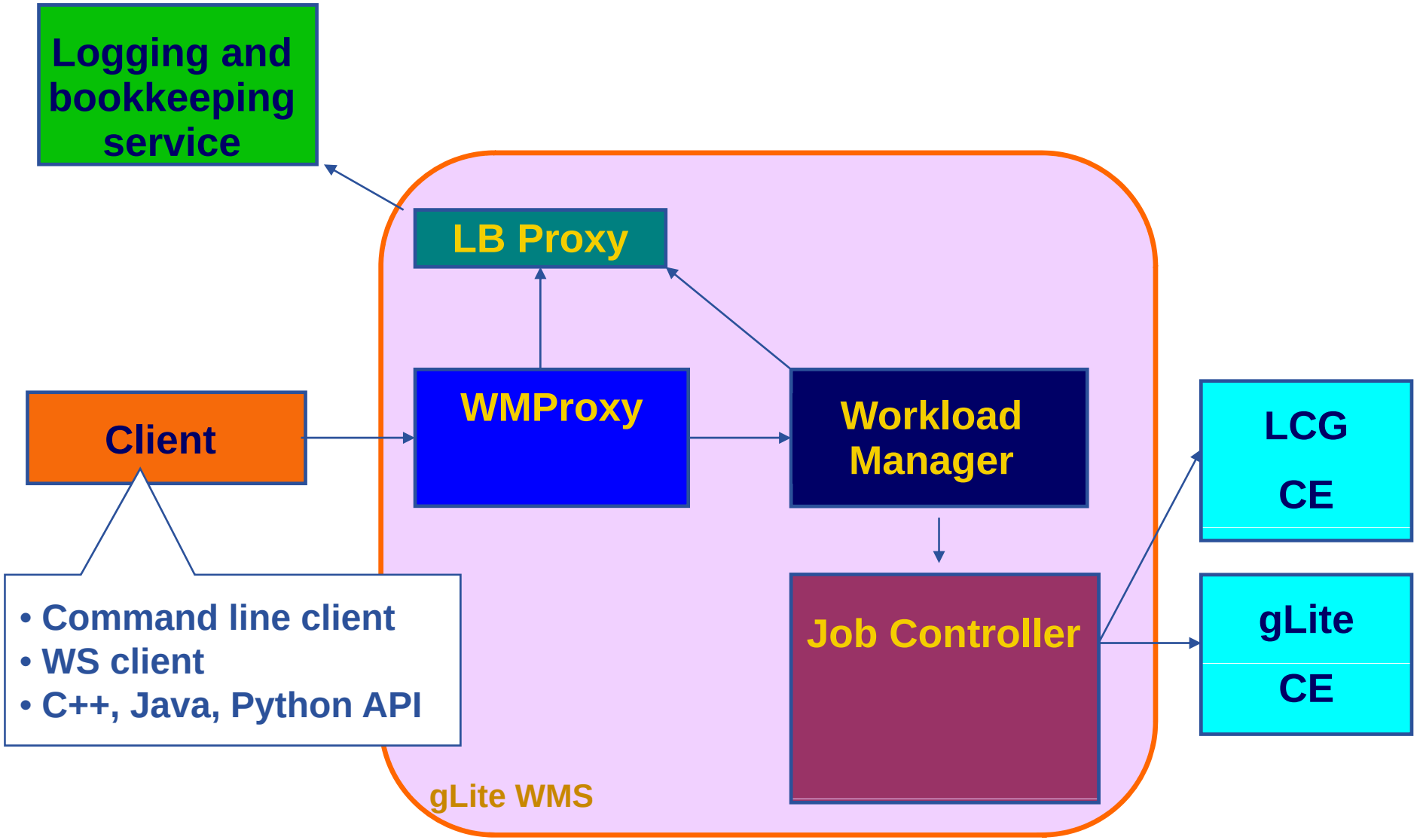
The WMProxy

- The WMProxy service provides access to the WMS functionality (broker) through a Web Service Interface
- The gLite WMProxy Server can be either accessed directly through the published WSDL, the command line interface, or the API
- In these two links you can find a guide about how to interact with the main services exposed by the WMProxy through the provided API Java and C++ API

  https://grid.ct.infn.it/twiki/bin/view/GILDA/ApiJavaWMProxy

  https://grid.ct.infn.it/twiki/bin/view/GILDA/WMProxyCPPAPI

WMS Architecture overview



Submit a new request to WMProxy /1

```
[larocca@glite-tutor:~/API]$ java WMProxyGetProxyAndSubmit

        WMProxyGetProxyAndSubmit.java
        "A simple client to interact with the WMProxy Server."

        Author: Giuseppe La Rocca (giuseppe.larocca@ct.infn.it)
                I.N.F.N. - Sez. of Catania - ITALY
                Via S.Sofia, 64 - 95123 Catania
                Phone: +39.095.378.53.74

Usage :
   java WMProxyGetProxyAndSubmit -h[elp]
   java WMProxyGetProxyAndSubmit <user_proxy> <delegation_id> <wmproxy_server> <InputSandboxFiles>
                                 <jdl_file> <CAcertsPath> [CAs paths (optional)]

where:
   <user_proxy>          ... the file containing the user's credentials
   <delegation_id>       ... the string used to save the user's delegation

   <wmproxy_server>      ... the entry point of the WMProxy Server to contact
                             (e.g. : https://glite-rb3.ct.infn.it:7443/glite_wms_wmproxy_server)

   <InputSandboxFiles>   ... The list of file(s) to transfer to the WMProxy Server
   <jdl_file>            ... the jdl file to submit to the grid

   <CAcertsPath>         ... the path location of the directory containing all the Certificate
                             Authorities files

Contacting... https://glite-rb2.ct.infn.it:7443/glite_wms_wmproxy_server with the proxy..
    /tmp/x509up_u512

Your job has been successfully submitted.
jobID = [ https://glite-rb2.ct.infn.it:9000/XAoY7FZgLJjgCp4U9grsBw ]
```

Submit a new request to WMProxy /2

```
for (int index = 0; index < InputSandboxFiles.length; index++)
{
 String toURL = front + "2811" + rear;
 toURL = toURL + "/" + InputSandboxFiles[index];
 fromURL = "file:///" + InputSandboxFiles[index];

 try {
        GlobusURL from = new GlobusURL(fromURL);
        GlobusURL to = new GlobusURL(toURL);

        UrlCopy uCopy = new UrlCopy();
        uCopy.setDestinationUrl(to);
        uCopy.setSourceUrl(from);
        uCopy.setUseThirdPartyCopy(true);

        uCopy.copy();
 } catch (Exception e) {System.err.println(e.getMessage());}
}
```

Specify the Destination and Source URL(s)

Copy file(s) from the UI to the Resource Broker

The script, thanks to the UrlCopy Class, performs the copy of the InputSandbox files to the reduced path of the WMS as you can see:

[root@glite-rb2 root]# cd /var/glite/SandboxDir/XA/

[root@glite-rb2 root]# ll https_3a_2f_2fglite-rb2.ct.infn.it_3a9000_2fXAoY7FZgLJgC4U9grsBw/input/

-rwxrwxr-x 1 gilda001 glite 30 Jan 11 09:05 start_hostname.sh

Retrieve output file(s)

With the job finishs you can retrieve the output file(s) as follow:

```
[larocca@glite-tutor:~/API]$ java WMProxyGetOutputAndPurge
Usage :
   java WMProxyGetOutputAndPurge -h[elp]
   java WMProxyGetOutputAndPurge <user_proxy> <wmproxy_server> <jobId>
                                 <dirPath> <CAcertsPath> [CAs paths (optional)]
+----------------------------------------------------------------------------+
WMProxy URL = [https://glite-rb2.ct.infn.it:7443/glite_wms_wmproxy_server]
proxyFile   = [/tmp/x509up_u512]
JobID       = [https://glite-rb2.ct.infn.it:9000/XAoY7FZgLJjgCp4U9grsBw]
dirPath     = [/home/larocca/API/]
CA certs    = [/etc/grid-security/certificates/]
+----------------------------------------------------------------------------+
```

List of file(s) retrieved from to the Resource Broker to the user's account

```
Start downloading output file(s)..
file n. 1
-------------------------------------------
name = [gsiftp://glite-rb2.ct.infn.it:2811/var/glite/SandboxDir/XA/https_3a_2f_2fglite-
    rb2.ct.infn.it_3a9000_2fXAoY7FZgLJjgCp4U9grsBw/output/hostname.err]
size = [0]

file n. 2
-------------------------------------------
name = [gsiftp://glite-rb2.ct.infn.it:2811/var/glite/SandboxDir/XA/https_3a_2f_2fglite-
    rb2.ct.infn.it_3a9000_2fXAoY7FZgLJjgCp4U9grsBw/output/hostname.out]
size = [28]
```

References

API Documentation

http://trinity.datamat.it/projects/EGEE/wiki/apidoc/3.1/htmljava/index.html

Datamat – WMProxy quick start

http://trinity.datamat.it/projects/EGEE/wiki/wiki.php?n=WMProxyClient.QuickStart

JDL Attributes guide for WMProxy

https://edms.cern.ch/document/590869/1

WMProxy user guide

https://edms.cern.ch/document/674643/1

SEE-GRID File management API

Overview

- SEE-GRID File Management Java API supports most of the data management operations offered by LFC and LCG_UTILS.

