# gLite Overview

*Riccardo Bruno*
*INFN*
*gLite Tutorial - BioinfoGrid*
*Bari, 8/3/06*

*Presented by David Fergusson*

www.eu-egee.org

Information Society

# Grid Aim

Grid Systems & Applications aim is to:

- ***Integrate***
- ***Virtualise***
- ***Manage***


**RESOURCE**s and **SERVICE**s across different **VO**s.

- **VO** – Individuals and/or Institutions having direct access to resources.

# Grid Requirements 1

- **Heterogeneous** (OSes, Devs, Apps.)
- **VO Resource Sharing** (Management, Security and Accounting)
- **Resource Utilisation** (Reservation, Metering, Monitoring and Logging)
- **Job Execution** (VO access, QoS, LCM, WFM, SLA)
- **Data Services** (Integration, Provisioning, Cataloguing)

# Grid Requirements 2

- **Security** (Authentication, Authorisation and Auditing) - provenance
- **Administrative Costs** (Provisioning, Deployment and Configuration)
- **Scalability**
- **Availability** (Disaster Recovery, Fault Management)
- **Specific Requirements**: (**EGEE**: HEP, BioMed)

https://savannah.cern.ch/support/?group=egeeptf

# gLite - Middleware

- Many **VO**s need **sharing of resources** through services
  - **Accessing**
  - **Allocating**
  - **Monitoring**
  - **Accounting**


- **Grid Middleware** - Layer between services and physical resources
- **gLite** - Lightweight Middleware for Grid Computing

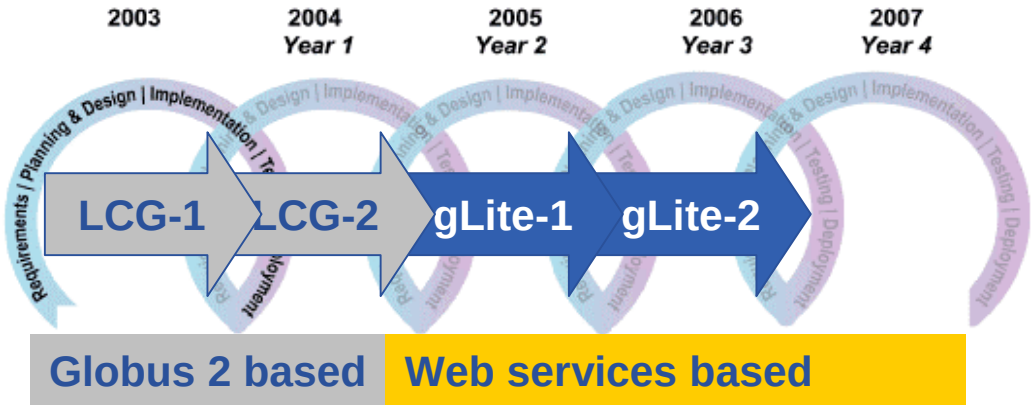
http://www.glite.org

# gLite - Background

## Other Grid Projects:

- Global Grid Forum - GGF
- Open Grid Services Architecture – OGSA
- EU DataGrid
- AliEn
- Globus
- Condor
- ...

## LCG:



# gLite – SOA Compliant 1

**The Past:**

Single Centralised System

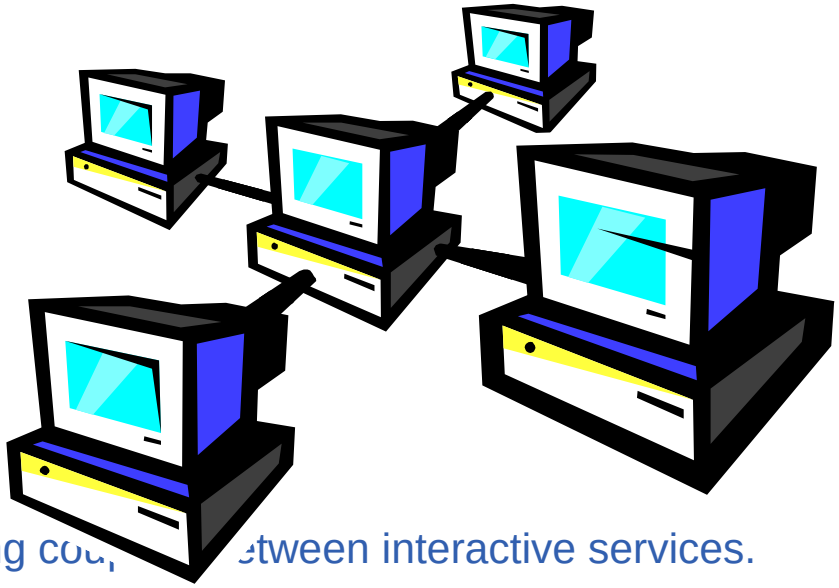
| | |
|---|---|
| Error Prone<br>Single point of failure | Simple control<br>- not scalable |
| Costly | inflexible |

# gLite – SOA Compliant 2

**Now:**

Applications and Services on Distributed Environments Through the Network

| | |
|---|---|
| Reuse | Flexible |
| Modifications | Extensible |
| | SW Quality |
| | Productivity |



- **SOA -** Distributed Systems as Services loosing cou[illegible]etween interactive services.
- **Loose coupling - flexibility, dynamic configurations, encapsulation**
- **SOA further abstraction of Code Reuse - closer to object orientation.**

# gLite – SOA Services

## Functions

- **Well-Defined**
- **Self-Contained**
- **Independent**
- **Message Based Interface**

