The gLite middleware

architecture and components

Ariel Garcia

Forschungszentrum Karlsruhe

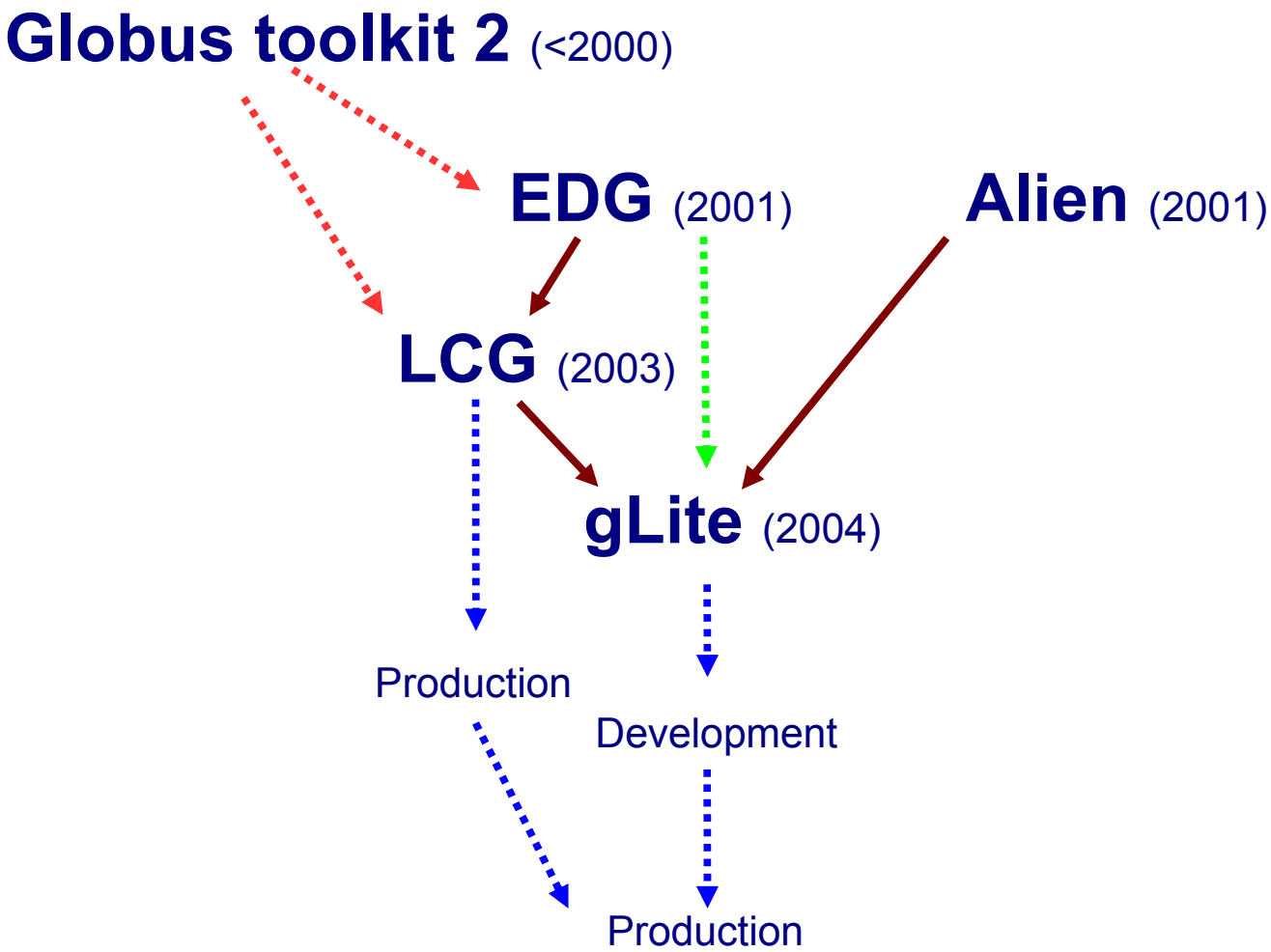
Outline

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

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

