



Enabling Grids for E-science

Resource integration in gLite based grid infrastructures

*G. Borges, M. David, J. Gomes, J.P. Martins,
M. Montecelo, N. Dias, J. Lopez, A. Simon, E. Garcia*

IBERGRID Conference

Porto, Portugal

12th-15th May 2008



■ gLite middleware...

- Deployed on top of a unique OS
- Interfaced with a limited set of LRMS
- Assumed to be installed on dedicated resources.

■ There is no support/information on...

- How institutes owning non-gridified resources should configure and install grid middleware without losing their local configurations

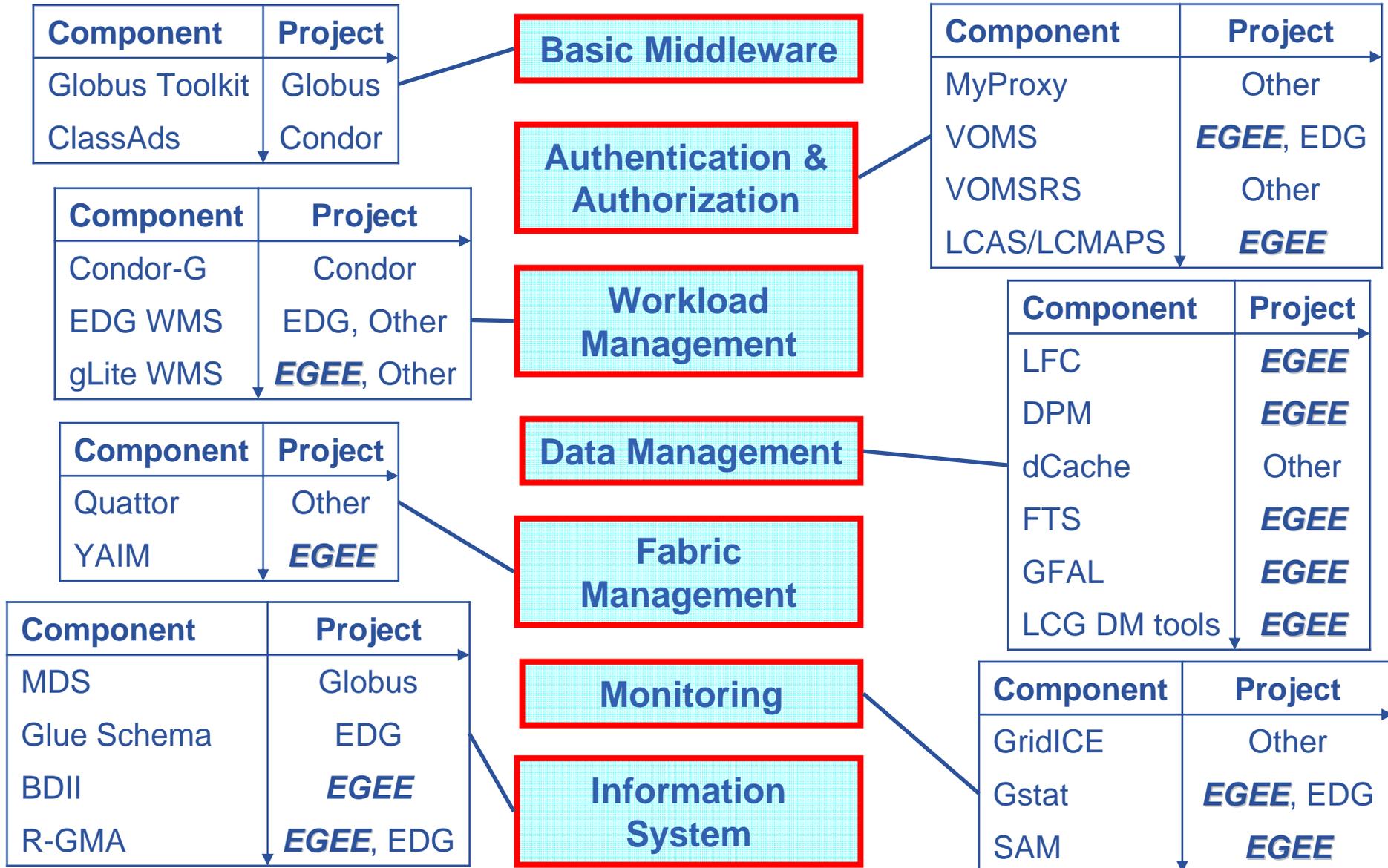
■ This presentation aims to...

