The gLite middleware

architecture and components

Fotis Georgatos <gef@grnet.gr>

(Thanks to Ariel Garcia & Evangelos Floros)

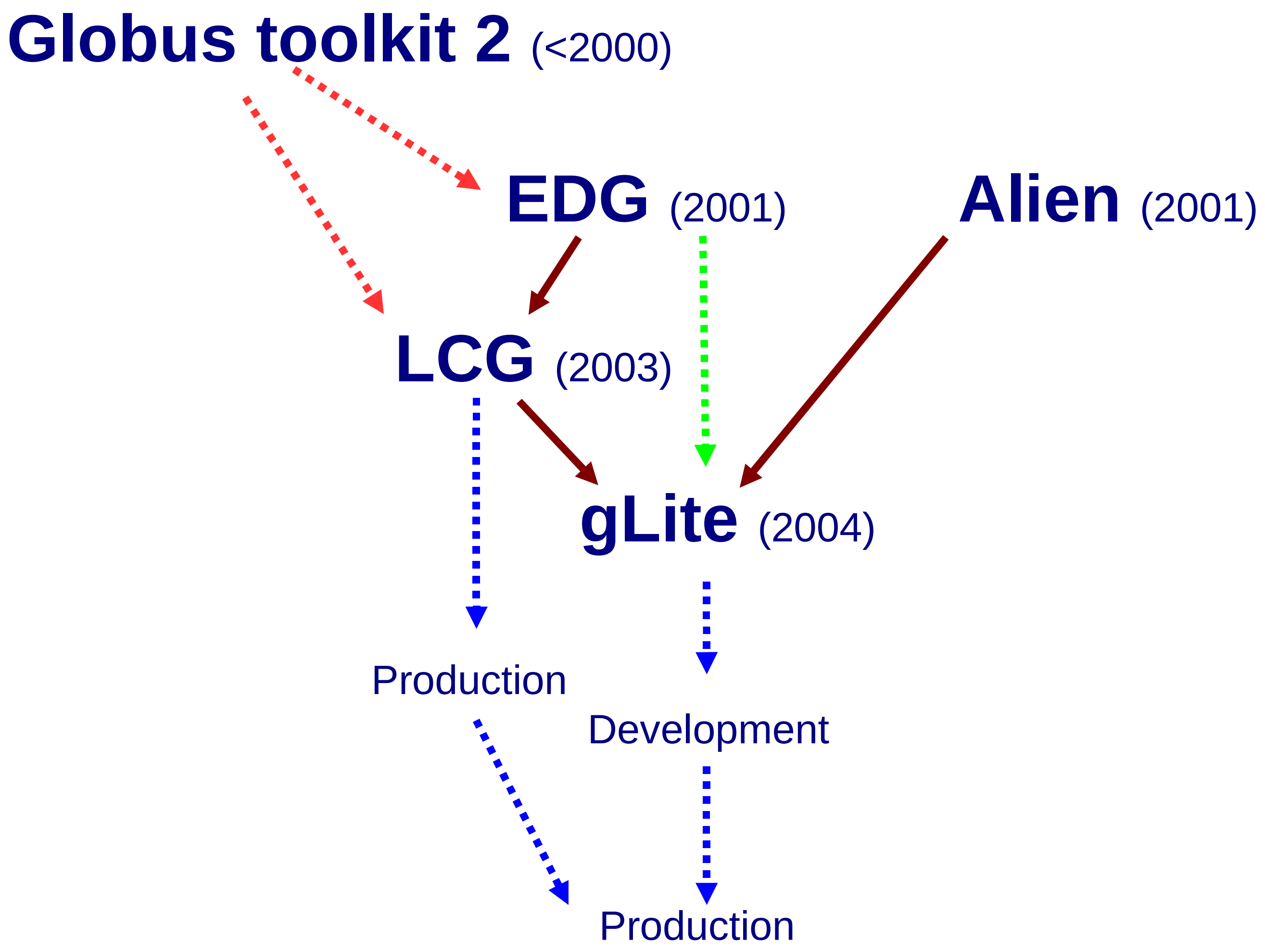
Information Society

Outline

- **Some history**
- **Grid and the middleware**

- **gLite components, functionality and architecture**
  - security
  - information
  - job management
  - data management
- **Conclusions**

Some history



Globus toolkit 2 (<2000)

EDG (2001)

Alien (2001)

LCG (2003)

gLite (2004)

