



Enabling Grids for E-scienceE

Comparison of LCG-2 and gLite v1.0

*Author E.Slabospitskaya
NA3 manager for Russia,
IHEP*

17 April 2005

www.eu-egee.org



Information Society



INFSO-RI-508833

- **What are LCG-2 and gLite?**
- **gLite Architecture Release 1.0 review**
 - What is gLite?.
 - Schema of gLite Services. The main subsystems of gLite. Comparison to LCG-2.
 - Security
 - Job management Services (CE,WMS)
 - Data Management
 - Information System
 - Clients (WN, UI, API)



Current status of gLite – first major release

- **Mapping to physical machines**

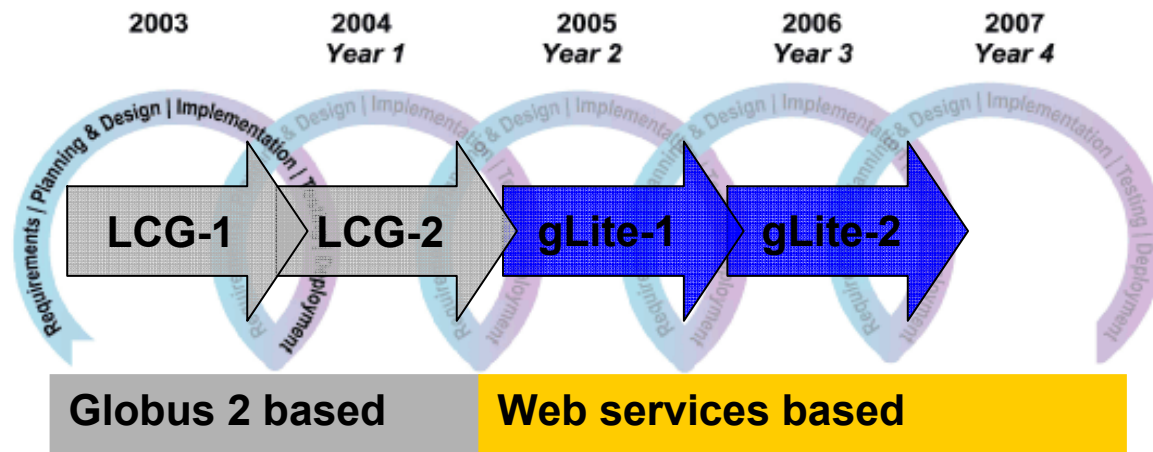
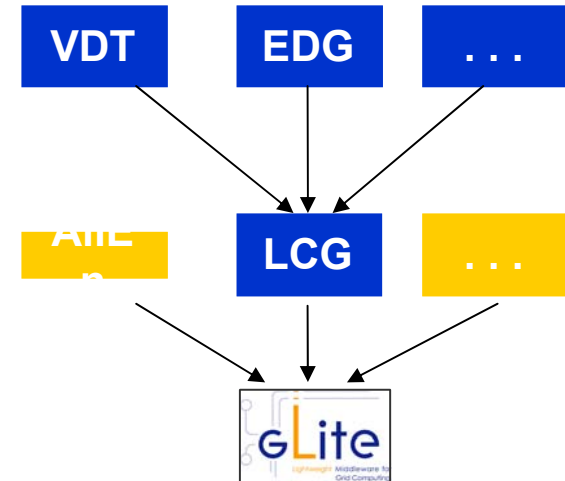
<http://glite.web.cern.ch/glite>

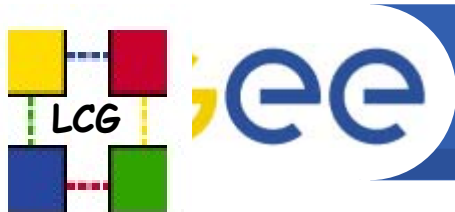
<http://www.glite.org>

- Architecture: <https://edms.cern.ch/document/476451>
- Design: <https://edms.cern.ch/document/487871>

gLite Software and association documentation. Release 1.