- Clarify generic issues that a site admin may be faced while installing/configuring a site for a gLite based grid infrastructure
- Clarify how to configure local non-gridified production resources as gLite services
 - Becoming able to serve both local and grid requests !!!



gLite middleware “puzzle” (components)

Enabling Grids for E-science



■ Two gLite versions

- gLite 3.0: Compiled on SLC3
- gLite 3.1: Compiled on SLC4

■ The middleware is distributed as RPM's

- Dependencies issues may appear while trying to deploy gLite under a different OS than SLC
 - Use Red Hat Enterprise Linux compatible OS
 - A few additional packages can only be found in SLC repositories

■ Your package manager software should be configured...

- To give priority to your original OS repository during the installation mechanism
- To fetch the missing packages from SLC repositories

■ yum: The gLite 3.1 package management software

```
# cat /etc/yum.repos.d/os.repo
```

```
[sl-base]
```

```
baseurl=http://linuxsoft.cern.ch/scientific/4x/i386/SL/RPMS
```

```
enabled=1
```

```
protect=1
```

SL OS repository is marked with **protect=1**
(highest priority)

```
[slc-base]
```

```
baseurl=http://linuxsoft.cern.ch/cern/slc4X/i386/yum/os
```

```
enabled=1
```

```
protect=0
```

SLC OS repository is marked with **protect=0**
(lowest priority)

```
[slc-update]
```

```
baseurl=http://linuxsoft.cern.ch/cern/slc4X/i386/yum/updates
```

```
enabled=1
```

```
protect=0
```



Enabling Grids for E-science

gLite central and local services

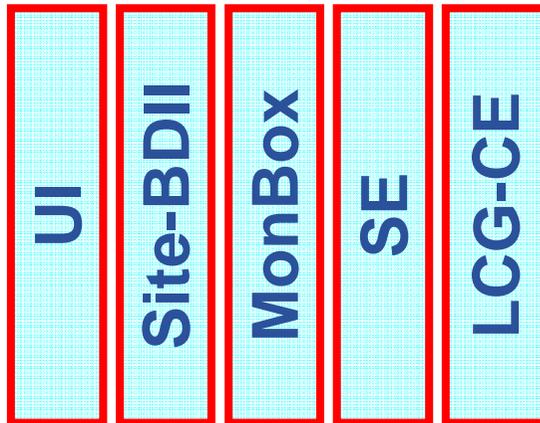
Core Services

Public ips



Local Services

Public ips



Public/Private ips



- **CE, SE, MON, SITE-BDII and UI**
 - Public IP addresses
 - Name resolution must be set up correctly to forward and reverse queries (via DNS or /etc/hosts)
 - Only the WN and the LRMS server may have private IPs

- **Network Time Protocol (NTP) properly configured and enabled in all machines**

- **Java SDK (≥ 1.5) is needed on the MON, CE, WN and UI**
 - R-GMA client tools

- **Host certificate (CE, SE and MON) deployed under “/etc/grid-security” with the correct permissions**

Node	Service	From	Port
All	NTP	NTP server	123/udp
SITE-BDII	BDII (LDAP)	All	2170/tcp
LCG-CE	GRAM	RBs, CEs	2119/tcp
	GSIFTP control	All	2811/tcp
	GSIFTP data	All	20000,25000/tcp
SE_DPM	BDII (LDAP)	Local SITE-BDII	2170
	SRMv1	All	8443/tcp
	SRMv2.2	All	8446/tcp
	GSIFTP control	All	2811/tcp
SE_DPM_POOL	BDII (LDAP)	Local site_BDII	2170/tcp
	GSIFTP	All	2811/tcp
MON	GSIFTP data	All	20000,25000/tcp
	RGMA-https	All site hosts	8443
	RGMA	MONs	8088
	ExGRIS (LDAP)	GridICE Server, Local SITE-BDII	2136
	GRIS (LDAP)	GridICE Server	2135