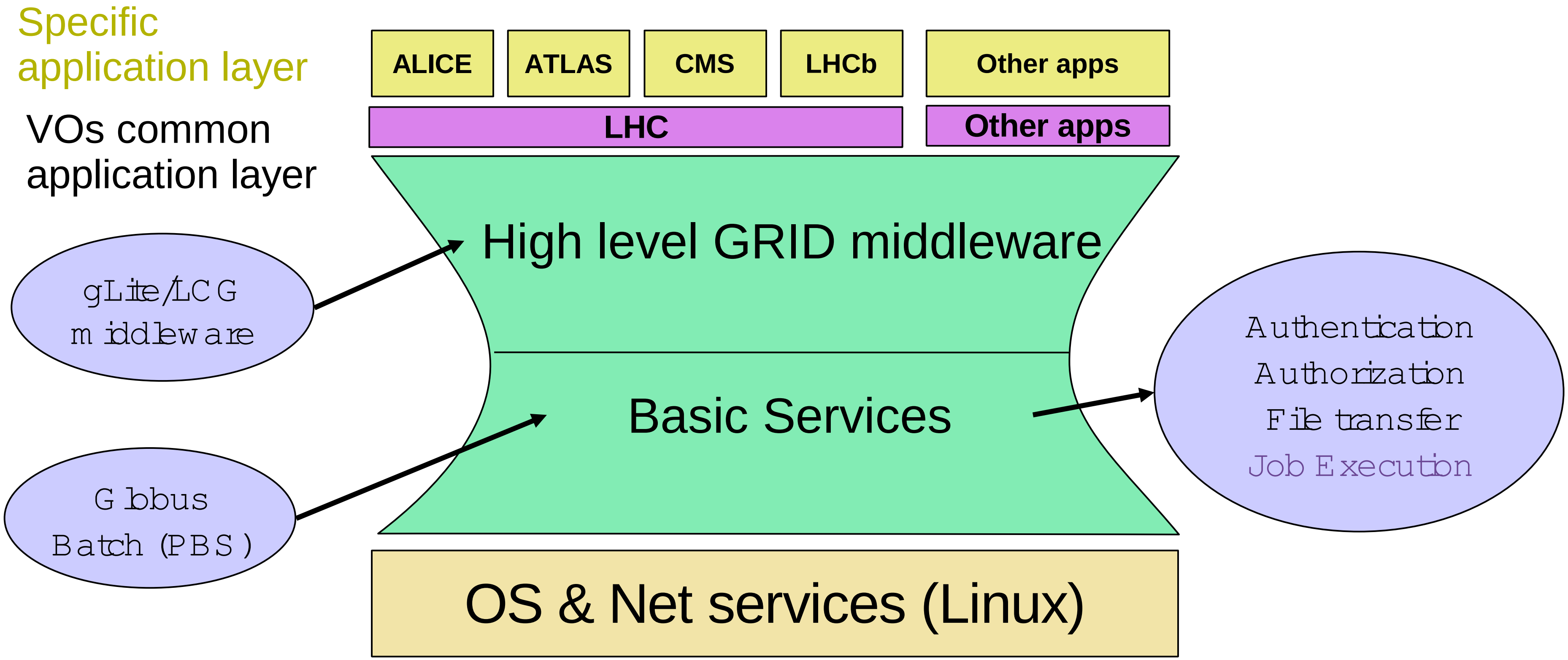
Production

Development

Production

Software layers

- Middleware keeps the grid together



Specific application layer

ALICE

ATLAS

CMS

LHCb

Other apps

VOs common application layer

LHC

Other apps

High level GRID middleware

Basic Services

gLite/LCG middleware

Globus Batch (PBS)

Authentication
Authorization
File transfer
Job Execution

OS & Net services (Linux)

The middleware

gLite:

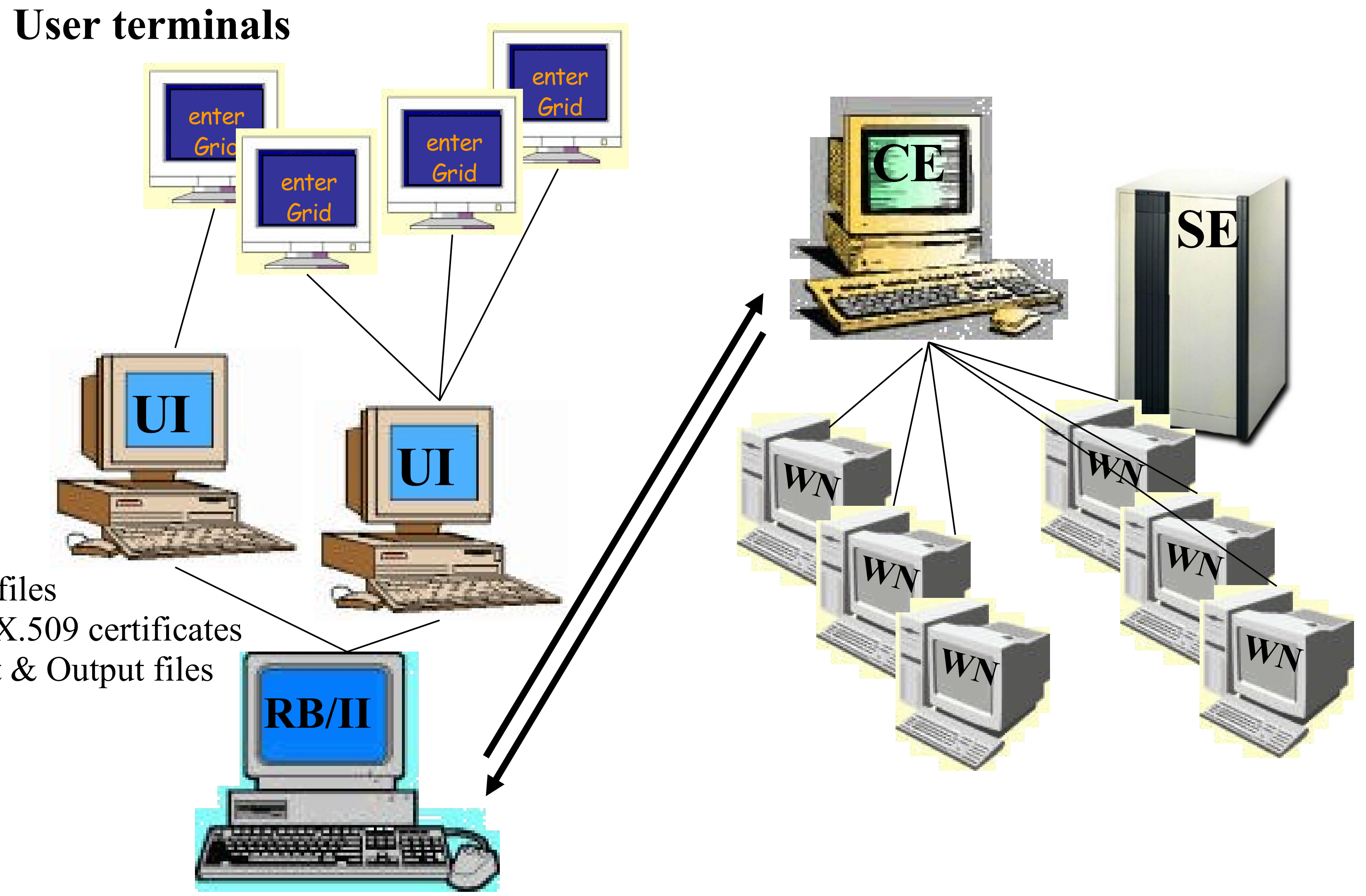
- **Next step in middleware development**
- **New standards adopted**
  - Web services
- **Reengineering / redesign**
  - Scalability
  - Performance
  - Interoperability
  - Modularity
  - (...) the perfect grid middleware ;-)
- **Functionality added - user requirements**
  - HEP / Biomedicine / generic application - users

What's new

New features:

- Increased modularity
  - services can be deployed independently
- XML based configuration
- Finer grained security (VOMS)
- Pull model for job management (lazy scheduling)
- POSIX I/O to grid files
- User friendly LFNs
- File transfer services (data management jobs)
- ...