- These Java API are compatible with LCG 2.7.x and gLite grid middleware.

Catalog management

Method Summary	
java.lang.String	**getCatalogType**() Returns type of used grid file catalogue.
Item	**getItem**(java.lang.String pathName) Returns the Item of the corresponding type for a given pathname.
DirectoryItem	**getRoot**() Returns root directory of grid file catalogue.
SEList	**getSEList**() Returns list of available storage elements.
java.lang.String	**getSeparator**() Returns default pathname-separator character for used grid file catalogue.
java.lang.String	**getVO**() Returns name of Virtual Organisation.

Directory management: DirectoryItem class

Method Summary	
boolean	**canExecute**() Test for execute permission.
boolean	**canRead**() Test for read permission.
boolean	**canWrite**() Test for write permission.
boolean	**copyAndRegister**(java.lang.String sourceFilePath, java.lang.String destinationSE) Copies and registers file in grid catalogue directory.
boolean	**copyAndRegister**(java.lang.String sourceFilePath, java.lang.String destinationFileName, java.lang.String destinationSE) Copies and registers file in grid catalogue directory.
boolean	**createNewAlias**(java.lang.String newAliasPathname) Creates the Item's alias with a given pathname.
boolean	**exists**() Test if the item denoted by pathname exists.
java.lang.String[]	**getAliases**() Returns the list of Item's aliases.
java.lang.String	**getComment**() Returns associated comment.
int	**getFileMode**() Returns the filemode value describing item's type and permissions.
int	**getGID**() Returns the Group ID (GID) of the group owning the Item.
java.lang.String	**getGroup**() Returns the name of the group owning the Item.

Directory management: DirectoryItem class

Item	**getParent**() Returns parent item.
java.lang.String	**getParentPathName**()
long	**getSize**() Returns size in bytes.
int	**getUID**() Returns the User ID (UID) of the user owning the Item.
java.lang.String	**getUser**() Returns the name of the user owning the Item.
boolean	**mkdir**(java.lang.String name) Creates subdirectory with the given name.
boolean	**mkdir**(java.lang.String name, LFCFileMode lfcFileMode) Creates subdirectory with the given name and permissions.
protected void	**populateChildren**() Fetches the items contained by the directory.
void	**refresh**() Refreshes the cached information about the directory.
boolean	**renameTo**(java.lang.String newPathName) Renames/moves the item to a given pathname.
void	**setComment**(java.lang.String comment) Assigns a new comment to the item.

File management: FileItem class

Method Summary

boolean	**canExecute**() Test for execute permission.
boolean	**canRead**() Test for read permission.
boolean	**canWrite**() Test for write permission.
boolean	**createNewAlias**(java.lang.String newAliasPathname) Creates the Item's alias with a given pathname.
boolean	**delete**() Deletes file.
boolean	**deleteReplicaFromSE**(java.lang.String se) Deletes replica of a file from specified Storage element
boolean	**deleteReplicaFromSurl**(java.lang.String surl) Deletes replica specified by surl.
boolean	**download**(java.lang.String destinationFile) Downloads file to local filesystem.
boolean	**download**(java.lang.String surl, java.lang.String destinationFile) Downloads file to local filesystem.
boolean	**exists**() Test if the item denoted by pathname exists.
java.lang.String[]	**getAliases**() Returns the list of Item's aliases.
java.lang.String	**getComment**() Returns comment associated with file.

File management: FileItem class

int	**getFileMode**() Returns the filemode value describing item's type and permissions.
int	**getGID**() Returns the Group ID (GID) of the group owning the `Item`.
java.lang.String	**getGroup**() Returns the name of the group owning the `Item`.
java.lang.String	**getGUID**() Returns guid of a file.
Item	**getParent**() Returns parent item.
java.lang.String	**getParentPathName**()
java.lang.String[]	**getReplicas**() Returns list of file's replicas.
long	**getSize**() Returns size in bytes.
int	**getUID**() Returns the User ID (UID) of the user owning the `Item`.
java.lang.String	**getUser**() Returns the name of the user owning the `Item`.
void	**refresh**() Refreshes the cached information about the file.
boolean	**renameTo**(java.lang.String newPathName) Renames/moves the item to a given pathname.
boolean	**replicate**(java.lang.String se) Replicates file.

References

SEE-GRID File Management Java API Documentation

http://grid02.rcub.bg.ac.yu/LFCJavaAPI/fles/docs/javadoc/version1.2/index.html

Source code (version 1.2)

http://grid02.rcub.bg.ac.yu/LFCJavaAPI/files/downloads/SEE-GRIDFileManagementAPIv1.2.zip

Source code (version 1.1)

http://grid02.rcub.bg.ac.yu/LFCJavaAPI/files/downloads/SEE-GRIDFileManagementAPIv1.1.zip

When to use command line, when to use API?

Command line

- **You want to interact with services not a program running on your behalf**
- **Your program is rather long-running: few seconds script overhead does not matter**
- **Your program needs interaction with grid services only at a few, well known points**
- **Your code is a legacy application**
  - You do not have the source
  - You do not understand the source
  - You are not allowed to modify the source

API

- **Your program needs frequent interaction with grid services**
- **Execution time is critical – any effort to minimize it is useful**
- **You can modify the source code or you do not have the source code yet**
- **You need some capability which is not available via command line**
  - E.g. GFAL API: download only part of a file instead of the whole file

AMGA metadata catalog

Metadata on the GRID

- **Metadata is data about data**
- **On the EGEE Grid: information about files**
  - Describes files
  - Locate files based on their metadata
- **You many have 1000's of files, being shared with other researchers**
  - Either:
    - You all access data by remembering lfns (or guids…)
    - .. And hope you know what is in the file…
  - Or
    - Have a metadata catalogue
    - Allow selection of files based on metadata
- **Metadata is fundamental to e-research**

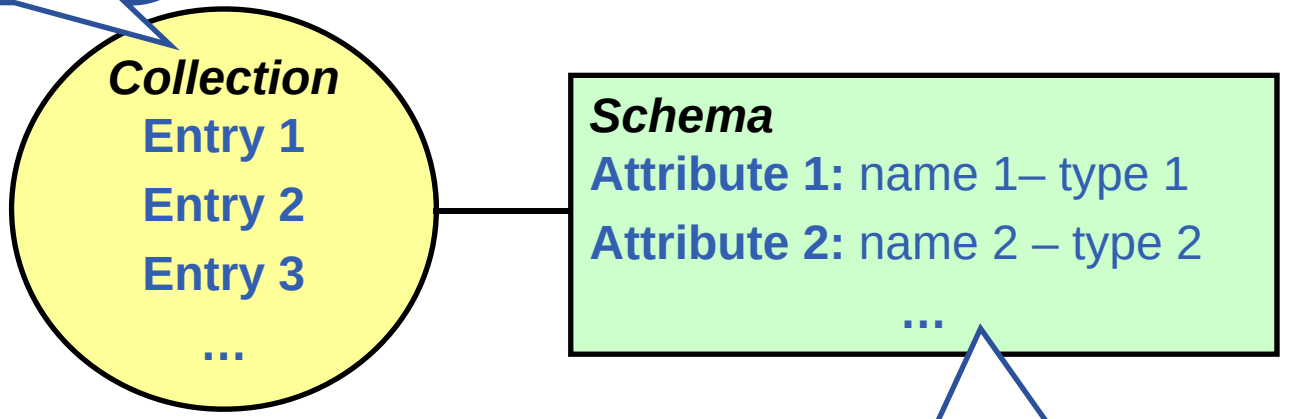
AMGA Implementation

- **AMGA – ARDA Metadata Grid Application**
  - ARDA: A Realisation of Distributed Analysis for LHC
    - Hundreds of millions of files
    - No special security requirements
    - Protection against DoS attacks
- **Now part of gLite middleware**
  - Official Metadata Service for EGEE
  - Also available as standalone component
- **Expanding user community**
  - HEP, Biomed, UNOSAT…
  - More on this later