## Messaging

Service interaction by messages having a common messaging infrastructure

**SOAP (Web Services) –** Std Protocol to manage Messaging among Services

**WSDL -** A language that expose the service interface.

## Policies

Security, QoS, Management

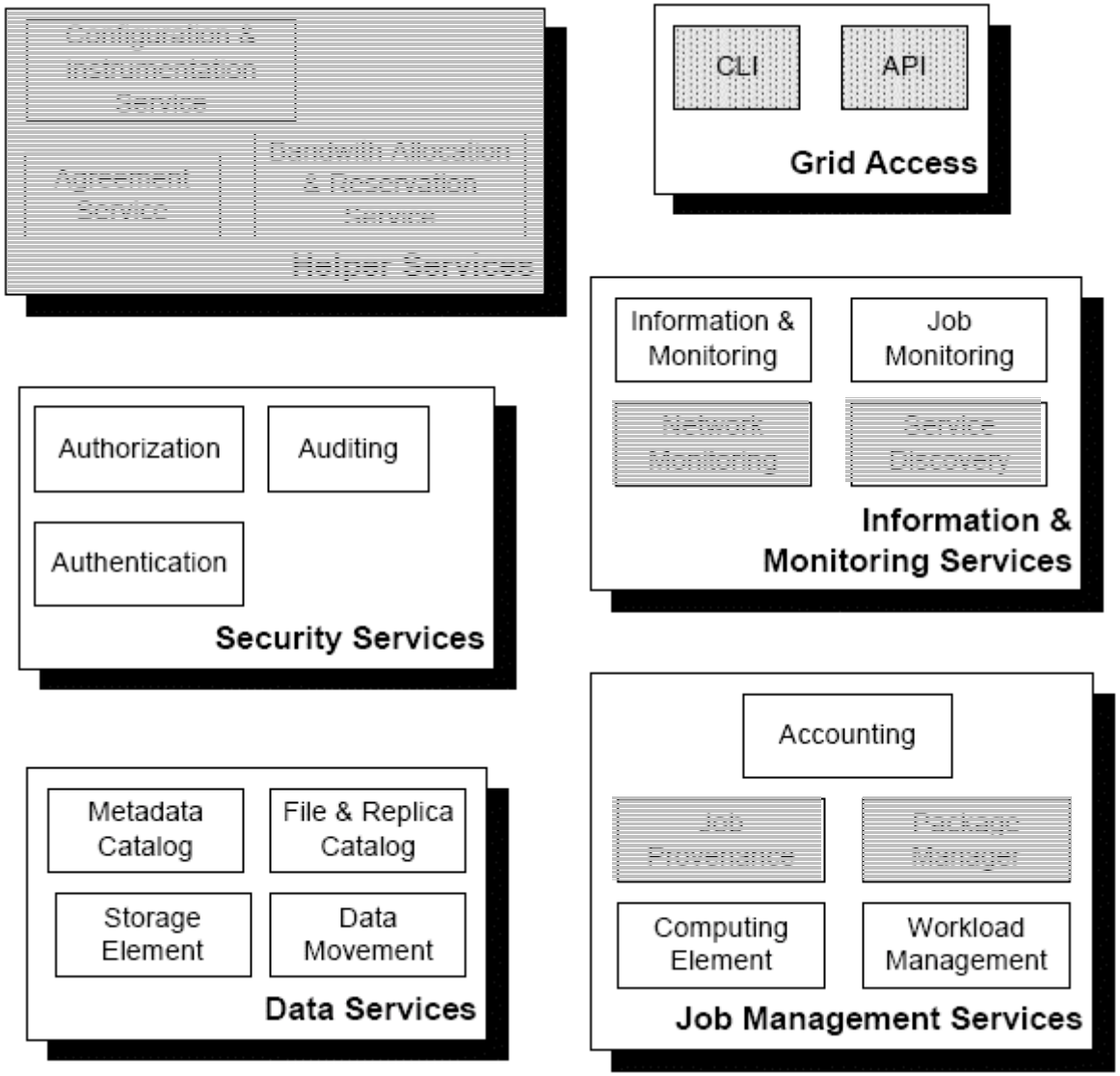
## State (Business Logic) - added to web services (by grid or new standards, WSDL)