WMS: an overview



- JDL files
- PKI X.509 certificates
- Input & Output files

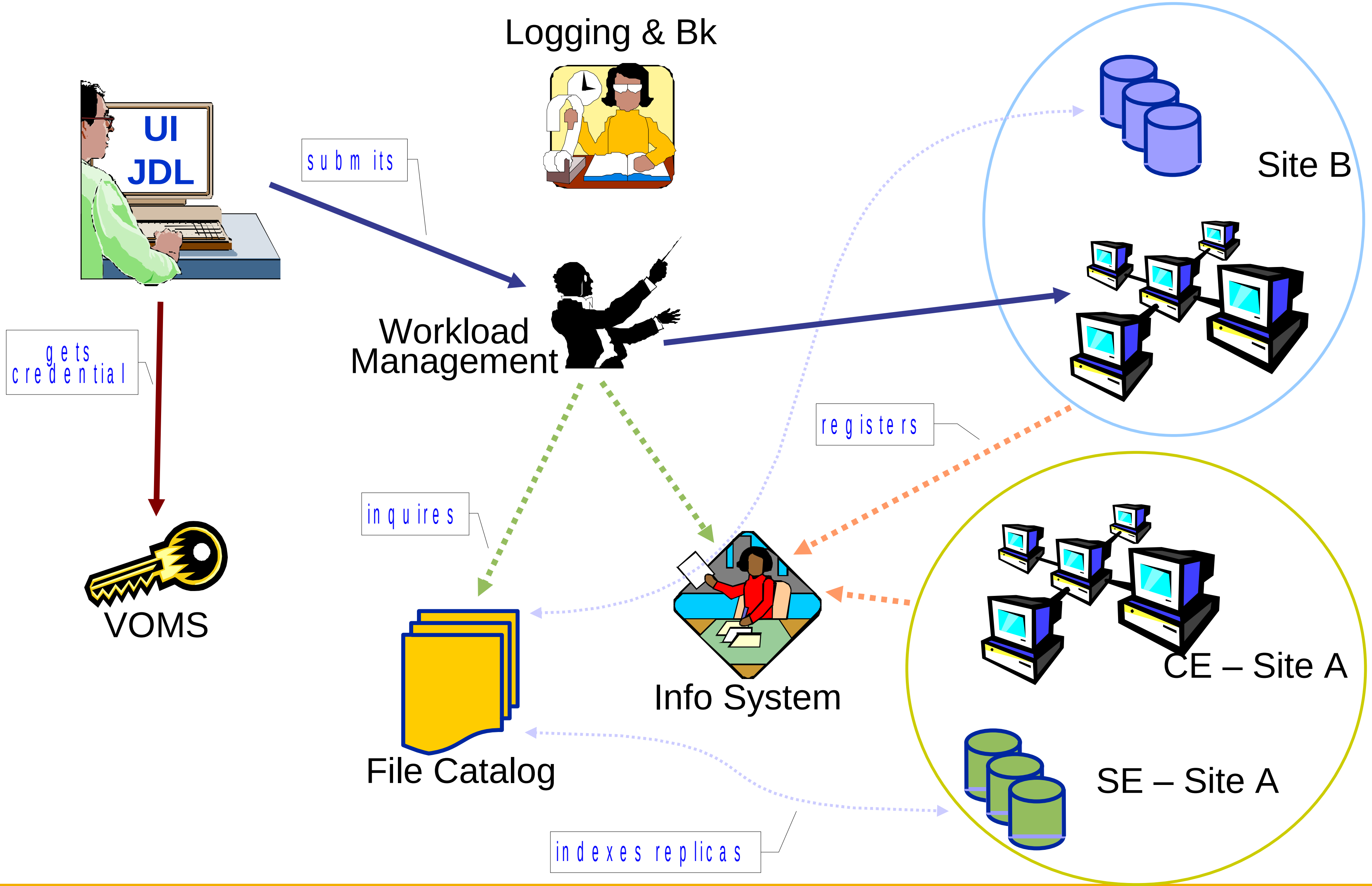
Components...

- **@site**
  - Computing Element (CE)
    - gateway to local computing resources (cluster of worker nodes)
  - Worker Nodes (WN)
  - Storage Element (SE)
    - gateway to local storage (disk, tape)
    - a gridftp server, an SRM interface, IO server
  - User Interfaces (UI)
    - user's access point to the grid
    - client programs using some/all grid services

CE & SE:

layer of abstraction, local peculiarities irrelevant

WMS: A detailed job workflow



...and more components

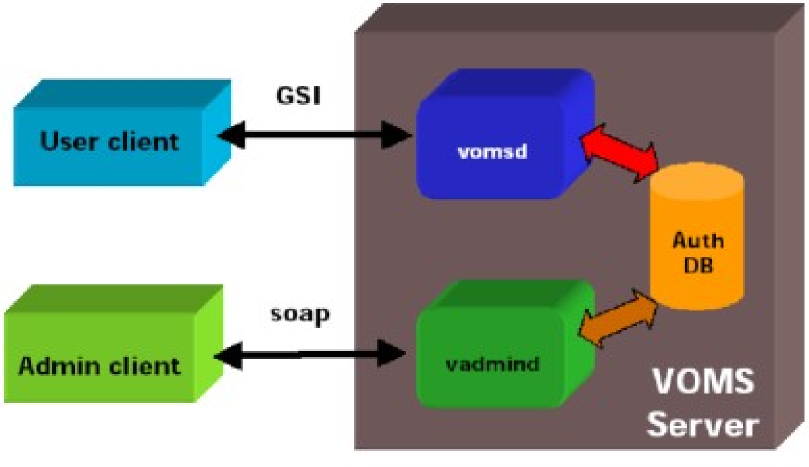
- **Grid- or VO-wide**
  - Security
    - Virtual Organization Server (VOMS)
    - MyProxy server (Proxy)
  - Information System (IS)
  - Job handling
    - Workload Management System (WMS)
    - Logging & Bookkeeping (LB)
  - Data management
    - File catalog (LFC)
    - File Transfer Service (FTS)
    - File Placement Service (FPS)

Enabling Grids for E-sciencE

Securing the access

Virtual Organization Membership Service

- Multiple VOs
- Multiple roles in VO
  - compatible X509 extensions
  - signed by VOMS server



- Web admin interface
- Supports MyProxy

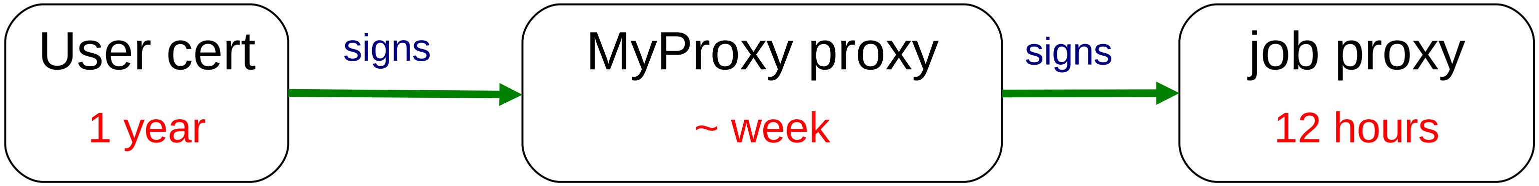
- Resource providers grant access to VOs or roles
- Sites map VO members/roles to local auth mechanism (unix user accounts)
  - allows for local policy

Layer of abstraction: individual members irrelevant

While I'm away...

