Data Management Services

Riccardo Bruno
INFN

gLite Tutorial at the First EGEE User Forum
CERN, 27-28.02.2006

Outline

- Grid Data Management Challenge
- Storage Elements, SRM
- LFC (LCG File Catalog)
- File and Replica Catalog (FiReMan)
- gLite I/O

Outline

- Grid Data Management Challenge
- Storage Elements (SRM)
- LFC (LCG File Catalog)
- File and Replica Catalog (FiReMan)
- gLite I/O

The Grid DM Challenge

NEEDS	REQUIREMENTS	SOLUTIONS
Heterogeneous : Data are stored on different storage systems using different technologies.	A common interface to storage resources is required in order to hide the underlying complexity.	**Storage Resource Manager** (SRM) interface; (gLite File I/O Server)
Distributed: Data are stored in different locations; in most cases there is no shared file system or common namespace.	Data need to be moved between different locations. There is need to keep track where data is stored.	**File Transfer Service** (FTS) – to move files among GRID sites. **Catalog** – to keep track where data are stored.
Data Retrieving: Applications are located in different places from where data are stored.	There is need of scheduled reliable file transfer service.	**File Transfer Service** •Data Scheduler •File Placement Service •Transfer Agent •File Transfer Library
Security: Data must be managed according to the VO membership access control policy.	Centralized Access control Service.	**File Authorization Service**

The Grid DM Challenge

- **DM** works with **files**, this assumption is due the following reasons:
  - **semantic** of file is very good understood by everyone
  - file is the **smallest granularity** of data**.**
- **EGEE**'s Specific Grid **Requirements**:
  - **HEP** - High Energy Physics
  - **Biomed**

Data services

- **File Access Patterns:**
  - Write once, read many
  - Rare append - only updates with one owner
  - Frequently updated at one source - replicas check/pull new version
  - (*NOT* frequent updates, many users, many sites)

- **File naming**
  - Mostly, see the “logical file name” (LFN)
  - LFN must be unique:
    - includes logical directory name
    - in a VO namespace
  - E.g. /gLite/myVOname.org/runs/12aug05/data1.res

Data Management Services

- **Storage Element** – common interface to storage
  - **S**torage **R**esource **M**anager — Castor, dCache, DPM, …
  - POSIX-I/O — gLite-I/O
  - Native Access protocols — rfio, dcap
  - Transfer protocols — gsiftp

- **Catalogs** – keep track where data are stored
  - File Catalog
  - Replica Catalog
  - File Authorization Service
  - Metadata Catalog

  → gLite File and Replica Catalog
  → **FireMan**
  → **LFC**
  → **AMGA Metadata Catalogue**

- **File Transfer** – schedules reliable file transfer
  - Data Scheduler — *(only designs exist so far)*
  - File Transfer Service (manages physical transfers)
  - File Placement Service (FTS and catalog interaction in a transactional way)

  → **gLite FTS – lcg-rep**

Outline

- Grid Data Management Challenge
- Storage Elements (SRM)
- LFC (LCG File Catalog)
- File and Replica Catalog (FiReMan)
- gLite I/O

SRM in an example

He is running a job which needs:

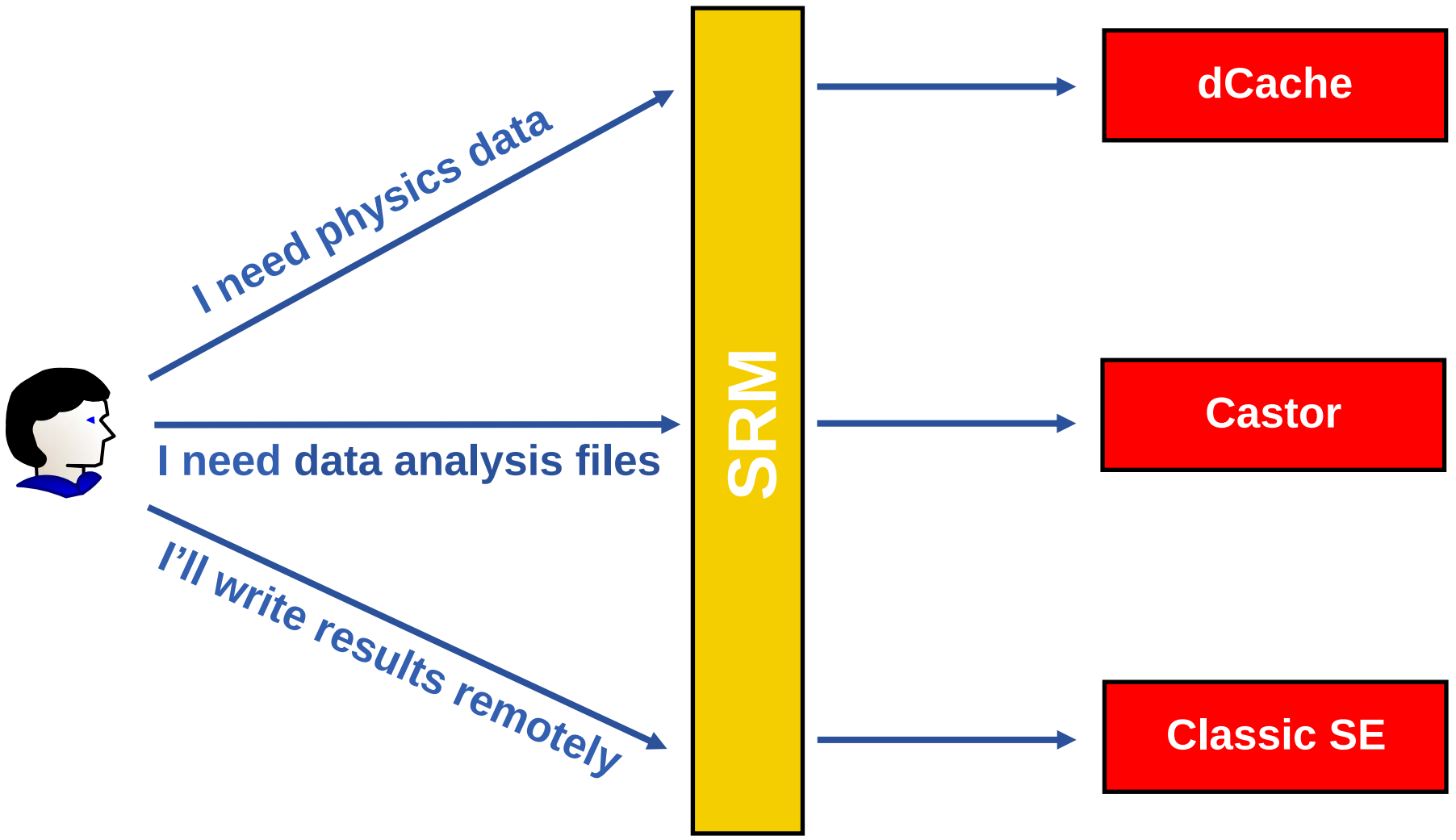
- Data for physics event reconstruction
- Some data analysis files
- He will write files remotely too

