LCG-2 and gLite Architecture and components

Author E.Slabospitskaya

Outline

- **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

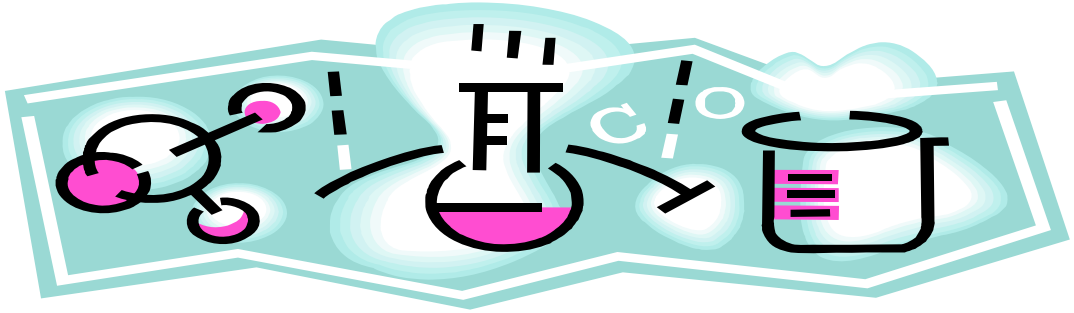
- **Mapping to physical machines**

What is gLite

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.

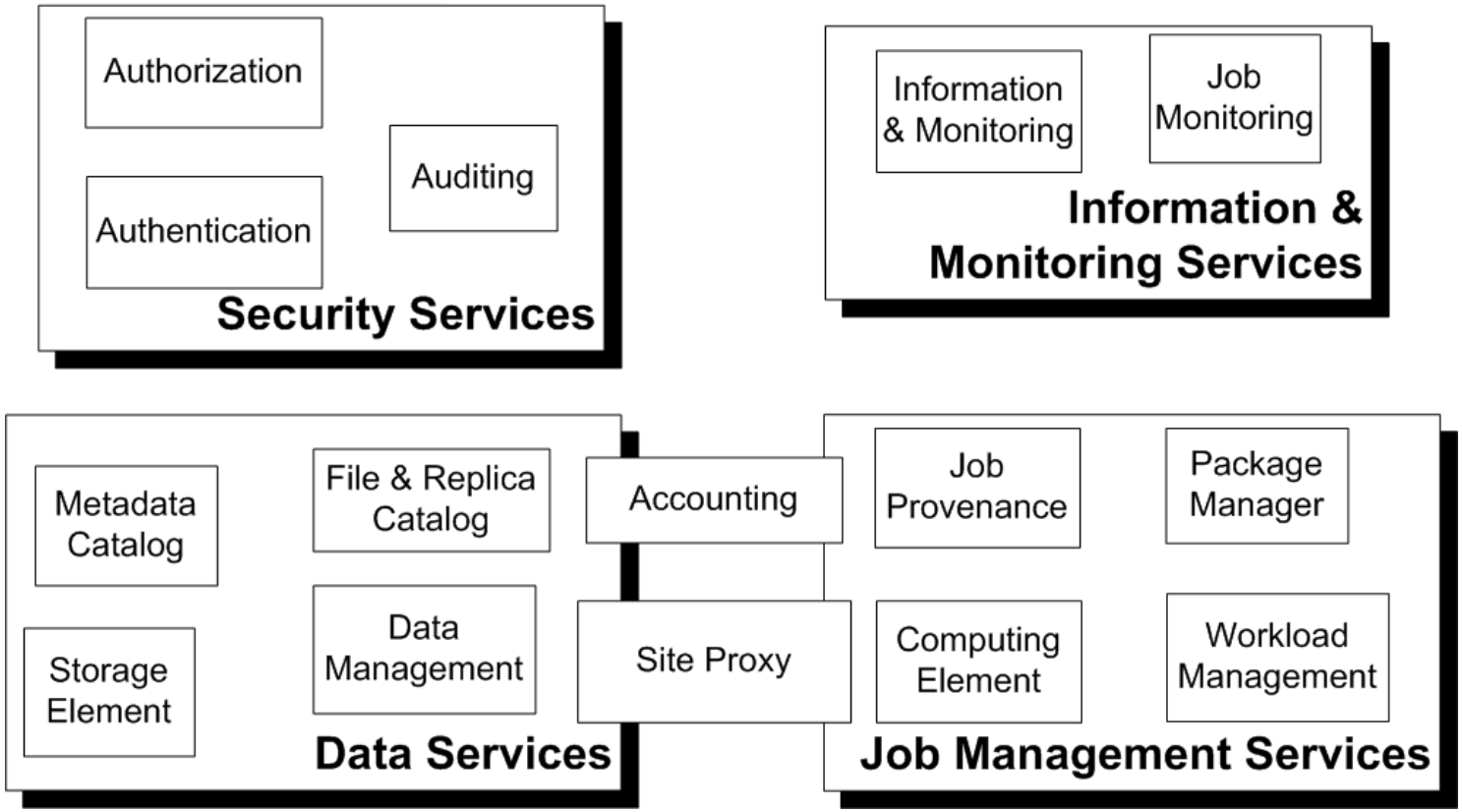


Service Oriented Architecture

- **Web Services widely used in Biomed community**

- **Existing services can be composed with grid services to provide a ‘simple’ way of bringing applications to the grid.**

gLite Services



Grid Security Infrastructure (GSI)

- 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 form 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

Overview of EGEE Middleware

- **Data Services**
  - **Storage Element**
  - **Catalog Services**
  - **Management**

- Closely related to the data services are
  - **the security-relatedservices**
  - **the Package Manager**
  -
- The granularity of the data is on the file level.
  - **generic enough to be extended to other levels of granularity.**
    - **Data sets or collections are a very common extension**
      - ***information about which files belong to a dataset maybe kept in an application metadata catalog.***

- EGEE data services users are supplied with the abstraction of a global file system.
  - **A client user application may look like a Unix shell (as in AliEn) which can seamlessly navigate this virtual file system, listing files, changing directories, etc.**

- The data in the files can be accessed through the Storage Element (SE).
- The access to the files is controlled by Access Control Lists (ACL)..

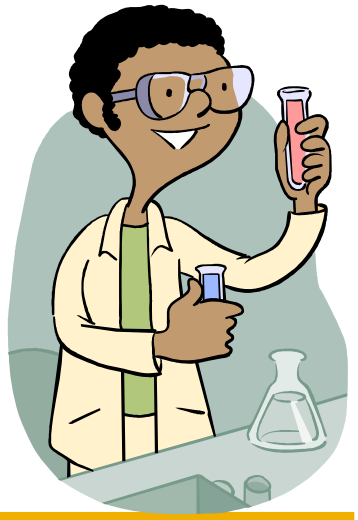
Overview of EGEE Middleware

- **Job Management Services**
  - **main services related to job management/execution are**
    - **computing element**
      - ***job management (job submission, job control, etc.), but it must also provide***
      - ***provision of information about its characteristics and status***
    - **workload management**
      - ***core component discussed in details***
    - **accounting**
      - ***special case as it will eventually take into account***
        - **computing, storage and network resources**
    - **job provenance**
      - ***keep track of the definition of submitted jobs, execution conditions and environment, and important points of the job life cycle for a long period***
        - **debugging, post-mortem analysis, comparison of job execution**
    - **package manager**
      - ***automates the process of installing, upgrading, configuring, and removing software packages from a shared area on a grid site.***
        - **extension of a traditional package management system to a Grid**

- Services communicate with each other as the job request progresses through the system
  - **a consistent view of the status of the job is maintained**

Provenance and Biomed

- **Provenance for jobs is central in biomed.**
- **Need to know how an experiment was done and how analysis was carried out**
  - Example – MAIME
  - Example – different results using different versions (dates) of databases (ie. EMBL, SWISSPROT, PDB releases).
- **Pharmaceutical companies absolutely require provenance for IP and regulatory reasons**



Workload Management System

- Workload Management System (WMS) comprises a set of Grid middleware components responsible for distribution and management of tasks across Grid resources
  - **applications are conveniently, efficiently and effectively executed.**

- Comparable services from other grid projects are, among others, the EDG WMS, Condor and the Eurogrid-Unicore resource broker.