- Your package manager (yum for gLite 3.1) should point to a specific set of repositories, in addition to the OS one

```
# gLite 3.1 middleware
http://linuxsoft.cern.ch/EGEE/gLite/R3.1
# Certification Authorities
http://grid-deployment.web.cern.ch/grid-deployment/yaim/repos/lcg-CA.repo
# JPackage
http://grid-deployment.web.cern.ch/grid-deployment/yaim/repos/jpackage.repo
# DAG
http://linuxsoft.cern.ch/dag/redhat/el4/en/\$basearch/dag
```

- For each node/service there is an **empty meta package containing all RPM dependencies**
 - The installation of a single RPM will pull all the necessary software
 - The CA's certificates have to be installed separately
- List of gLite service combinations known to work:
 - UI/MON, UI/SE_dpm, CE/SITE-BDII/TORQUE_server

■ Install the service(s) meta-package with “yum”

```
# yum install <Service_Meta_Package>
```

■ Don't forget the CA's certificates...

```
# yum install lcg-CA
```

■ Configure the services with “YAIM”

- gLite tool to configure grid services
- The proper YAIM module is installed by the meta-packages
- Deployed under “/opt/glite/yaim/”
- Needs some configuration files to be filled by the site admin
 - site-info.def, users.conf, groups.conf

```
# /opt/bin/yaim/bin/yaim -c -s site-info.def -n <Service_Target_1> -n ...
```

Meta packages & Configuration Targets

```
# /opt/bin/yaim/bin/yaim -c -s site-info.def -n <Config_Target>
```

```
# yum install <Meta_Package>
```

<i>Service Description</i>	<i>Meta Package</i>	<i>Config Target</i>
LCG Computing Element	lcg-CE	lcg-CE
Utilities for Torque	glite-TORQUE_utils	TORQUE_utils
gLite worker node	glite-WN	WN
Torque client tools	glite-TORQUE_clients	TORQUE_clients
Torque batch system server	glite-TORQUE_server	TORQUE_server
Berkeley Database Information Index	glite-BDII	BDII_site
gLite DPM with mysql backend	glite-SE_dpm_mysql	SE_dpm_mysql
gLite DPM pool	glite-SE_dpm_disk	SE_dpm_disk
gLite User Interface	glite-UI	UI
Monitoring Box	glite-MON	MON
CA's certificates	lcg-CA	lcg-CA

■ Assumed cenario...

- Torque server already in production locally
 - Installed in a dedicated/isolated machine
- Torque clients already in production
 - Local cluster execution machines
- Different Torque version than the one distributed with gLite 3.1
 - Torque-2.1.9-4cri.slc4
- MAUI is not used !!!

■ Defined strategy...

- Deploy the lcg-CE on a dedicated machine
 - And properly link it to the current Torque Server
- Install glite-WN middleware (except the distributed torque tools) on top of the local cluster execution machines
 - Without losing local configurations

■ lcg-CE deployment...

- Install the “lcg-CE” meta-package: The middleware itself
- Install the glite-Torque-utils metapackage
 - Torque interfaces with lcg-CE: Information plugin, JobManager, etc ...
 - Remove unnecessary Torque packages

```
# yum install lcg-CE lcg-CA glite-TORQUE_utils
# rpm -e --nodeps glite-yaim-torque-server torque torque-client
```

- Install the preferred Torque version
 - *Binary client tools should be available on the CE*
- Declare the CE host has an allowed Torque submission machine
- Set up a mechanism to transfer Torque accounting logs to the CE
 - *Through a daily cron job to the directory defined in \$BATCH_LOG_DIR in the site-info.def YAIM configuration file*
 - *Through NFS*