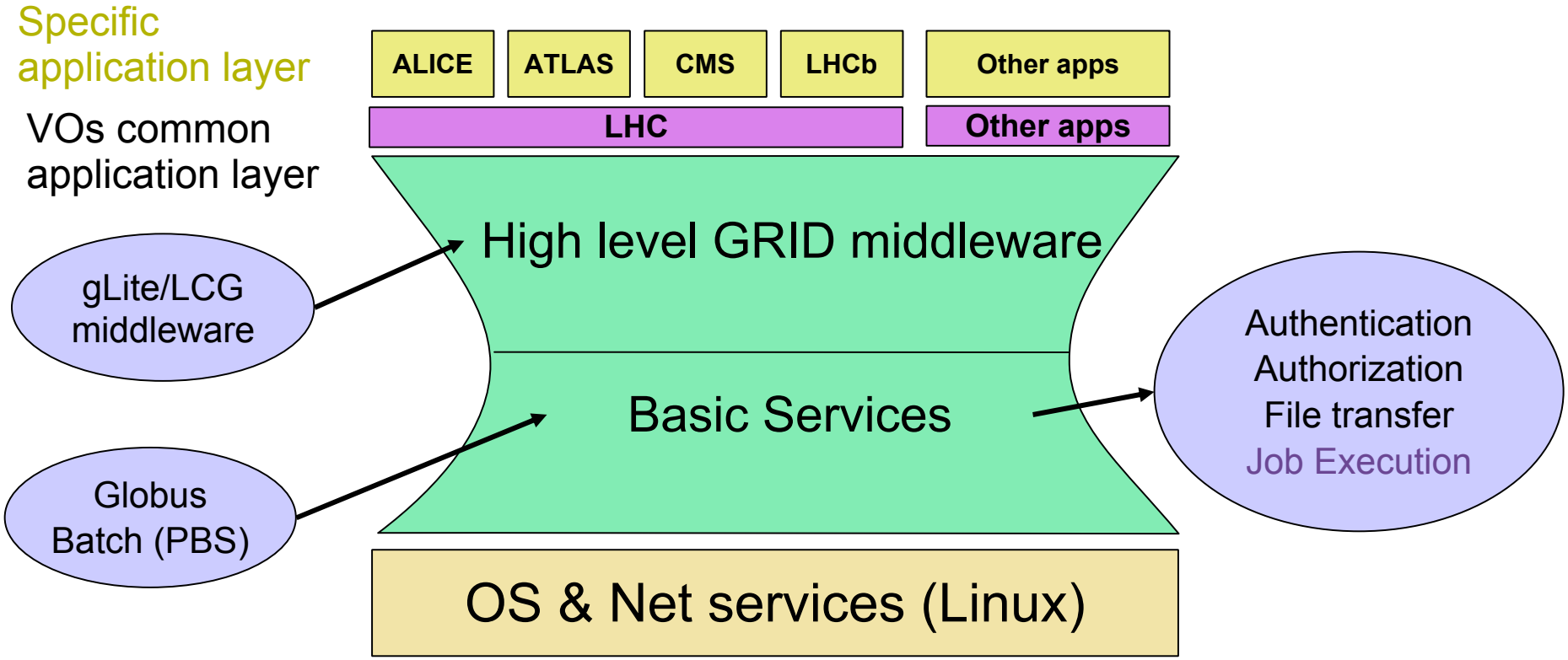
- **Conclusions**

Some history



Software layers

- **Middleware keeps the grid together**



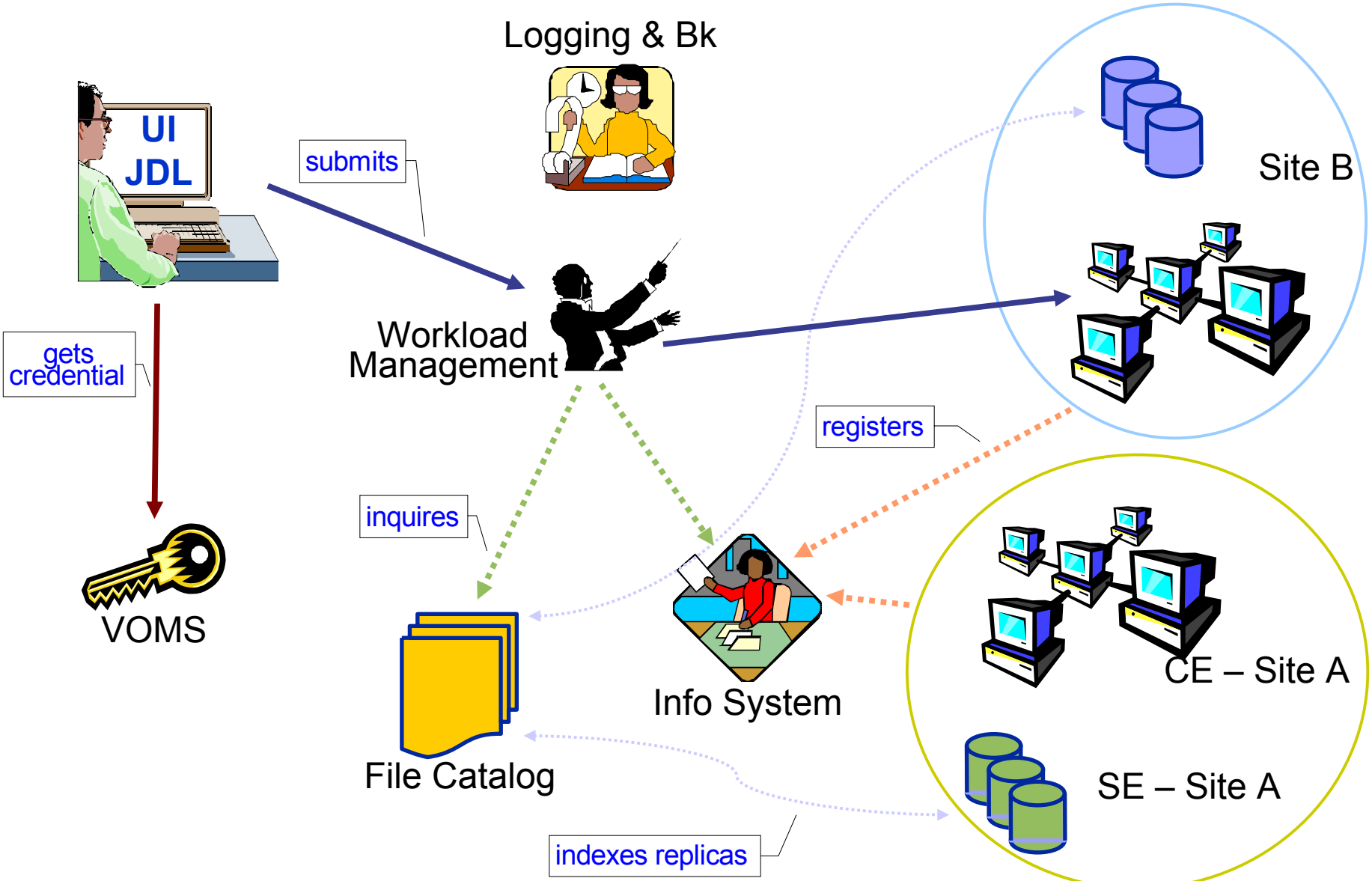
gLite:

- **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

New features:

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

A typical job workflow



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

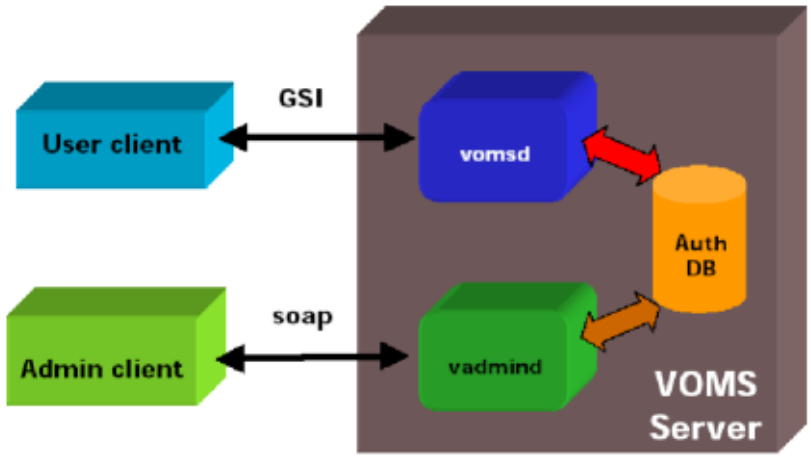
CE, SE: layer of abstraction, local peculiarities irrelevant

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 (FiReMan)
    - File Transfer Service (FTS)
    - File Placement Service (FPS)

Virtual Organization Membership Service

- Multiple VOs
- Multiple roles in VO
  - compatible X509 extensions
  - signed by VOMS server
- Web admin interface
- Supports MyProxy

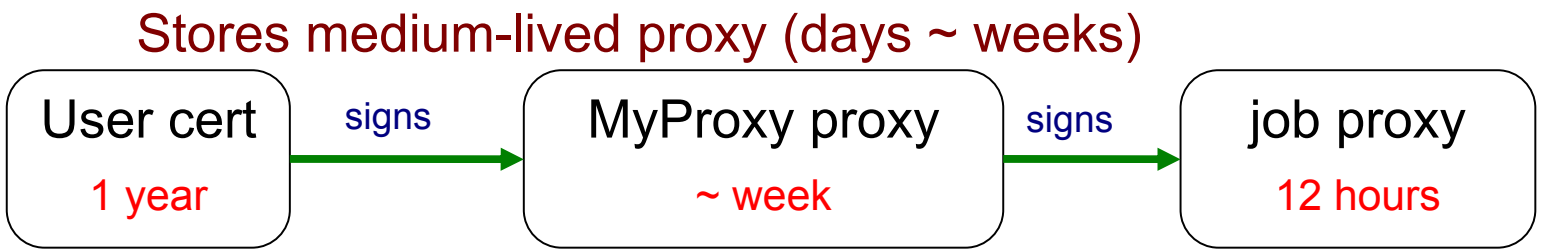


- Resource providers grant access to VOs or roles
- Sites map VO members/roles to local auth mechanism (unix user accounts)
  - allows for local policy

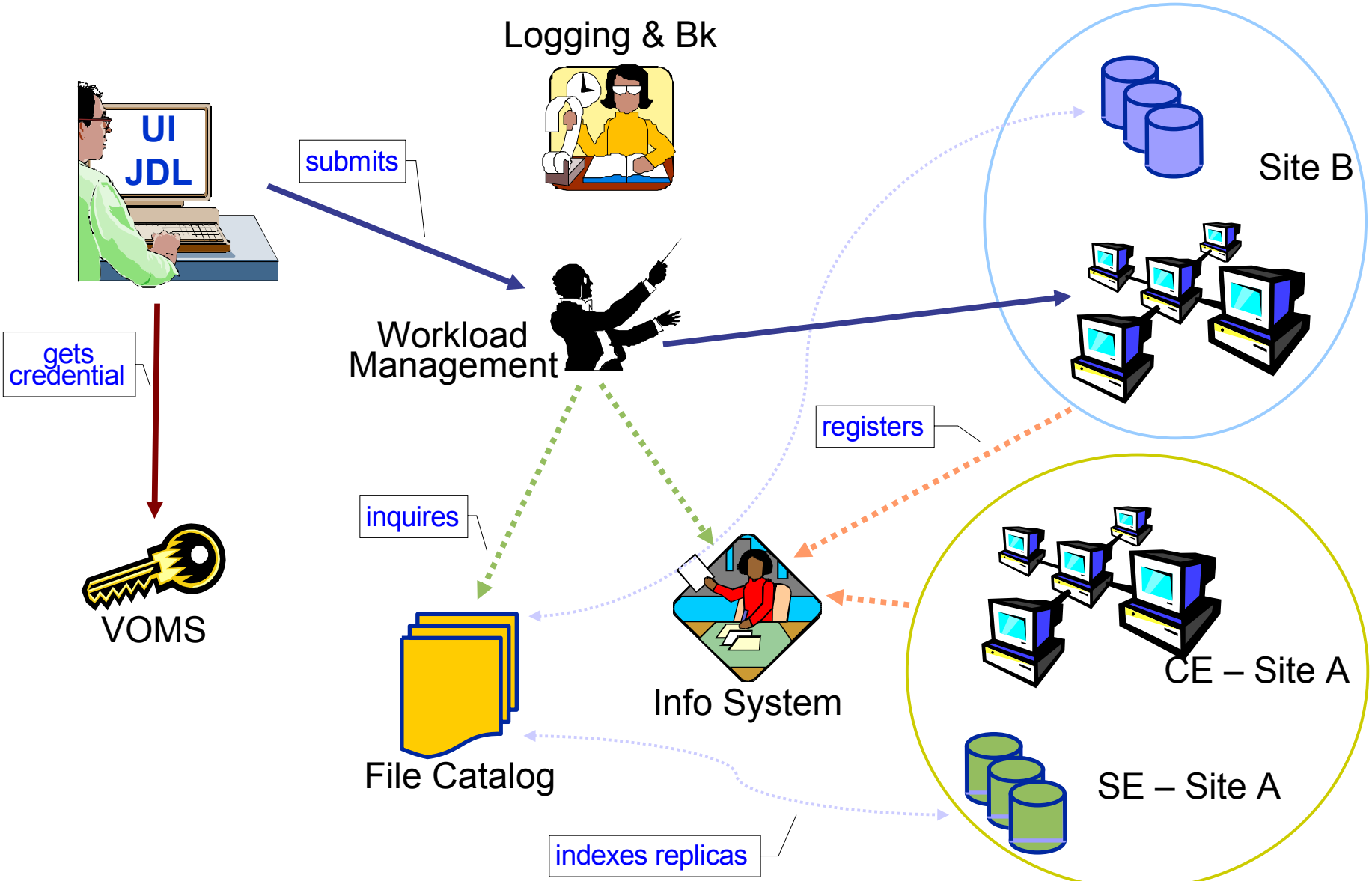
Layer of abstraction: individual members irrelevant