Workload Management System

- Purpose of Workload Manager (WM) is accept and satisfy requests for job management coming from its clients
  - **meaning of the submission request is to pass the responsibility of the job to the WM.**
    - **WM will pass the job to an appropriate CE for execution**
      - ***taking into account requirements and the preferences expressed in the job description***
- The decision of which resource should be used is the outcome of a *matchmaking* process between submission requests and available resources
  - **availability of resources for a particular task depends**
    - **on the state of the resources**
    - **on the utilisation policies**
      - ***assigned for the VO the user belogs***

THE INFORMATION SUPERMARKET

- ISM represents one of the most notable improvements in the WM as inherited from the EU DataGrid (EDG) project
  - **decoupling between the collection of information concerning resources and its use**
    - **allows flexible application of different policies**
- The ISM basically consists of a repository of resource information that is available in read only mode to the matchmaking engine
  - **the update is the result of**
    - **the arrival of notifications**
    - **active polling of resources**
    - **some arbitrary combination of both**
  - **can be configured so that certain notifications can trigger the matchmaking engine**
    - **improve themodularity of the software**
    - **support the implementation of lazy scheduling policies**

THE TASK QUEUE

- The Task Queue represents the second most notable improvement in the WM internal design
  - **possibility to keep a submission request for a while if no resources are immediately available that match the job requirements**
    - **technique used by the AliEn and Condor systems**
- Non-matching requests
  - **will be retried either periodically**
    - **eager scheduling approach**
  - **or as soon as notifications of available resources appear in the ISM**
    - **lazy scheduling approach**

JOB LOGGING AND BOOKKEEPING

- L&B tracks jobs in terms of *events*
  - **important points of job life**
    - **submission, finding a matching CE, starting execution etc**
      - ***gathered from various WMS components***
- The events are passed to a physically close component of the L&B infrastructure
  - ***locallogger***
    - **avoid network problems**
      - ***stores them in a local disk file and takes over the responsibility to deliver them further***
- The destination of an event is one of *bookkeeping servers*
  - **assigned statically to a job upon its submission**
    - **processes the incoming events to give a higher level view on the job states**
      - **Submitted, Running, Done**
    - **various recorded attributes**
      - ***JDL, destination CE name, job exit code***
- Retrieval of both job states and raw events is available via legacy (EDG) and WS querying interfaces
  - **user may also register for receiving notifications on particular job state changes**

JOB SUBMISSION Services

WMS components handling the job during its lifetime and performing the submission

- **Job Adapter**
  - **is responsible for**
    - **making the final touches to the JDL expression for a job, before it is passed to CondorC for the actual submission**
    - **creating the job wrapper script that creates the appropriate execution environment in the CE worker node**
      - ***transfer of the input and of the output sandboxes***
- **CondorC**
  - **responsible for**
    - **performing the actual job management operations**
      - ***job submission, job removal***
- **DAGMan**
  - **meta-scheduler**
    - **purpose is to navigate the graph**
    - **determine which nodes are free of dependencies**
    - **follow the execution of the corresponding jobs.**
  - **instance is spawned by CondorC for each handled DAG**
- **Log Monitor**
  - **is responsible for**
    - **watching the CondorC log file**
    - **intercepting interesting events concerning active jobs**
      - ***events affecting the job state machine***

triggering appropriate actions.

Biomed and queues

- **Can be useful to have the ability to get ‘low’ priority jobs**
  - ie. Get them run on large scale resources when these are not used for something else

Computing Element

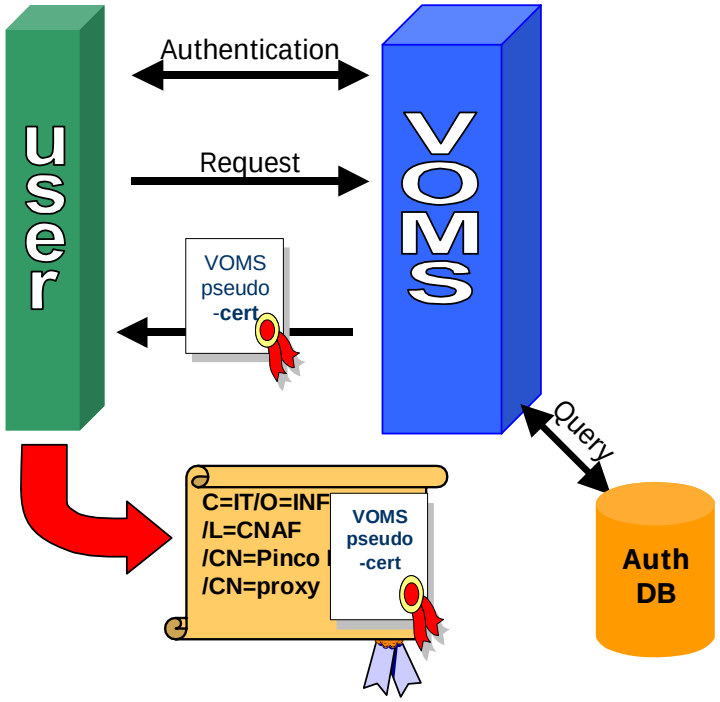
- **Service representing a computing resource**
- **Refers to a Cluster of Computational Resources, also heterogeneous.**
- **Main functionality: job management**
  - Run jobs
  - Cancel jobs
  - Suspend and resume jobs, send signals to them, get status or notification.
  - Provide info on “quality of service”
    - How many resources match the job requirements ?
    - What is the estimated time to have the job starting its execution ? (ETT)
- **Used by the WM or by any other client (e.g. end-user)**
- **CE architecture accommodated to support both push and pull model**
  - Push model: the job is pushed to the CE by the WM
  - Pull model: the CE asks the WM for jobs
- **These two models are somewhat mirrored in the resource information flow**
  - In order to 'pull' a job a resource must choose where to 'push' information about itself (CE Availability message)

WMS's Scheduling Policies

- WM can adopt
  - **eager scheduling**
    - **a job is bound to a resource as soon as possible and, once the decision has been taken, the job is passed to the selected resource for execution**
  - **lazy scheduling**
    - **foresees that the job is held by the WM until a resource becomes available, at which point that resource is matched against the submitted jobs**
      - ***the job that fits best is passed to the resource for immediate execution.***
- Varying degrees of eagerness (or laziness) are applicable
  - **match-making level**
    - **eager scheduling**
      - ***implies matching a job against multiple resources***
    - **lazy scheduling**
      - ***implies matching a resource against multiple jobs***