Consistent, Accurate and Durable

# gLite – Service Decomposition

**5 High level services**

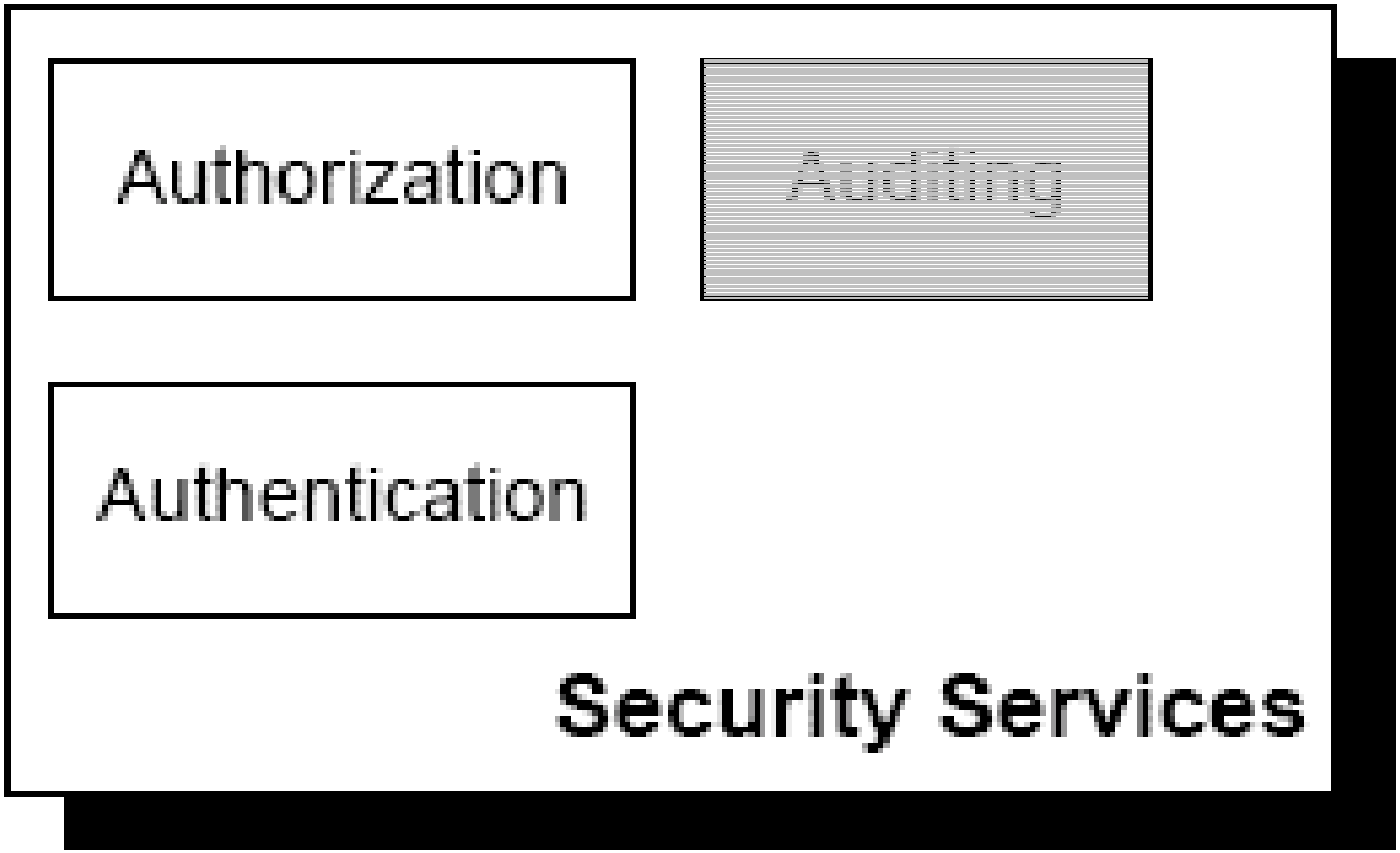
**+ CLI & API**




Legend:

- Available
- Soon Available

# gLite – Security Services



**Identify entities** (users, systems and services) when establishing a context for message exchange (Who are you?).

**Aim** - Provide a Credential having a universal value that works for many purposes across many infrastructures, communities, VOs and projects.

gLite uses: the **Private Key Infrastructure** (X.509) infrastructure using **Certification Authorities** as trusted third parties.

gLite uses**:** ***MyProxy*** **(http://grid.ncsa.uiuc.edu/myproxy/) extended by VOMS**.

**Trust domain**: The set of all EGEE CAs is our Trust Domain.

- **Revocation**: Identities must be removed in a timely manner
- **Credential Storage**: Local or delegated credential (Services or Users)
- **Privacy Preservation**: Use of personal data
- **Security Consideration**: Trusted computing env. Do not serve a whole VO.

# Virtual Organisation Management Service

- **VOMS**
- **Allows fine grained control of access**
  - Roles - define policies towards particular groups
- **Supports Access Control Lists**
  - Like UNIX permissions (eg. Owner, group, world)

**Allows or denies access** to services, based on **policies.**

- **Agent**: The user interacts with a centralized Authorization Server
- **Push**: Authorization Services issue Tokens.
- **Pull**: The resource asks to the Authorization Services.

Authorization Sources:

- **Attribute Authority (AA):** User **<->** Set of Attibutes. (**VOMS**)
- **Policy Assertions:** Third party policies. (CAS)

- **Policy combination and Evaluation**: Combine policy information from a number of different sources. (XACML)

- **Mutual Authorization**: Client-Service (i.e. issuing sensitive data)

# gLite – Security Services
## Auditing, Delegation, Sandboxing

**Auditing** - Monitoring and Post-Mortem analysis of security related events.
In computational grids It goes hand by hand with the accounting.

- **Who did what?**
- **Where and when?**
- *In case of accounting:*
  - ***For how long?***
  - ***For how much?***

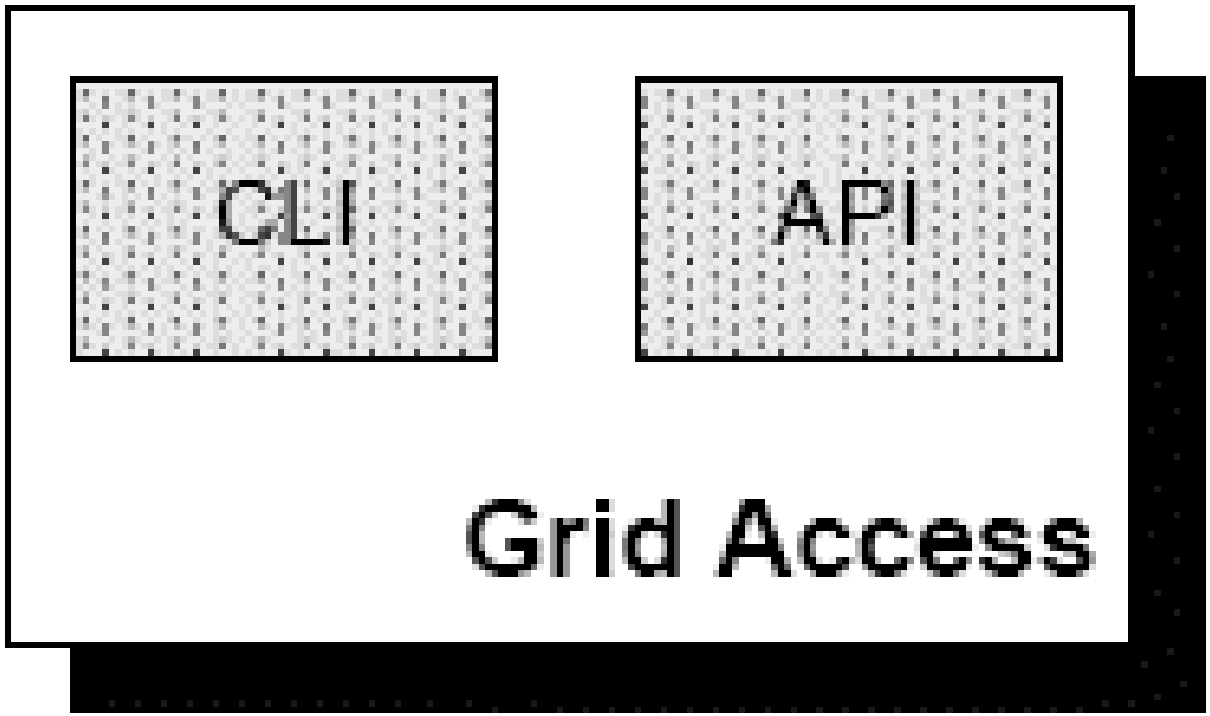
- **Delegation:** The need of delegate privileges to other entities is done by **Proxy Certificates**. This is the most widely adopted mechanism by Grid communities. (Also: Single Sign-On, Dynamic entity identification).