■ Set up your site-info.def relevant variables

```
CE_HOST=<CE hostname>.$MY_DOMAIN
BATCH_SERVER=<LRMS Server hostname>.$MY_DOMAIN
JOB_MANAGER=lcgpbs
CE_BATCH_SYS=torque
BATCH_BIN_DIR=<Path to torque binaries>
BATCH_VERSION=<torque version>
BATCH_LOG_DIR=<path for accounting records>
VOS=<list of VOs supported at the site>
QUEUES=<list of VO queues supported at the site>
GRID_GROUP_ENABLE=<VOs names and VOs user roles>
```

○ Configure your lcg-CE

```
# /opt/glite/yaim/bin/yaim -c -s site-info.def -n lcg-CE -n TORQUE_utils
```

○ If you are not using MAUI...

- Comment the following line in **lcg-info-dynamic-scheduler.conf** file

```
vo_max_jobs_cmd: /opt/lcg/libexec/vomsjobs.maui -h <hostname>
```

■ Follows the same strategy as for the lcg-CE deployment

- Install the glite-WN and glite-Torque_client meta-packages
- Remove the unnecessary/unwanted software

```
# yum install glite-WN lcg-CA glite-TORQUE_client  
# rpm -e --nodeps torque-mom torque torque-client
```

- Install the required Torque client tools
- Configure the service

```
# /opt/glite/yaim/bin/yaim -c -s site-info.def -n WN -n TORQUE_client
```

- Ensure that the “grid users” home dirs, UIDs and GIDs are the same as in the CE
- Ensure that a “grid user” can perform passwordless ssh to the CE from a WN
- Configure the VO grid queues on the Torque Server

- (Limited!) Icg-CE integration with other LRMS: **SGE, LSF, Condor**
- The support is based on the same software architecture strategy:
 - Interface meta package called "glite-<LRMS>_utils"
 - APEL <LRMS> parser, <LRMS> Jobmanager, <LRMS> information script
 - YAIM plugin used to configure Icg-CE interaction with the LRMS
 - Client meta package called "glite-<LRMS>_client"
 - YAIM plugin used to configure glite-WN interaction with the LRMS clients
 - Server meta package called "glite-<LRMS>_server"
 - YAIM plugin used to configure the LRMS server
- Site admins should install the LRMS server and client commands