<https://edms.cern.ch/document/567624>



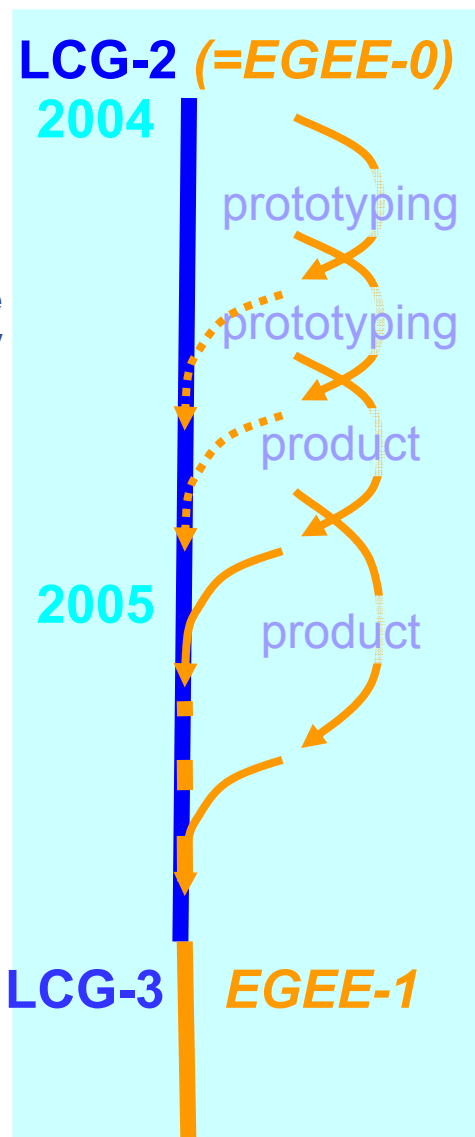


gLite and LCG-2

LCG-2

focus on production, large-scale data handling

- The service for the 2004 data challenges
- Provides experience on operating and managing a global grid service
- Development programme driven by data challenge experience
 - Data handling
 - Strengthening the infrastructure
 - Operation, VO management
- Evolves to LCG-3 as components progressively replaced with new middleware
 - target is to minimise the discontinuities of migration to the new generation
- Aim for migration plan by end of year



gLite

focus on analysis

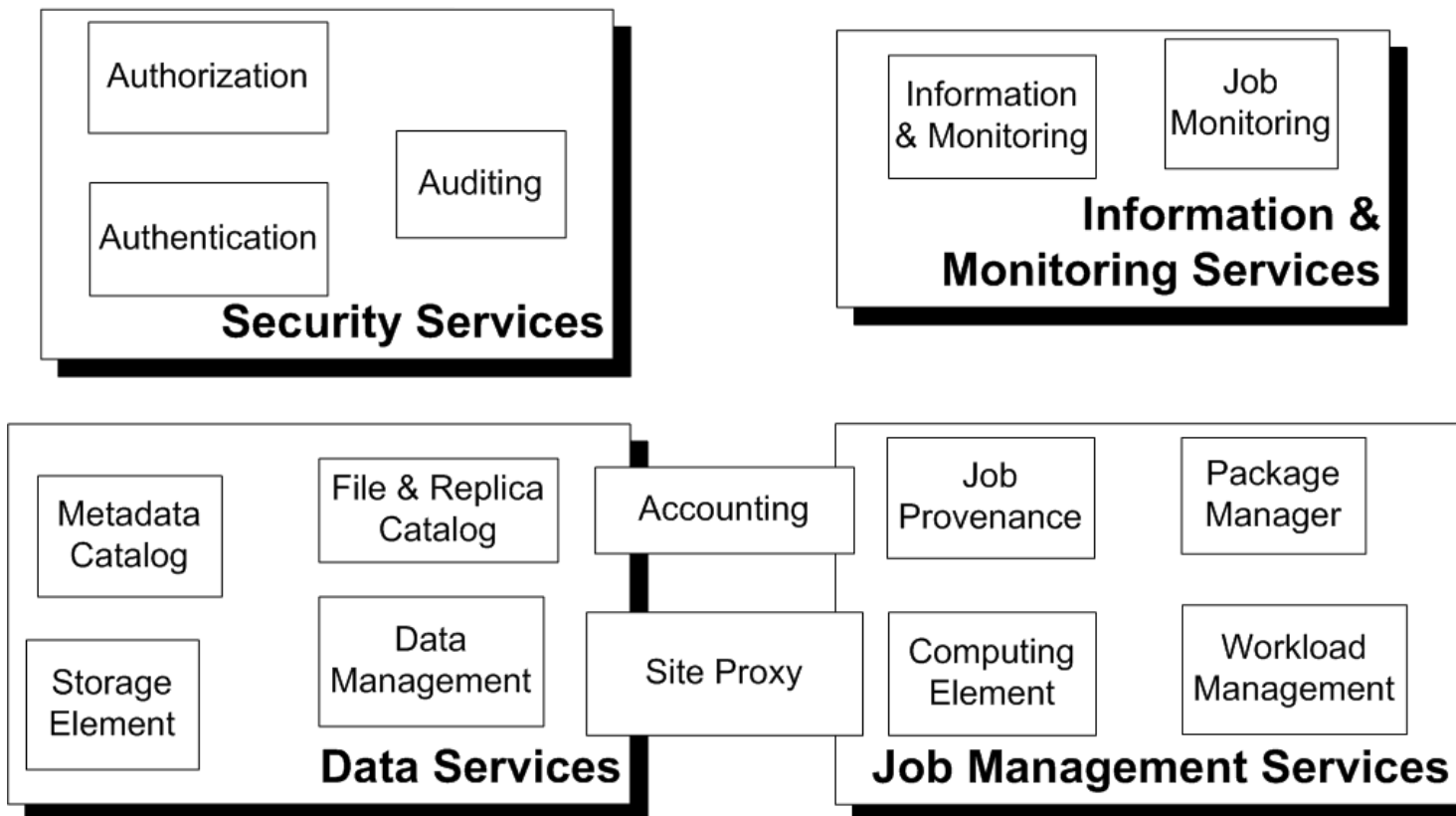
- Developed by EGEE project in collaboration with VDT (US)
- LHC applications and users closely involved in prototyping & development (ARDA project)
- Short development cycles
- Co-existence with LCG-2
- Profit as far as possible from LCG-2 infrastructure, experience
- Ease deployment – avoid separate hardware
- As far as possible - completed components integrated in LCG-2
- improved testing, easier displacement of LCG-2