Metadata concepts

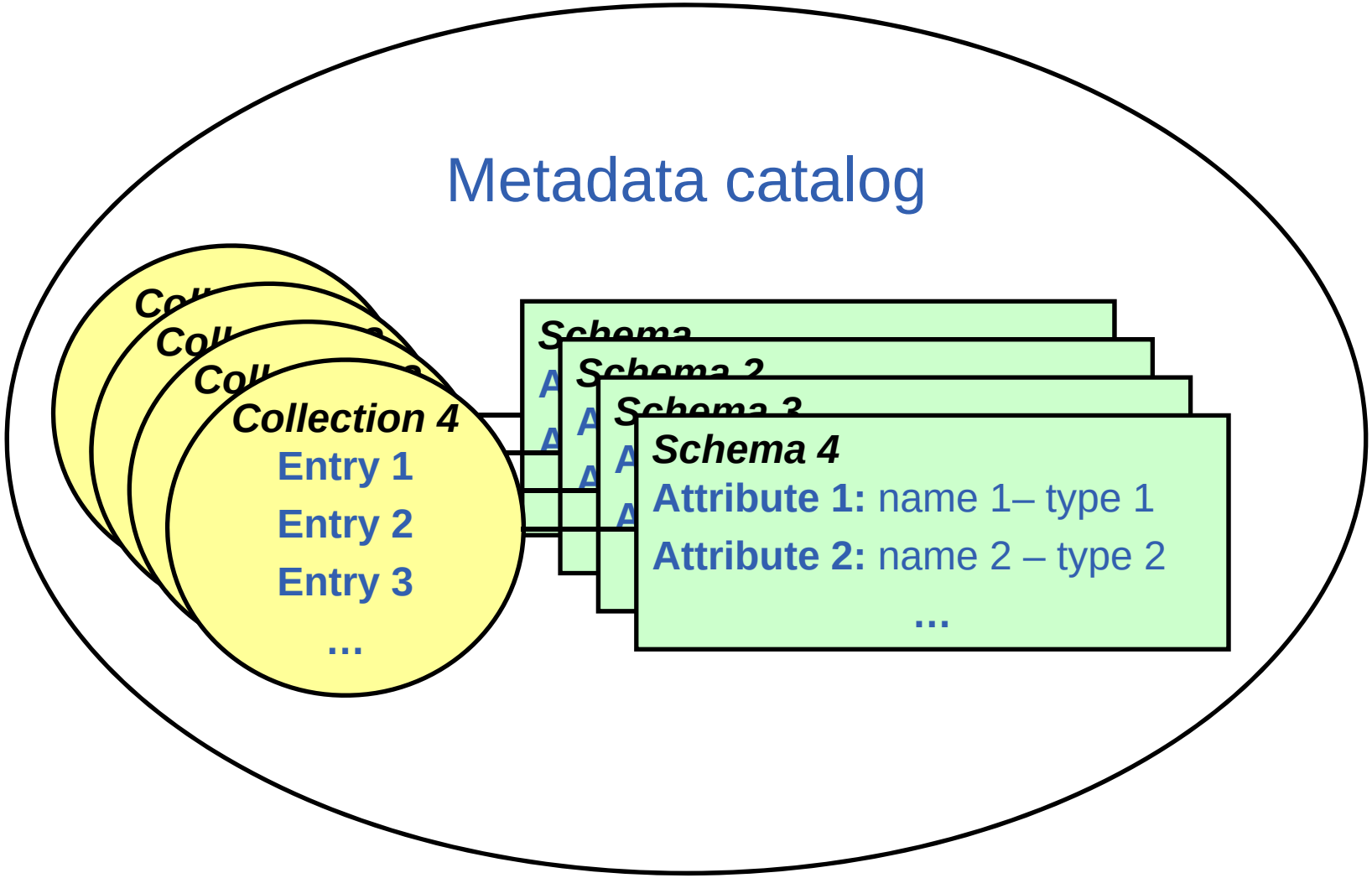
A set of entries.