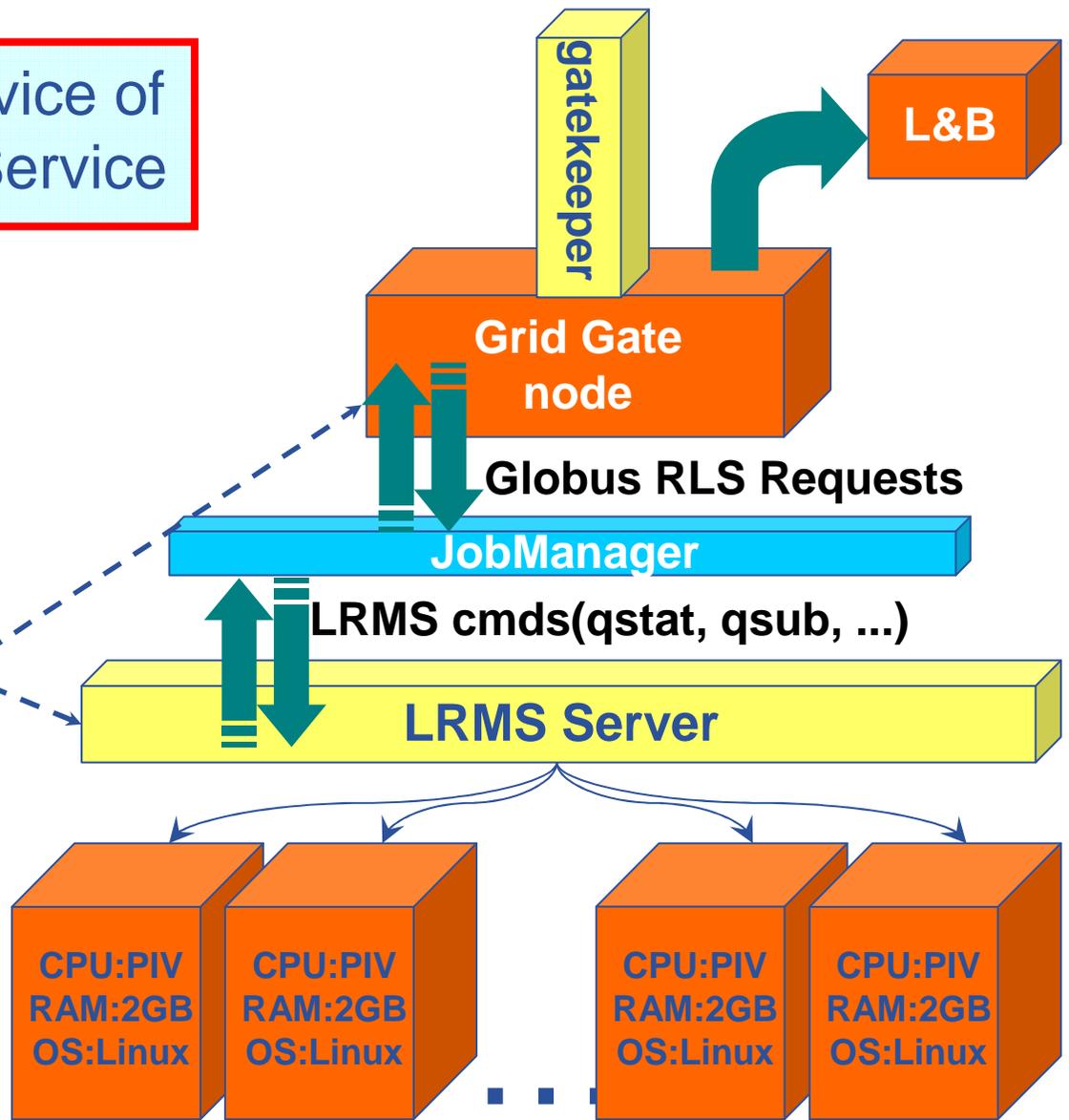
```
# /opt/glite/yaim/bin/yaim -c -s site-info.def -n Icg-CE -n <LRMS>_utils
```

```
# /opt/glite/yaim/bin/yaim -c -s site-info.def -n WN -n <LRMS>_client
```

```
# /opt/glite/yaim/bin/yaim -c -s site-info.def -n <LRMS>_server
```

The JM is the core service of the Globus GRAM Service

Submits jobs to the LRMS based on Globus requests and through a jobwrapper script