MyProxy

- **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



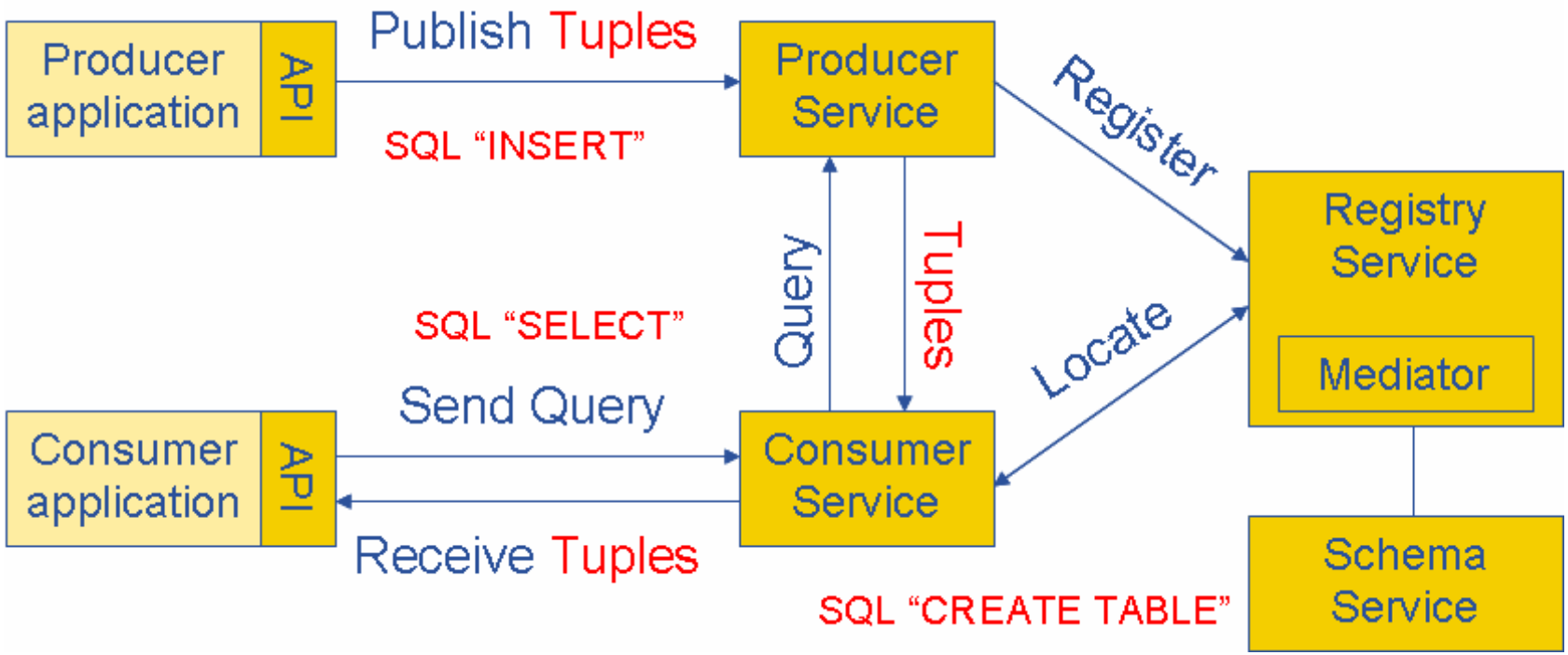
A typical job workflow



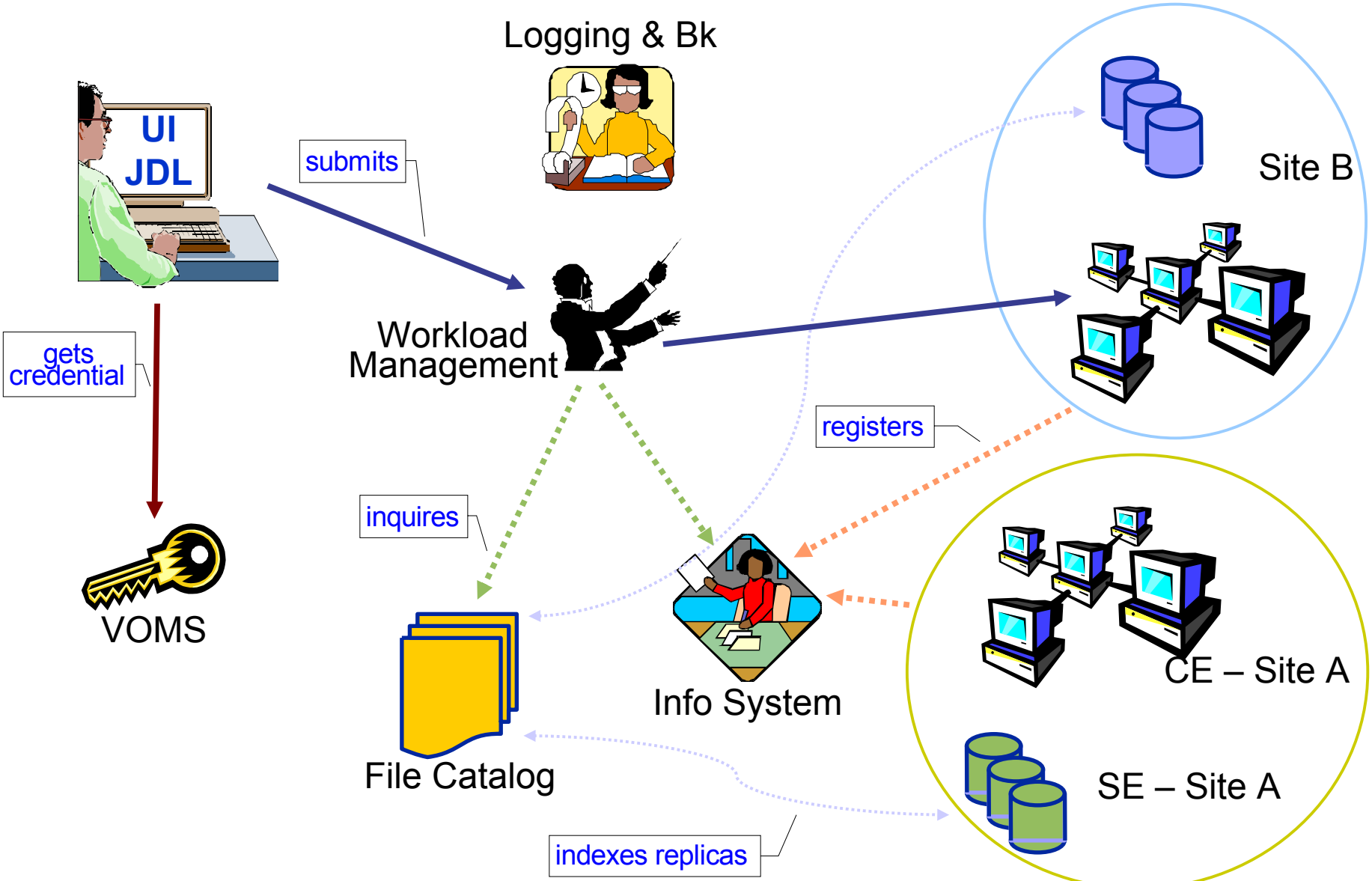
Information system

- **Based on R-GMA**
  - relational (database-like) implementation of the GGF Grid Monitoring Architecture
  - 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