Enabling Grids for E-science in Europe

VOMS Operations

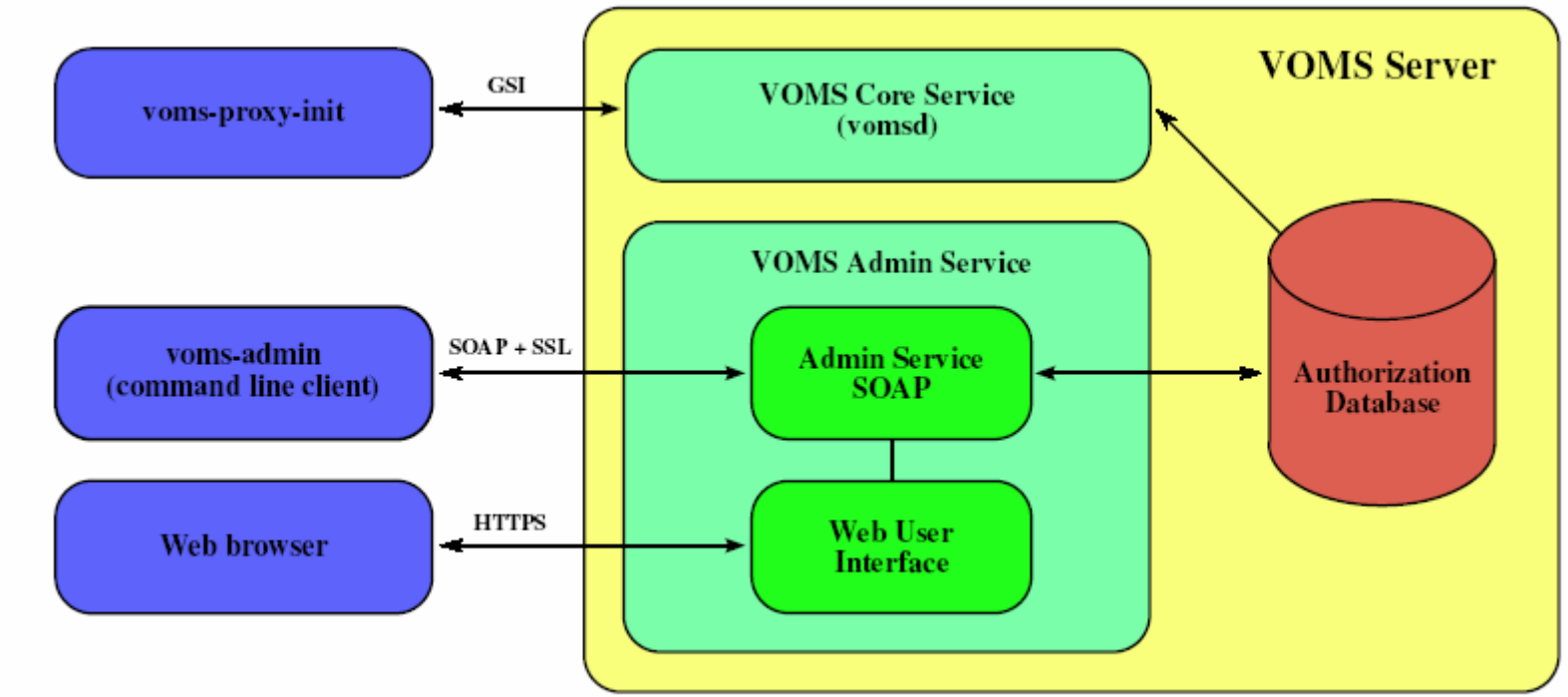


1. Mutual authentication Client-Server
   - Secure communication channel via standard Globus API
2. Client sends request to Server
3. Server checks correctness of request
4. Server sends back the required info (signed by itself) in a "Pseudo-Certificate"
5. Client checks the validity of the info received
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

gLite VOMS


voms-proxy-init
GSI
VOMS Server
VOMS Core Service
(vomsd)
VOMS Admin Service
voms-admin
(command line client)
SOAP + SSL
Admin Service
SOAP
Authorization
Database
Web browser
HTTPS
Web User
Interface

Job Management

.

•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

WMS in LCG2

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).

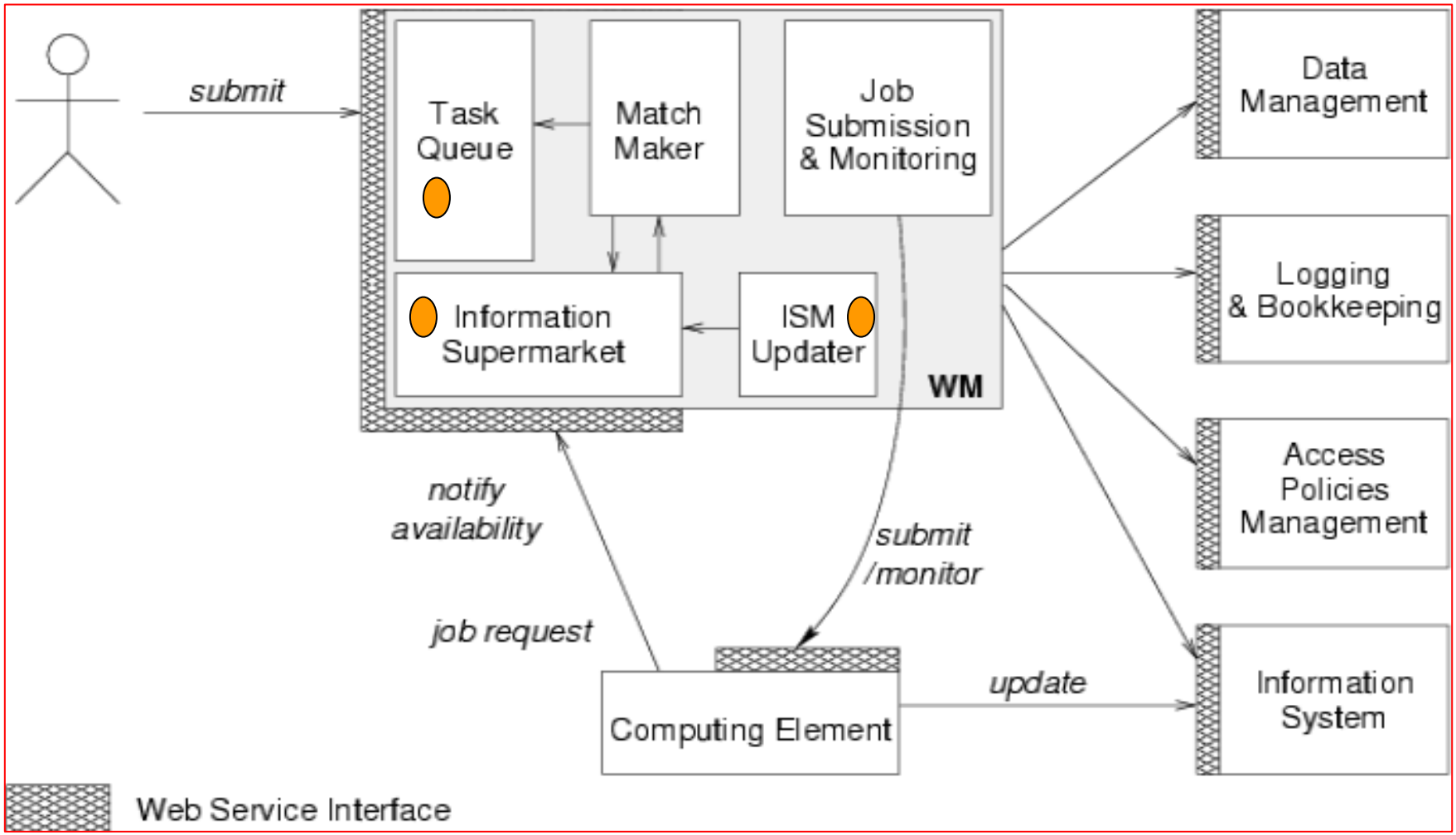
.Job Adapter, which prepares the environment for the job and its final description, before passing it to the Job Control Service.