les robertson - cern-it-4

The gLite middleware is a Service Oriented Grid middleware providing services for managing distributed computing and storage resources and the required security and information services.

gLite is a set of middleware services which, although supposed to work together in a concerted manner, can be used independently.

Target server platform is Red Hat Linux 3.0 or any binary compatible distribution, such as Scientific Linux and Windows.



- In industry, several security standards exist:
 - **Public Key Infrastructure (PKI)**
 - PKI keys
 - SPKI keys (focus on authorisation rather than certificates)
 - RSA
 - **Secure Socket Layer (SSL)**
 - SSH keys
 - **Kerberos**
- Need for a common security standard for Grid services
 - Above standards do not meet all Grid requirements (e.g. delegation, single sign-on etc.)
- Grid community mainly uses **X.509 PKI** for the Internet
 - Well established and widely used (also for www, e-mail, etc.)

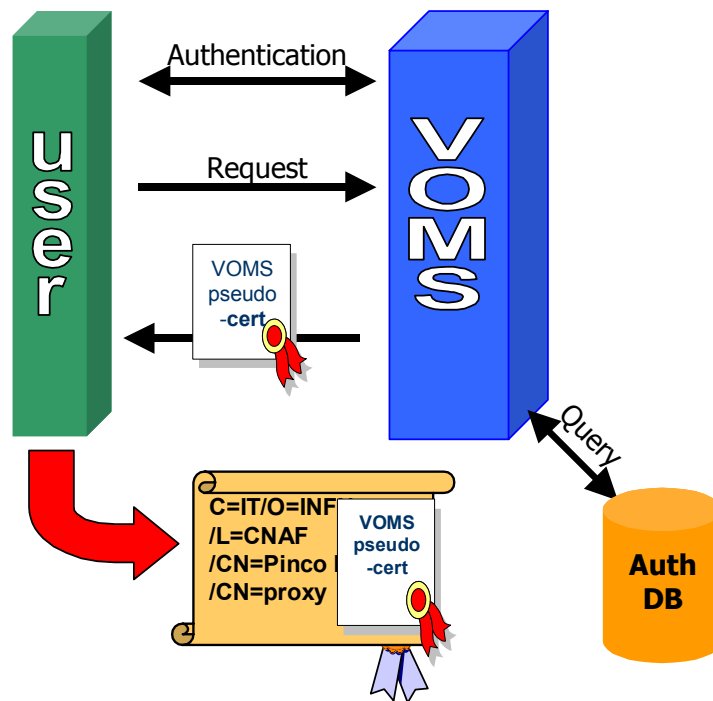
Flavia Donno

Induction: Overview – May 18th, 2004 - 8

- Globus Toolkit™ proposed and implements the Grid Security Infrastructure (GSI)
 - Protocols and APIs to address Grid security needs
- GSI protocols **extend standard public key protocols**
 - Standards: X.509 & SSL/TLS
 - Extensions: X.509 Proxy Certificates (single sign-on) & Delegation
- GSI extends standard GSS-API (Generic Security Service)
 - The GSS-API is the IETF standard for adding authentication, delegation, message integrity, and message confidentiality to applications.
- **Proxy Certificate:**
 - Short term, restricted certificate that is derived from a long-term X.509 certificate
 - Signed by the normal end entity cert, or by another proxy
 - Allows a process to act on behalf of a user
 - Not encrypted and thus needs to be securely managed by file system