A typical job workflow



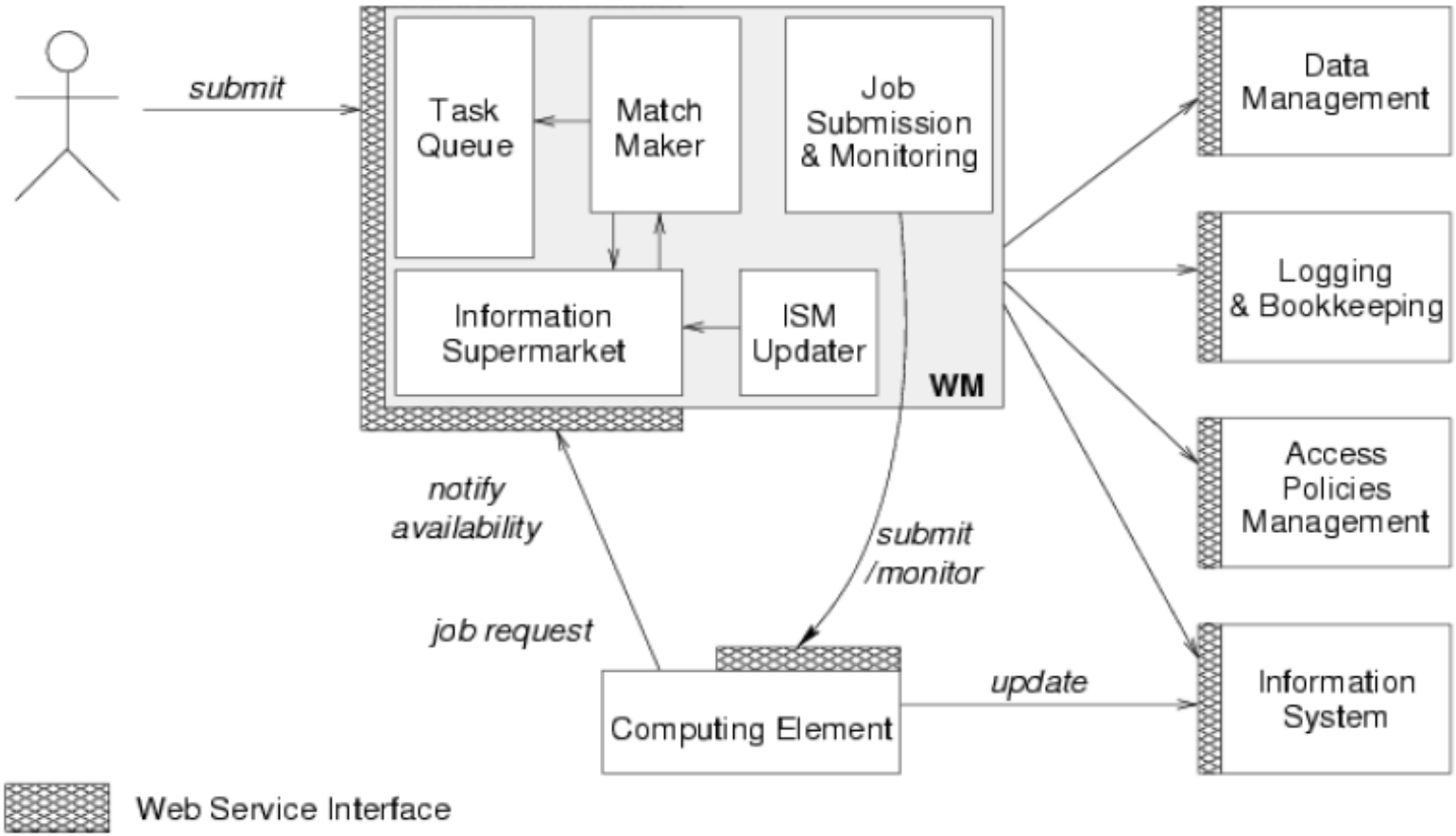
Job Submission

- **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**

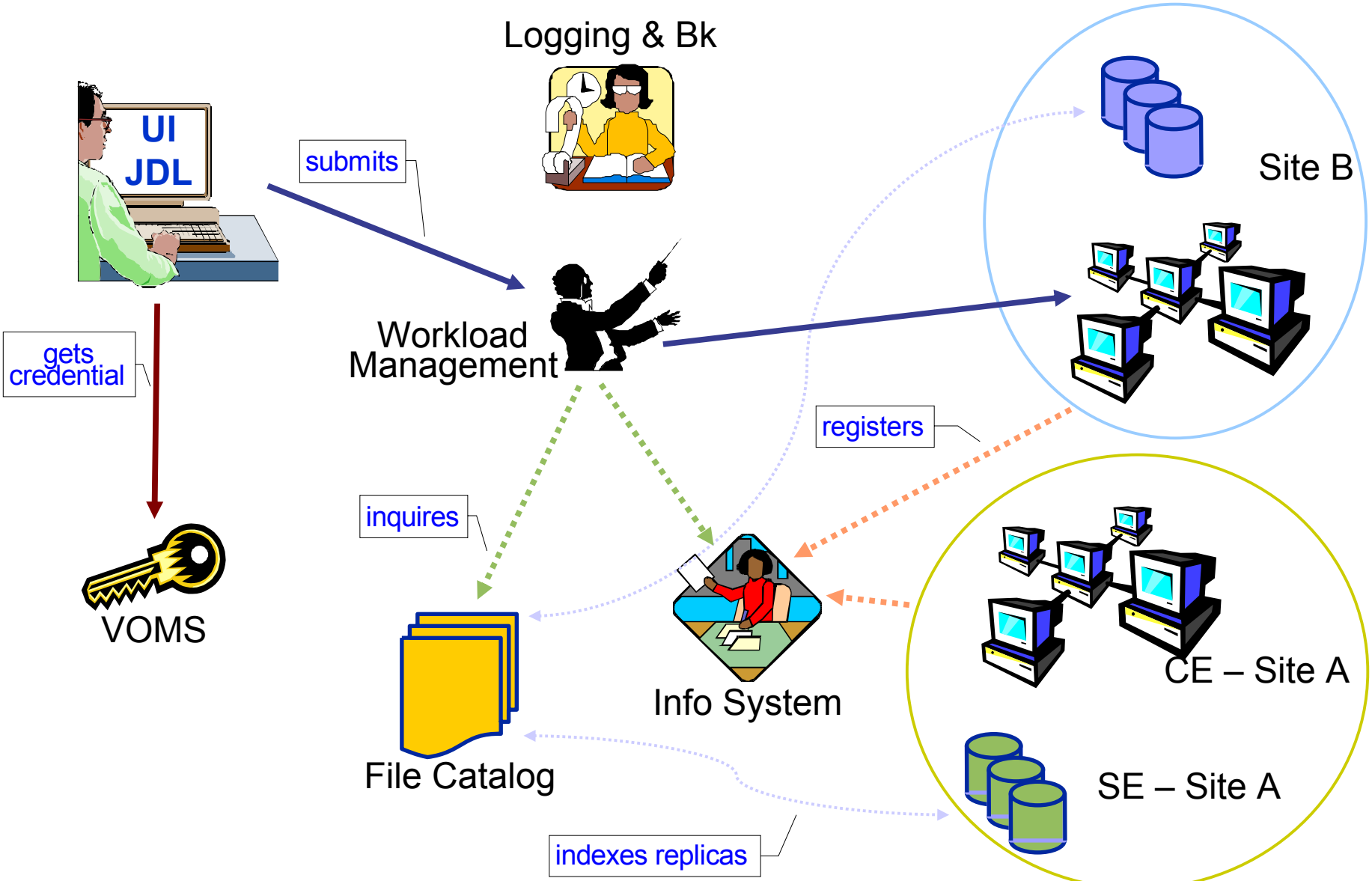
- **WMS is primary job execution interface for users**
- **each server allows only certain VOs / groups**

Layer of abstraction: sites irrelevant

Job Management Services



A typical job workflow



Data Management Services

- **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
  - File authorization service
  - Metadata catalog

  gLite FiReMan catalog (MySQL and Oracle)