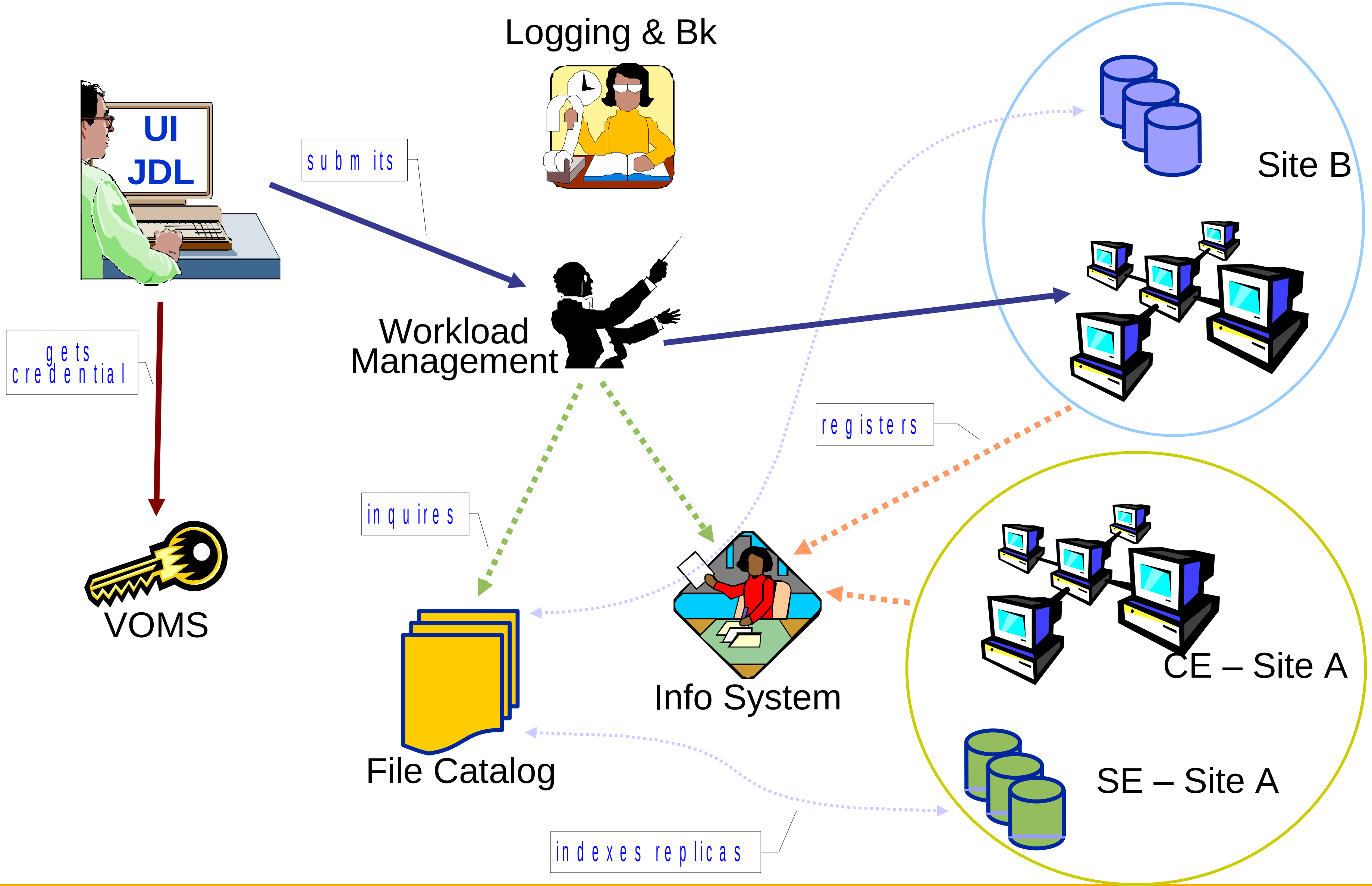
- **MyProxy**
  - allows longer lived jobs / increases security
    - WMS renews proxy
    - users should not produce long lived proxies :-)
  - allows for secure user mobility
    - user does not need to copy globus-keys around

Stores medium-lived proxy (days ~ weeks)



User cert
1 year
signs
MyProxy proxy
~ week
signs
job proxy
12 hours

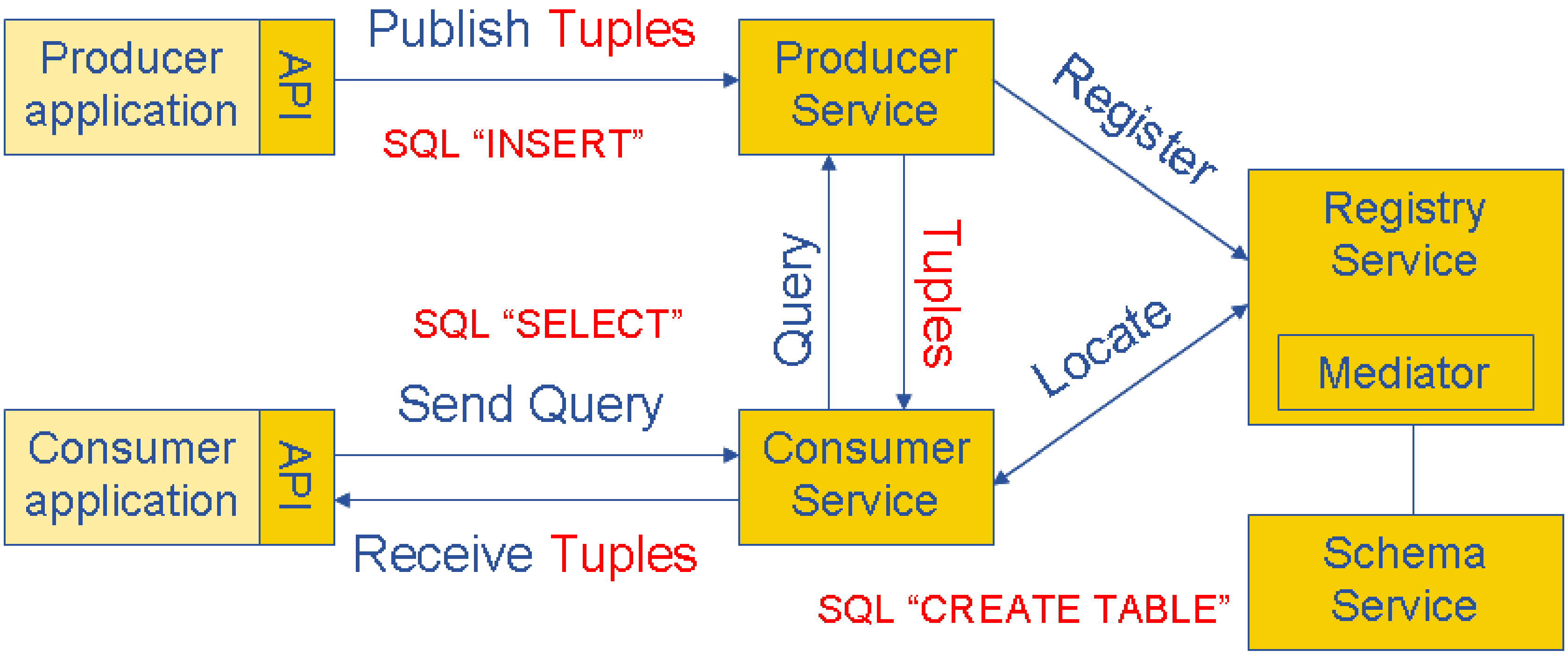
IS: Who knows what?