They are at CERN in dCache

They are at Fermilab in a disk array

They are at Nikhef in a classic SE

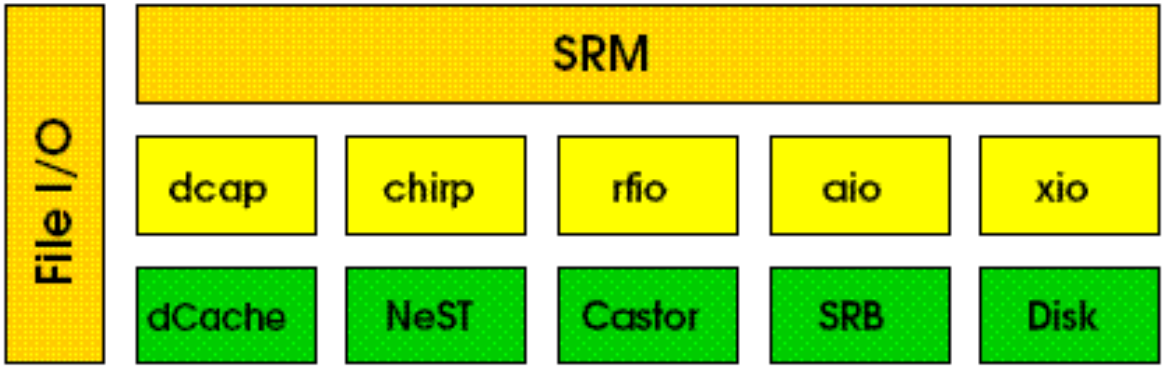
SRM in an example



Storage Resource Management

- Data are stored on **Disk Pool Servers** or **Mass Storage Systems**
- **S**torage **R**esource **M**anagement needs to take into account
  - Transparent access to files (migration to/from disk pool)
  - Space reservation
  - File status notification
  - Life time management
- **SRM** (Storage Resource Manager) takes care of all these details
  - **SRM** is a **Grid Service** that takes care of **local storage** interaction and provides a Grid **interface** to outside world.
- Interaction with the **SRM** is hidden by higher level services