- **Sandboxing** - Grid applications need the isolation of assigned resources in a transparent fashion by Security services: AuthN and AuthZ. (Virtualisation).

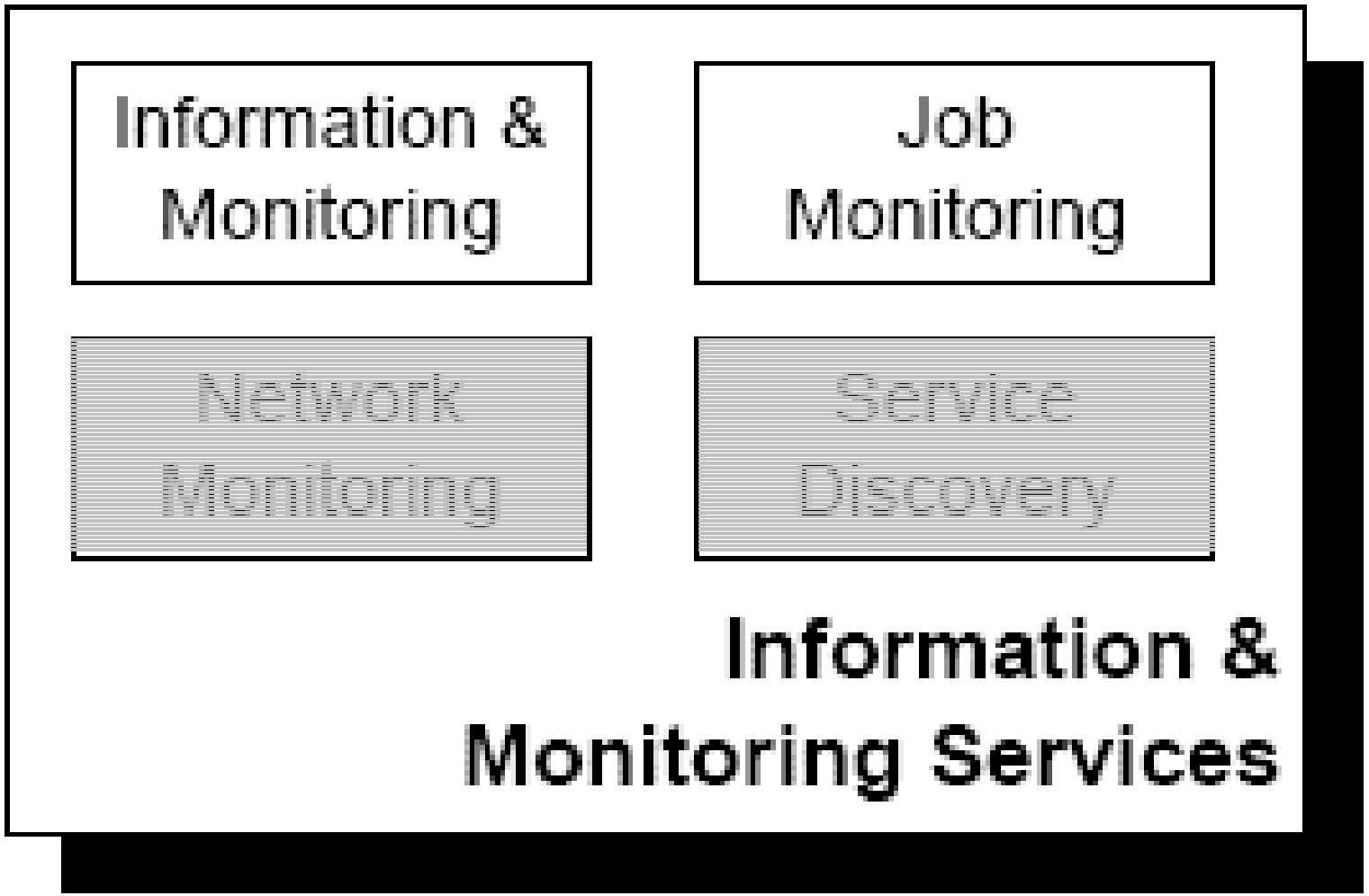
# gLite – Grid Access

## Two possibilities: **API**s and **CLI**.

The use of web-services allows the automatic generation of APIs (error prone, lack of tools).