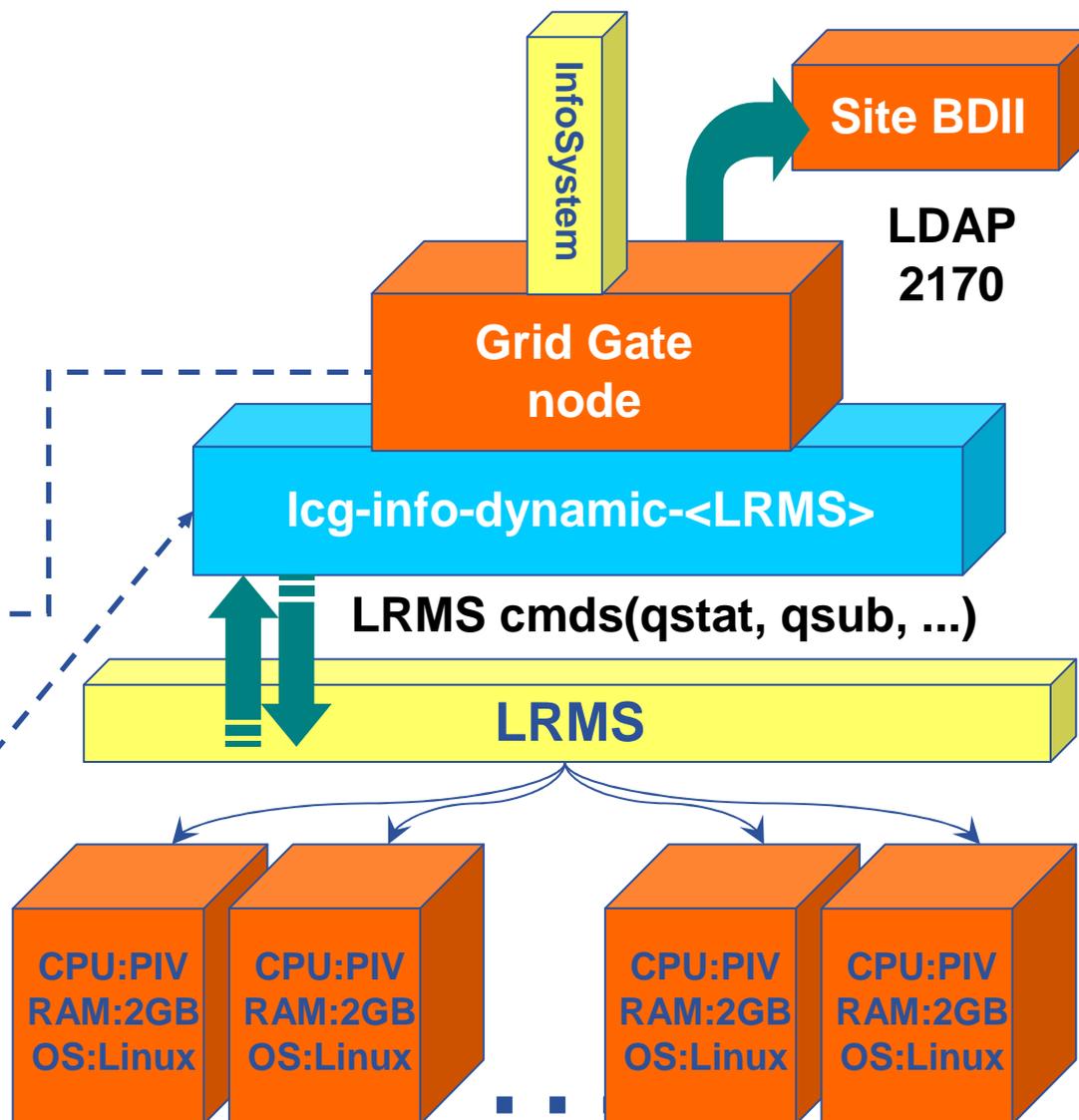


Other LRMS integration steps: Info Plugin

The information dynamic provider should be a standalone Information plugin script that examines the queuing system state

