



Enabling Grids for E-scienceE

Data Management System gLite – LCG – FiReMan

Salvatore Scifo

INFN Catania

BioinfoGrid

Bari, March 8th- 10th 2006

www.eu-egee.org



- **Data Management System introduction**
- **Objectives and challenges about data management**
- **GRID File Naming conventions**
- **Storage Element (types and interfaces)**
- **DMS in LCG-2 middleware**
 - **LFC - LCG File Catalog**
- **DMS in gLite 1.4 middleware**
 - **FiReMan - Combined Catalog Implementation**

- Data Management System is the subsystem of the GRID infrastructure which takes care about **file manipulation** for both, all other GRID services and user applications.
- Main capabilities provided by the DMS are locating, accessing and moving files.
- Simply, DMS provides all operation that all of us are used to perform on the data.
 - **Uploading/downloading files**
 - **Creating file/directories**
 - **Renaming file/directories**
 - **Deleting file/directories**
 - **Moving file/directories**
 - **Listing directories**
 - **Creating symbolic links**

- In a grid environment, data can be replicated to many different sites depending on where the data are needed.
- Users don't need to know where data are located, they only know and use just logical file name (LFN).
- First assumption: DMS works with **files**, this assumption is due to following reasons:
 - Semantic of file is very good understood by everyone
 - Historical reason : HEP (High Energy Physics) and Biomedical community are the two first Virtual Organization that used Grid.

- Data Management System provides two main capabilities:
 - **File Management**
 - **Metadata Management**

- What is a file?
 - file is the **smallest granularity** of data

- What are metadata?
 - Metadata are **data that describe data**

File Management

- storage (save, copy, read, list, ...)
- movement (replica, transfer,)
- security (access control,);

Metadata Management

- cataloguing
- secure database access
- database schema virtualization

DMS – Grid Data Management Challenge

NEEDS	Requirements	LCG	gLite
Heterogeneity: Data are stored on different storage systems using different access technologies.	A common interface to storage resources is required in order to hide the underlying complexity.	SRM to interface RFIO + dCAP (local ccess)	SRM to interface RFIO + dCAP (local access) gLite File I/O Server
Distribution: Data are stored in different locations; in most cases there is no shared file system or common namespace.	A uniformed view of the storage is needed. There is need to keep track about data location.	LFC	File Catalog Replica Catalog Metadata Catalog Storage Index
Data Retrieving: Applications are located in different places from where data are stored.	Data need to be accessed in a easy way independently on the data location. There is need of scheduled reliable file transfer service.	GFAL to interface RFIO + dCap (remote access) GSIFTP	gLite File I/O Server POSIX-Like Library File Transfer Service
Security: Data must be managed according to the Virtual Organization membership access control policy.	Centralized Access control Service.	Secure RFIO dCAP (gsi compliant)	File Authorization Service

- **LFN – Logical File Name**

- it is a logical (human readable) identifier for a file.
- LFN is unique but mutable
- the namespace of the LFNs is a global hierarchical namespace.
- each VO has its own namespace
- Syntax :
 - **lfn:<anything_you_want>**
 - **lfn:/home/gilda/bari/tutorial.txt**

- **GUID – Grid Unique Identifier**

- it is a logical identifier
- It is unique by construction (based on a UUID mechanism). GUID and LFN are associated with a 1:1 relationship
- Syntax :
 - **guid:<40_bytes_unique_string>**
 - **guid:38ed3f60-c402-11d7-a6b0-f53ee5a37e1d**

- **SURL – Site URL**

- it is used by the SRM interface to identify file replica stored within a site. GUID and SURLs are associated with a 1:n relationship.

- Syntax :

- **sfn://<SE_hostname><SE_Accesspoint><VO_path><filename>**
- **srm://egee016.cnaf.infn.it:8443/srm/managerv1?SFN=/dpm/cnaf.infn.it/home/gilda/bari/tutorial.txt**

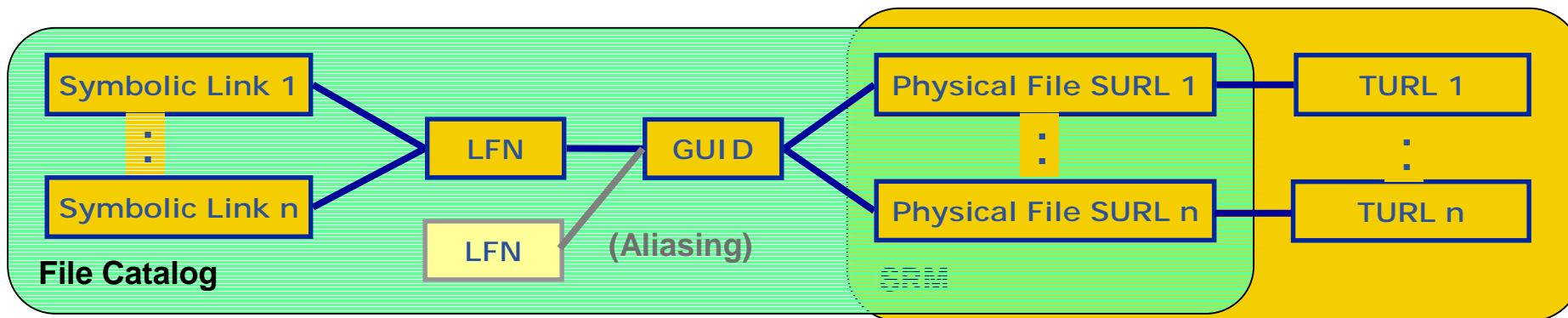
- **TURL – Transport URL**

- it is a fully qualified URL used to transfer a file by mean any standard transport protocol.