Flavia Donno

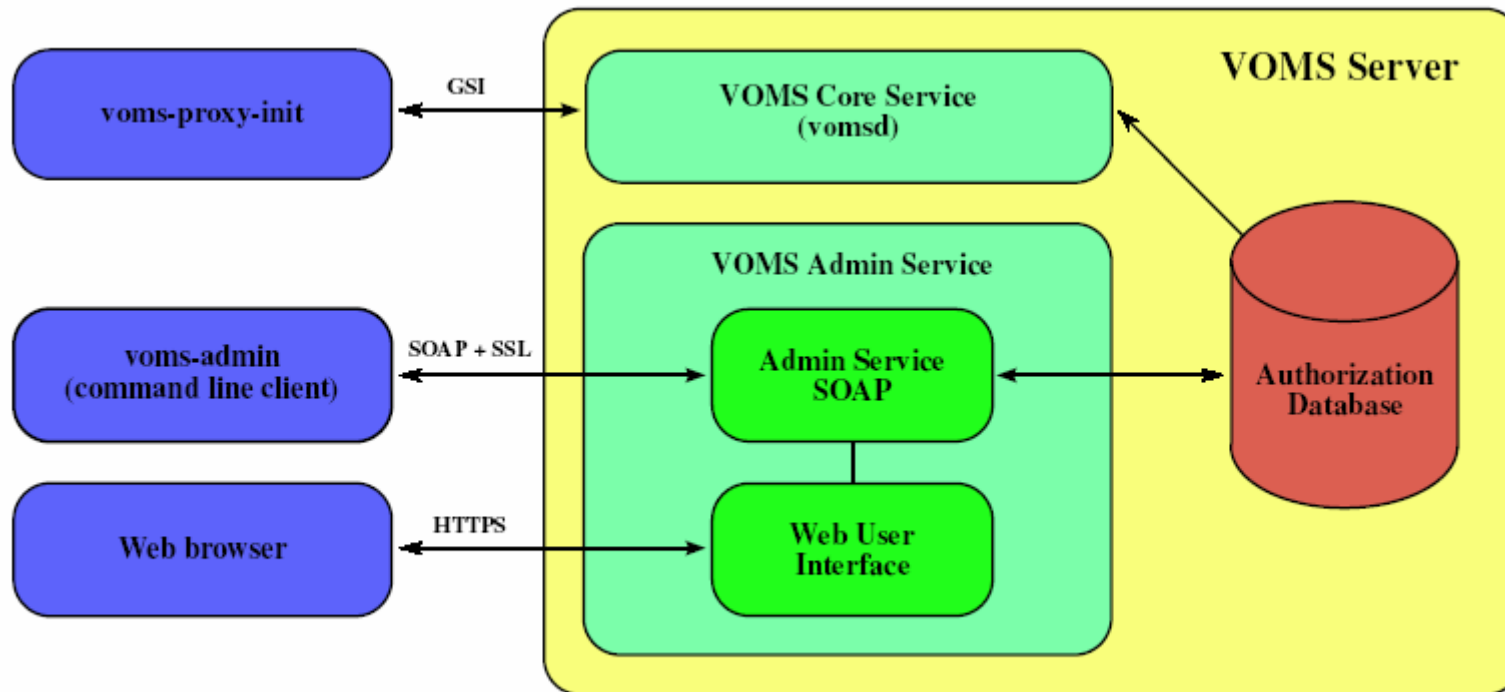
VOMS Operations



1. Mutual authentication Client-Server
 - Secure communication channel via standard Globus API
2. Client sends request to Server
3. Server checks correctness of request
4. Server sends back the required info (signed by itself) in a "Pseudo-Certificate"
5. Client checks the validity of the info received
6. Optionally: [Client repeats process for other VOMS's]
7. Client creates proxy certificates containing all the info received into a (non critical) extension
8. Client may add user-supplied auth. info (kerberos tickets, etc...)

Flavia Donno

Based on: <http://www.slac.stanford.edu/econf/C0303241/proc/pres/317.PPT>



- A Computing Element interfaces the local resource management system (e.g. LSF, PBS) to the Grid middleware.
- The Worker Nodes behind the local resource management system host all the necessary clients to interact with the Grid middleware from within a job.
- Workload Management System (WMS)
- Logging and Bookkeeping Server

Workload Management System (WMS) is usually run at Resource Broker.

[Network Server \(NS\)](#), which accepts the incoming job requests from the UI, and provides for the job control functionality.

- [Workload Manager](#), which is the core component of the system.