Entries: The objects (e.g. files) that need to be described with metadata



Schema: a set of attributes. Defines the structure of the metadata

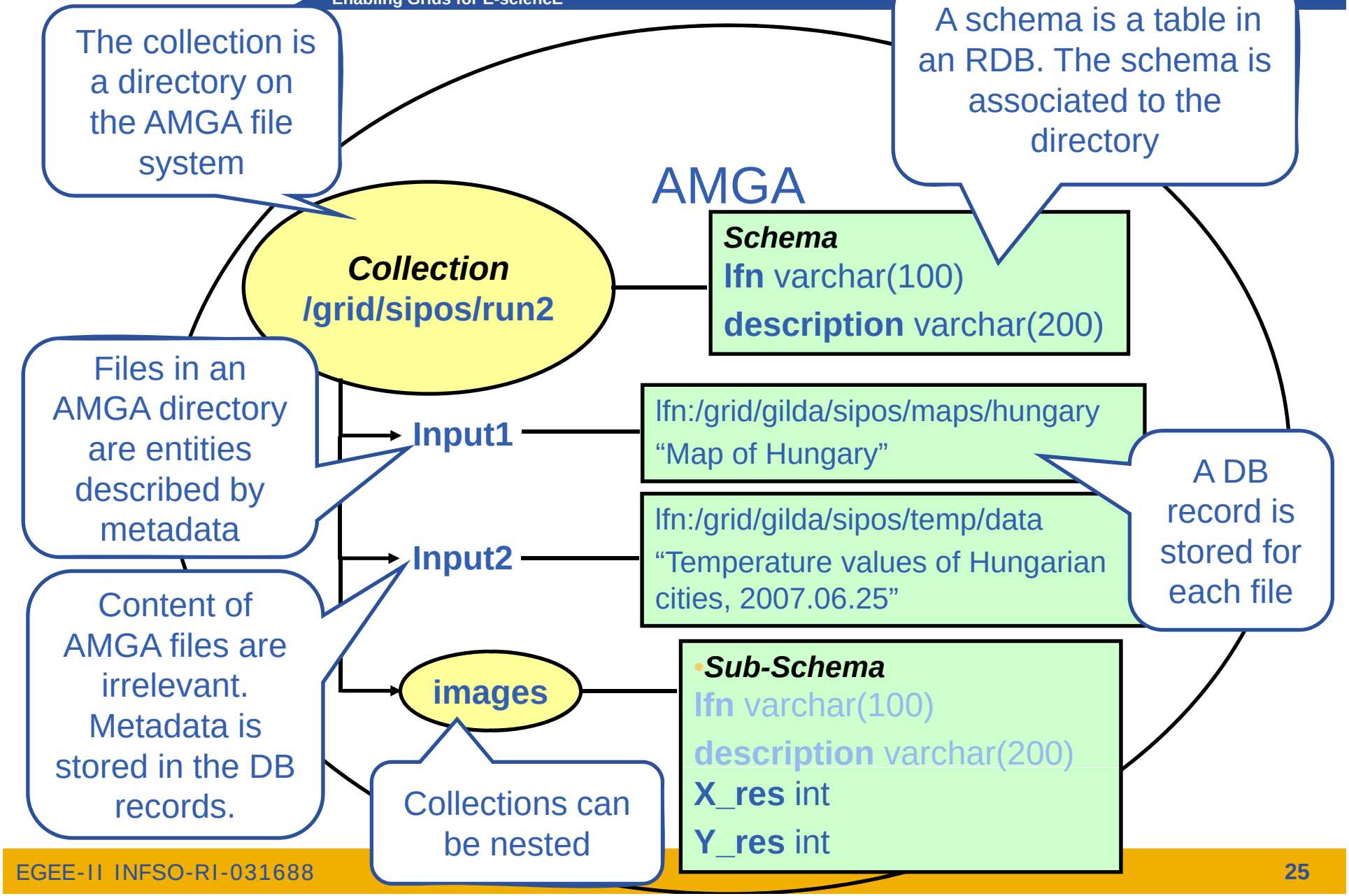
Metadata concepts



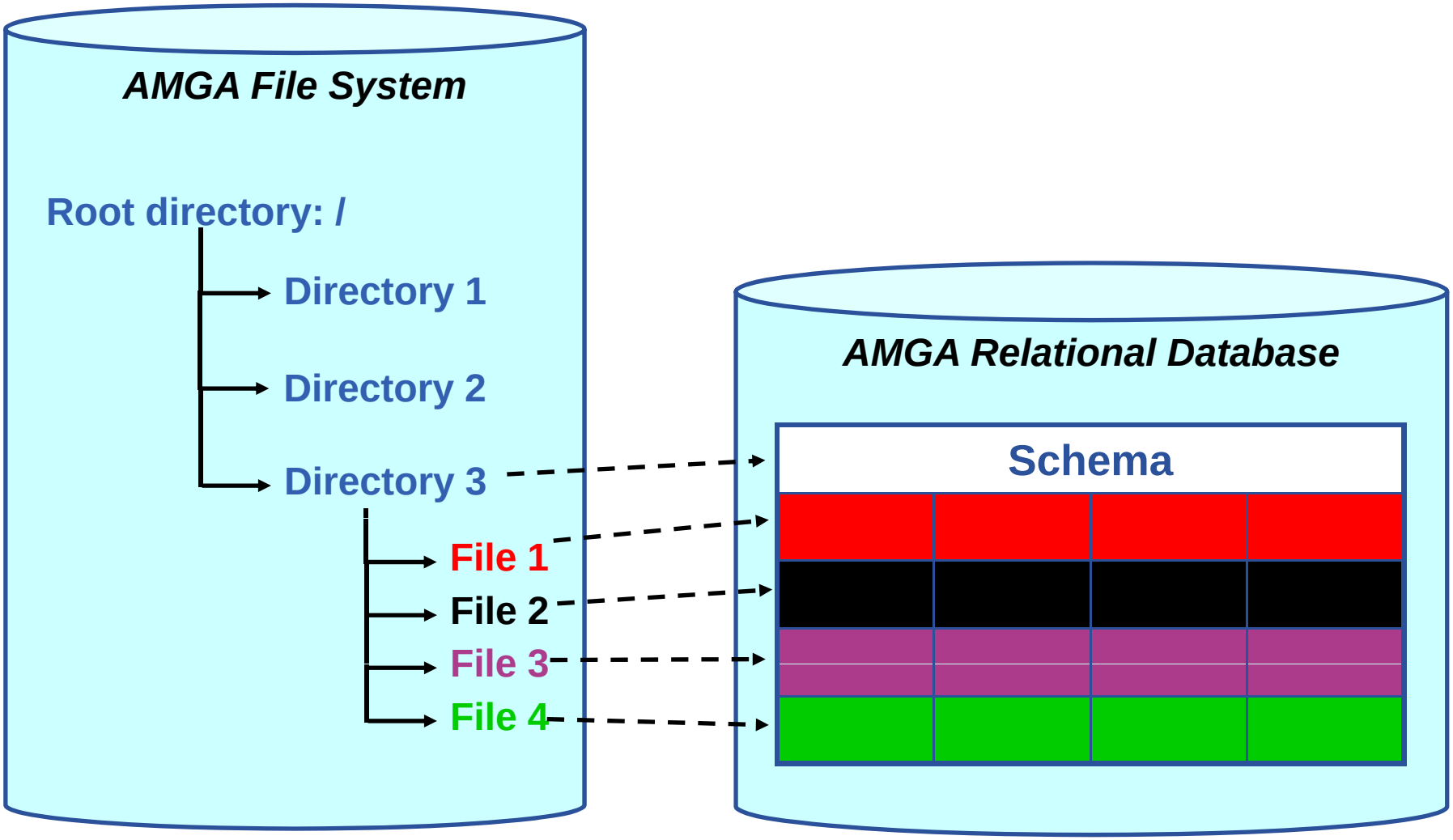
Metadata Concepts

- **Some Concepts**
  - Metadata - List of attributes associated with entries
  - Attribute – name/value pair with type information
    - Type – The type (int, float, string,…)
    - Name – The name of the attribute
    - Value - Value of an entry's attribute
  - Schema – A set of attributes
  - Collection – A set of entries associated with a schema
  - Think of schemas as tables, attributes as columns, entries as rows

Implementation of the concept in AMGA

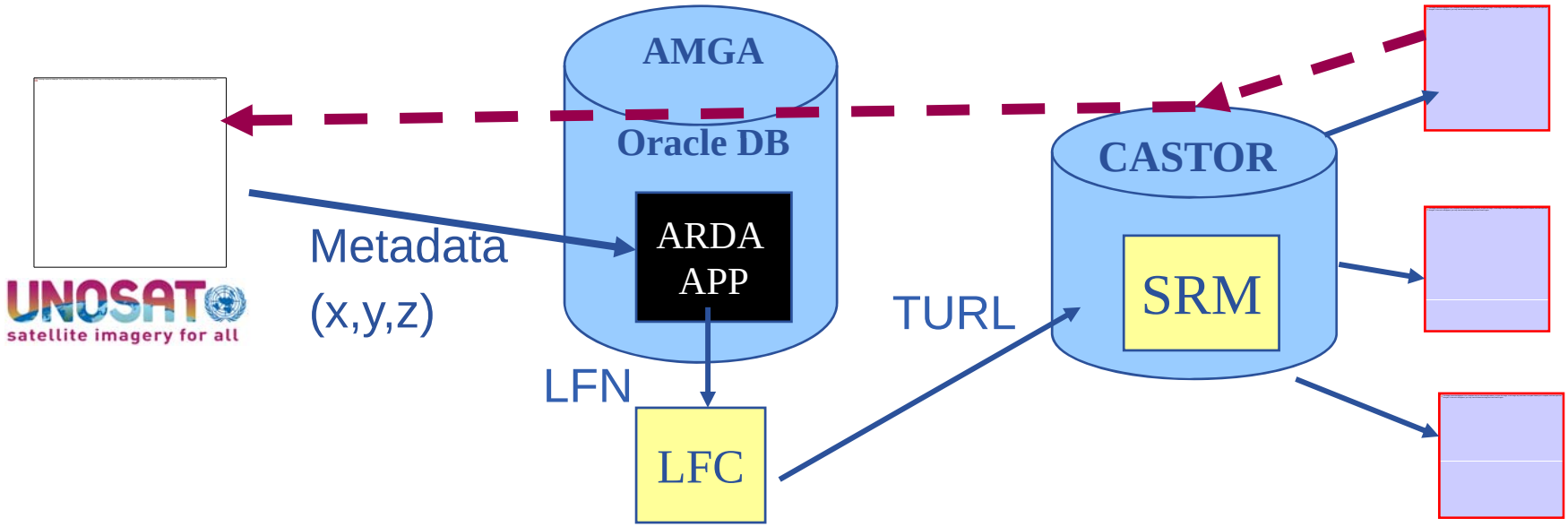


Data distribution overview



An example

- LFC Catalogue
  - Mapping of LFN to TURL
- UNOSAT requires
  - User will give as input data certain coordinates (x, y, z)
  - As output, want the satellite image file for downloading
- The ARDA Group assists us setting up the AMGA tool for UNOSAT



Not only metadata….

- **But also….**

  simplified DB access on the Grid

  - Many Grid applications need structured data
  - Many applications require only simple schemas
    - Can be modelled as metadata
  - Main advantage: better integration with the Grid environment
    - Metadata Service is a Grid component
    - **Grid security**
    - Hide DB heterogeneity