- Syntax :

- **<protocol>://<some_string>**
- **gsiftp://tbed0101.cern.ch/flatfiles/SE00/dteam/generated/2004-02-26/file3596e86f-c402-11d7-a6b0-f53ee5a37e1d**

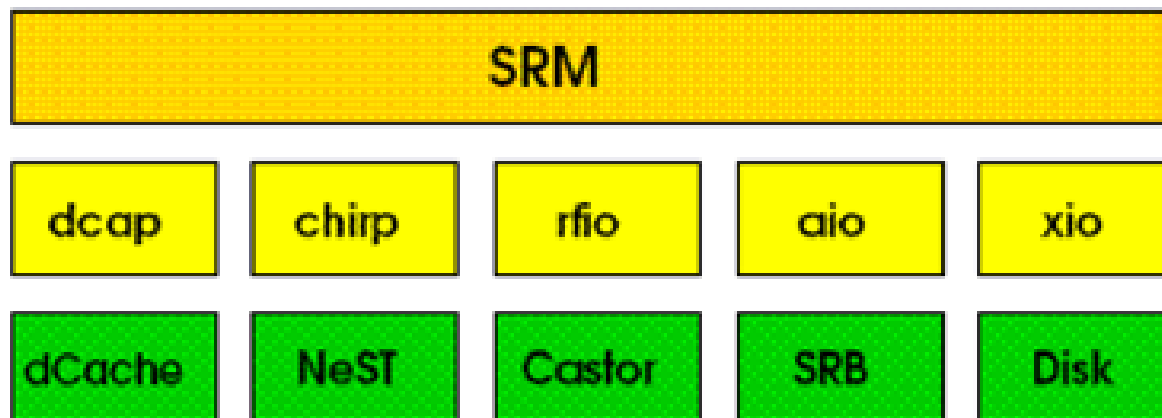
- The following table shown the relationship among file names in the GRID environment:
 - LFN e GUID 1:1 (gLite middleware)
 - LFN e Symbolic Link 1:N
 - GUID e SURL 1:N
 - SURL e TURL 1:1
- Notice** that **Alias** are allowed in LCG middleware then the **GUID** is referred by '**n**' LFNs.