R-GMA: The eye of the Grid

- **Based on GMA**
  - relational (database-like) implementation of the GGF Grid Monitoring Architecture (GMA)
  - distributed
- **Aggregates service information from multiple grid sites**
  - hosts, resources (CPU, storage)
  - accepted VOs
  - based on Glue schema
- **Used by WMS (= RB's) to collect information on sites**
  - defines WMS's view of the Grid!
- **Generic Service Discovery API**
  - used by replica management tools to locate SEs, Catalogs
- **R-GMA system also used for monitoring :-)**

R-GMA ~ Distributed r-DB



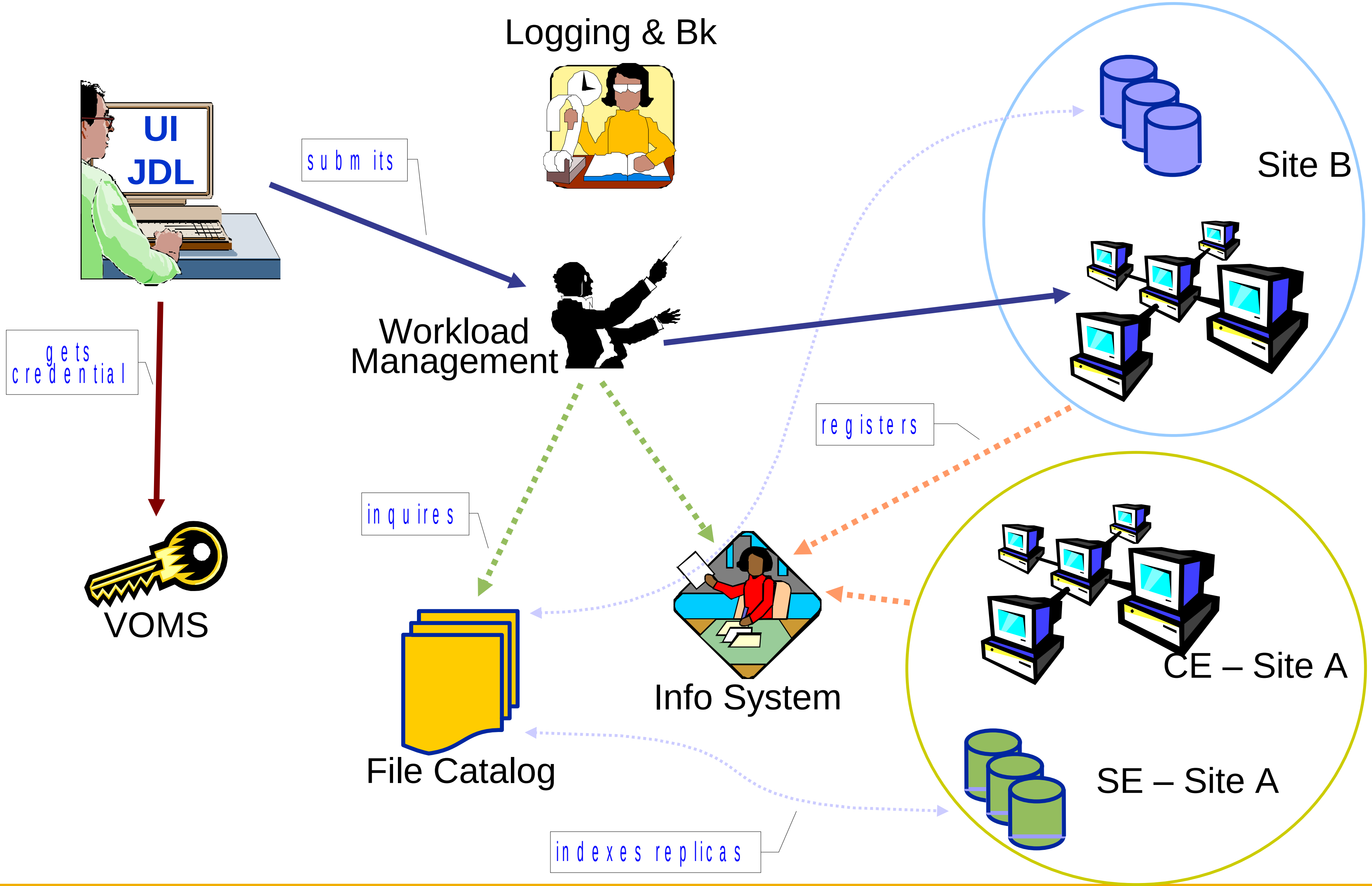
Workload Management System

- **Helps the user accessing computing resources**
  - resource brokering
  - management of input and output
  - management of complex workflows

- **Support for MPI job even if the file system is not shared between CE and Worker Nodes (WN) – easy JDL extensions**

- **Web Service interface via WMProxy**

I have a job...