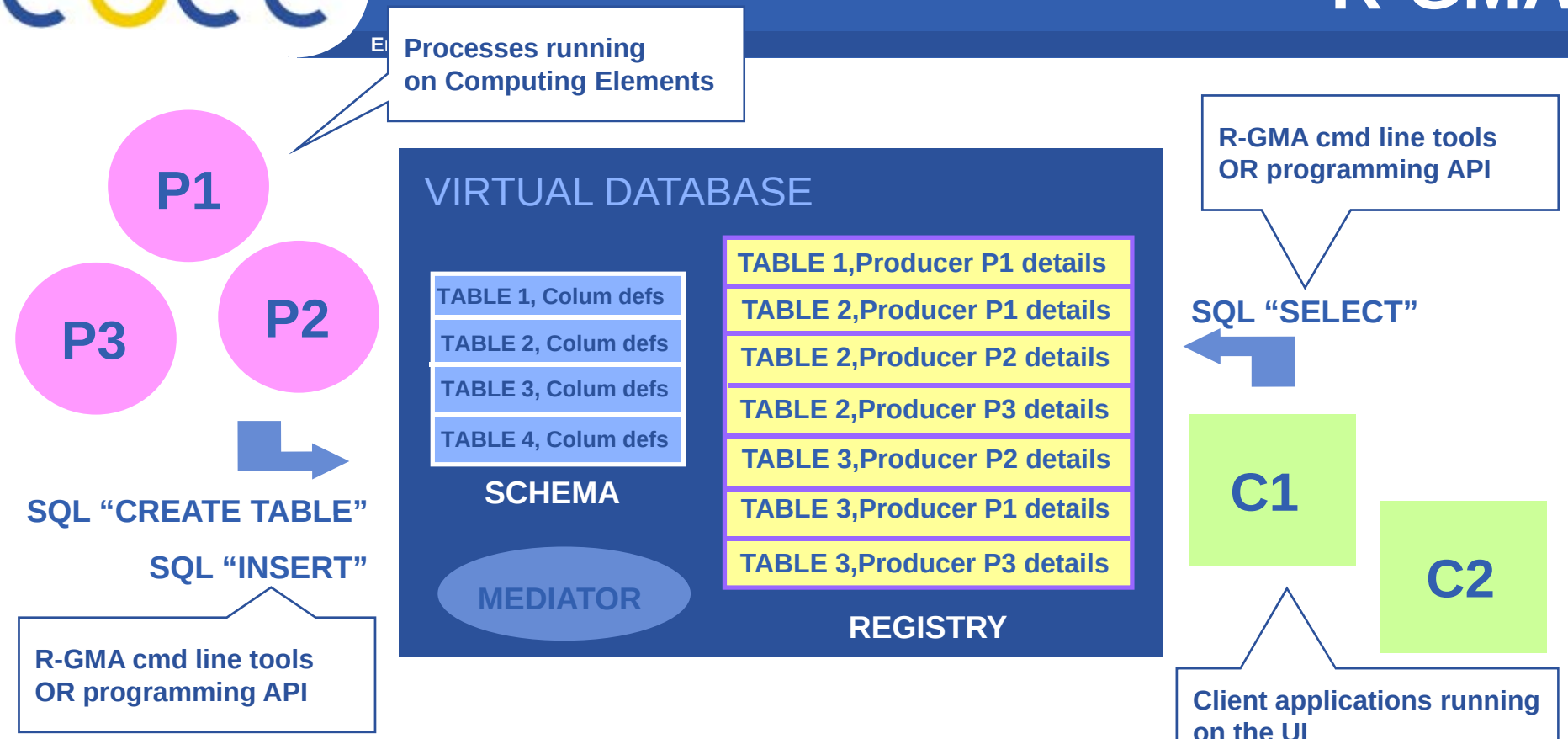
- **Interfacing with AMGA**
  - Command line interpreter (See AMGA practical linked to agenda)
  - Programming API

R-GMA

What is R-GMA ?

- **Uniform method to access and publish both information and monitoring data.**
- **From a user's perspective, an R-GMA installation currently appears similar to a single relational database.**
- **GMA (Grid Monitoring Architecture) was developed by the Global Grid Forum (Predecessor of Open Grid Forum – www.ogf.org)**
- **R-GMA (Relational GMA) was created:**
  - To simplify use of GMA
  - To give a relational view

R-GMA



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

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

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

Producers publish data.

Consumers read data published.

R-GMA practicals

- **Linked to the agenda**
  - Command line practical
    - Query R-GMA tables on GILDA
    - Create new tables, add new entries
  - API practical
    - Write a data producer with RGMA API (monitored grid application)
    - Write a data consumer with RGMA API (monitor application)

More information

- **R-GMA overview page.**
  - http://www.r-gma.org/
- **R-GMA in EGEE**
  - http://hepunx.rl.ac.uk/egee/jra1-uk/
- **R-GMA command line tool**
  - http://hepunx.rl.ac.uk/egee/jra1-uk/glite-r1/command-line.pdf
- **R-GMA Browser Home Page**
  - https://rgmasrv.ct.infn.it:8443/R-GMA/

EGEE NA4 RESPECT initiative

***R**ecommended **E**xternal **S**oftware **P**ackages for **E**gee **C**ommuni**T**ies*

www.eu-egee.org

www.glite.org

Information Society and Media

Motivations

- EGEE gLite middleware: implementations of base grid services that serve as a platform for high-level services
- Not aim to provide a comprehensive solution for any particular grid application
- Software from external providers must be used in conjunction with gLite to satisfy fully the needs of the user community

- The array of available grid software is vast!
- RESPECT (Recommended External Software Packages for EGEE Communities) program aims identifying useful, well-supported software for EGEE users

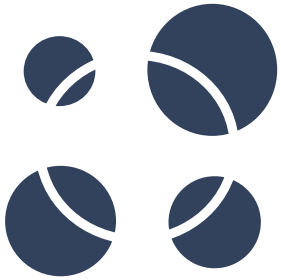
Benefits to EGEE users

- Having a set of external software packages that **enhances the functionality of the gLite middleware**
- Rduces the amount of application development, and generally **accelerates the adoption of grid technologies**
- **Reduce the pressure on the EGEE middleware activity** to provide solutions for services outside of the core functionality
- Integration and testing activities in **EGEE can concentrate on core gLite issues**
- Increasing the number of users via a **more attractive platform** and having more varied services

RESPECT tools

- **RESPECT provides list of software that focuses on those packages that are genuinely useful for an EGEE application and that are generic enough to be useful to other applications**
- **The RESPECT program is not a general repository of grid software**
- **Current RESPECT tools:**
  - GANGA
  - GridWay
  - P-GRADE Portal

- **Further information:**
  http://egeena4.lal.in2p3.fr/ → "Grid software" menu



The GridWay Metascheduler

GridWay

GridWay

one of the tools recognised by EGEE's RESPECT program

Alternative to WMS

Higher level command line UI

Examples of use:

Alternative broker – no need for close CE-SE

Many similar jobs

Resources outside EGEE also to be used

User-site-specific policies are required (priorities of users' jobs)

…

What is GridWay?

GridWay is a meta-scheduler that works on top of Globus-based services (e.g. GRAM, MDS & GridFTP, CE)

For the user

A Local resource management system-like (LRMS) environment for submitting, controlling & monitor jobs

A way to execute your applications on the Grid, without having to worry about resource brokering, file staging or failures