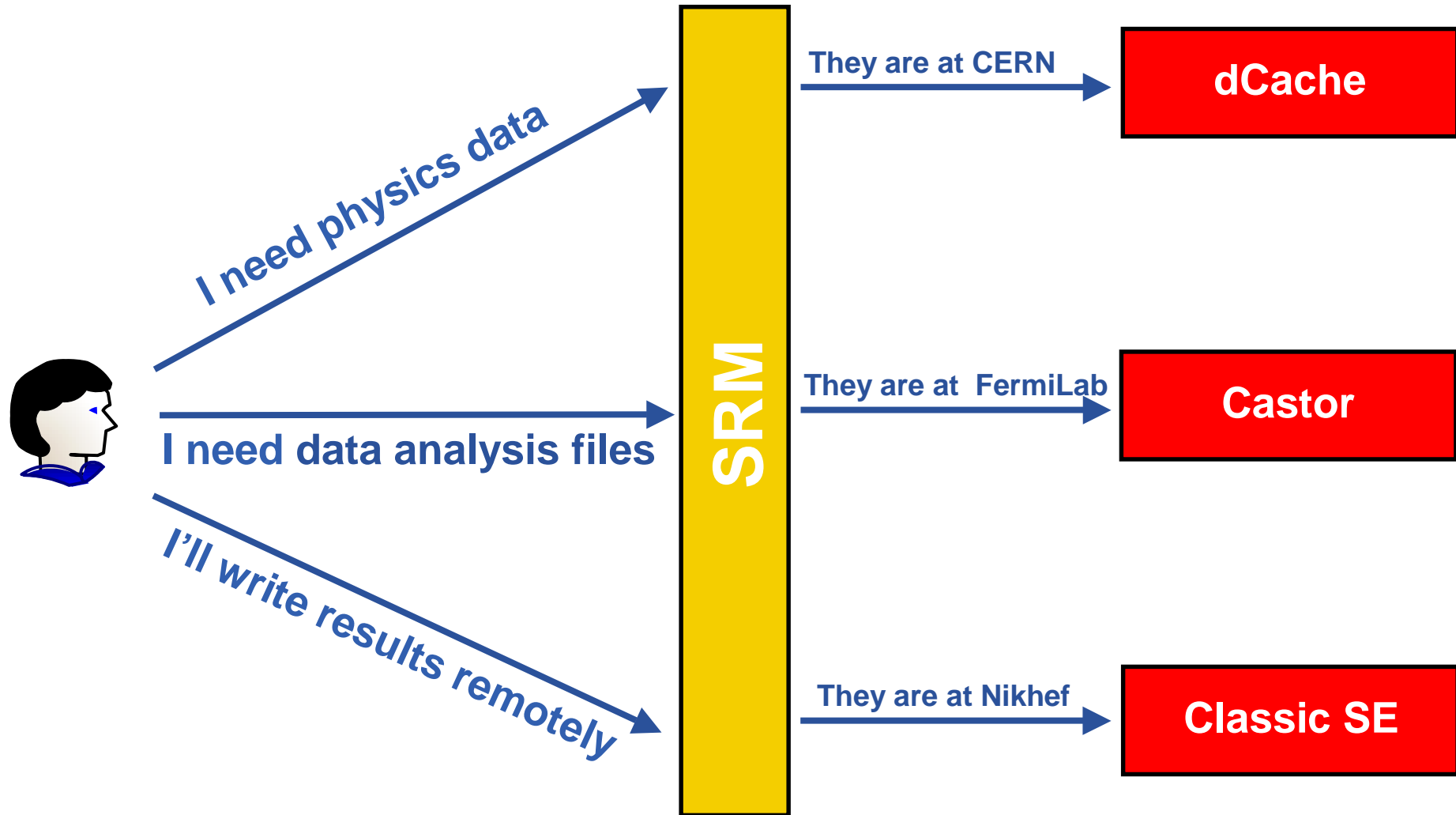


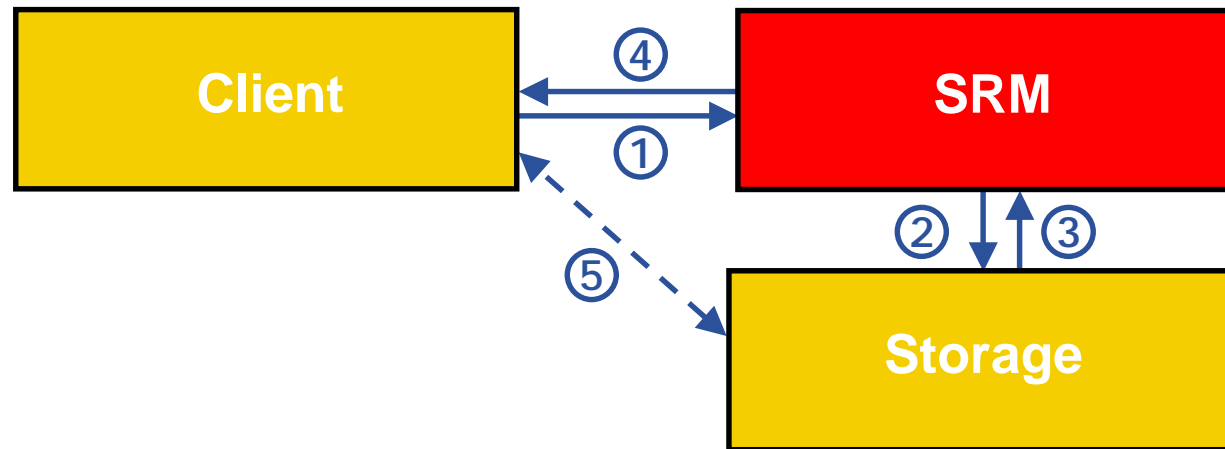
- **File Access**
 - RFIO remote file I/O (developed to access Cern Advanced STORage devices)
 - GSIDCAP – GSI dCache Access Protocol
- **GFAL – Grid File Access Library**
 - It hides interactions with RFIO and DCAP and presents a Posix-like interface for the I/O operations on remote file
 - RFIO and GSI DCAP are commonly used to access local and remote file (used to make workernode able to work on file remotely)
- **Transfer**
 - GSIFTP is the most used file transfer protocol and it is supported by any SE

Protocol	Type	Security
GSIDCAP – GSI dCache Access Protol (Access Protocol for dCache devices)	File I/O	GSI enabled
RFIO - Remote File I/O (Access Protocol for CASTOR devices)	File I/O	Secure + Insecure
GSI FTP	File Transfer	GSI enabled

- SRM has been developed to be the single interface for the management of disk and tape storage devices.
- SRM is a collection of several native *storage access protocol* depending on the storage device and it is designed to hide their complexity to the end user (human that use the client tool or any other grid services).







1. The client asks the SRM for the file providing a SURL (Site URL)
2. The SRM asks the storage system to provide the file
3. The storage system notifies the availability of the file and its location
4. The SRM returns a TURL (Transfer URL), i.e. the location from where the file can be accessed
5. The client interacts with the storage using the protocol specified in the TURL

- CLASSIC SE

- It consists of a **GSIFTP** server and an insecure **RFIO** daemon as front end for the a physical disk or disk array.
- The only security capabilities are provided by the GSIFTP server, than the RFIO daemon acts only within the LAN accessibility.
- Classic SE does not support **SRM** interface and the most common disk management features are delegated to the site administrators duties. (For example, space reservation for VOs is performed by disk partition).

- dCACHE DISK POOL MANAGER

- It consist of a **dCache** server and one or more pool nodes. The server represent the single access point to the SE and presents files in the pool disk under a single virtual file system tree.
- It uses **GSIFTP** to ensure data movements capability and **GSIDCAP** to ensure data management.
- It provides also a SRM interface. Disks can be added dynamically to the pool manager at any time.

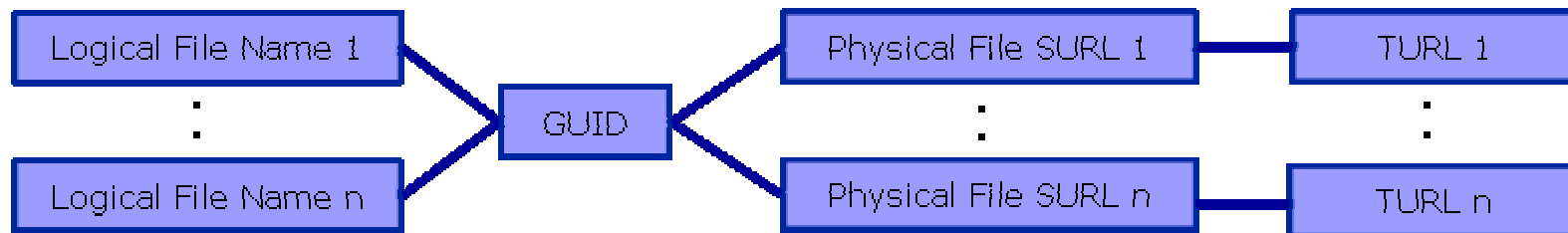
- LCG Disk Pool Manager
 - It is an lightweight alternative to the dCache DPM. It is easy to install and manage and, although not so powerful as dCache, it is a good choice for small sites.
 - Disk can be added dynamically to the pool manager at anytime. It provides an SRM interface and, in the same way of the dCache DPM, It uses GSIFTP to ensure data movements capability and gsidcap to ensure data management.
 - It is a good solution to convert quickly a classic SE into a LCG DPM with only one disk into the pool server.
 - disk quota allocation for VOs is provided.