Information expected to be reported is based on queues

The info reporter read a copy of a static Idif file

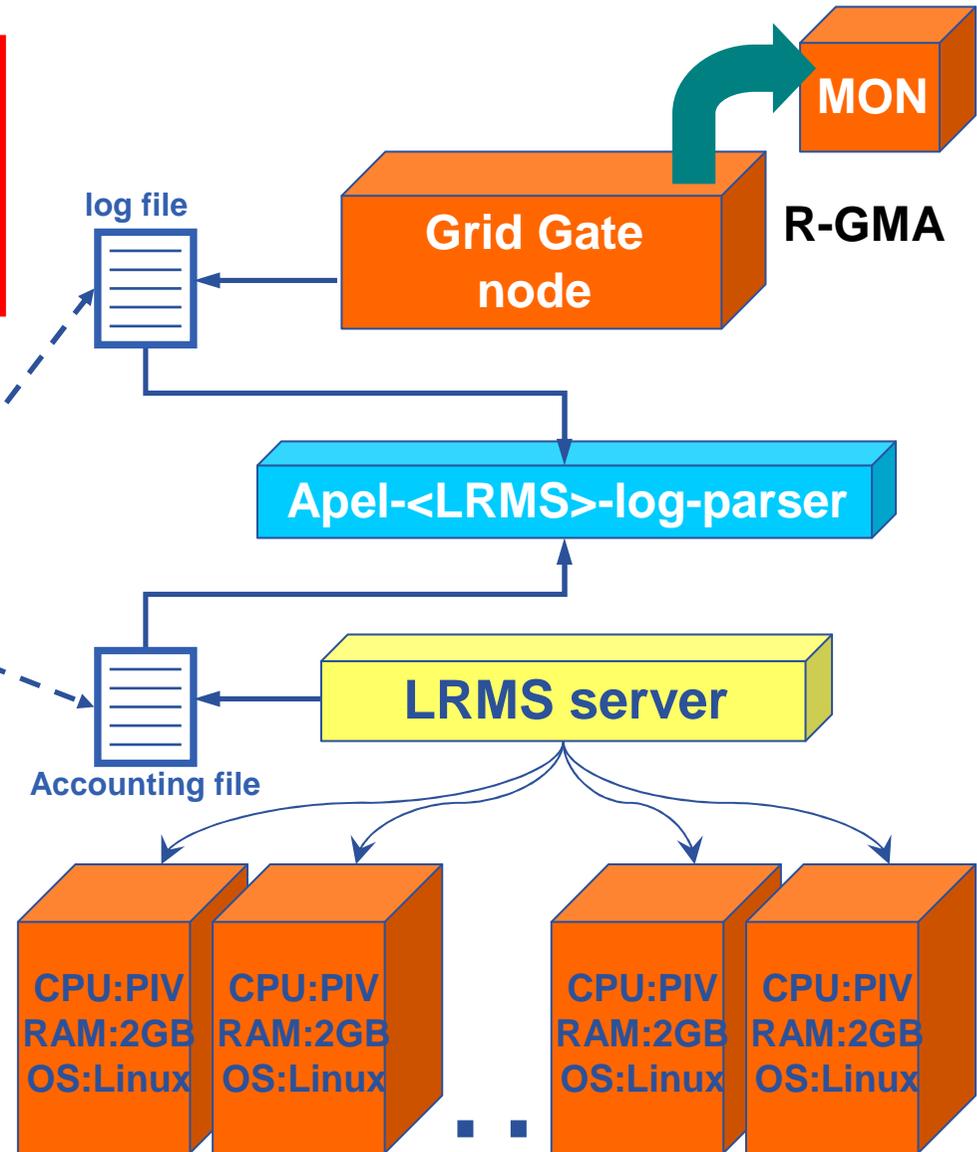


Other LRMS integration step: Apel Plugin

APEL LRMS plug-in is a log processing application used to produce CPU job accounting records

Interprets GK & batch system logs which together with the gridinfo mappings from the JM are joined together to form accounting records

- Published using R-GMA to an accounting database.



- **Two types of SEs are part of the gLite stack**
 - DPM and dCache: Both are based on the implementation of a virtual namespace stored in a DB exposed to the user
 - Hides the actual physical location of the files

- **The “Disk Pool Manager (DPM)”**
 - Suited to small sites (less than 10 TB of disk based storage)
 - Only supports pool of disks
 - Suited to grid users using GSI authentication
 - Kerberos authentication or UNIX UID/GID not well supported

- **The “dCache” system**
 - Developed outside the grid world
 - Good support for GSI, Kerberos and UNIX authentication
 - Supports pool of disks connected to Mass Storage Systems
 - dCache namespace “PNFS” can be mounted on clients via NFS
 - Standard unix operations are allowed: ls, mkdir
 - IO is performed via dedicated dCache library called “dcap”

■ Nothing is so easy as it sounds...

- Real life could be much more dramatic!
- Be afraid, be very afraid... And prepared!

■ These instructions are aimed to...

- Clarify some of the generic issues while installing and configuring grid services
- Provide some usefull guidelines to allow local resources to be also configured as grid services
 - Allow interoperability with local non-gridified infrastructures
- Provide some directives on how to implement LRMS plugins different from the ones distributed with gLite
 - Deploy LRMS different from TORQUE/MAUI