Logging & Bk
UI
JDL
submits
Workload
Management
gets
credential
VOMS
Site B
registers
inquires
Info System
File Catalog
CE – Site A
SE – Site A
indexes replicas

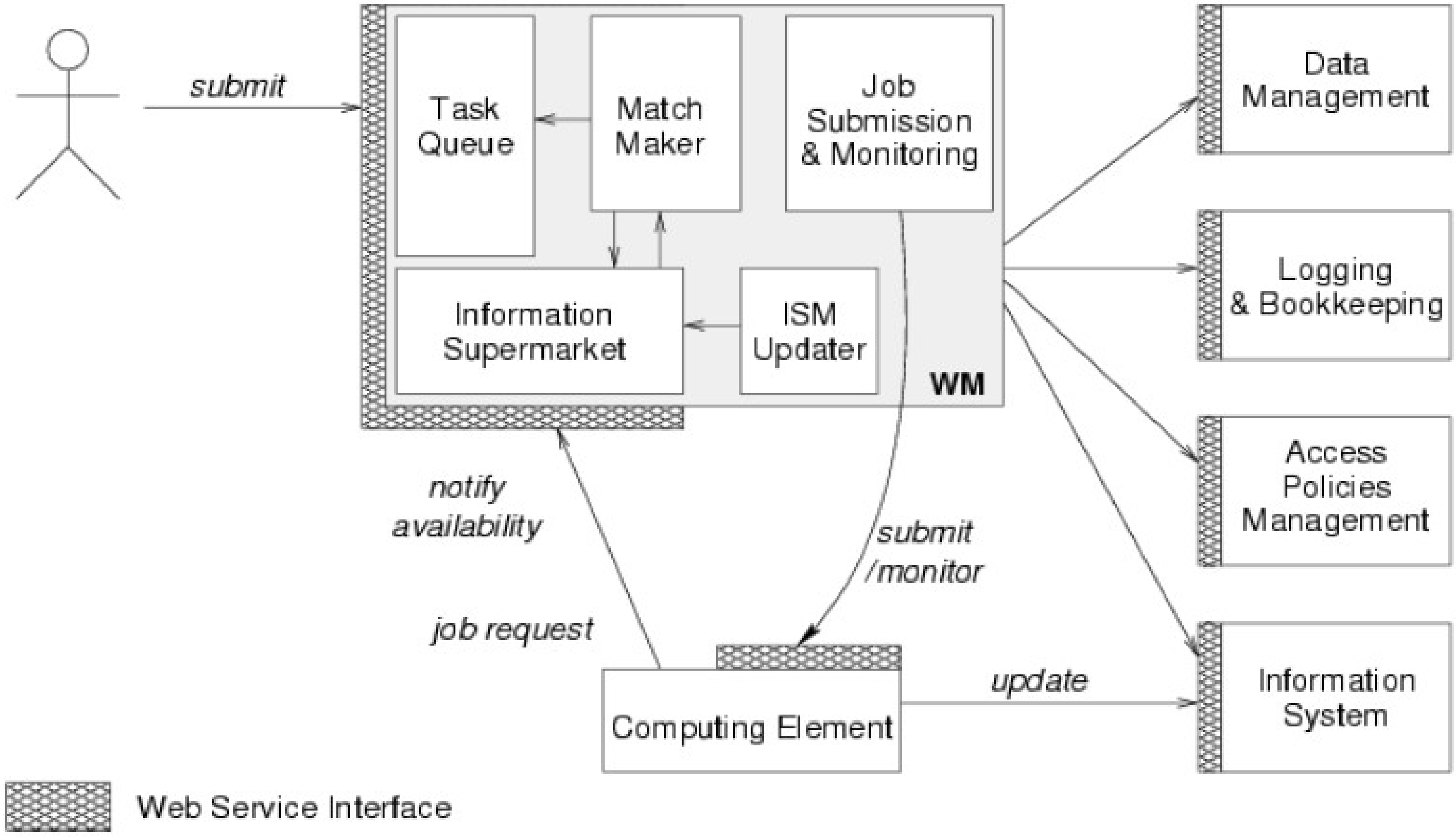
Who cares about my job?

- **WMS finds best location for job**
  - considering job requirements and available resources (CPUs, files)
    - Push model: WMS pushes job to CE
    - Pull model: CE asks the WMS for jobs
  - gets resource information from IS and File Catalogs
- **JSS (Condor) provides reliable submission system**
- **LB keeps track of job's status**

- **WMS is primary job execution interface for users**
- **each server allows only certain VOs / groups**