- GILDA test bed provides different SE:
 - classic SE (grid009)
 - dCache DPM (aliserv)
 - CASTOR is coming soon

- MSS - MASS STORAGE SYSTEM
 - It consists of a Hierarchical Storage Management System for files that need to be migrated between **front end disk** and **back end tape storage devices**.
 - This mechanism that provides migration is called **STAGER** process.
 - MSS provides the **GSIFTP** for file transfer and **insecure RFIO** for CASTOR or **gsidcap** for dCache implementation for file accessing as a classic interface.
 - LCG supports only the **SRM interface** for the CASTOR MSS, therefore, it is up to the site to provide the right SRM interface for their own MSS.

- Files are stored in **Storage Element** (SE)
- **Catalog Service** is provided to manage file and replicas on different SEs.
- User can interact to Data Management services towards the Data Management *Client Tools*.
- There is no policy for **volatile** and **permanent** space; all data are considered permanent therefore it is user responsibility to manage available space into the SE.

- Users and applications need to locate files (or replicas) on the whole Grid. The **File Catalog** is the service which allows it and it maintains the mappings between LFNs, GUIDs and SURLs.
- In LCG-2, file cataloguing operations are provided by the **LFC** (LCG File Catalog); it is the best substitute of the oldest **RLS** (Replica Location Server).

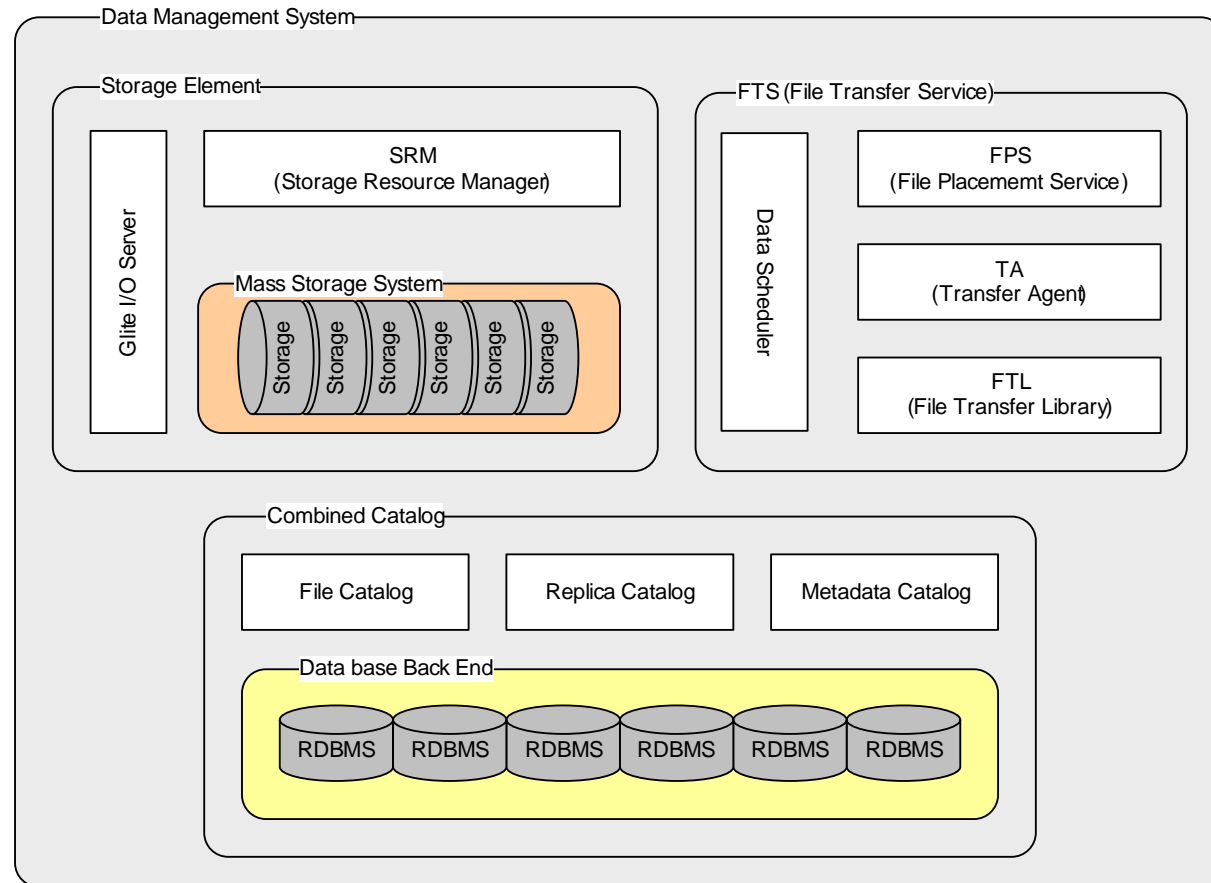


- The past
 - RLS is the first catalog used in LCG middleware
 - It works with 2 sub services: LRC (Local Replica Catalog) maps LFN onto GUID and the RMC (Replica Metadata Catalog) maps GUID into SURLs.
- The present
 - LFC is deployed as a centralized service and its endpoint is published on the Information Service in order to be found by the LCG DMS tools and/or other GRID services.
- Note1: endpoint is the URL of the service.
- Note2: if in the site are deployed both RLS and LFC, remember that they are not mirrored, therefore it is user responsibility to ensure data consistency among different catalogs entries.

- **LFC** was developed to **improve performance** and **security** and it implements new functionalities such as **transaction**, **roll-back** and **hierarchical name space** for LFNs.
- It supports **metadata management**, but although it is full compliant to handle system metadata, just a single string is provided as user metadata.
 - this has been improved in **gLite middleware catalog implementation**, (see FiReMan and AMGA for more details).
- How user can interact with LCG DMS?
 - There are available two kinds of tools: high level tools (**lcg-utils** or **lgc-* commands**) and low level tools (**edg-gridftp-*** or **globus-url-copy**).