SE – Storage Resource Manager



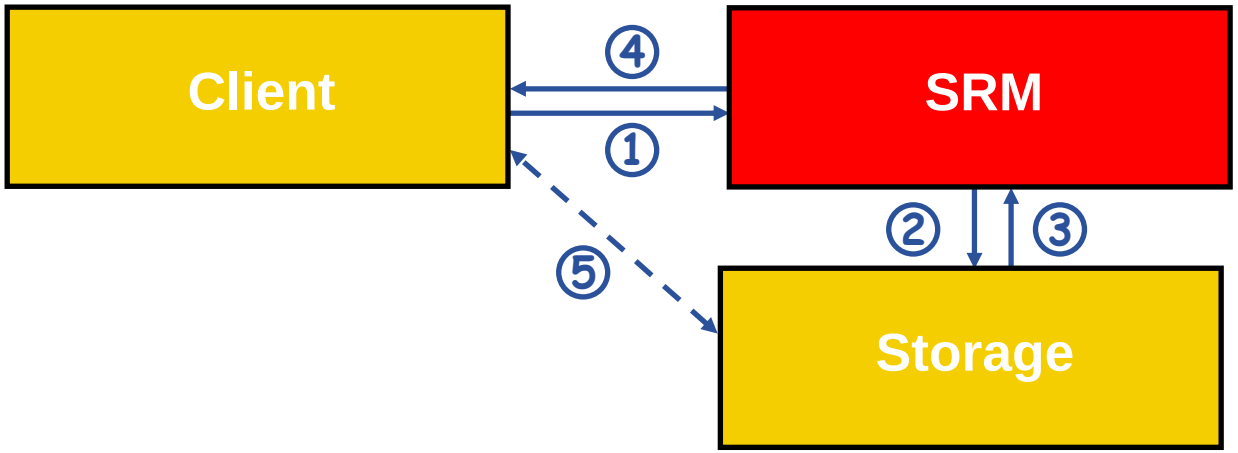
Main supported MSS

- dCache
- NeST
- CASTOR
- SRB
- Disk

Main supported protocols

- dCap
- Chirp
- Rfio
- Aio
- xio

SRM Interactions



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

Grid Storage Element

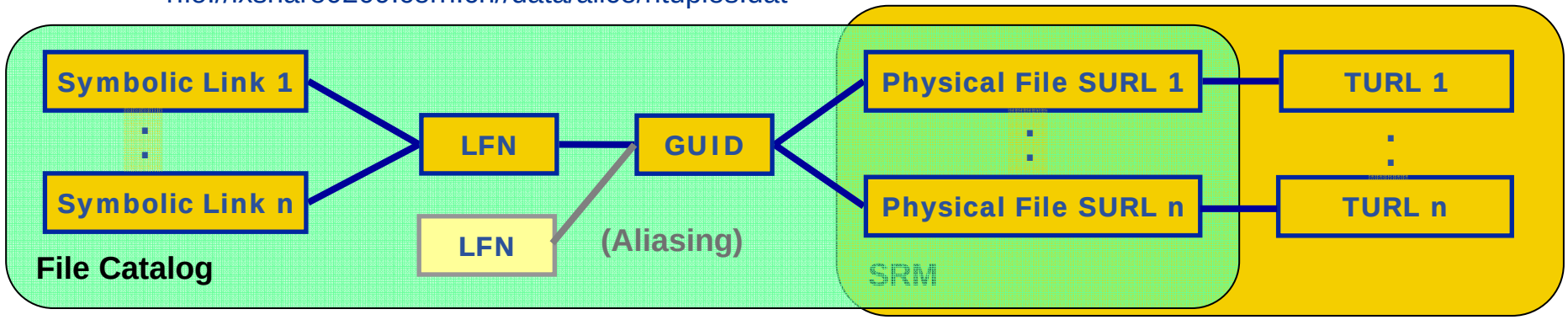
- **Provide an SRM interface**
  - Specific Storage Solution: HPSS, CASTOR, DiskeXtender (UNITREE), DPM, dCache
- **Support basic file transfer protocols**
  - **GridFTP** mandatory
  - Others if available: https, ftp, etc.
- **Support a native I/O access protocol**
  - **POSIX** (like) I/O client library for direct access of data (rfio, dcap, gsidcap)
- **Other auxiliary services**
  - Accounting

Outline

- Grid Data Management Challenge
- Storage Elements (SRM)
- LFC (LCG File Catalog)
- File and Replica Catalog (FiReMan)
- gLite I/O

Files & replicas: Naming Conventions

- **Symbolic Link** in logical filename space
- **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.
    "rfio://lxshare0209.cern.ch//data/alice/ntuples.dat"



LFC - Description

The LCG File Catalog fixes the performance and scalability problems of EDG (European Data Grid) file catalogs.

Provides

- **Bulk operations.**
- **Cursors for large queries.**
- **Timeouts and retries for client operations.**

Added features :

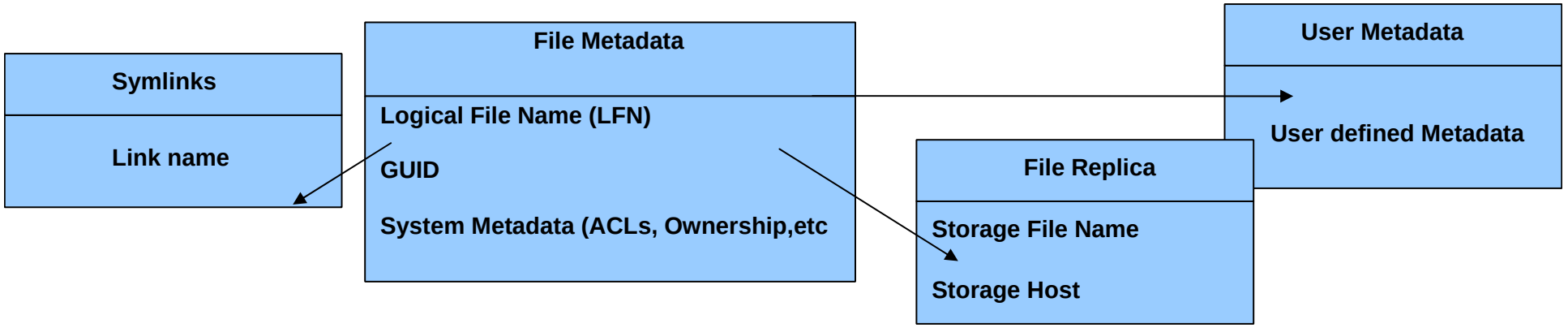
- **User exposed transaction API.**
- **Hierarchical namespace and namespace operations.**
- **Integrated GSI Authentication and Authorization.**
- **Access Control Lists (Unix Permissions and POSIX ACLs).**
- **Checksums.**

Supported database backends: Oracle and MySQL

GFAL integration and support to lcg-* done by Grid Deployment group

LFC - Architecture

- LFC stores both **logical** and **physical** mappings for the file in the same database → Speed up of operations
- Treats all entities as files in a **UNIX-like** filesystem.
- File **API** also similar to UNIX (create(), mkdir(), chown()….)
- Hierarchical namespace of **LFNs** mapped to the **GUIDs**
- **GUIDs** mapped to the physical locations of file **replicas** in the storage
- **System attributes** of files (creation time, file size and checksum…) stored as LFN attributes
- One field for **user-defined metadata**
- Multiple LFNs per GUID allowed as **symbolic links** to the primary LFN.