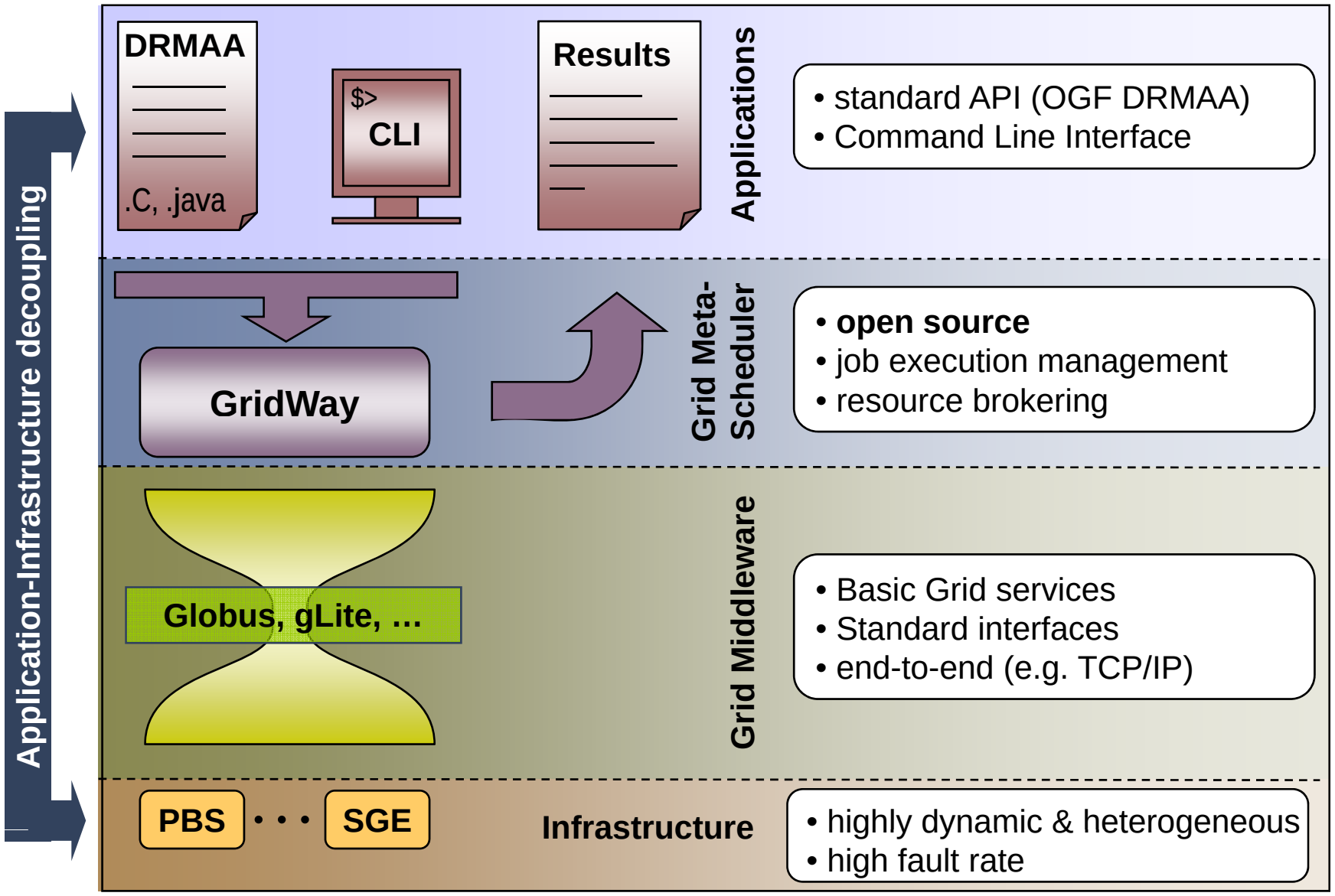
For the Grid Application Developer

- A standard-base development framework for Grid Applications
- JAVA and C bindings of DRMAA API – Programming API to manage jobs

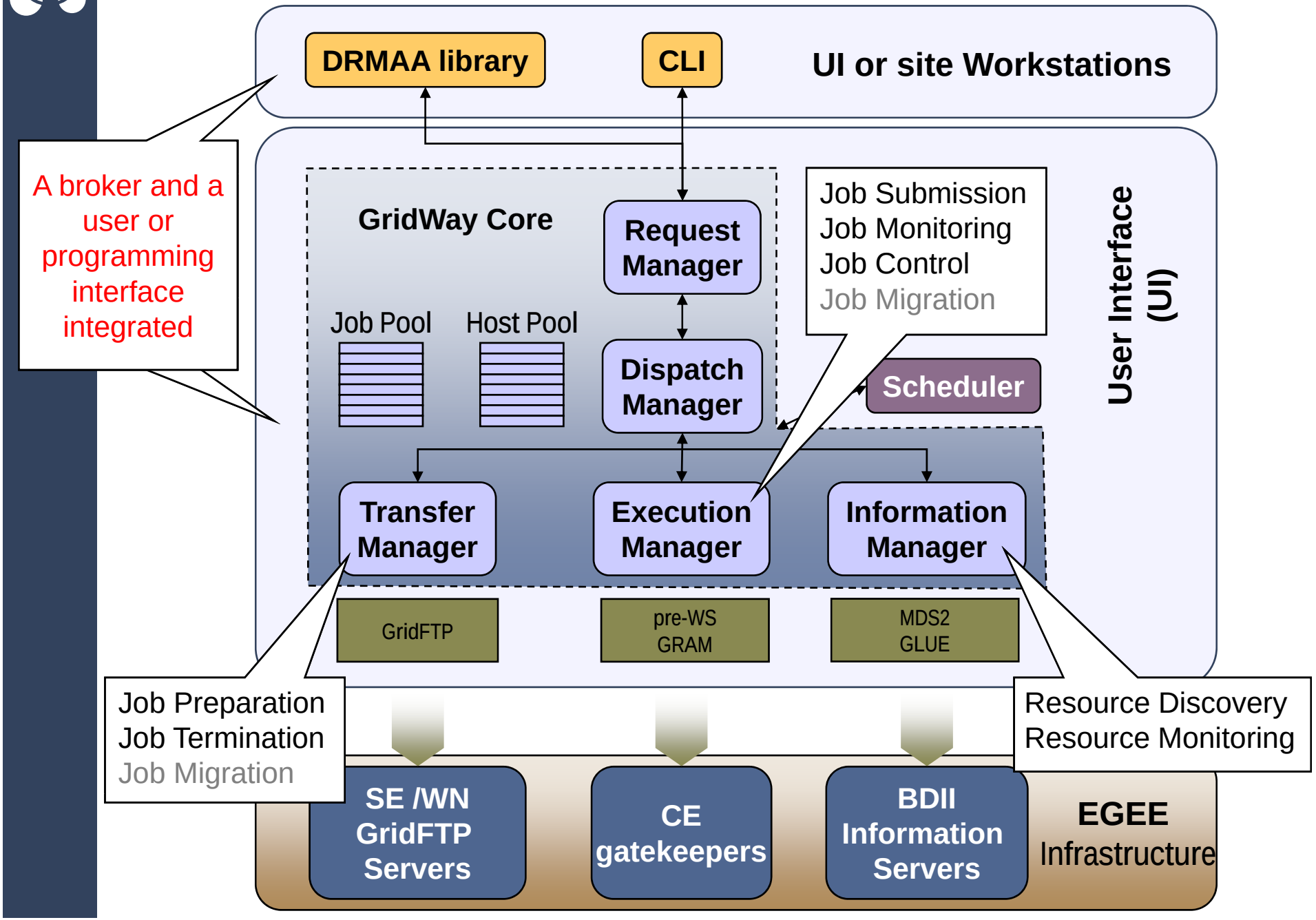
For the System Administrator

- A policy-driven job scheduler, implementing a wide range of access and Grid-aware policies.

What is GridWay?



How do I use GridWay in EGEE?



How do I use GridWay in EGEE?

Job Template – similar to, but not the same as JDL!

```
# Execution variables
EXECUTABLE  = job
ARGUMENTS   = ${TASK_ID} ${TOTAL_TASKS} 100000
ENVIRONMENT = LD_LIBRARY_PATH=/usr/local/lib

# Resource selection parameters
REQUIREMENTS = HOSTNAME= "*.dacya.ucm.es"
RANK         = CPU_MHZ

# I/O files
INPUT_FILES  = my_inputfile
OUTPUT_FILES = my_outputfile

# Standard streams
STDOUT_FILE = stdout_file.${TASK_ID}
STDERR_FILE = stderr_file.${TASK_ID}
```

Parameter study jobs

Parameter study jobs

How do I use GridWay in EGEE?

`gwps`: display job information and status

USER	JID	AID	TID	DM	EM	START	END	EXEC	XFER	EXIT	NAME	HOST
ruben	0	--	--	done	----	15:31:57	15:44:08	0:10:01	0:01:26	0	job1.jt	cluster.pnpi.nw.ru
rgh	1	--	--	done	----	15:31:58	15:44:11	0:09:59	0:01:26	0	MPI.jt	e1.egee.fr.cgg.com
rgh	2	--	--	done	----	17:07:44	17:21:09	0:11:27	0:01:28	0	maratra.jt	aquila.dacya.ucm.es
nacho	3	--	--	prol	----	17:07:47	--:--:--	0:11:19	0:01:43	--	maratra.jt	e1.egee.fr.cgg.com
rgh	4	--	--	done	----	17:41:29	17:55:07	0:11:29	0:01:27	0	maratra.jt	heplnx201.pp.ac.uk
rgh	5	--	--	done	----	17:41:32	17:54:05	0:10:24	0:01:28	0	test.jt	e1.egee.fr.cgg.com
jlvazq	6	--	--	pend	----	10:58:38	--:--:--	0:54:06	0:58:37	--	test.jt	gridgate.cs.tcd.ie

`gwhost`: display resources information and status

HID	OS	ARCH	MHZ	%CPU	MEM(F/T)	DISK(F/T)	N(U/F/T)	LRMS	HOSTNAME
0	Scientific	i686	1001	0	513/513	0/0	0/169/224	jobmanager-lcgpbs	cg02.ciemat.es
1	Scientific	i686	1000	0	1536/1536	0/0	0/2/30	jobmanager-lcgpbs	lcgce01.jin.ru
2	Scientific	i686	2800	0	2048/2048	0/0	0/1/98	jobmanager-lcgpbs	lcg6.smsu.ru
3	Scientific	i686	1266	0	2048/2048	0/0	0/0/6	jobmanager-pbs	ce1.cgg.com
4	Scientific	i686	3000	0	2048/2048	0/0	0/0/56	jobmanager-pbs	cluster.nw.ru
5	Linux2.6.16	x86	3216	73	862/2027	114643/118812	0/1/1	Fork	cygnus.ucm.es
6	Linux2.6.16	x86	2211	0	671/1003	76882/77844	0/2/2	SGE	aquila.ucm.es
7	Linux2.6.16	x86	3215	0	133/2027	109735/118812	0/1/1	Fork	draco.ucm.es
8	Linux2.6.16	x86	3200	0	513/513	0/0	0/1/2	SGE	ursa.ucm.es
9	Linux2.6.16	x86	2211	100	673/1003	76876/77844	0/2/2	PBS	hydrus.ucm.es

How do I use GridWay in EGEE?