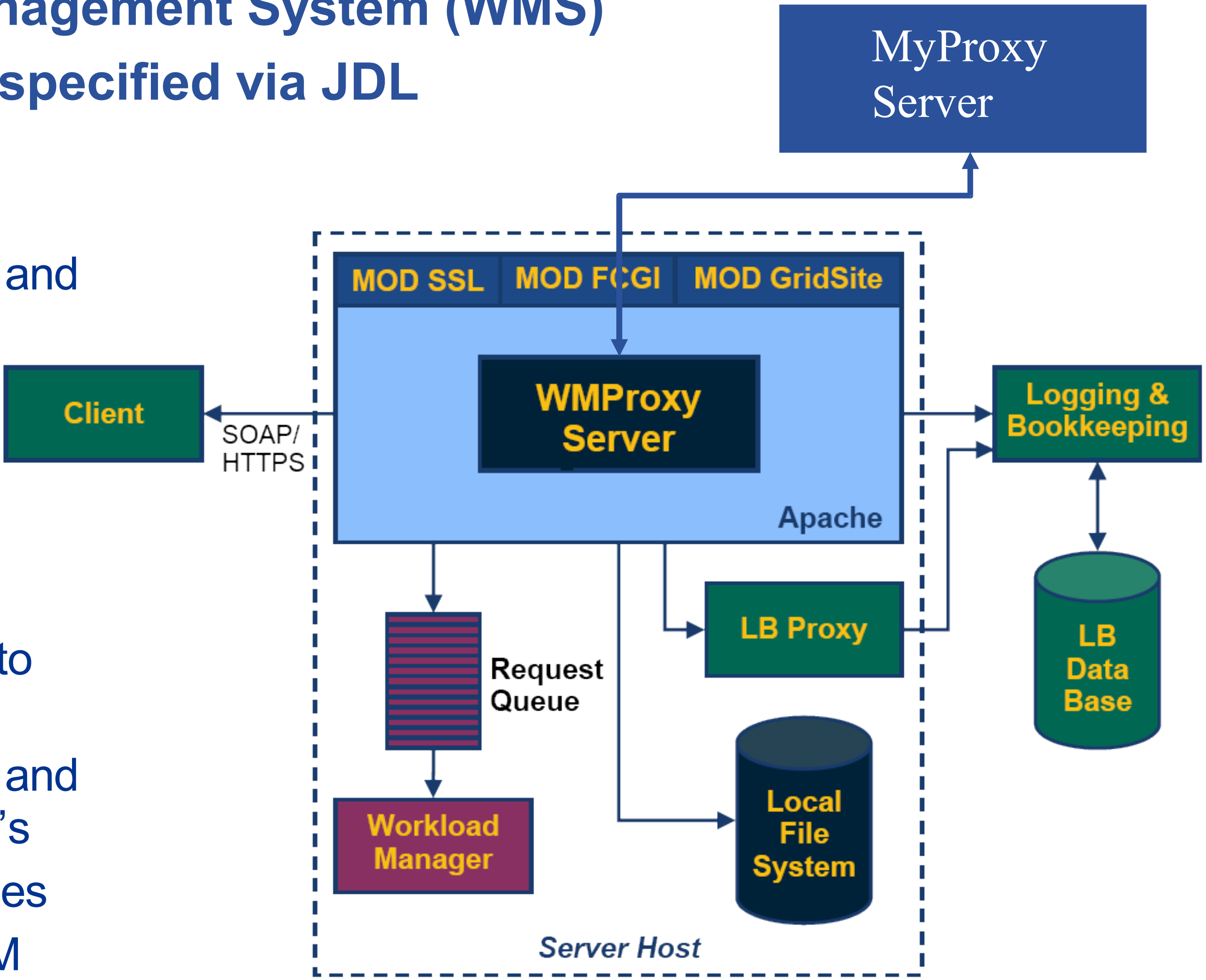
Layer of abstraction: sites irrelevant

Inside WMS


submit
Task Queue
Match Maker
Job Submission & Monitoring
Information Supermarket
ISM Updater
WM
Data Management
Logging & Bookkeeping
Access Policies Management
Information System
notify availability
job request
submit /monitor
update
Computing Element
Web Service Interface

WMProxy

- **WMProxy is a SOAP Web service providing access to the Workload Management System (WMS)**
- **Job characteristics specified via JDL**
  - jobRegister
    - create id
    - map to local user and create job dir
    - register to L&B
    - return id to user
  - input files transfer
  - jobStart
    - register sub-jobs to L&B
    - map to local user and create sub-job dir's
    - unpack sub-job files
    - deliver jobs to WM



Data Management

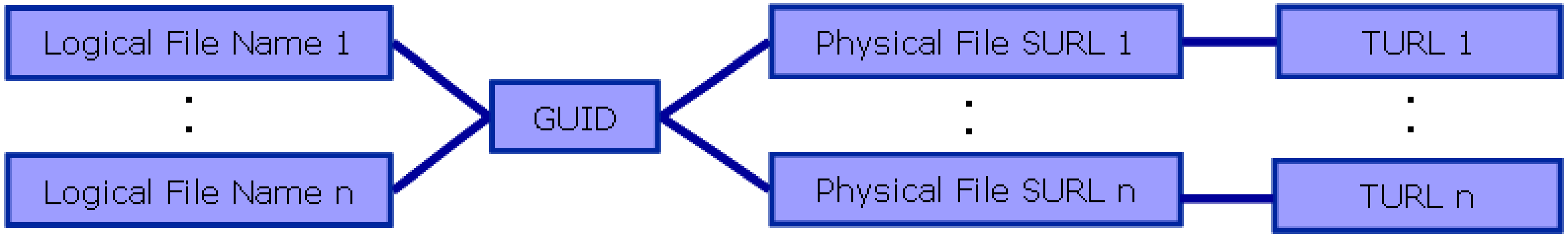
- **User and programs produce and require data**

- **Data may be stored in Grid datasets (files)**
  - Located in Storage Elements (SEs)
  - Accessed/Transferred eg. using GSIFTP
  - Several replicas of one file in different sites
  - Accessible by Grid users and applications from “anywhere”
  - Locatable by the WMS (data requirements in JDL)

- **Also…**
  - WMS can send (small amounts of) data to/from jobs: Input and Output Sandbox
  - Data may be copied from/to local filesystems (WNs, UIs) to the Grid

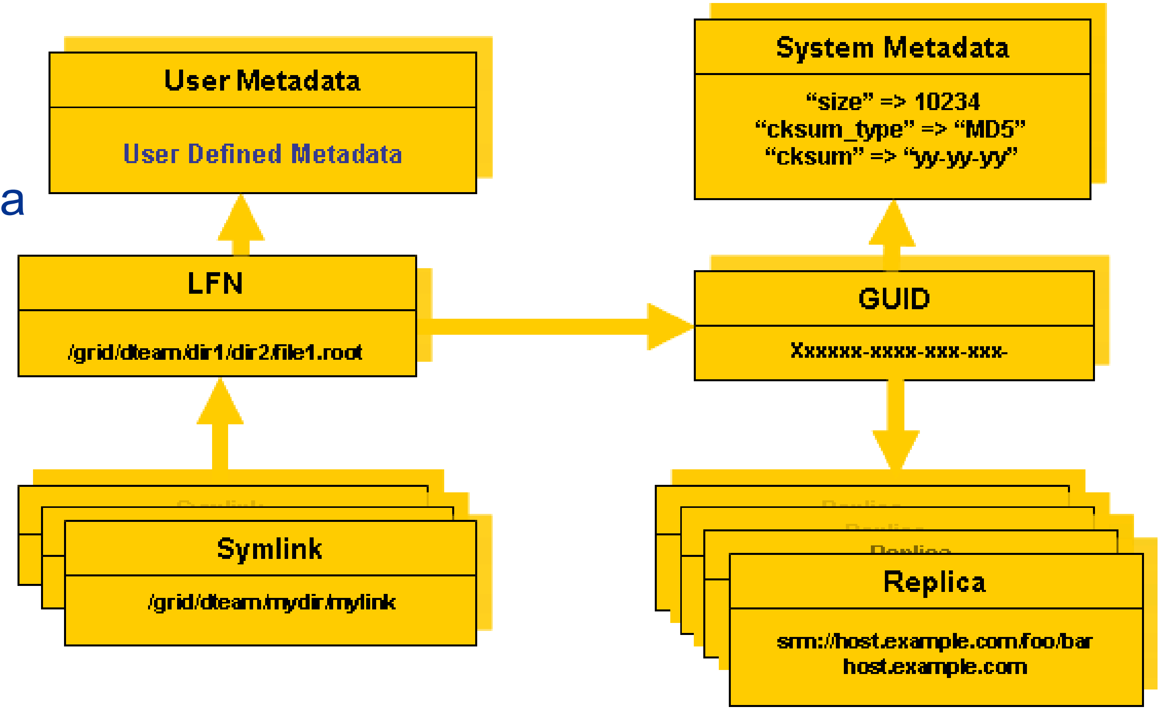
Name conventions

- Logical File Name (LFN)
  - An alias created by a user to refer to some item of data, e.g. “lfn:cms/20030203/run2/track1”
- Globally Unique Identifier (GUID)
  - A non-human-readable unique identifier for an item of data, e.g. “guid:f81d4fae-7dec-11d0-a765-00a0c91e6bf6”
- Site URL (SURL) (or Physical File Name (PFN) or Site FN)
  - The location of an actual piece of data on a storage system, e.g.
    “srm://pcrd24.cern.ch/flatfiles/cms/output10_1” (SRM)
    “sfn://lxshare0209.cern.ch/data/alice/ntuples.dat” (Classic SE)
- Transport URL (TURL)
  - Temporary locator of a replica + access protocol: understood by a SE, e.g.
    “gsiftp://lxshare0209.cern.ch//data/alice/ntuples.dat”



The Logical File Catalog

- **Manages the identification, sharing and replication of data in the gLite Grid.**
- **LFN acts as main key in the database. It has:**
  - Symbolic links to it (additional LFNs)
  - Unique Identifier (GUID)
  - System metadata
  - Information on replicas
  - One field of user metadata



Who does what with files?