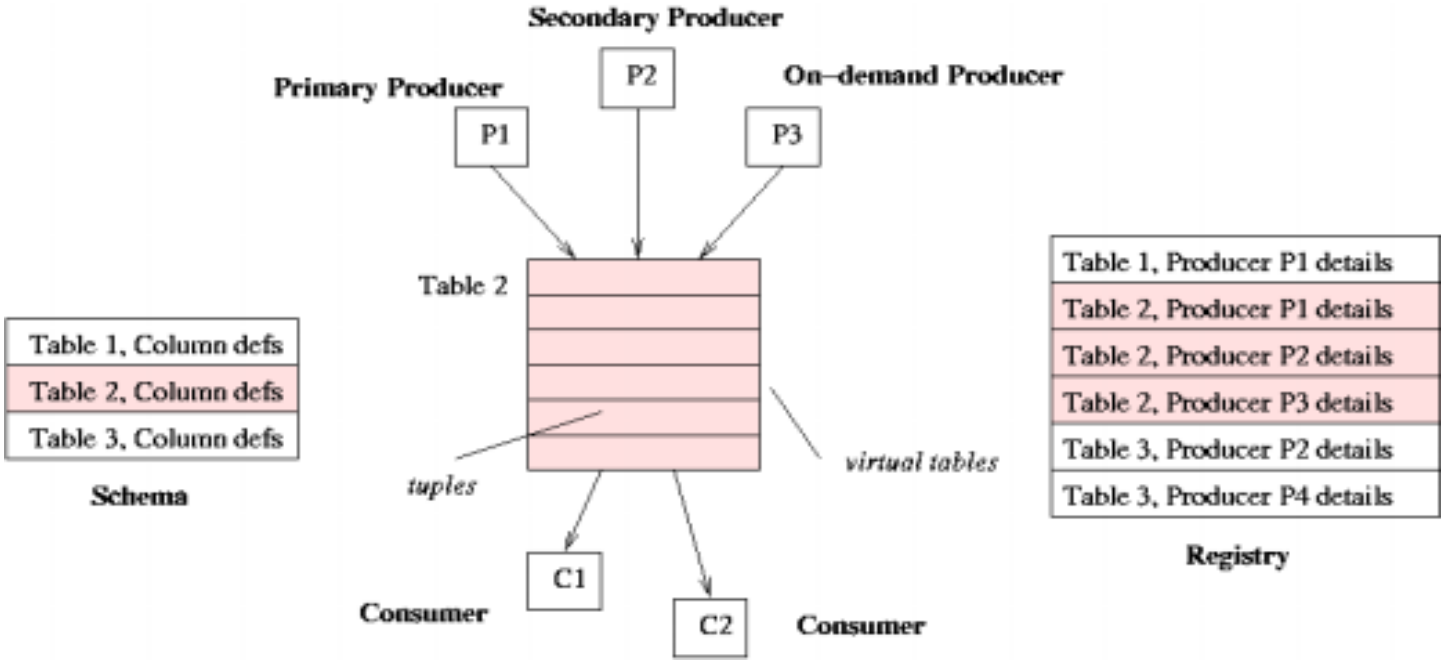
Information services are vital low level component of Grids.



# gLite – Information and Monitoring Services

## Basic info and monitoring services (RGMA)

- Information is provided by a **Publish** and **Consume** mechanism.
- Appearance of a single federated database to query through the SQL.
- Each **VO** has a **Virtual DataBase**.
  - **Schema** - Contains tables (***GLUE***)
  - **Registry** – List of available sources of information (Mediation)
  - **Producers** – Source of information (Primary, Secondary, On-demand)
  - **Consumers** – Make queries against tables (Continuous, Latest, History)