Other Commands

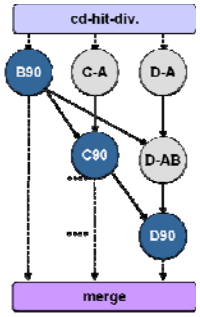
- **gwhistory:** display job execution history

```
HID  START     END       PROLOG   WRAPPER  EPILOG   MIGR     REASON  QUEUE   HOST                 .
2    15:40:22  15:44:11  0:00:15  0:03:15  0:00:19  0:00:00  ----    fusion  e1.egee.fr.cgg.com
1    15:36:22  15:40:09  0:00:09  0:03:21  0:00:17  0:00:00  err     fusion  e2.egee.cesga.es
0    15:32:22  15:36:11  0:00:07  0:03:23  0:00:19  0:00:00  err     fusion  ce-egee.bifi.unizar
```

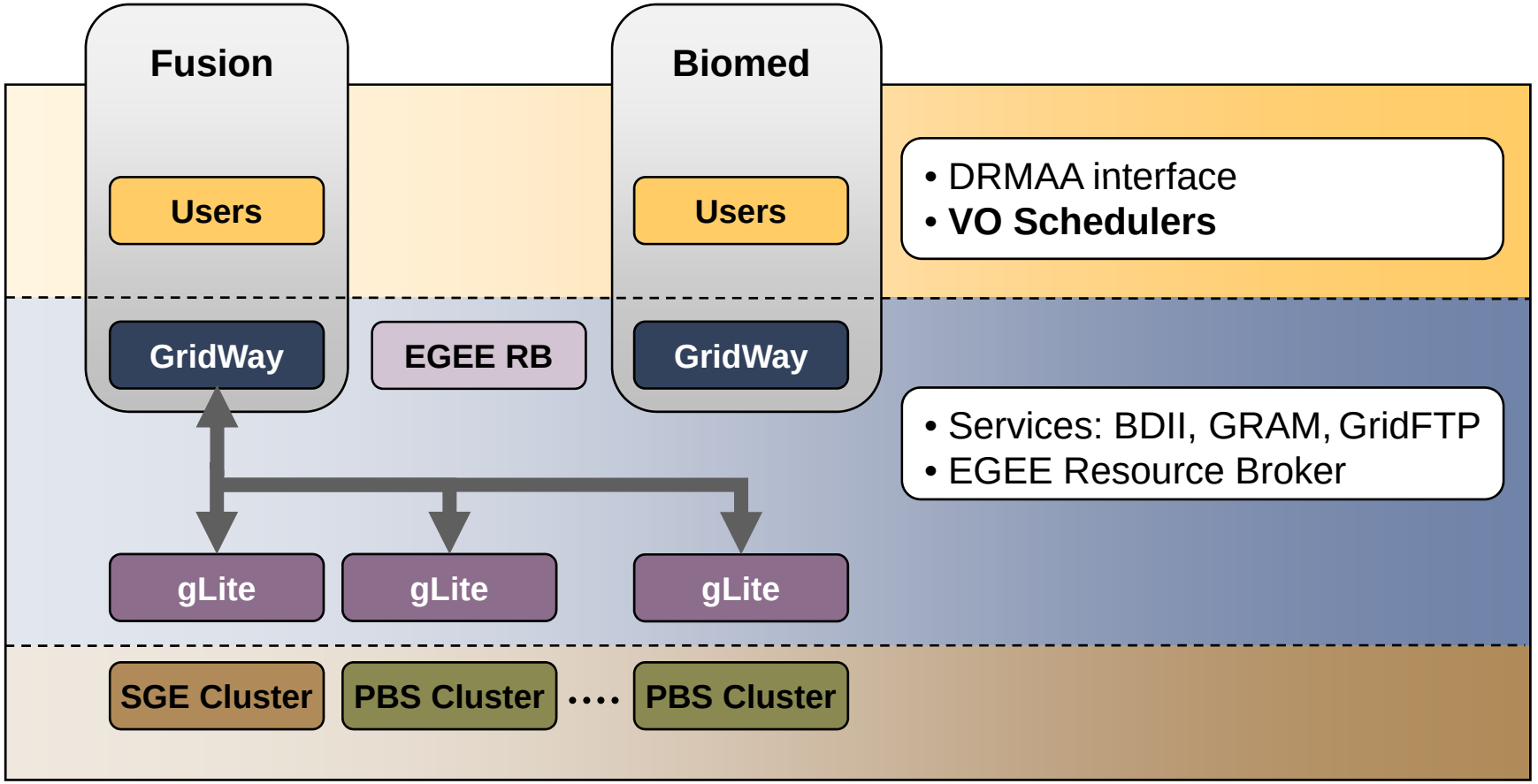
- **gwkill:** signals a job (kill, stop, resume, reschedule)
- **gwsubmit:** submits a job, or an array job
- **gwwait:** waits for zombie state of a job (any, all, set)
- **gwuser:** displays information about users
- **gwacct:** prints accounting information

Who is using GridWay in EGEE?

Massive **Ra**y **Tra**cing



CD-HIT workflow



Where can I get GridWay?

Download the software

- From the Gridway webpage: www.gridway.org
- From the ETICS repository
- From the Globus CVS repository (cvs.globus.org)

Install the software

- Install it on your desktop computer OR
- Ask your institute to make a central installation OR
- Ask your VO to make a central installation

More Information

- Gridway webpage: **www.gridway.org**
- Application porting with GridWay

http://www.gridway.org/successstories/applicationporting.php

- Infrastructures using GridWay

http://www.gridway.org/successstories/projectsinfrastructures.php

GANGA

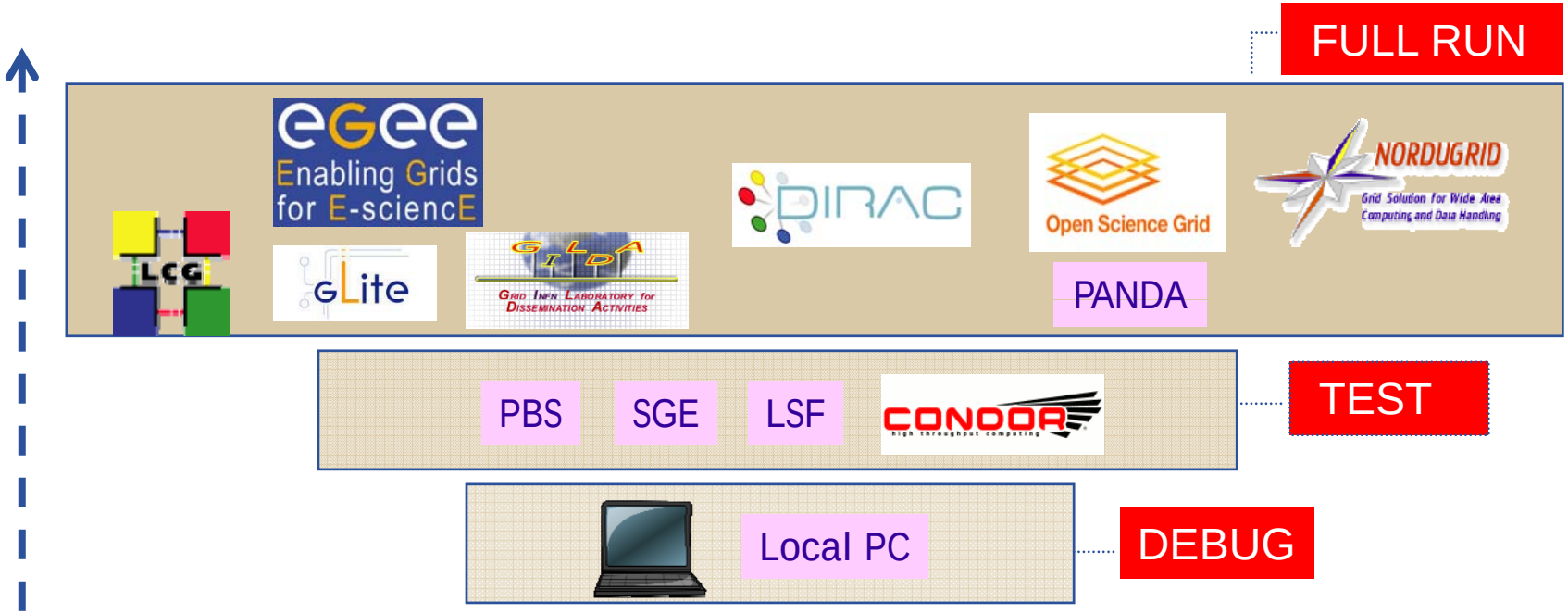
Introduction

Goals:

- provide a simple and consistent way of preparing, organising and executing jobs on different computing infrastructures
- provide a clean interface which can be used:
  - interactively (CLI / python interpreter)
  - as a Python API for scripting
  - through a GUI
- Make it easy and integrated with application environment
- Allow quick transition between local PC, cluster, Grid...
- Organize work, keep history of jobs,...