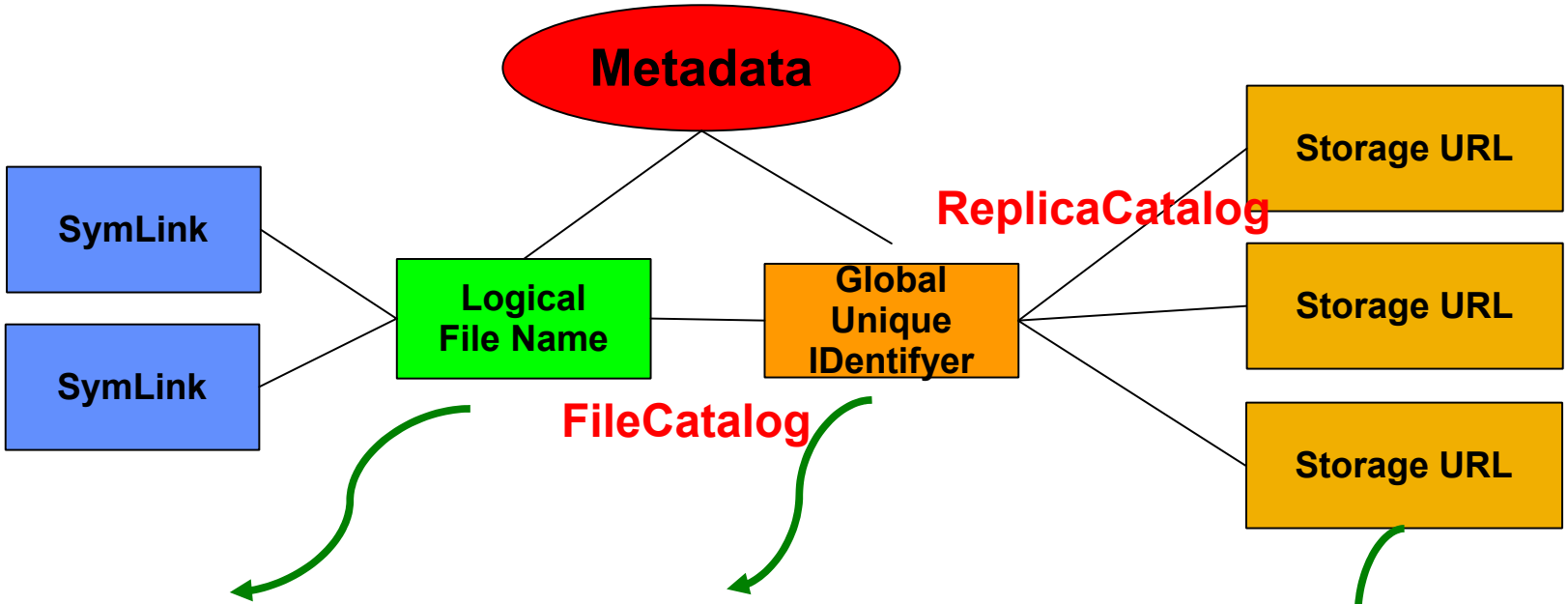
  gLite standalone metadata catalog
- **File Transfer**
  - File Transfer Service
  - File Placement Service