.Job Control Service (JCS), which finally performs the actual job management operations (job submission, removal...)

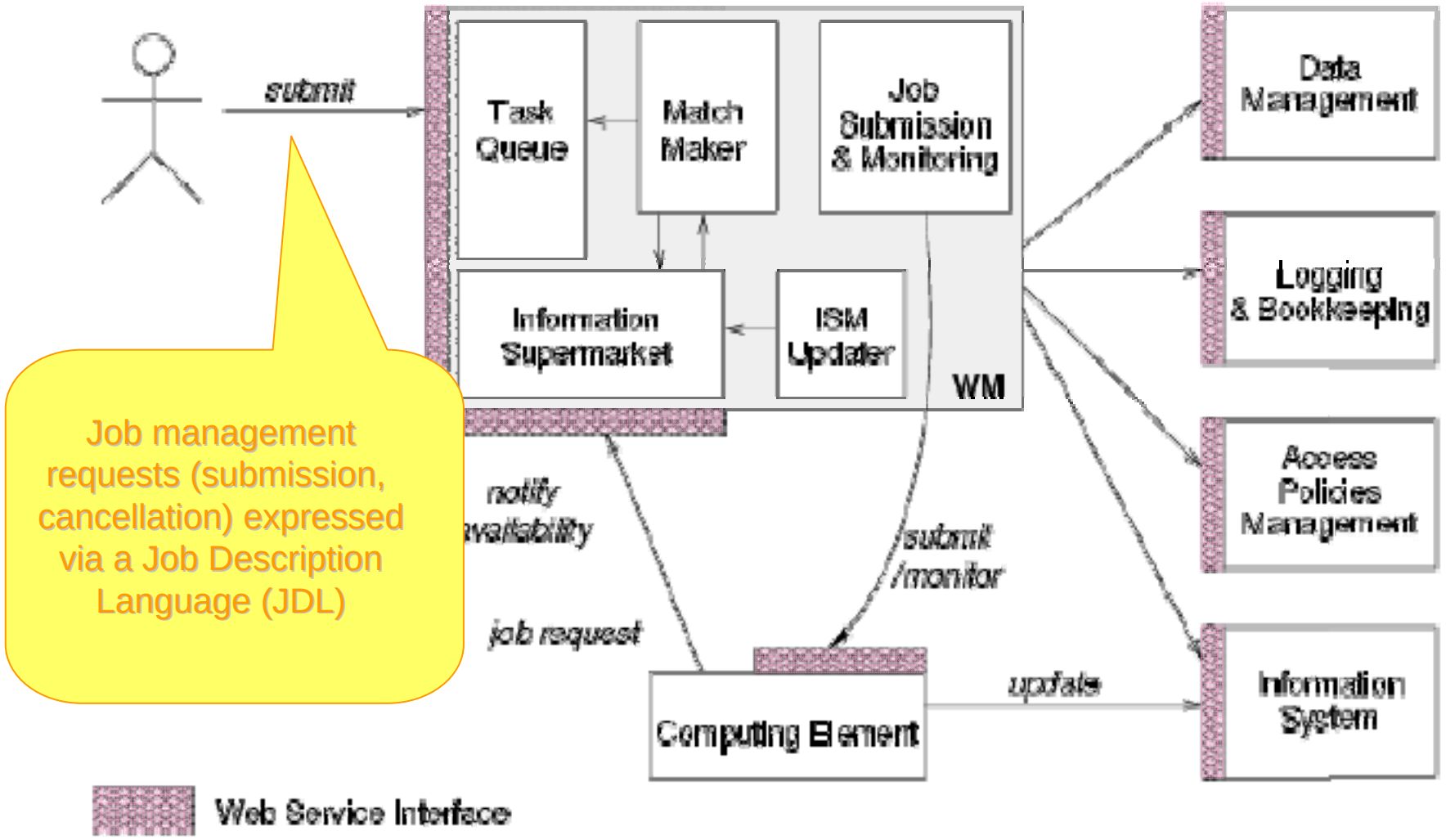
.Logging and Bookkeeping service (LB) . The LB logs all job management Grid events, which can then be retrieved by users or system administrators for monitoring or troubleshooting.