LFC - Requirements

If a site acts as a central catalog for several VOs, it can either have:

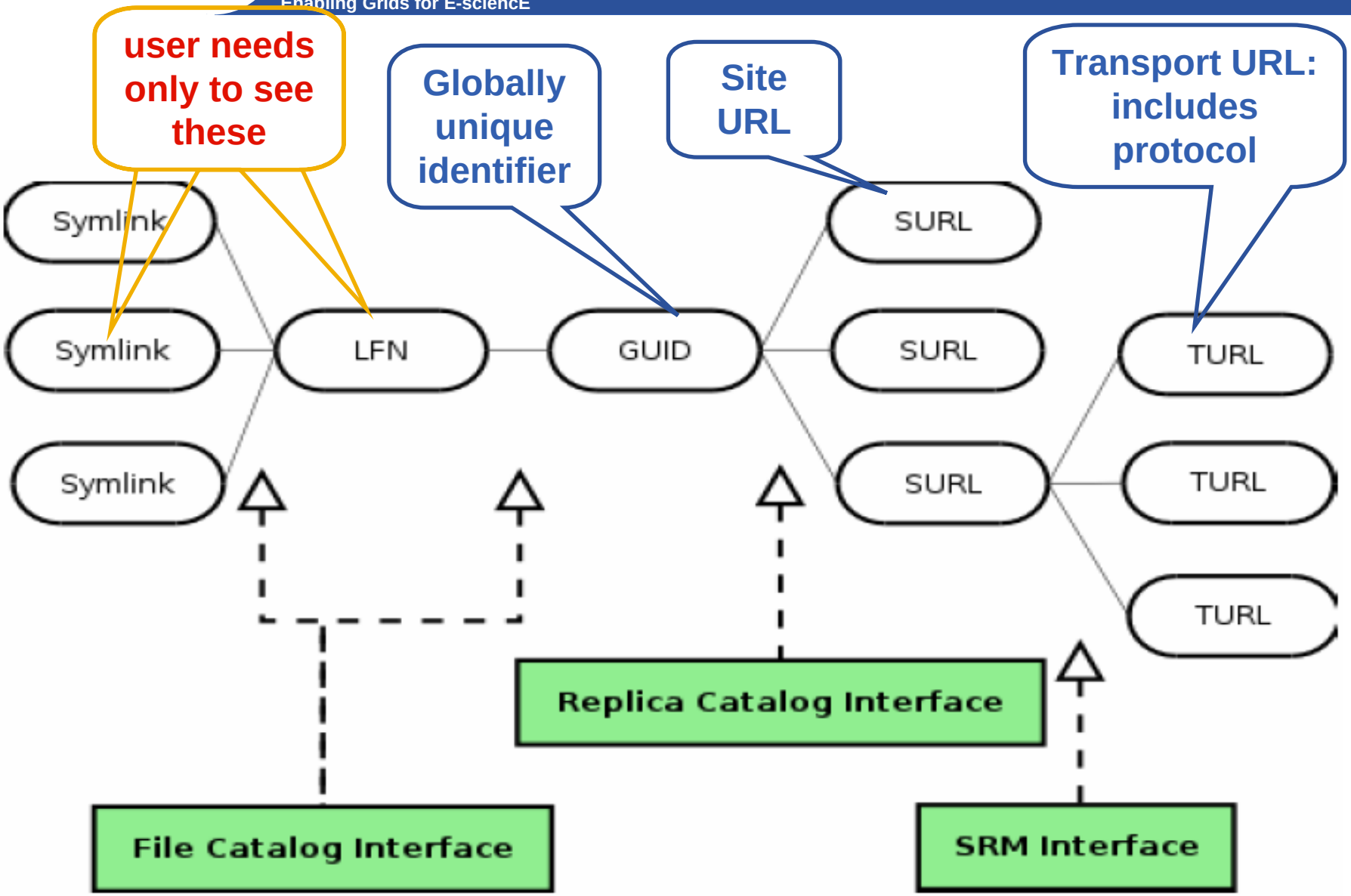
- **One LFC server, with one DB account containing the entries of all the supported VOs. You should then create one directory per VO.**
- **Several LFC servers, having each a DB account containing the entries for a given VO.**

Both scenarios have consequences on database backup policies.