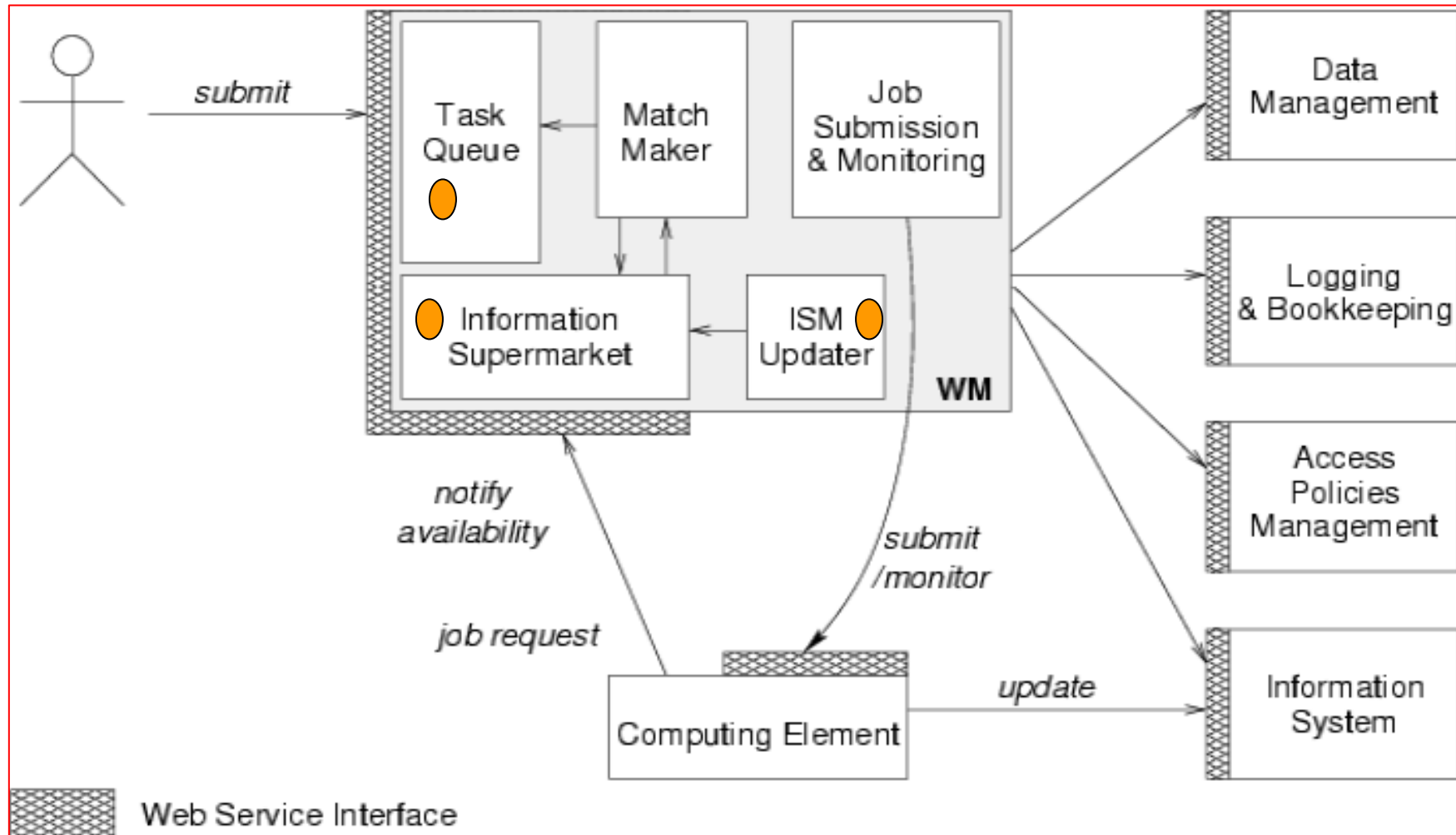
- [Match-Maker](#) (also called Resource Broker), whose duty is finding the best resource matching the requirements of a job (match-making process).

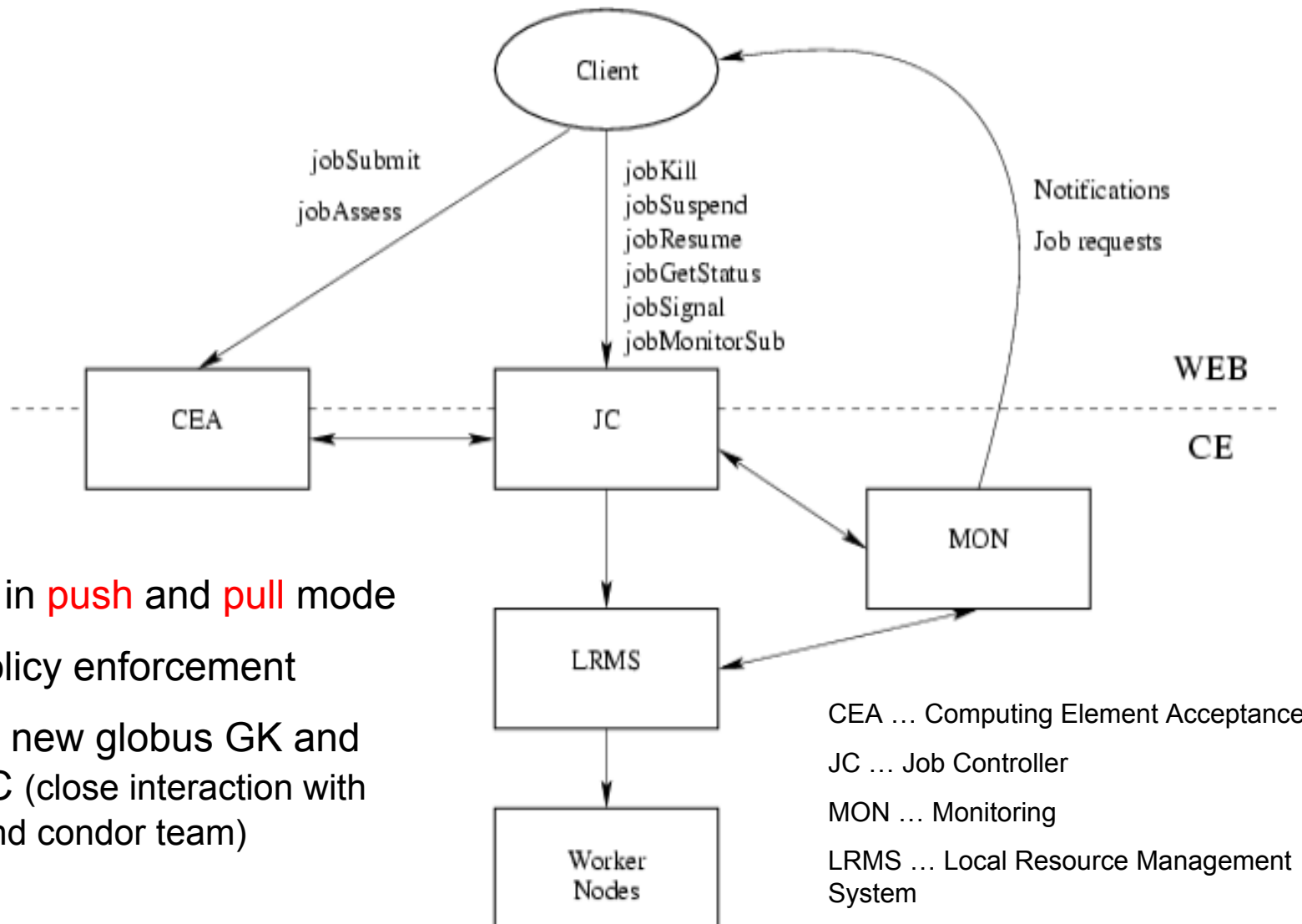
- [Job Adapter](#), which prepares the environment for the job and its final description, before passing it to the Job Control Service.