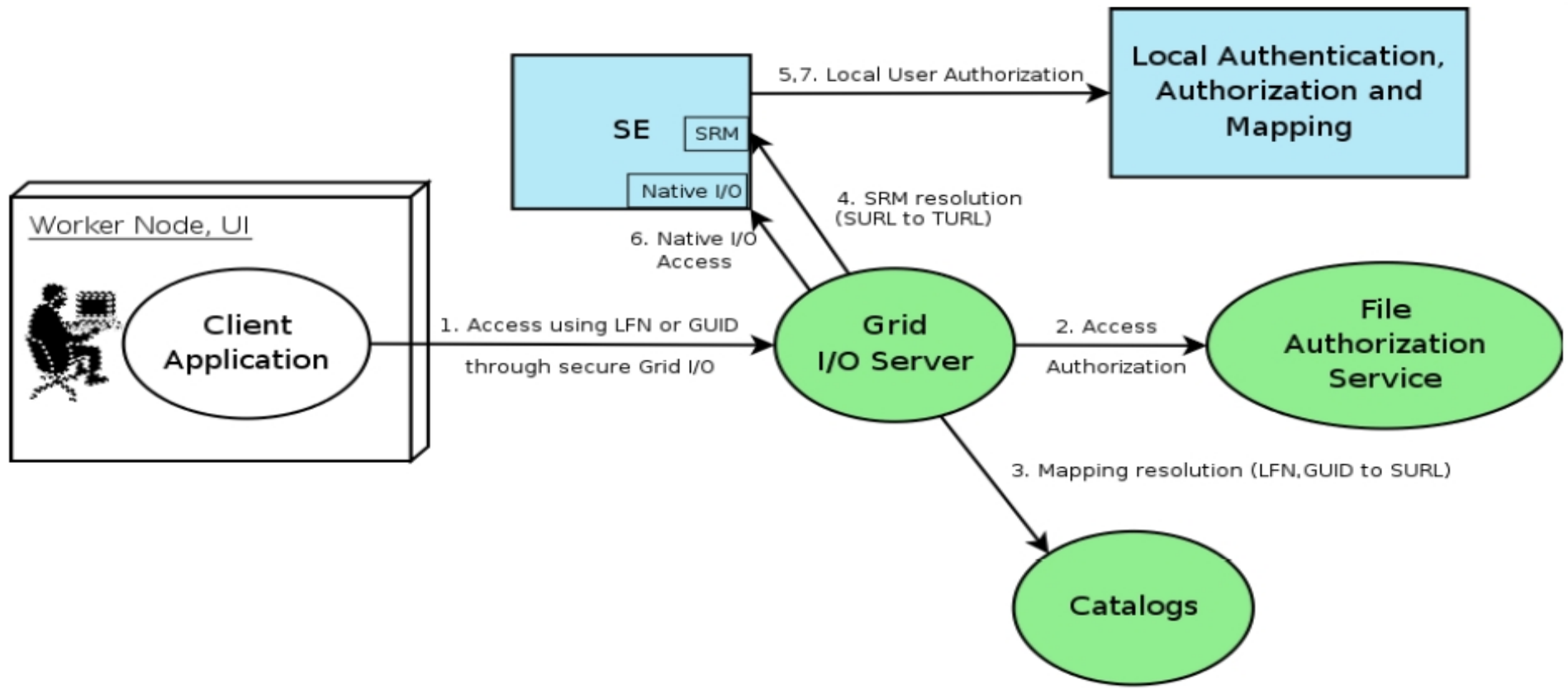
- In to the gLite middleware, Data Management System functionalities are provided by a set of **web services** according to a **service oriented architecture (SOA)**.
- They have to respect 4 keywords:
 - Interoperability
 - Portability
 - Scalability
 - Modularity
- Data Management System is composed by three main modules: **Storage Element, Catalog and File Transfer Service**.