WMS in gLite

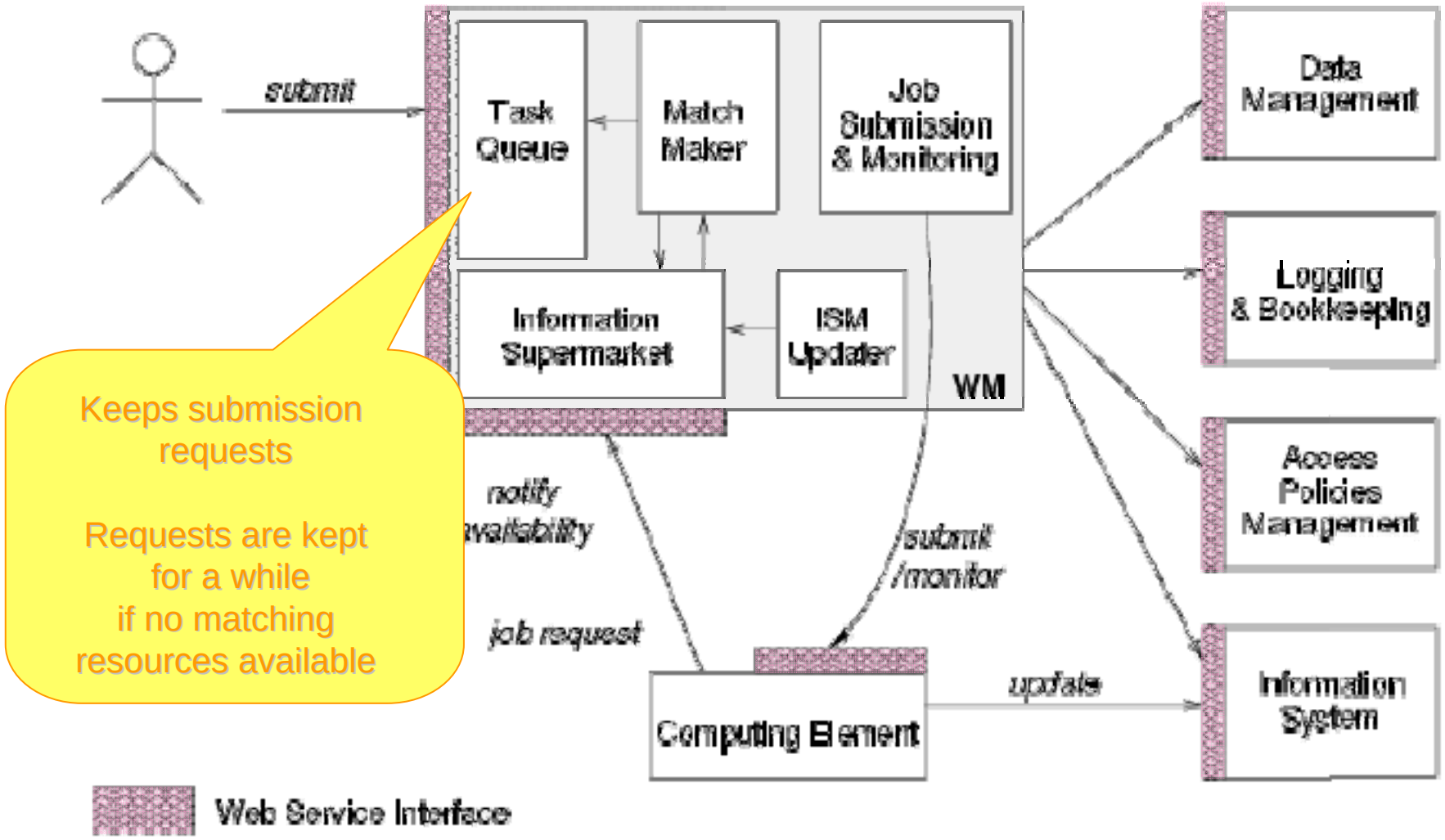
- Differ from LCG-2



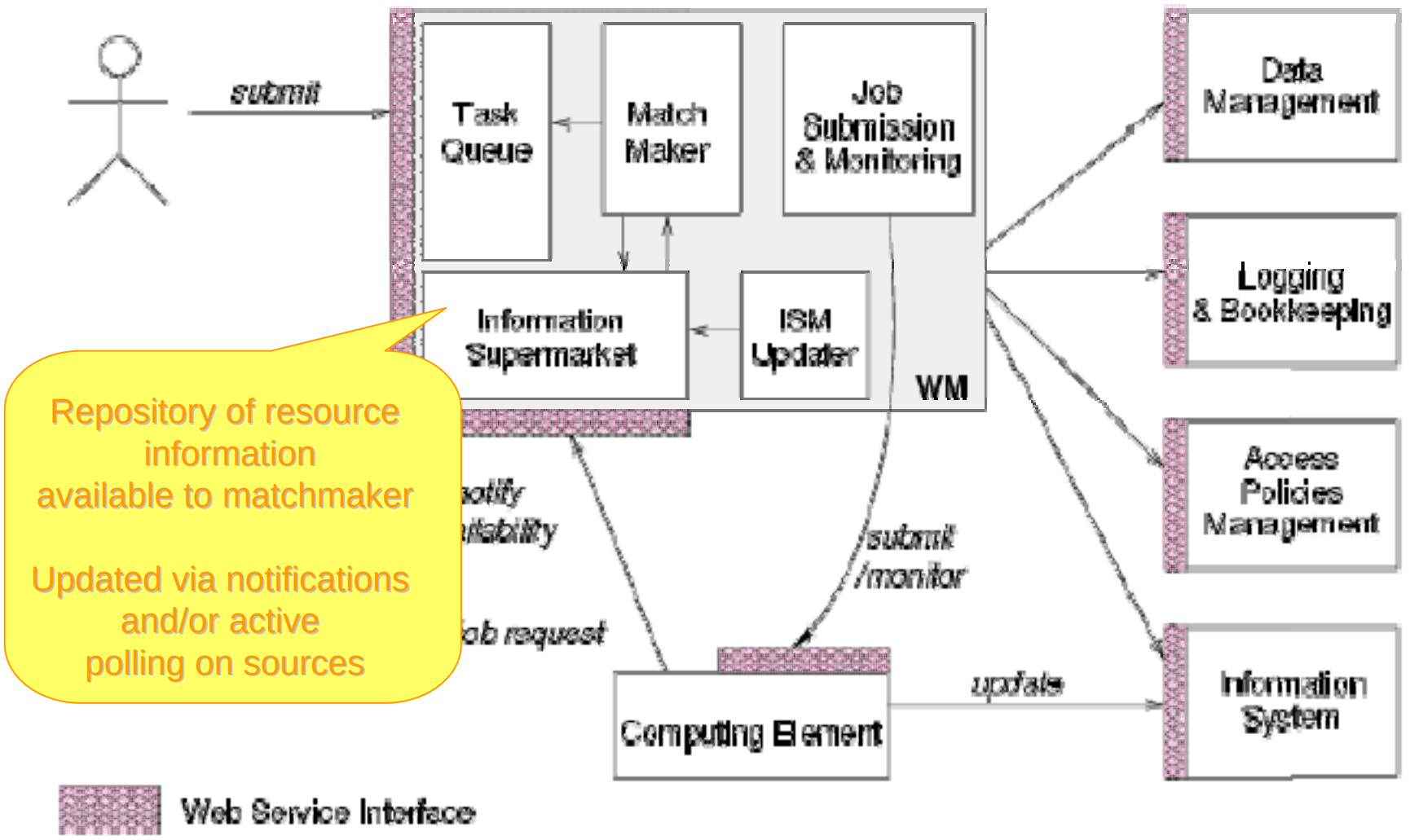
WMS's Architecture



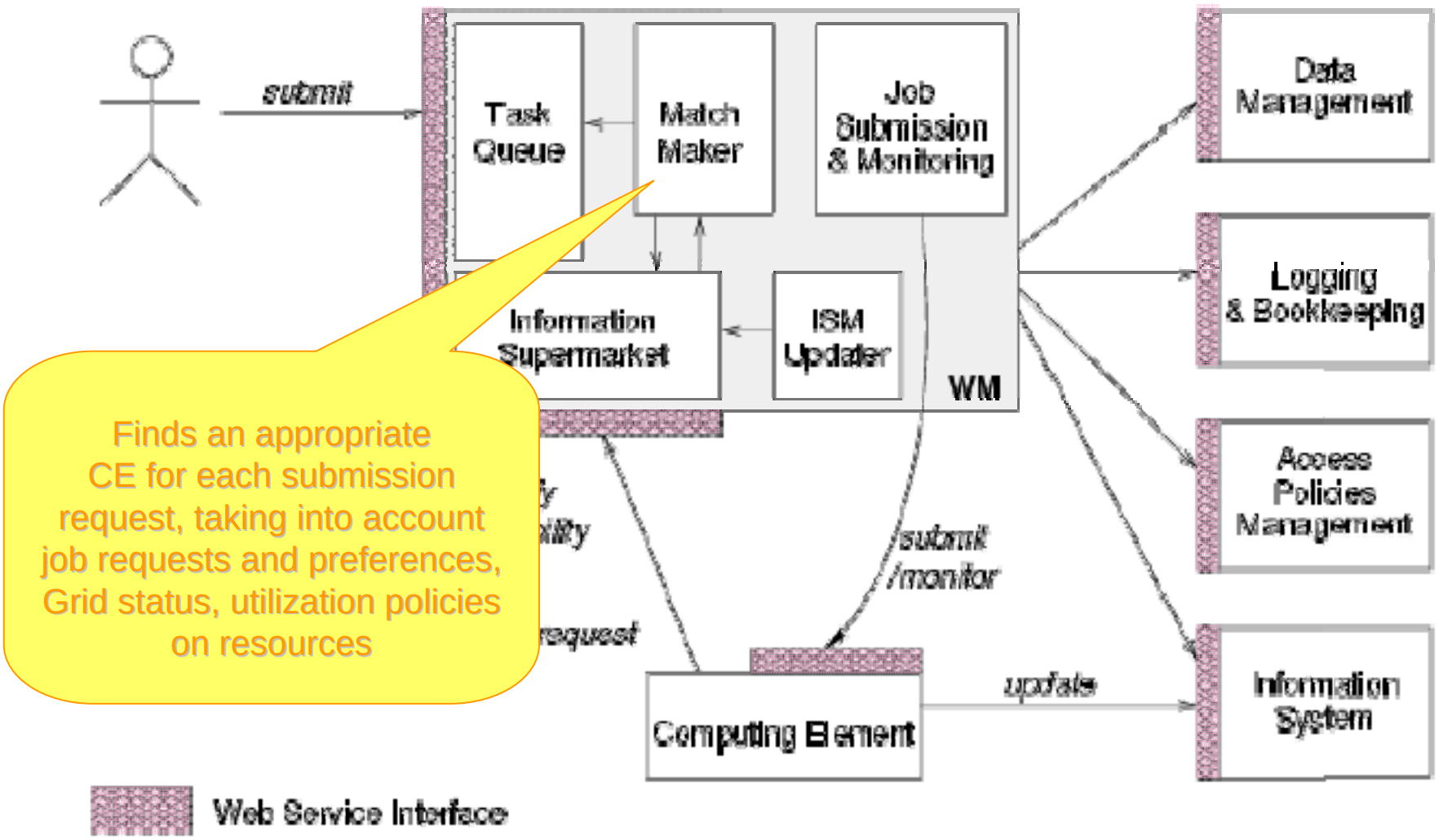
WMS's Architecture



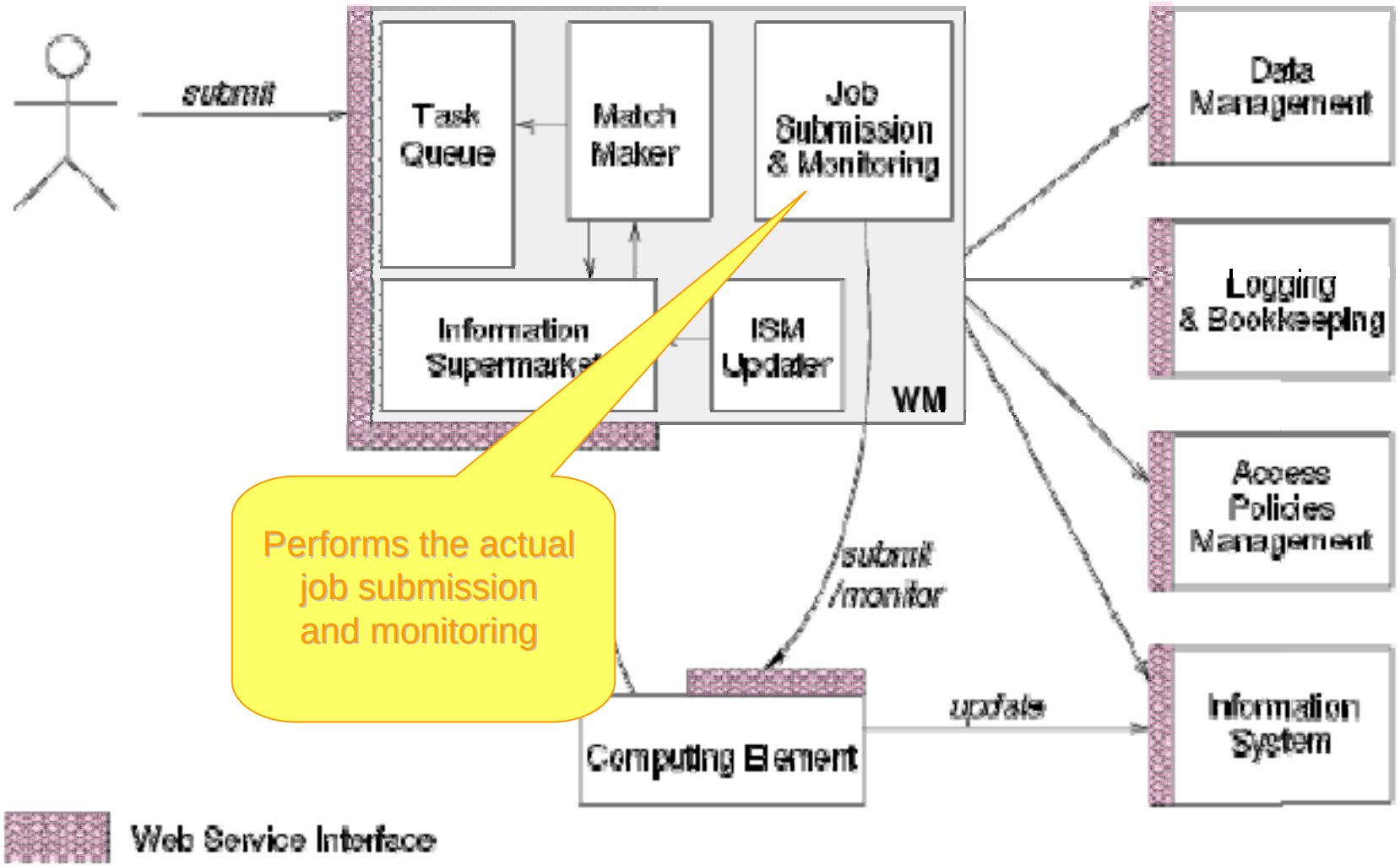
WMS's Architecture



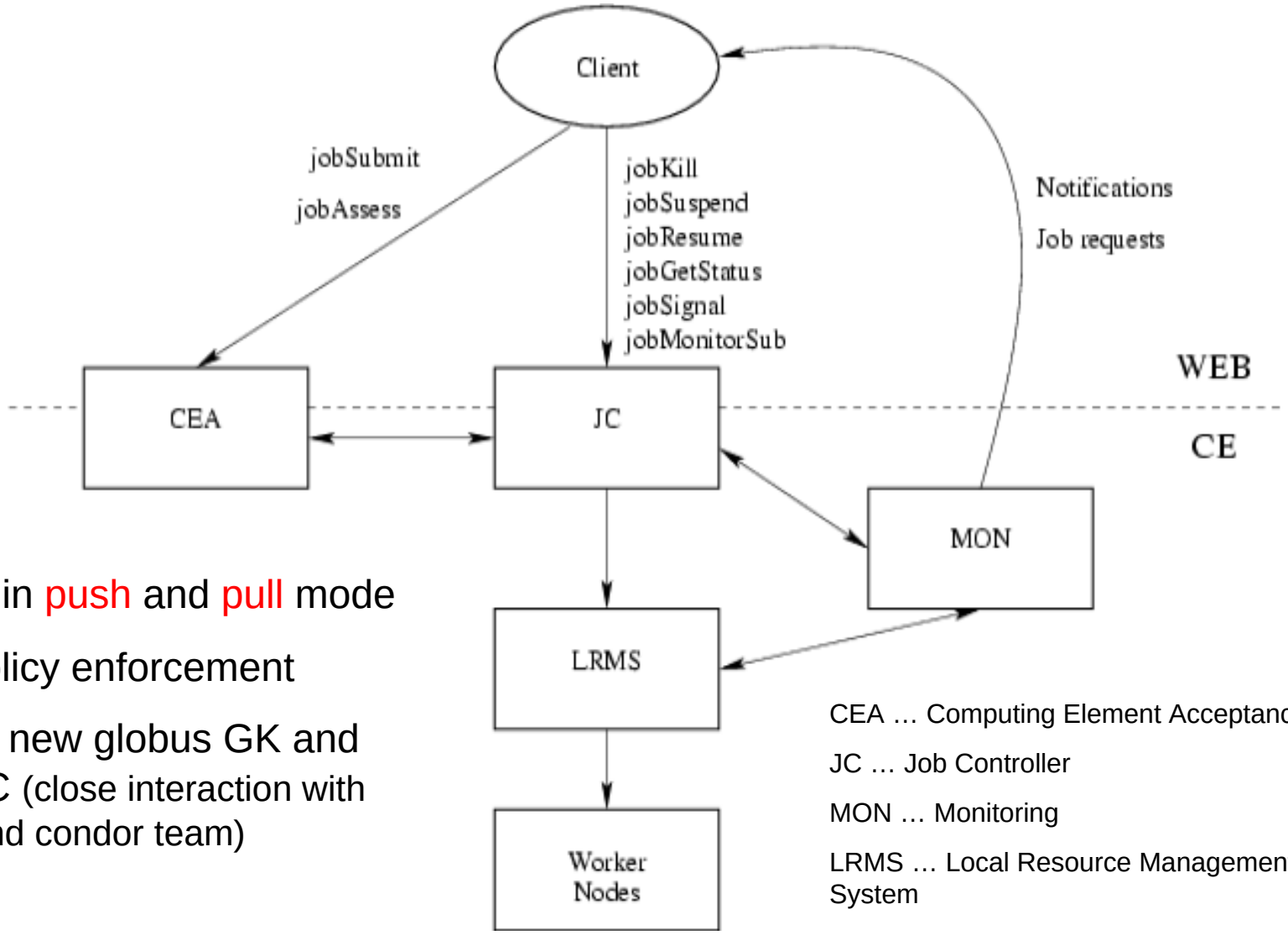
WMS's Architecture



WMS's Architecture



gLite Computing Element