- [Job Control Service \(JCS\)](#), which finally performs the actual job management operations (job submission, removal...)

- [Logging and Bookkeeping service \(LB\)](#) . The LB logs all job management Grid events, which can then be retrieved by users or system administrators for monitoring or troubleshooting.

- Differ from LCG-2





- Works in **push** and **pull** mode
- Site policy enforcement
- Exploit new globus GK and CondorC (close interaction with globus and condor team)

REPLICA MANAGEMENT SYSTEM (RMS)

The main services offered by the RMS are: the **Replica Location Service** (RLS) and the **Replica Metadata Catalog** (RMC).

The **RLS** maintains information about the physical location of the replicas (mapping with the GUIDs). It is composed of several Local Replica Catalogs (LRCs) which hold the information of replicas for a single VO.

The **RMC** stores the mapping between GUIDs and the respective aliases (LFNs) associated with them, and maintains other metadata information (sizes, dates, ownerships...)

The last component of the Data Management framework is the **Replica Manager**. The Replica Manager presents a single interface for the RMS to the user, and interacts with the other services.

Initially, 4 obvious differences between the gLite and LCG2 models are noted in data management:

1. ***Outbound connectivity***: gLite provides the File Placement Service (FPS) to **asynchronously** copy files to remote sites, via a request-submit poll-status pattern. lcg-utils has a **synchronous** remote copy function (e.g.gridFTP from WN)
2. ***Security model***: In LCG-2 the user's proxy is used directly to access to storage. It is expected to authorise the user. In gLite user owns all the files on the storage. Access to files is mediated via services that uses ACL.
3. ***Information System and Service discovery***.
4. ***gLite storage*** is always assumed to be **SRM**. The Classic SE is not supported.

3 main service groups that relate to data and file access are:

- Storage Element
- Catalogue Services
- Data Transfer Scheduling .

Closely related to the data services are the security-related services and the Package Manager.

- Data is stored in an SRM based storage system.
- The gLite-I/O server allows posix-like access to files stored in the SRM
- The local catalogue (LC) keeps track of the LFN:GUID:SURL mapping of local files

The file transfer/file placement service is used for moving files.