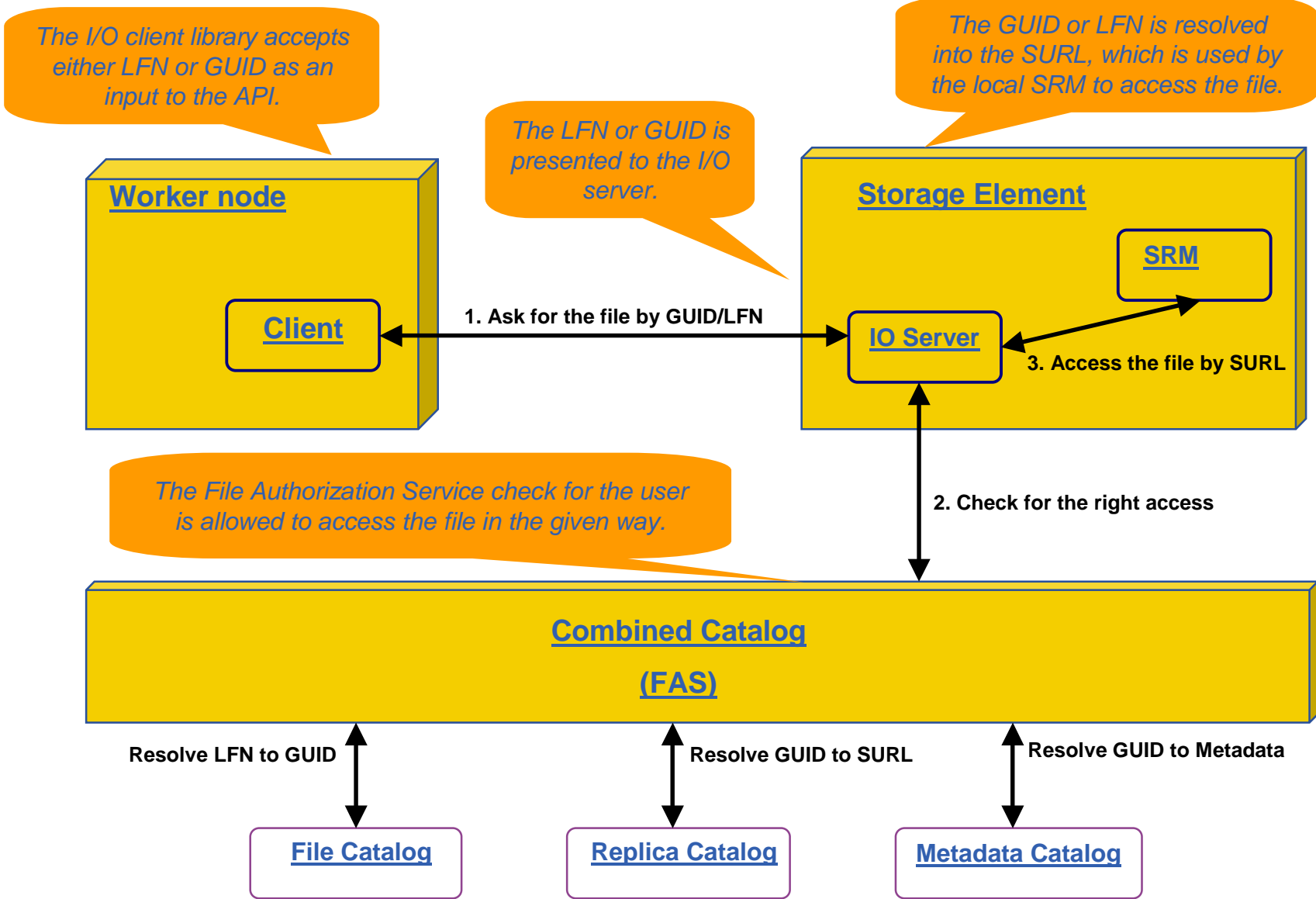
- The Storage Element takes care about data manipulation in order to make user and/or application to manage its own files.
 - **Storage Resource Manager (SRM)**
 - to dialogue to the provided Mass Storage System.
 - **gLite File I/O server:**
 - It is an interface for accessing both SE and Catalog
 - It provides a POSIX-like File I/O API to make user able to implement client applications.
- The Catalog has in charge to keep trace about file location into the distributed file system and store any kind of information about files.
- The File Transfer Service enables the GRID to move file from/to a site. It is a kind of intra-GRID moving file service.



Provided by site

Provided by VO





Co-location assumption

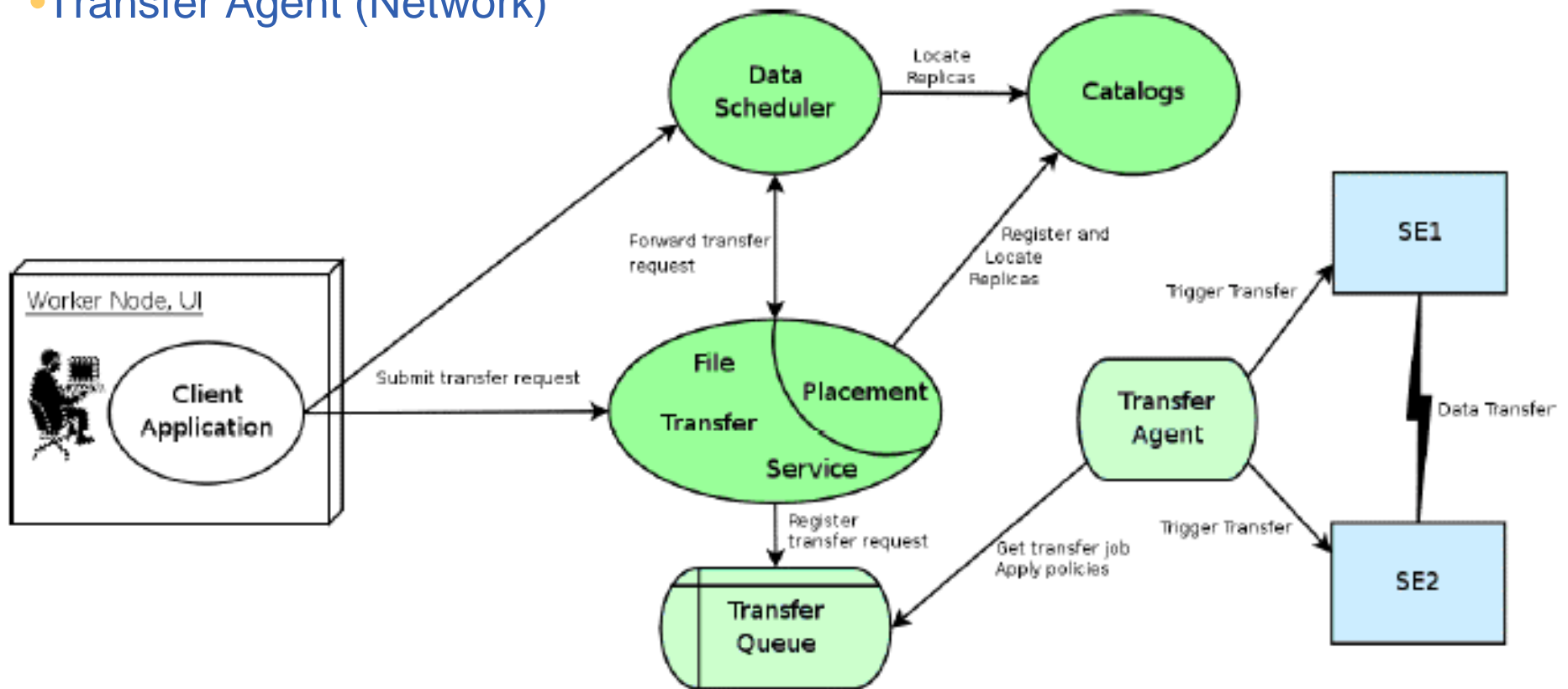
- The Grid assumes that data are co-located with the applications that use them. This co-location implies the co-scheduling of the given data in order to make sure applications that their data are available.