Outline

- Grid Data Management Challenge
- Storage Elements (SRM)
- LFC (LCG File Catalog)
- File and Replica Catalog (FiReMan)
- gLite I/O

File names and identifiers in gLite

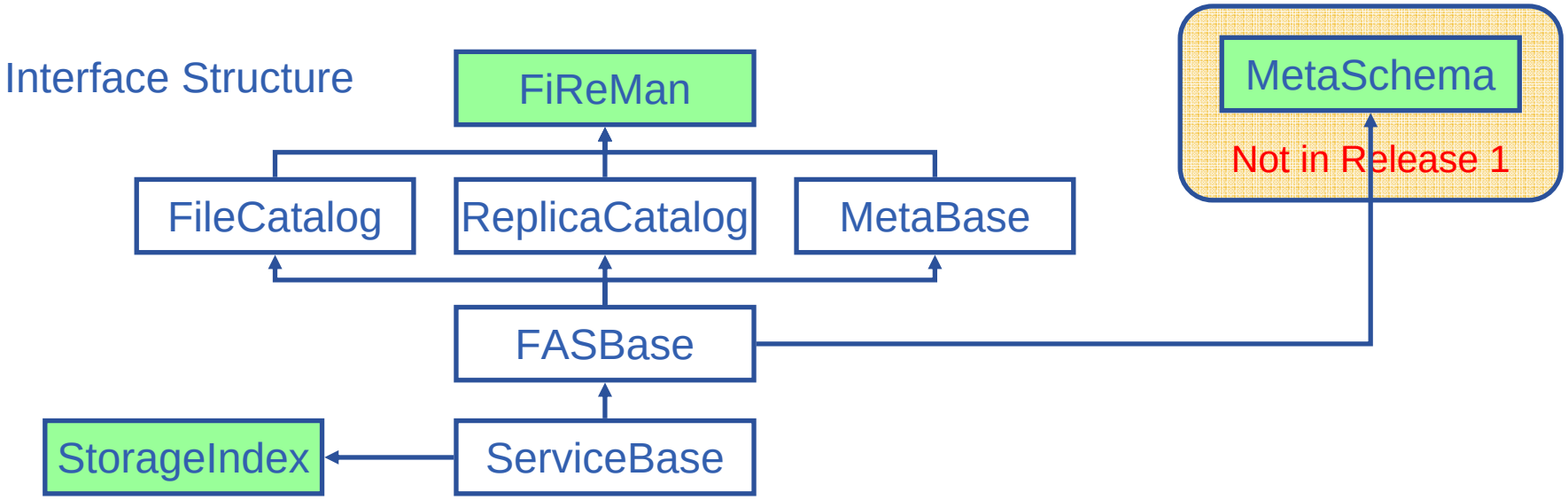


FiReMan: File and Replica Manager

- **File Catalog**
  - Allows operations on the logical file namespaces that it manages (e.g.: making directories, renaming files, creating symbolic links)
  - Manages LFNs, keeping internally **LFN-GUID** mappings
- **Replica Catalog**
  - Exposes operations concerning the replication aspect of the grid files (e.g.: listing, adding and removing replicas of a file identified by its GUID)
  - Gives access to the **GUID-SURL** mappings
- **File Authorization Service (FAS)**
  - Request authorization - based on the DN and the Groups from the user's delegated credentials
- **StorageIndex**
  - Allows WMS interactions (file location for the RB)
- **Fireman = File and Replica Manager**
  - Provides all the previous services

FiReMan Catalog Interface

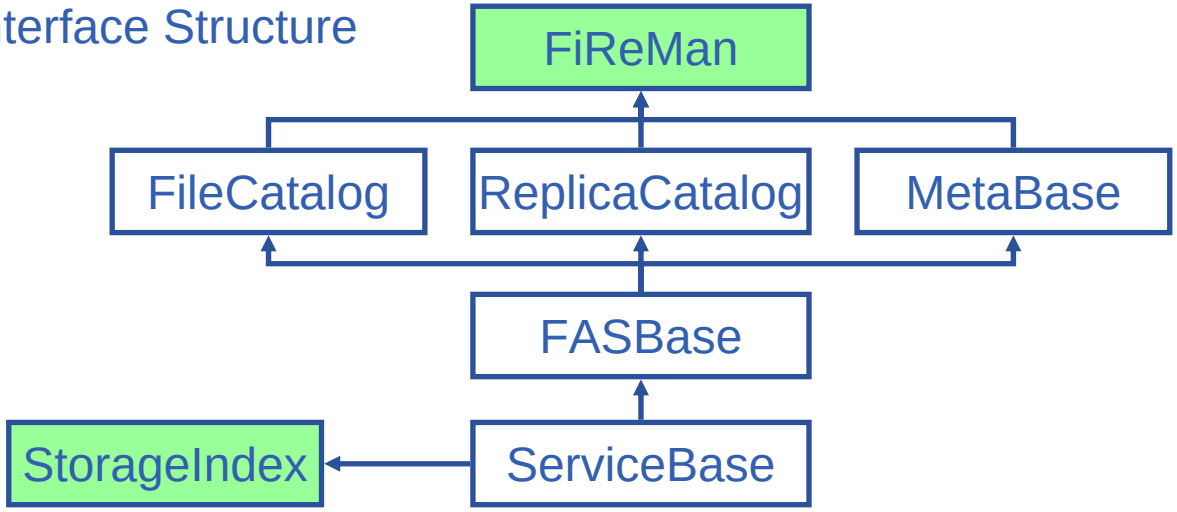
- Logical File Namespace management — **FileCatalog**
- Replica locations — **ReplicaCatalog**
- File-based metadata — **MetaBase**
- Metadata Management — **MetaSchema**
- Authentication and Authorization information (ACLs) — **FASBase**
- Service Metadata — **ServiceBase**
- WMS interaction and global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



gLite FiReMan Catalog details

- **Web Service interface (WSDL)**
- **Bulk operations**
- **Stateless interaction**
- **No transactions outside Bulk**

Interface Structure



- StorageIndex: file location for broker
- FAS: File Access Service (ACLs)
- File Catalog: directory structure in LFN namespace
- Replica Catalog: location of replicas
- Meta: additional (user defined metadata)