- **File Catalog**

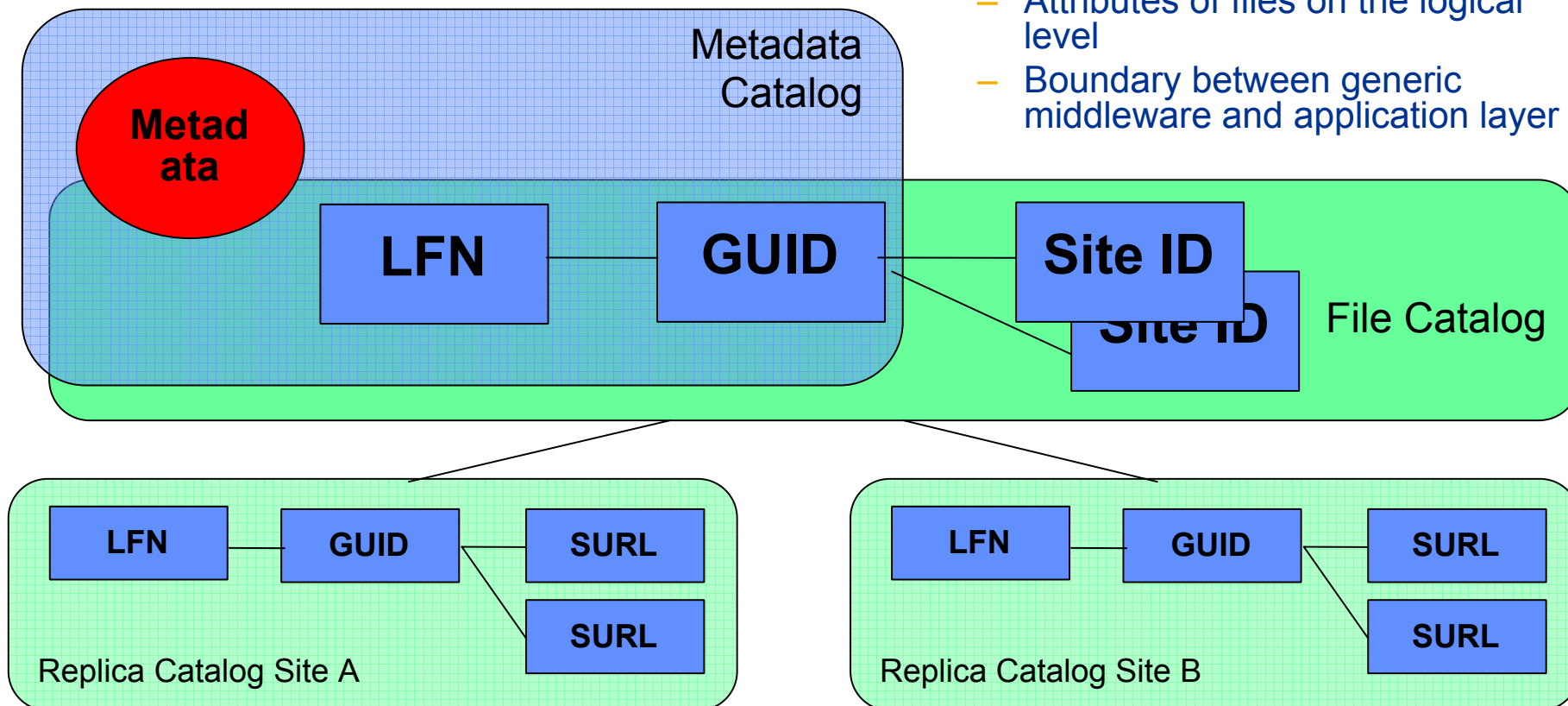
- Filesystem-like view on logical file names
- Keeps track of sites where data is stored
- Conflict resolution

- **Replica Catalog**

- Keeps information at a site

- **(Meta Data Catalog)**

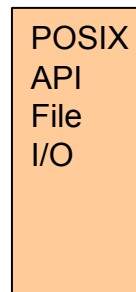
- Attributes of files on the logical level
- Boundary between generic middleware and application layer



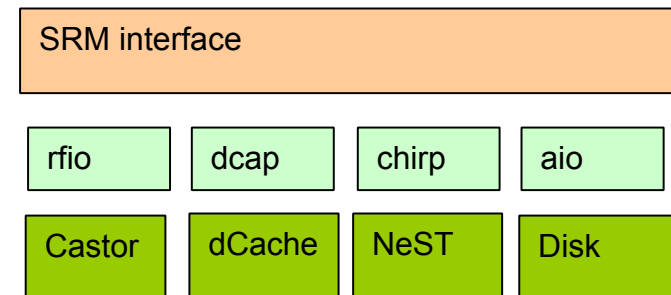
- **SRM interface**
 - Management and control
 - SRM (with possible evolution)

- **Posix-like File I/O**
 - File Access
 - Open, read, write
 - Not real posix (like rfio)