Paradigm

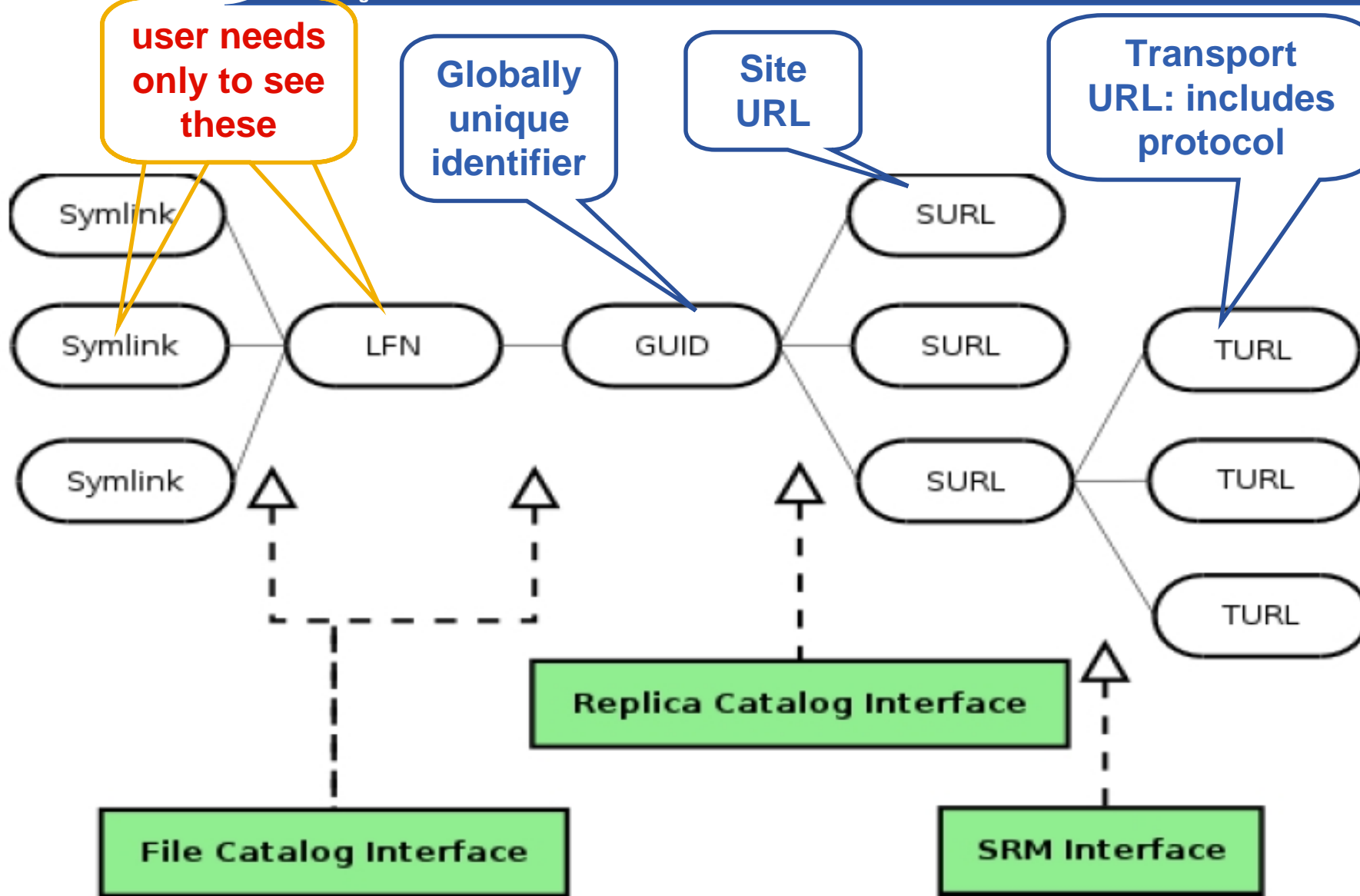
- The **destination SE** is always the local SE, the source SE may be inside or outside the site boundaries.
- Logically, for each SE there is a **File Transfer Service** to control all data incoming from other SEs.

- File transfer process is ensured by the interaction of three modules:
- **Data Scheduler**
 - data scheduler is responsible for data scheduling and it keeps track of the data transfers among multiple sites.
 - *(Note : Data Scheduler is not yet implemented).*
- **File Placement Service**
 - It polls the Data Scheduler in order to fetch all transfers whose destination is the local site for the given VO and it inserts the new requests into the internal work **Queue**.
 - FPS also ensures that File and Replica Catalogs are properly updated.
- **Transfer Agent**
 - It maintains the state for each transfer. It uses the **File Transfer Library** (the low level transfer tool) to perform the file moving.

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



- The catalog module takes care about **file organization** within the SE.
- The **ubiquity requirement** is maintained towards file replica mechanism at many GRID sites.
- The catalog provides the **authorization information** to access all the files stored into the local SE.
- The catalog manages the relationship between the **LFN**, (given by the user at creation time), and the **GUID**, (created by the GRID), and between the **GUID** and the **SURLs** (identifiers of the many replica file).



- **File Catalog**
 - It allows for operations on the logical file namespaces that it manages (ex: making directories, renaming files, creating symbolic link)
 - It keeps the LFN-GUID mappings
- **Replica Catalog**
 - It provides operations concerning the replication aspect of the grid files (ex: listing, adding and removing replicas to a file identified by its GUID)
 - It keeps the GUID-SURL mappings
- **Storage Index**
 - It allows WMS interactions (for example: file location for the Resource Broker)

- **Metadata Catalog**
 - It allows to collect and store data which describe files stored into the SE.
 - It provides operations to query and manipulate metadata.
- **Combined Catalog**
 - It keeps trace about all operations that are performed across catalogs in order to make sure that the operations occur in a synchronized manner.
- **FileReplicaManager**
 - It is an implementation of combined catalog type for gLite middleware.
 - It supports File Catalog, Replica Catalog and Metadata Catalog capabilities.
 - It also provides a simple StorageIndex interface, for job matchmaking.