Data management

- **Catalog 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

FiReMan Catalog



- Unique
- User-defined
- Mutable

/grid/me/test.txt

- Unique
- System-defined
- Immutable GUID

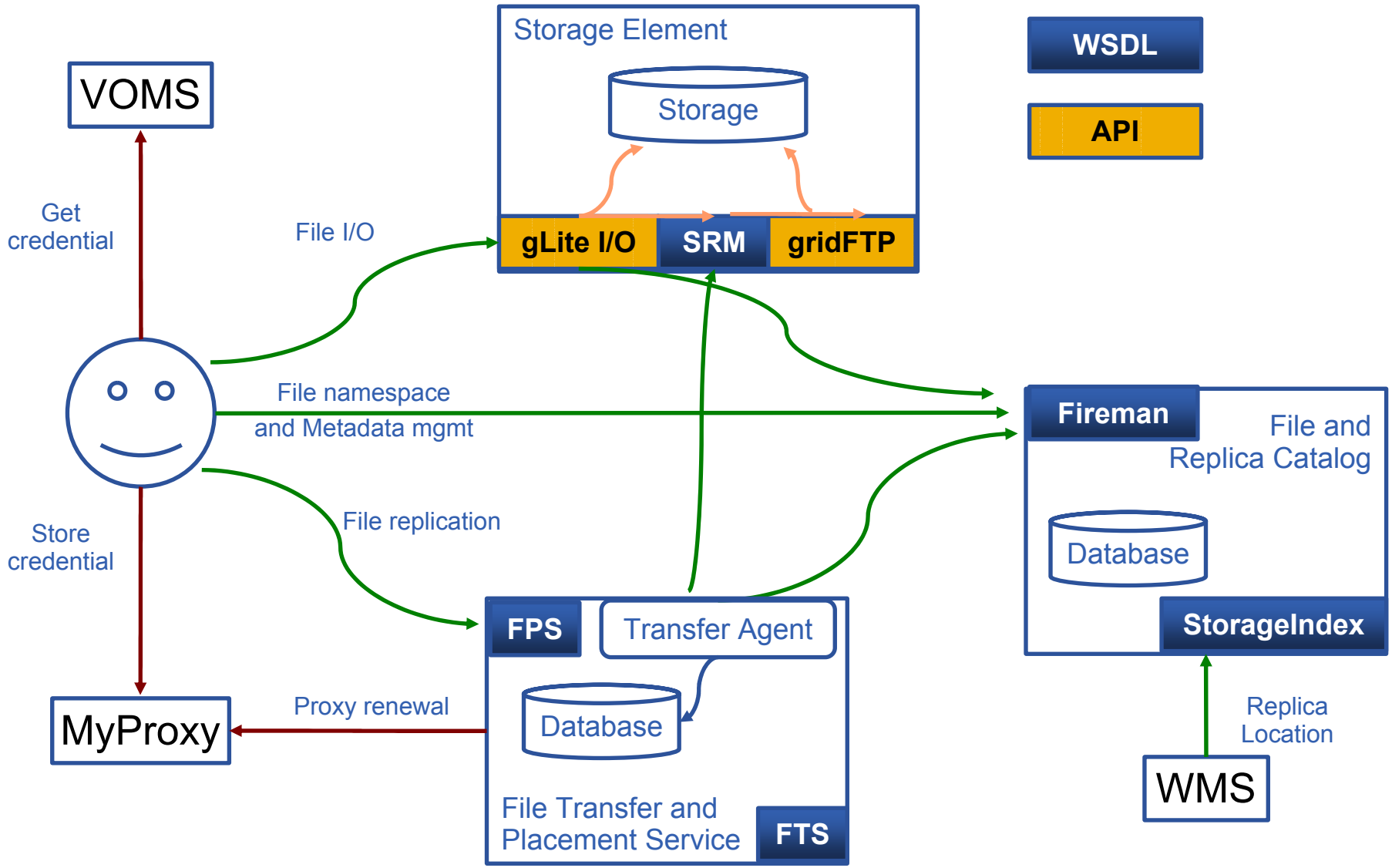
000-000-001-002

- Allows file retrieval

srm://host.net:8443/srm?SFN=/srm/

my.site/myvo/grid/me/test.txt

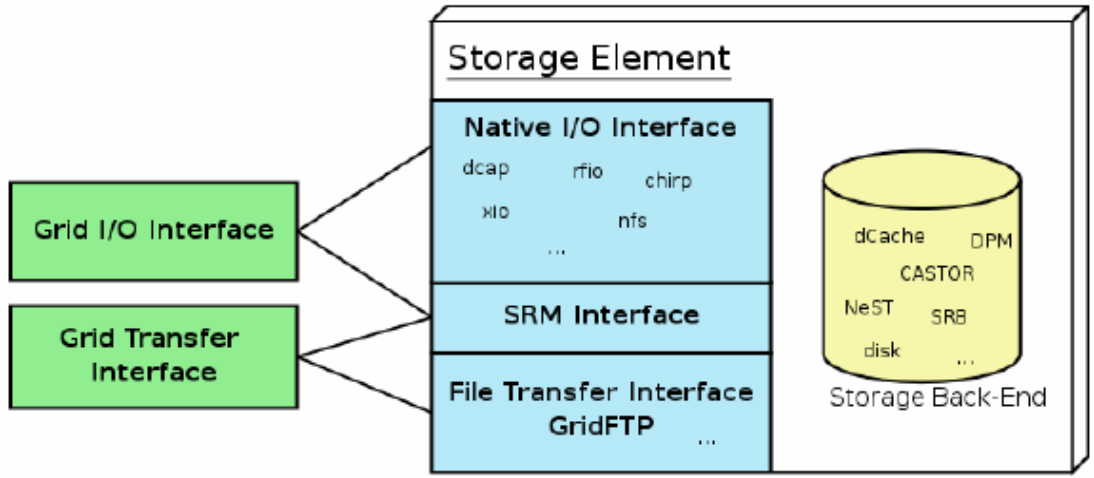
DM Interactions

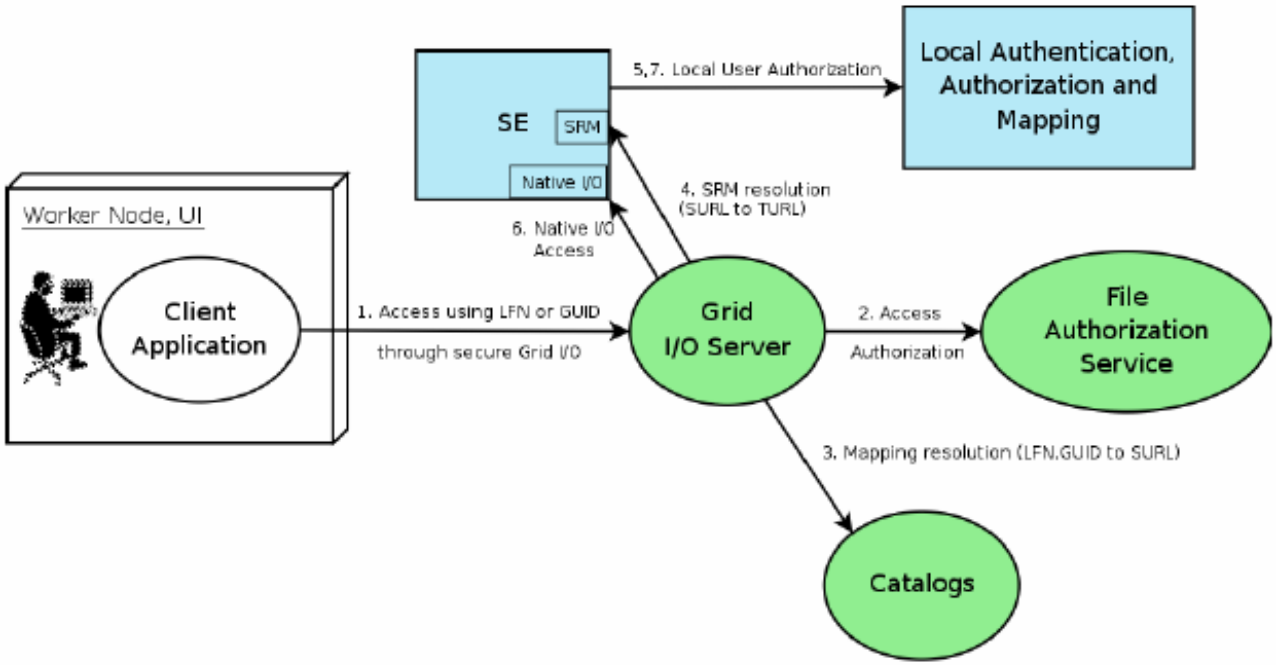


gLite-I/O

Provides POSIX-like access to grid files

- both CLI & API
- GUID or LFN can be used, i.e. open("/grid/myFile");





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**

File Transfer Service

- **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 Fireman
- **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

- Still moving target
- Current: gLite 1.3
- Expected soon: gLite 1.4

The end

Thanks for listening!