- **Storage Element**
  - Storage Resource Manager — not provided by gLite
  - POSIX-I/O — gLite-I/O
  - Access protocols — gsiftp, https, rfio, …
- **Catalogs**
  - File catalog
  - Replica catalog — gLite LFC catalog
  - File authorization service — (MySQL or Oracle)
  - Metadata catalog — gLite standalone metadata catalog
- **File Transfer**
  - File Transfer Service
  - File Placement Service

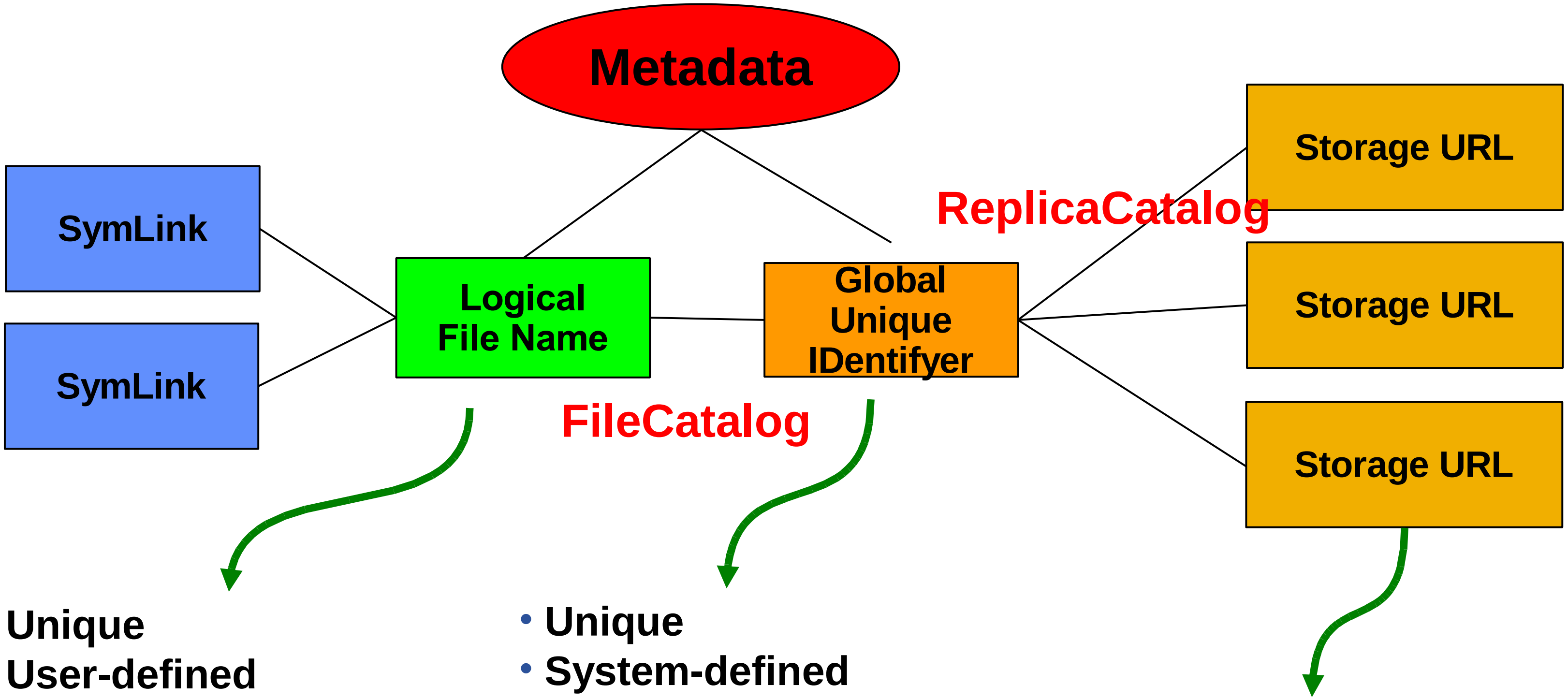
Where are my files?

- **Catalog (eg. LFC) remembers locations of files**
  - only deals with their locations (not data, not tranfers!)
  - data transfer handled separately: PFNs point to actual storage location and access protocol
- **Files can be replicated on multiple SEs**
- **Each file registered has a unique ID**
  - same file gets different IDs when registered multiple times

- **LFNs are names that make sense to you**

Layer of abstraction: file location irrelevant

Identifying files!?



Metadata

SymLink

SymLink

Logical File Name

Global Unique IDentifyer

ReplicaCatalog

FileCatalog

Storage URL

Storage URL

Storage URL

- Unique
- User-defined
- Mutable

/grid/me/test.txt

- Unique
- System-defined
- Immutable GUID

000-000-001-002

- Allows file retrieval

srm://host.net:8443/srm?SFN=/srm/my.site/myvo/grid/me/test.txt

File Transfer Service

- **Handles data management jobs**
  - “RB” for data jobs
- **Responsible for reliable file transfers between grid sites**
  - transfers (sets of) files between 2 SE's
  - endpoints with same protocol (gsiftp, ...)
- **Can be shared among VOs**

FTS, how to move files around

- **Transfer jobs**
  - identifier
  - state
  - files (source/destination PFN pairs)
  - support MyProxy
    - glite-transfer-submit
    - glite-transfer-status
- **Channels**
  - point to point (cern.ch – fzk.de) queues
  - state
  - bandwidth
  - concurrent tranfers
  - can be managed
    - production channels
    - default channel (free internet)

File Placement Service

- **Understands logical source files**
  - copy lfn:///grid/myvo/mytest.txt
- **Understands logical destination**
  - transfer to cern.ch
- **Updates the File Catalogs**
  - registers new replica SURL in LFC
- **Builds on FTS**

Conclusions

- More standards compliant (WS)
- More security, virtualization of resources

- Some components evolving keeping compatibility
- Commands renamed, same functionality
- New / rearchitected components

- Several required features implemented
- Some requirements still pending
- New features expected

- Current: gLite 3.0.5 (for most sites)
- Expected soon: gLite 3.0.10

Q & A