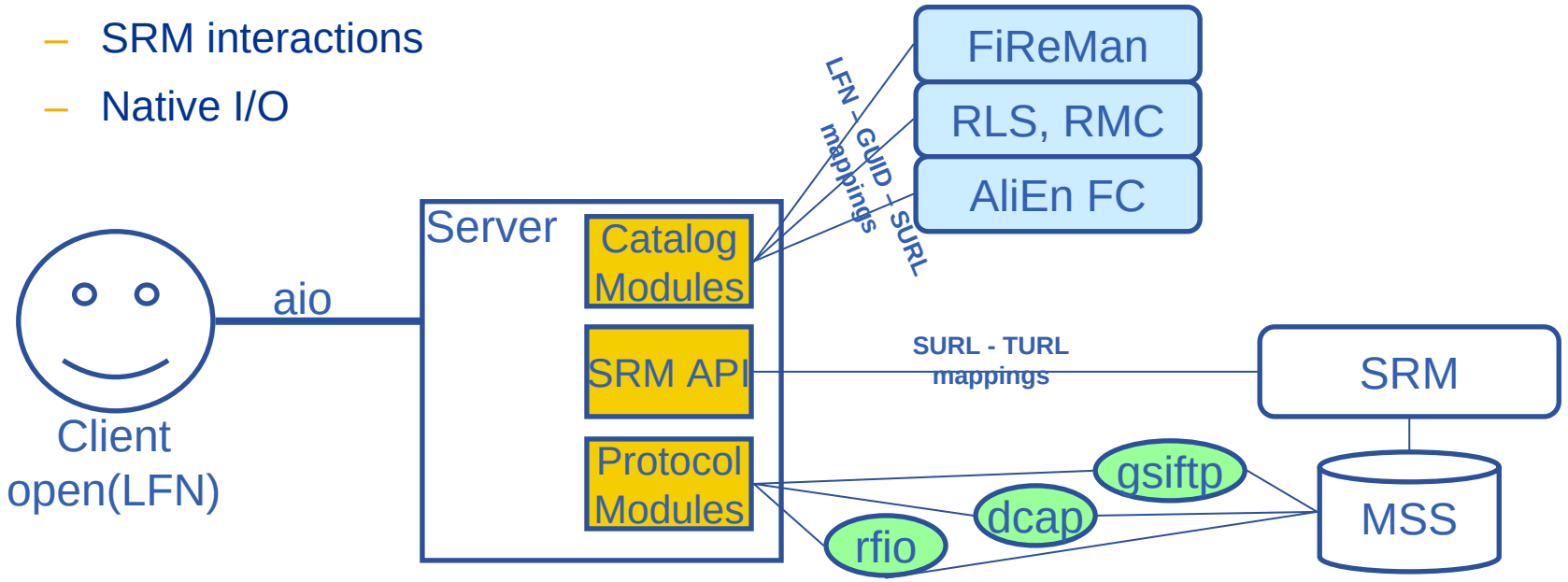
Implemented on top of Oracle and MySQL

Outline

- Grid Data Management Challenge
- Storage Elements (SRM)
- LFC (LCG File Catalog)
- File and Replica Catalog (FiReMan)
- gLite I/O

gLite-I/O

- **Client only sees: a simple API library and a Command Line Interface.**
  - GUID or LFN can be used, i.e. open("/grid/myFile")
- **GSI Delegation to gLite I/O Server**
- **Server performs all operations on User's behalf**
  - Resolve LFN/GUID into SURL and TURL
- **Operations are pluggable:**
  - Catalog interactions
  - SRM interactions
  - Native I/O