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

CEA … Computing Element Acceptance

JC … Job Controller

MON … Monitoring

LRMS … Local Resource Management System

Physics and data

- Note – remember that physists and biomed tend to have a different view of data.
- For physics data is mainly about files
- For biomed is mainly about databases

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 metada 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.

Differences between gLite and LCG-2 data management

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 .

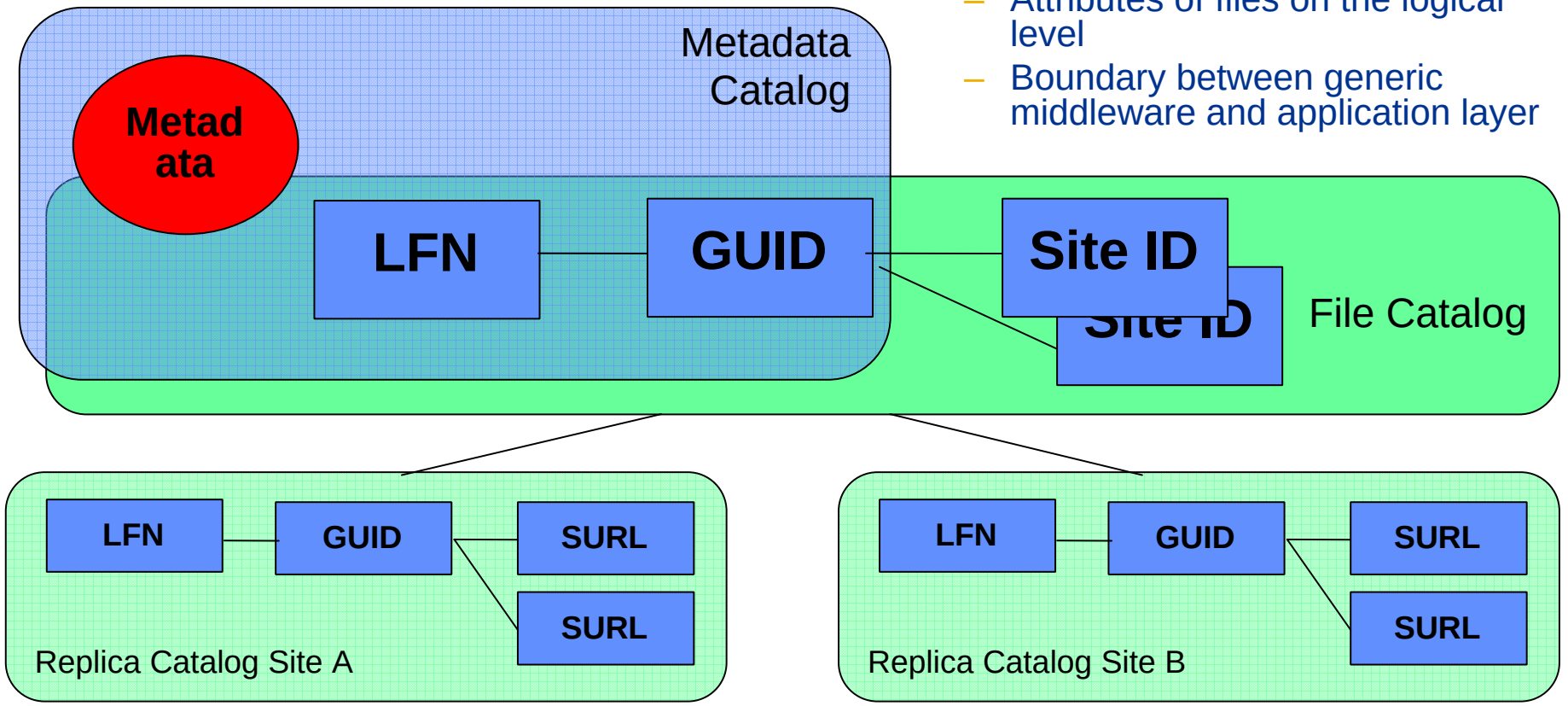
Closely related to the data services are the security-related services and the Package Manager.

- 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.