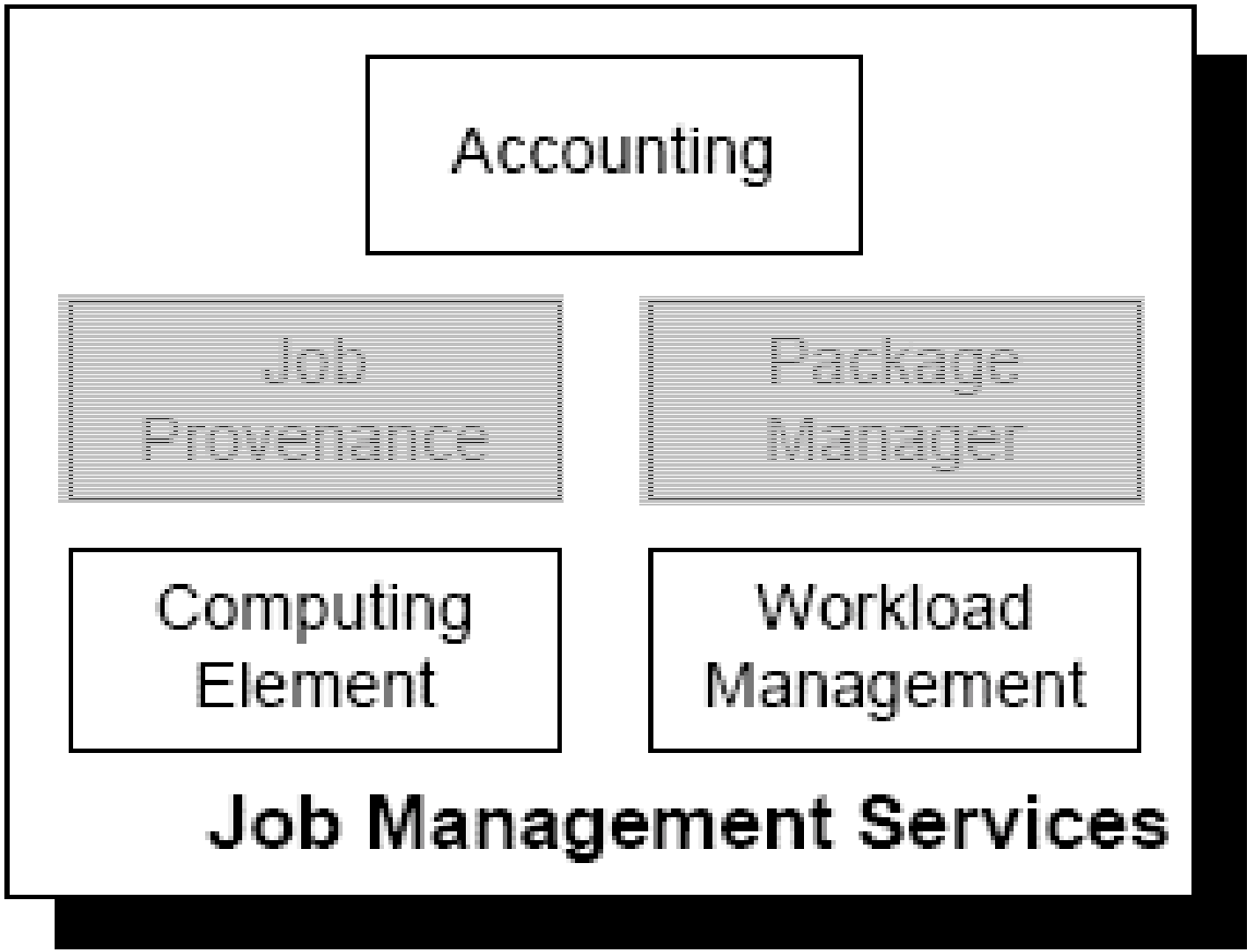


- **Job Monitoring** – Java logging service, log4j, Apache/Chainsaw (for other languages).

- **Service Discovery** – Locates suitable services for both users and services (Library!).

- **Network Performance Monitoring** – Many network monitoring frameworks. Aim: perform a standard interface to those frameworks.

# gLite – Job Management Services



# gLite – Job Management Services

## Accounting

- Accumulates information about the **resource usage** done by users or groups of users (VOs).
- Information on Grid Services/Resources needs **sensors** (Resource Metering, Metering Abstraction Layer, Usage Records).
- Records are collected by the **Accounting System** (Queries: Users, Groups, Resource)
- Grid services should  register themselves with a pricing service when accounting for billing purposes.

# gLite – Job Management Services

## Computing Element

- Service that represents the **computing resource** that is responsible of the job management: (submission, control, etc.)

- The CE may be used by a **Generic Client**: an end-user interacting directly with the Computing Element, or the **Workload Manager**, which submits a given job to an appropriate CE found by a **matchmaking** process.

- Two job submission models (accordingly to user requests and site policies):

  - **PUSH** (**Eager** *Scheduling*) (jobs pushed to CE),
  - **PULL** (**Lazy** Scheduling) (jobs coming from WMS when CE has free slots)

- CE must also provide information describing itself.
- CE responsible to collect accounting information.

# gLite – Job Management Services

## Workload Management

- **WMS** set of middleware components responsible of **distribution** and **management** of **jobs** across Grid resources.

- Two core components of WMS:

  - **WM**: **accepts** and **satisfy requests** for job management.
    Matchmaking is the process of assigning the best available resource.

  - **Logging & Bookeeping**: **keeps track** of job execution in term of **events:**
    (Submitted, Running, Done,...)