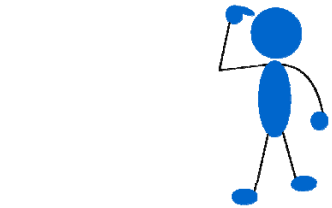
Motivation

- **In practice users deal with multiple computing backends**



Motivation

- **FAQ: running applications on multiple computing backends**



I must learn many interfaces

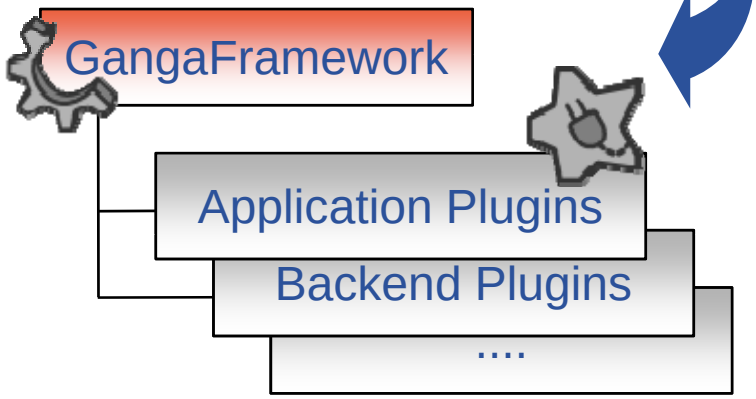
How to configure my applications?

Do I get a consistent view on all my jobs?

Ganga

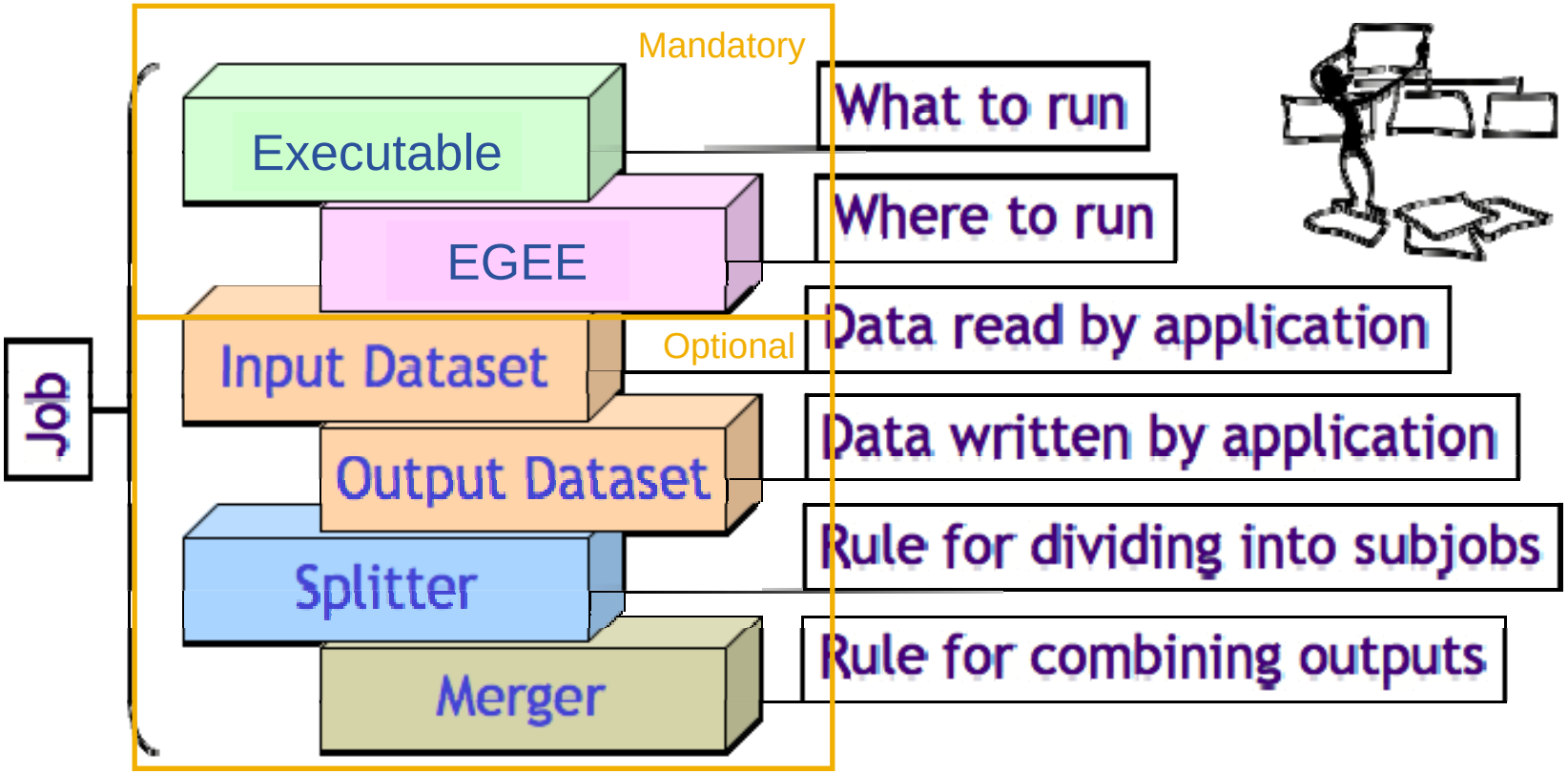
- **Ganga: Job Management Interface**
  - a utility which you download to your computer
    - or it is already installed in your institute in a shared area
      - *for example: /nfs/sw/ganga/install/4.3.2*
    - it is an **add-on** to installed software
  - comes with a set of plugins for some applications
    - **open** - other applications and backend may be easily added
      - *even by users*

Ganga
Application Software
LSF Client
LCG UI



Ganga Job

Where the Ganga journey starts …



Example Application: the ATLAS data analysis application

Scripting mode

```
$ ganga athena \
--inDS myInputDataset.txt\
--outputdata myOutput.root \
--split 3 \
--maxevt 100 \
--lsf \
jobOptions.py
```

quick

flexible

CLIP mode

```
j = Job()
j.application=Athena()
j.application.prepare()
j.application.option_file='jobOptions.py'

j.inputdata=DQ2Dataset()
j.inputdata.type='DQ2_LOCAL'
j.inputdata.dataset="myInputDataset.txt"

j.outputdata=DQ2OutputDataset()
j.outputdata.outputdata=['myOutput.root']

j.splitter = AthenaSplitterJob(numsubjobs=3)
j.merger = AthenaOutputMerger()

j.backend = LSF()
j.submit()

j2 = j.copy()
j2.backend=LCG( CE='ce102.cern.ch:2119/jobmanager-lcglsf-grid_2nh_atlas' )
j2.submit()
```

application

inputdata

outputdata

Splitter & Merger

More information

- **Ganga Home:**
  http://cern.ch/ganga

- **Official Ganga User's Guide:**
  http://cern.ch/ganga/user/html/GangaIntroduction/

- **GangaTutorial GPI Reference Manual :**
  http://ganga.web.cern.ch/ganga/release/4.3.2/reports/html/Manuals/GangaTutorialManual.html

- **Looking for help:**
  project-ganga-developers@cern.ch

Summary

- **gLite services can be accessed through programming APIs too**
  - GFAL API, WMProxy, SEE-GRID File Management
- **AMGA**
  - Metadata management on the Grid
- **R-GMA**
  - Relational database that mediates between your job and you
- **RESPECT program: EGEE NA4 initiative to identify useful tools that work and has user support**
  - List of software is at http://egeena4.lal.in2p3.fr/
  - GridWay: Broker and higher level command line client
    - Alternative to WMS; parametric jobs
  - GANGA
    - Object oriented cmd line interface for WMS; parametric jobs, splitter, merger components
  - P-GRADE:
    - Web portal, workflow and parameter study support

Questions?