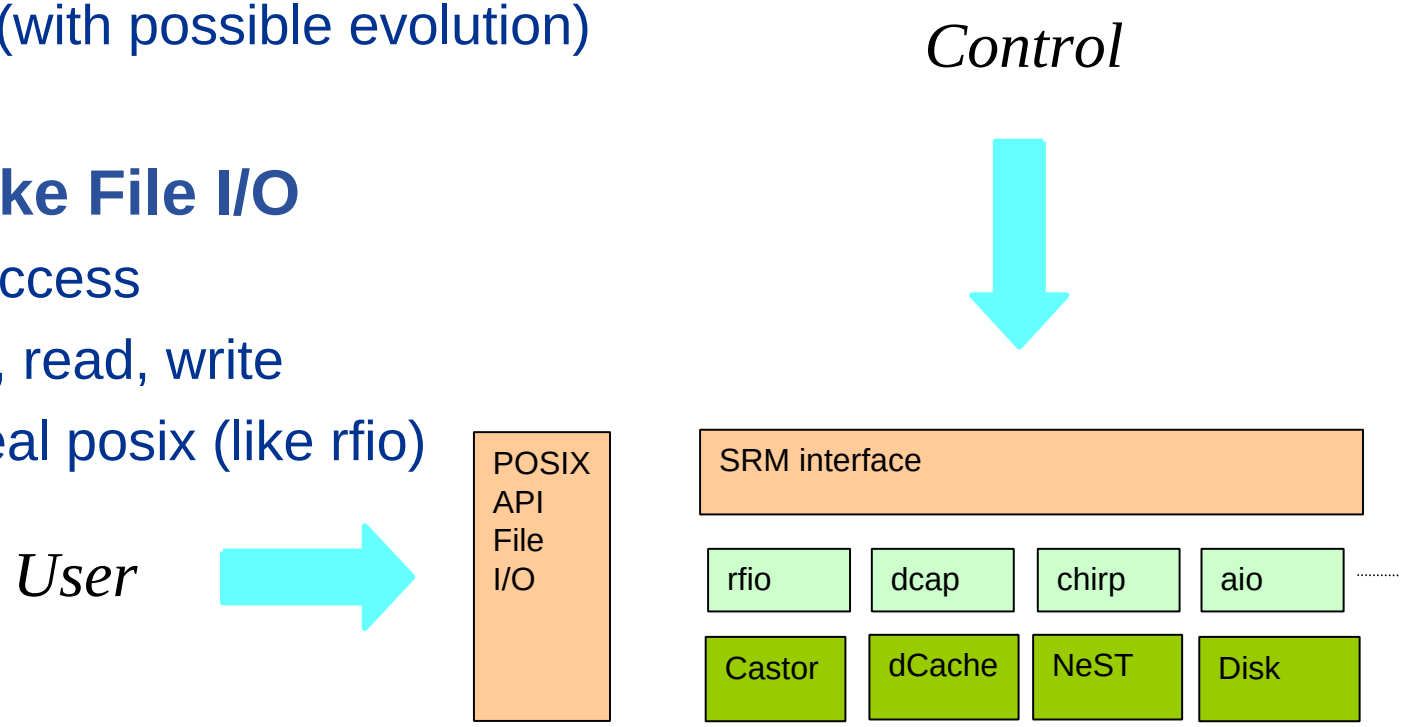
gLite Catalogs

- **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



Storage Element Interfaces

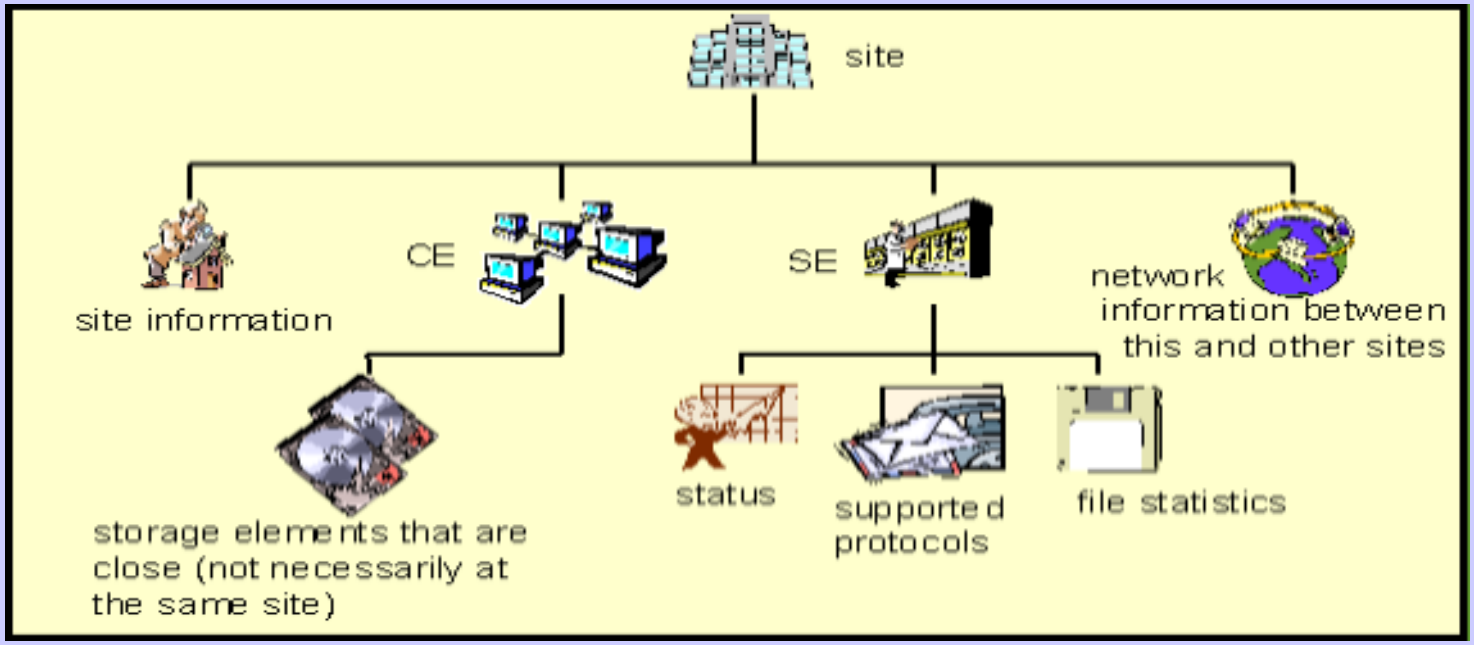
- **SRM interface**
  - Management and control
  - SRM (with possible evolution)
- **Posix-like File I/O**
  - File Access
  - Open, read, write
  - Not real posix (like rfio)



LCG-2 Information System

Directory Information Tree

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



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

gLite Information System(1)

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.

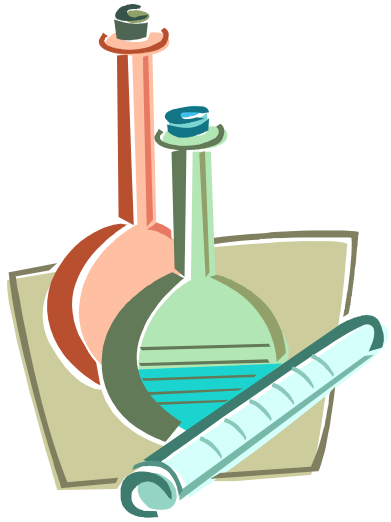
d. R-GMA service tool -regularly scans config files and updates a current Service Status R-GMA Table. CLI exists for modify these files and query the service table.

gLite Information System(2)

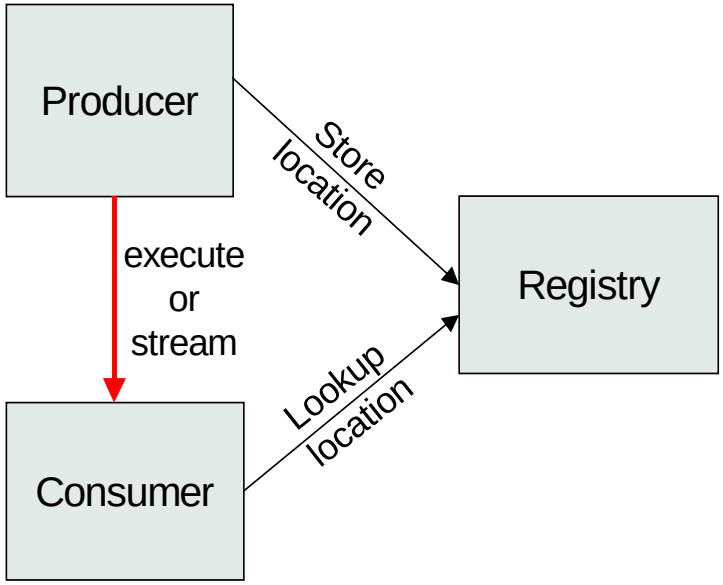
R-GMA is based on the Grid Monitoring Architecture (GMA) from the Grid Global Forum (GGF), which is a simple Consumer-Producer model that models the information infrastructure of a Grid as a set of consumers (that request information), producers (that provide information) and a central registry which 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,



R-GMA



- 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

Mapping gLite Services onto physical machines