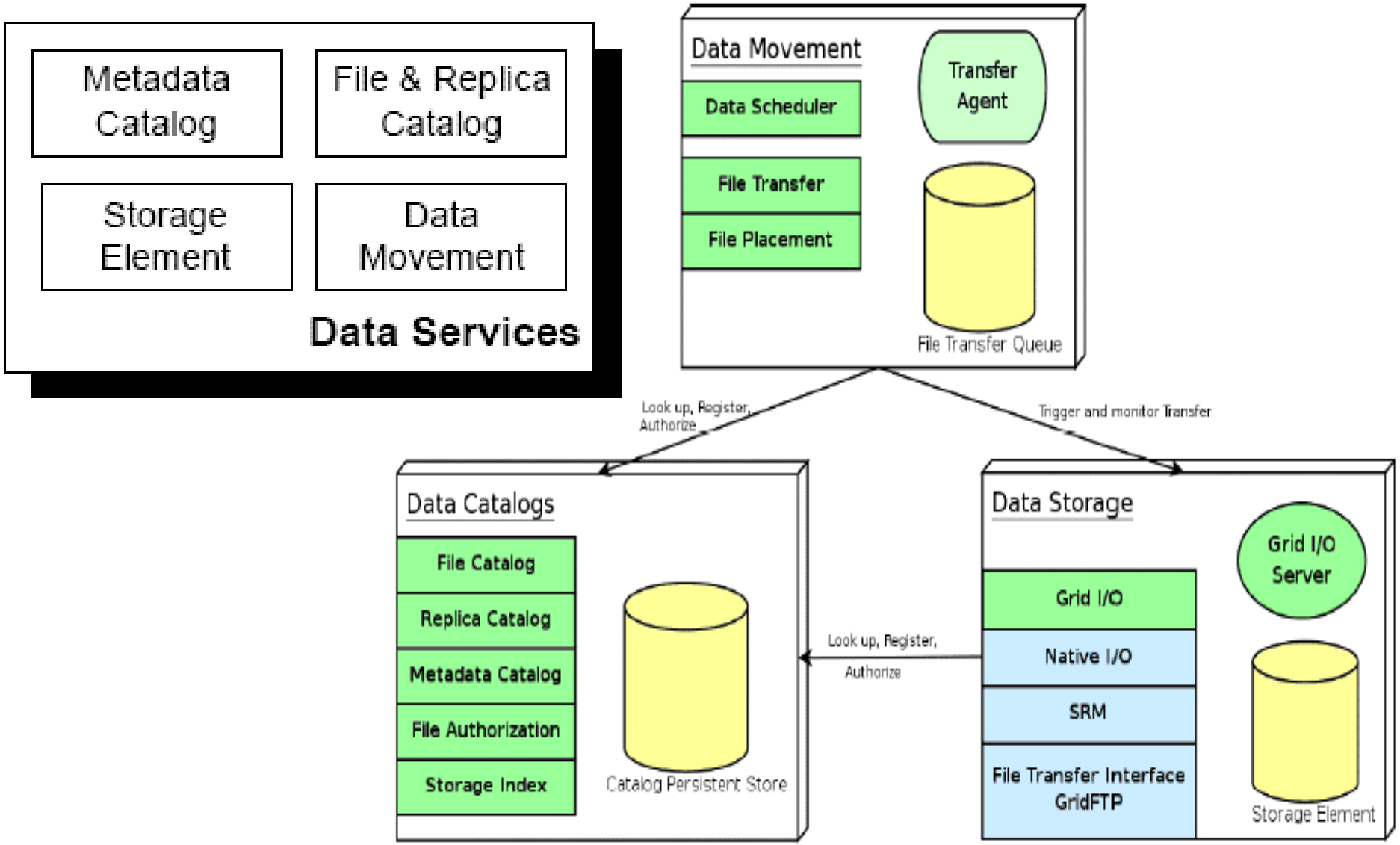
# gLite – Job Management Services

## Job Provenance, Package Manager

- **Job Provenance -** Keeps track of submitted jobs for long periods (months, years).

- **Package Manager –** Helper service to automate: installing, configuring, updating and removing of software components. (RPM, dpkg/APT, Portage, …)

# gLite – Data Services



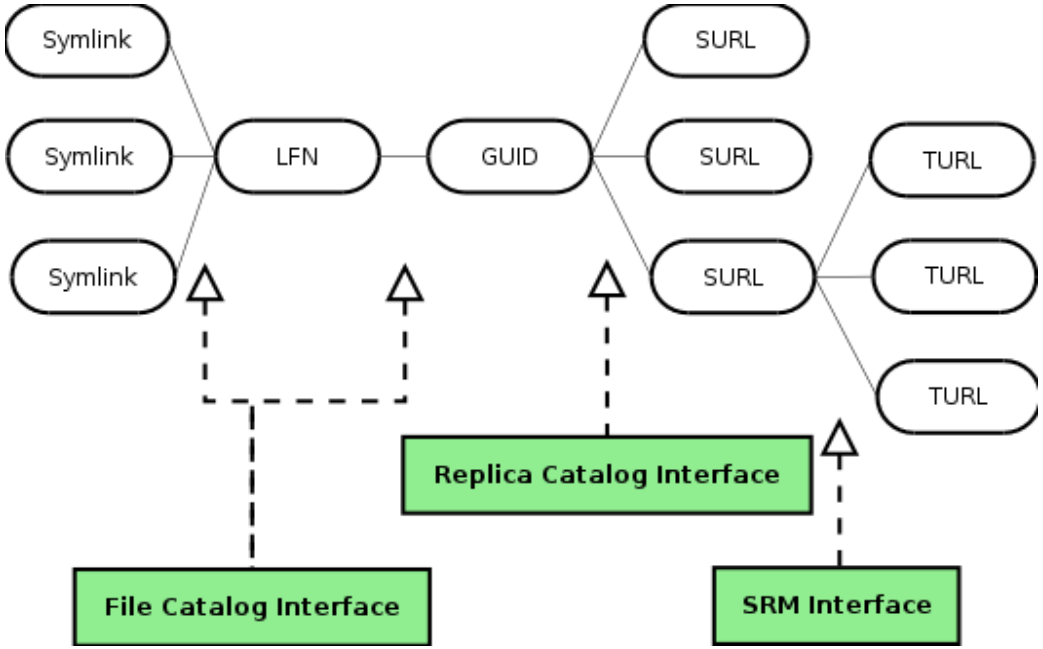
## Needed Service are at least:

- Storage back-end (Drivers and Hardware)
- Standard SRM Interface  (Storage Specific)
- Transfer service (GridFTP)
- Native POSIX like file I/O API (gLite-I/O, LFC)
- Auxiliary Accounting and Logging services

# gLite – Data Services
## Catalogs - keeping track of files

**OS like file access metaphor.**

- **LFN** (Logical file name)
- **GUID** (Grid unique identifier)
- **SymLinks**
- **SURL** (Site URL)
- **TURL** (Transfer URL)