User



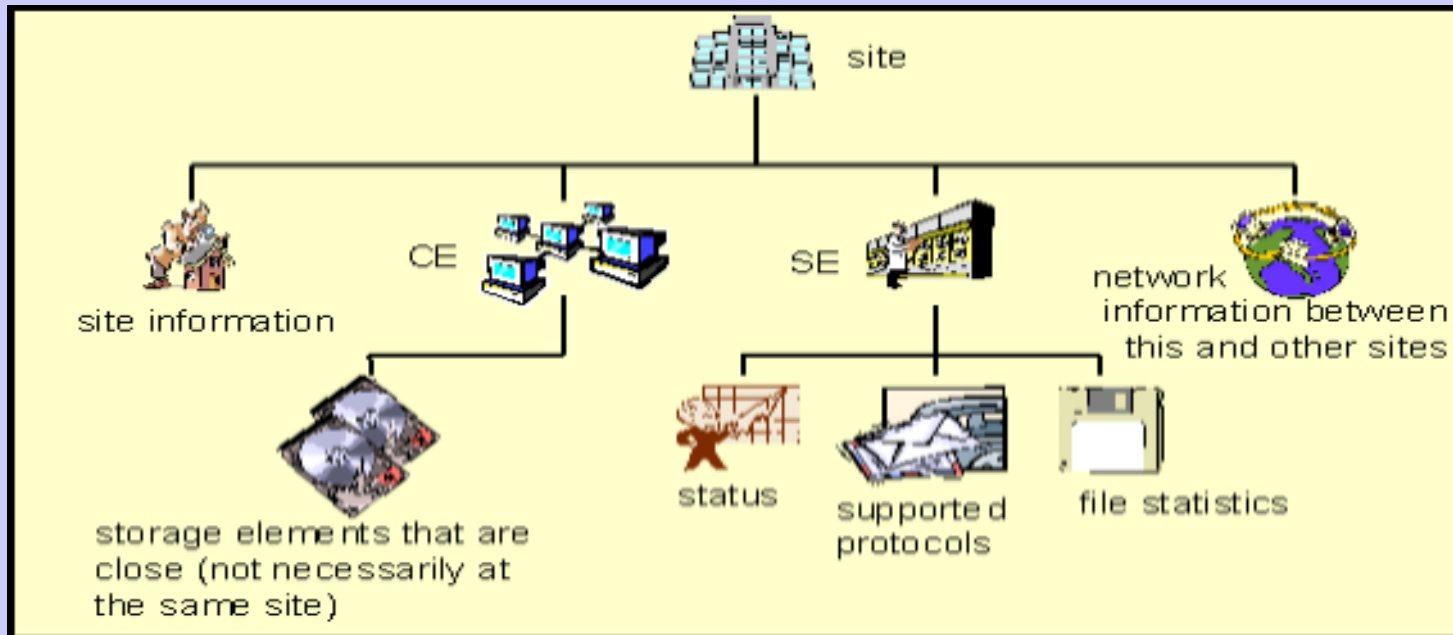
Control



MDS

Directory Information Tree

A LDAP Information System is based on entries. Each entry describes an object – person, computer etc and has unique Distinguished Name (DN). Which kind of information can be stored in each entry is specified in an LDAP schema



Directory Information Tree (DIT) – a tree of directory entries

The **R-GMA (Relational Grid Monitoring Architecture) servlet** accepts connection from clients (producers), i.e. the services publishing and user jobs publishing information, and forwards the information to the appropriate consumers.

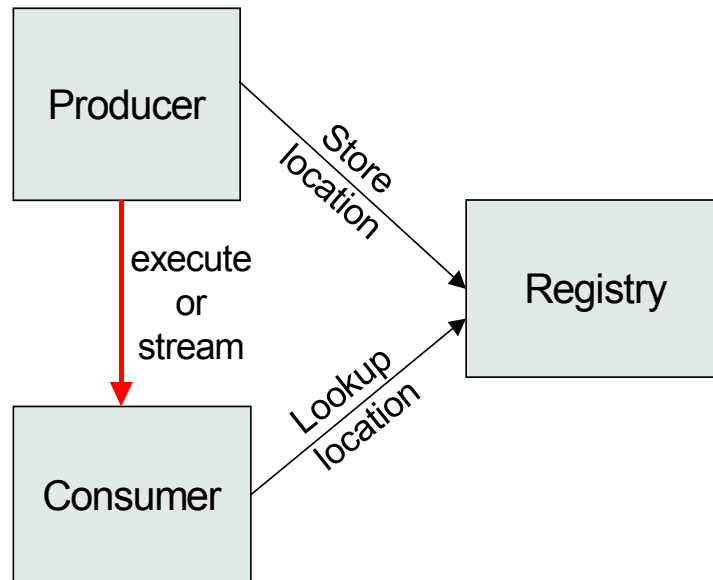
R-GMA is composed of the following services:

- a. **R-GMA server**
- b. **R-GMA client** is a set of client API in C,C++,Java and Python for the access the information and monitoring functionality of the R-GMA system
- c. **R-GMA site-publisher** (each site) is responsible for publishing site information to the R-GMA server.
- d. **R-GMA service tool** -regularly scans config files and updates a current Service Status R-GMA Table. CLI exists for modify these files and query the service table.

R-GMA is based on the Grid Monitoring Architecture (GMA) from the Grid Global Forum (GGF), which is a simple Consumer-Producer model that models the information infrastructure of a Grid as a set of consumers (that request information), producers (that provide information) and a central registry which mediates the communication between producers and consumers. R-GMA offers a global view of the information as if each Virtual Organization had one large relational database.

Producers contact the registry to announce their intention to publish data, and consumers contact the registry to identify producers, which can provide the data they require. The data itself passes directly from the producer to the consumer: it does not pass through the registry.

R-GMA adds a standard query language (a subset of SQL) to the GMA model,



- Use the GMA from GGF
- A relational implementation
 - Powerful data model and query language
 - All data modelled as tables
 - SQL can express most queries in one expression
- Applied to both information and monitoring

Flavia Donno