gLite I/O commands and API

Summary of the gLite I/O command line tools

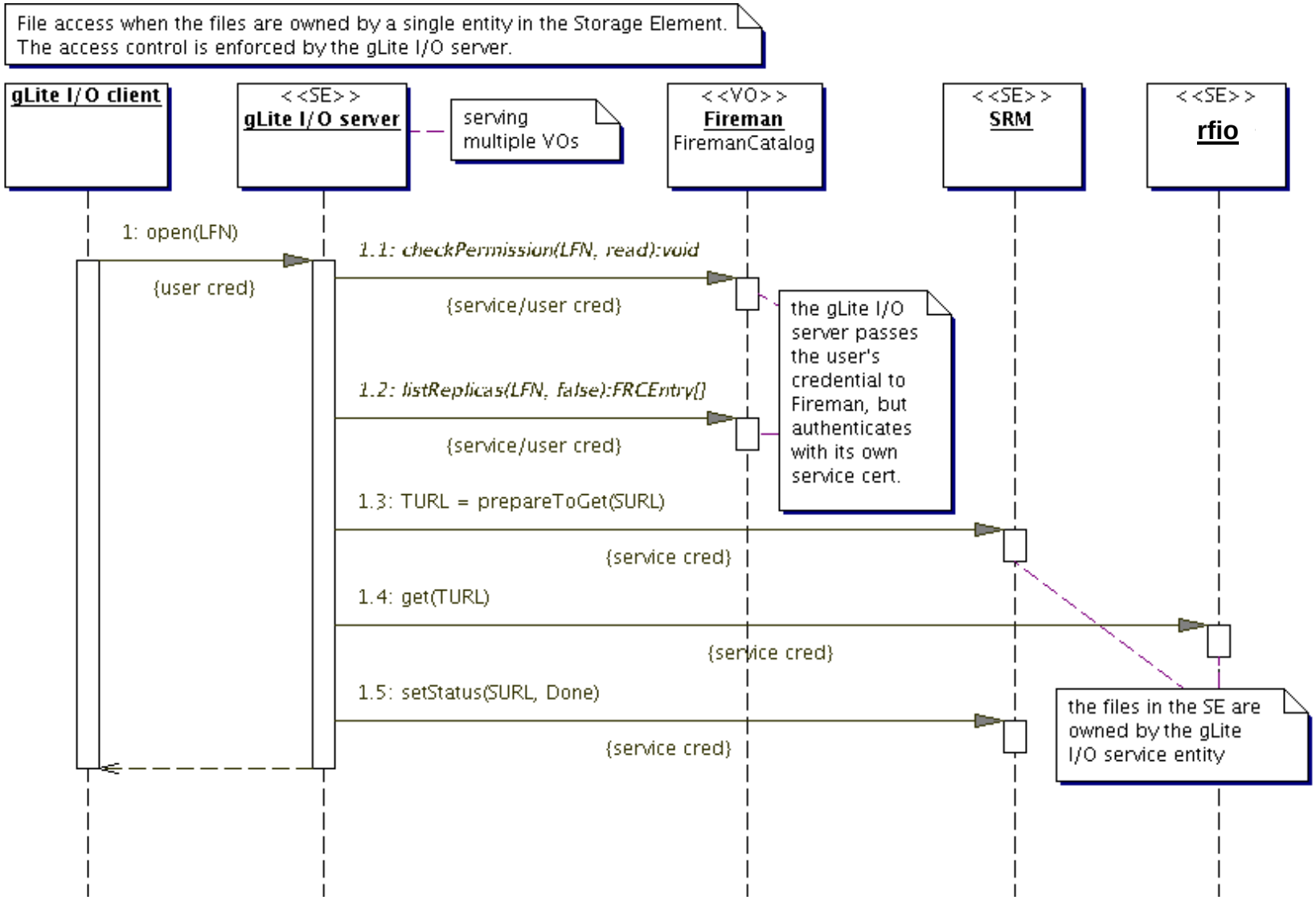
glite-get	Retrieve a file from the Grid using LFN or GUID
glite-put	Put a local file into the Grid, assigning LFN
glite-rm	Remove a file (replica!) from the Grid using LFN or GUID

Summary of the gLite I/O API calls (C only)

glite_open
glite_read
glite_write
glite_creat
glite_fstat
glite_lseek
glite_close
glite_unlink
glite_error
glite_strerror

glite_posix_open
glite_posix_read
glite_posix_write
glite_posix_creat
glite_posix_fstat
glite_posix_lseek
glite_posix_close
glite_posix_unlink
glite_filehandle