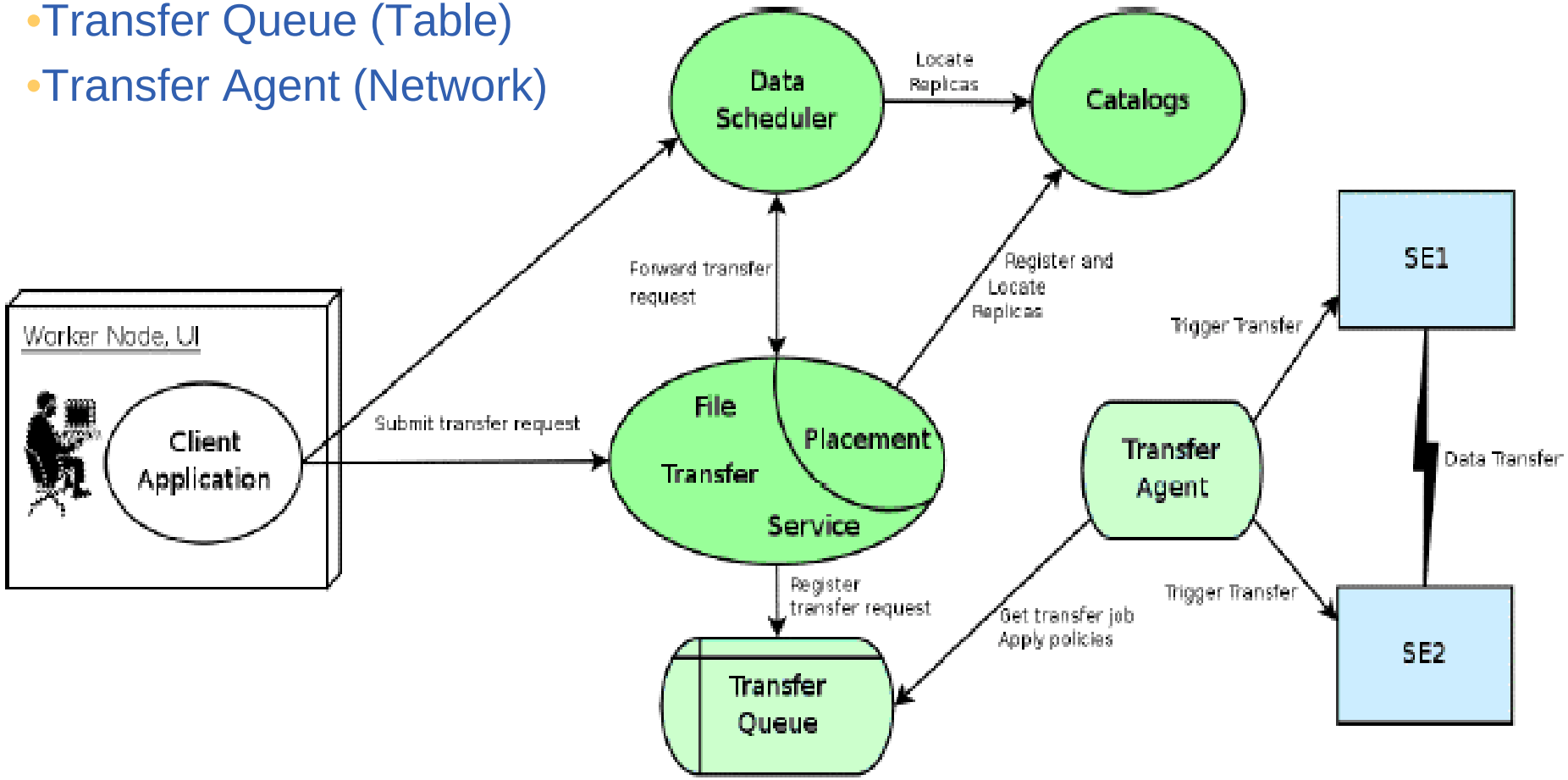

## Catalogs:

- Authorization Base
- Metadata Base
- Metadata Schema
- **Replica Catalog**
- **File Catalog**
- **File Authorization**
- Metadata
- **Combined Catalog**
- Storage Index

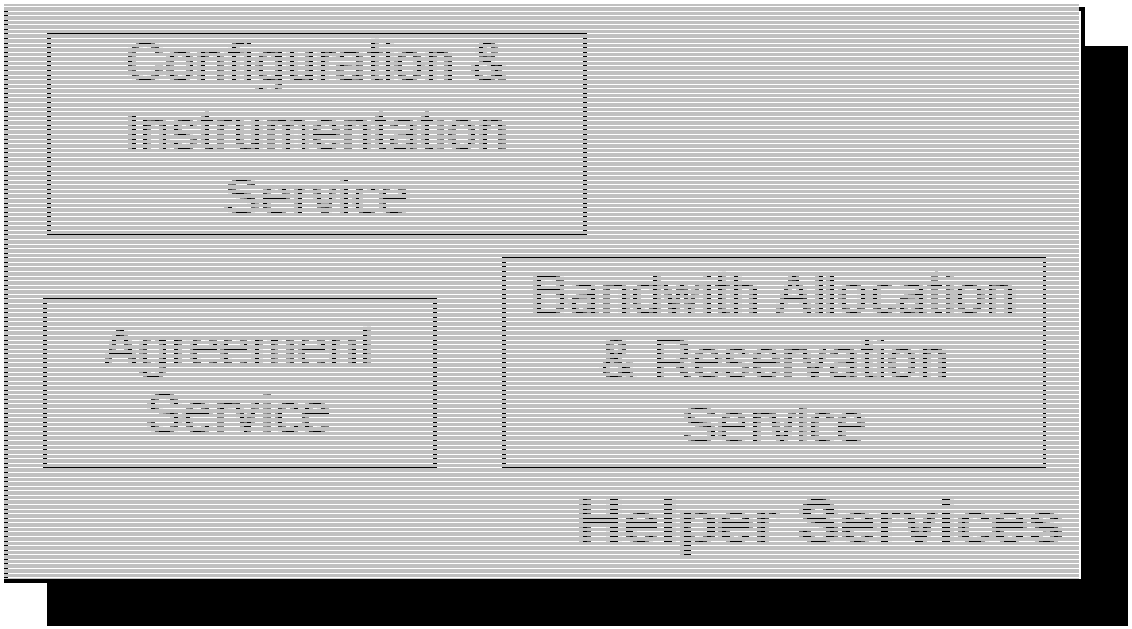
**gLite - (FireMAN)**

# gLite – Data Services
## Data Movement

- Data Scheduler (**DS**) Keeps track of user/service transfer requests
- File Transfer/Placement Service (**FTS/FPS**)
- Transfer Queue (Table)
- Transfer Agent (Network)



# gLite – Helper Services




- Configuration and Instrumentation Service – Queries service state.
- Agreement Service – Implements a communication protocol for the **Service Level Agreements**.
- Bandwidth Allocation & Reservation service (**BAR**) – Controlling, Balancing and Manage Network flows.

# References

- **gLite homepage**
  - http://www.glite.org

- **gLite Architecture Document**
  - https://edms.cern.ch/file/476451/1.0/architecture.pdf

# Questions…