- **Using Grid service**

- Grid services share the same security infrastructure. All users have to acquire an X.509 credential from their Certificate Authority (CA).
- If any user wants to access any Grid service must register itself to a Virtual Organization; this will allow him to use the Grid Service according to the VO access control policy.
- The access to the files are controlled towards **Access Control Lists** (ACLs) mechanism.
- Before accessing files user has to acquire a short-live proxy certificate provided by the **VOMS** (Virtual Organization Membership Service) that will append all users rights to the proxy certificate.

- End user is able to interact to the DMS towards the provided **GRID UI**.
 - **UI looks like a Unix shell and it allows user to navigate the remote virtual file system in the familiar manner (listing directory, creating directories and files, renaming them, etc).**
- In order to allow users to build their own application, DMS provides the **gLite-I/O API** to interface the SE and the **catalogs API** to interact to the Catalog.

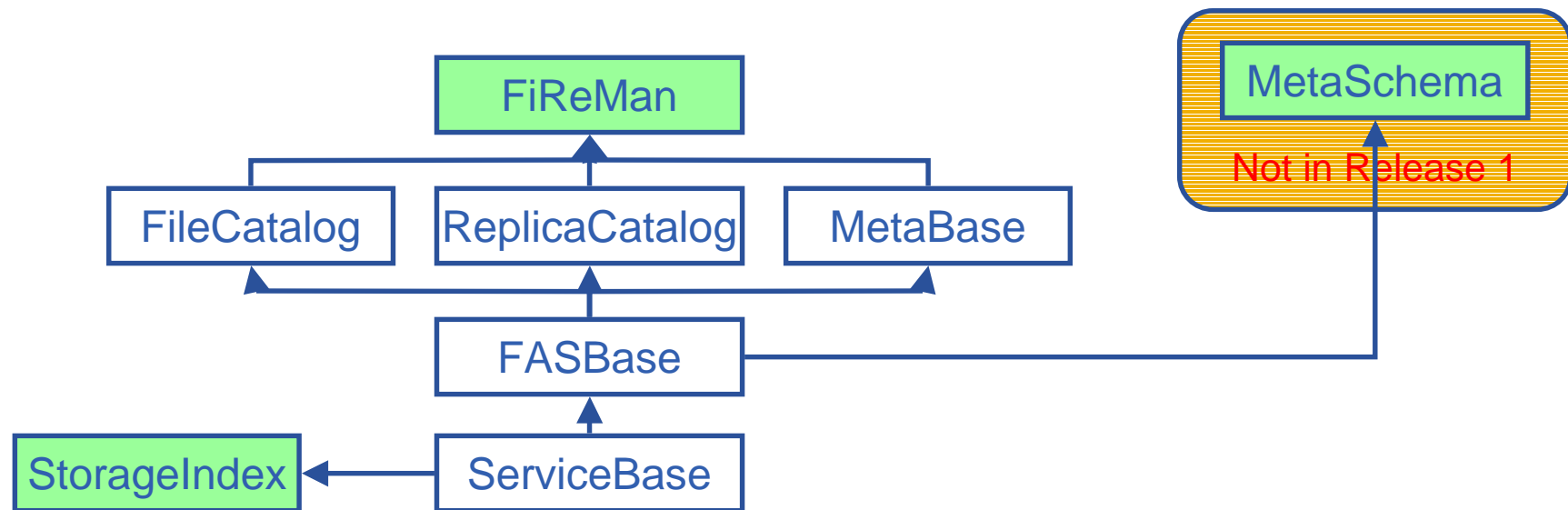
- It is an implementation of **Combined Catalog** type for gLite middleware.

- It provides the following capabilities:

– Logical File Namespace Management	FileCatalog
– Replica Locations	ReplicaCatalog
– File-based metadata	MetaBase
– Metadata Management	MetaSchema
– Authentication and Authorization information (ACLs)	FASBase
– Service Metadata	ServiceBase
– WMS interaction and global file location	StorageIndex

- StorageIndex is a simple interface, for job matchmaking function of the Workload Management System.

- Interface Structure



- **Implemented on top of Oracle and MySQL**
- **Web Service interface (WSDL)**
- **Mostly Bulk operations**
- **Stateless interaction**
- **No transactions outside Bulk**

- **DMS** : *Data Management System*
- **MMS** : *Mass Storage System*
- **FiReMan** : *File Replica Manager*
- **SRM** : *Storage Resource Manager*
- **FTS** : *File transfer service*
- **PFS** : *Placement file Service*
- **ACLs** : *Access Control List*
- **ACE** : *Access Control Element*
- **VOMS** : *Virtual Organization Membership Service*
- **LFN** : *Logical File Name*
- **GUID** : *Grid Unique Identifier*
- **URL** : *Universal Resource Locator*
- **SURL** : *Site URL*
- **TURL** : *Transport URL*
- **LHC** : *Large Hadron Collider*
- **LCG** : *LHC Computing Grid*
- **LFC** : *LCG File Catalog*
- **RFIO** : *Remote File I/O*
- **GSIFTP** : *Grid Security Infrastructure FTP*
- **CASTOR** : *CERN Advanced Storage Manager*
- **dCAP** : *dCache Access Protocol*

- **gLite homepage**
 - <http://www.glite.org>
- **DM subsystem documentation**
 - <http://egee-jra1-dm.web.cern.ch/egee-jra1-dm/doc.htm>
- **FiReMan catalog user guide**
 - <https://edms.cern.ch/file/570780/1/EGEE-TECH-570780-v1.0.pdf>
- **gLite-I/O user guide**
 - <https://edms.cern.ch/file/570771/1.1/EGEE-TECH-570771-v1.1.pdf>