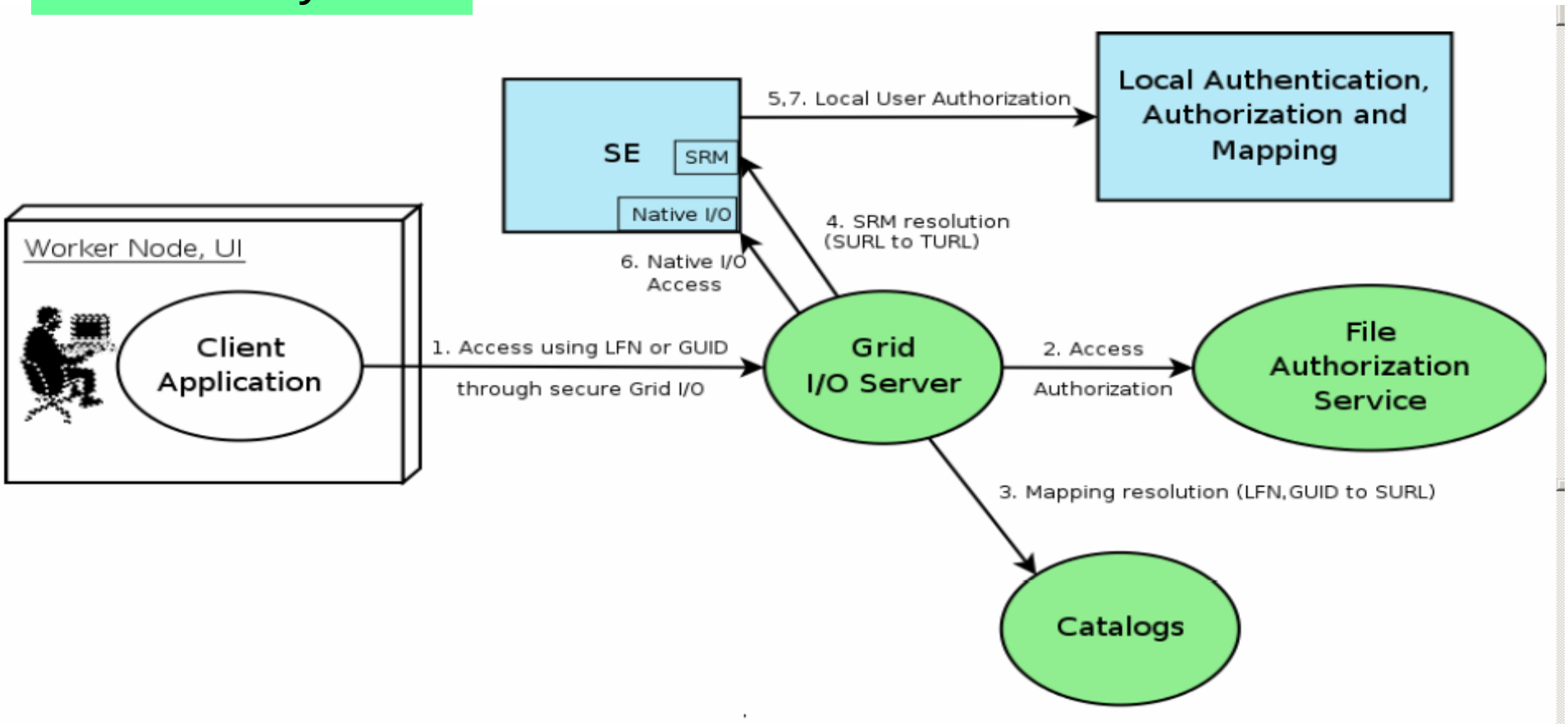
File Open


File access when the files are owned by a single entity in the Storage Element.
The access control is enforced by the gLite I/O server.
gLite I/O client
<<SE>>
gLite I/O server
serving multiple VOs
<<VO>>
Fireman
FiremanCatalog
<<SE>>
SRM
<<SE>>
rfio
1: open(LFN)
{user cred}
1.1: checkPermission(LFN, read):void
{service/user cred}
the gLite I/O server passes the user's credential to Fireman, but authenticates with its own service cert.
1.2: listReplicas(LFN, false):FRCEntry[]
{service/user cred}
1.3: TURL = prepareToGet(SURL)
{service cred}
1.4: get(TURL)
{service cred}
1.5: setStatus(SURL, Done)
{service cred}
the files in the SE are owned by the gLite I/O service entity

I/O server interactions

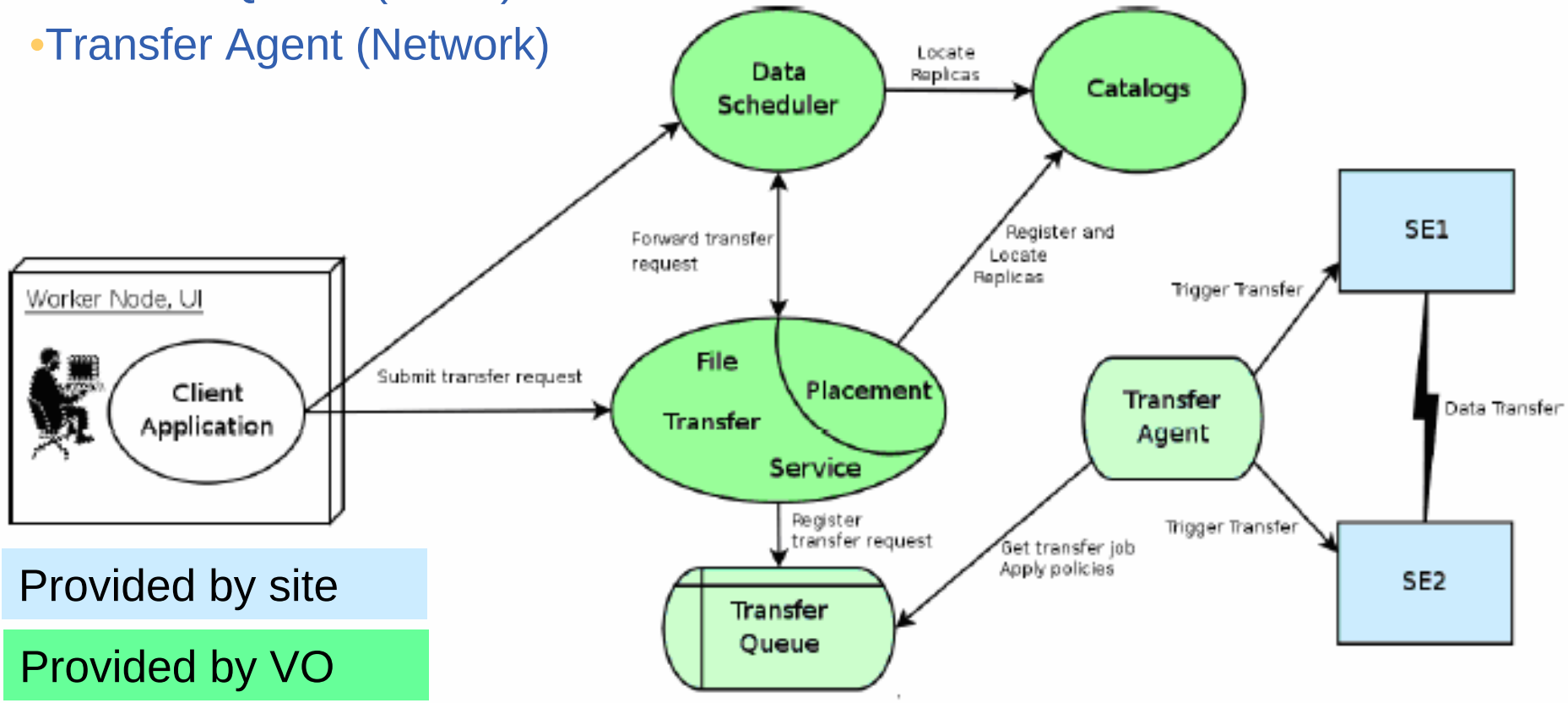
Provided by site

Provided by VO



Data Movement

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



Provided by site

Provided by VO

References

- **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

Questions…