gLite Data Management System

Donvito Giacinto
INFN-BARI

NETTAB Cagliari, 11.07.2006

Thank's to Giorgio Emidio

Outline

- Grid Data Management Challenge
- Storage Elements, SRM and gfal I/O
- File and Replica Catalogs (LFC)
- Data Movement (File Transfer Components)

The Grid DM Challenge

- **Heterogeneity**
  - Data are stored on different storage systems using different access technologies
  - Need common interface to storage resources
    - Storage Resource Manager (SRM)
- **Distribution**
  - 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
  - Need to keep track where data is stored
    - File and Replica Catalogs
  - Need scheduled, reliable file transfer
    - File transfer and placement services

Data Management Services Overview

- **Storage Element** – save date and provide a common interface
  - Storage Resource Manager (SRM) — Castor, dCache, DPM, …
  - Native Access protocols — rfio, dcap, nfs, …
  - Transfer protocols — gsiftp, ftp, …
- **File access library** – provides a POSIX-I/O interface to user — gfal-I/O
- **Catalogs** – keep track where data are stored
  - File Catalog
  - Replica Catalog — **LCG File Catalog (LFC)**
  - File Authorization Service
  - Metadata Catalog — **AMGA Metadata Catalogue**
- **File Transfer** – schedules reliable file transfer
  - Data Scheduler — (only designs exist so far)
  - File Transfer Service — gLite FTS
    (manages physical transfers)
  - File Placement Service — gLite FPS
    (FTS and catalog interaction in a transactional way)

Data services in gLite

- **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. /grid/myVOname.org/runs/12aug05/data1.res
- **3 service types for data**
  - Storage
  - Catalogs
  - Movement

SRM in an example

She is running a job which needs:
Data for physics event reconstruction
Simulated Data
Some data analysis files
She 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

dCache
Own system, own protocols and parameters

classic SE
Independent system from dCache or Castor

Castor
No connection with dCache or classic SE

SRM

I talk to them on your behalf
I will even allocate space for your files
And I will use transfer protocols to send your files there

Storage Resource Management

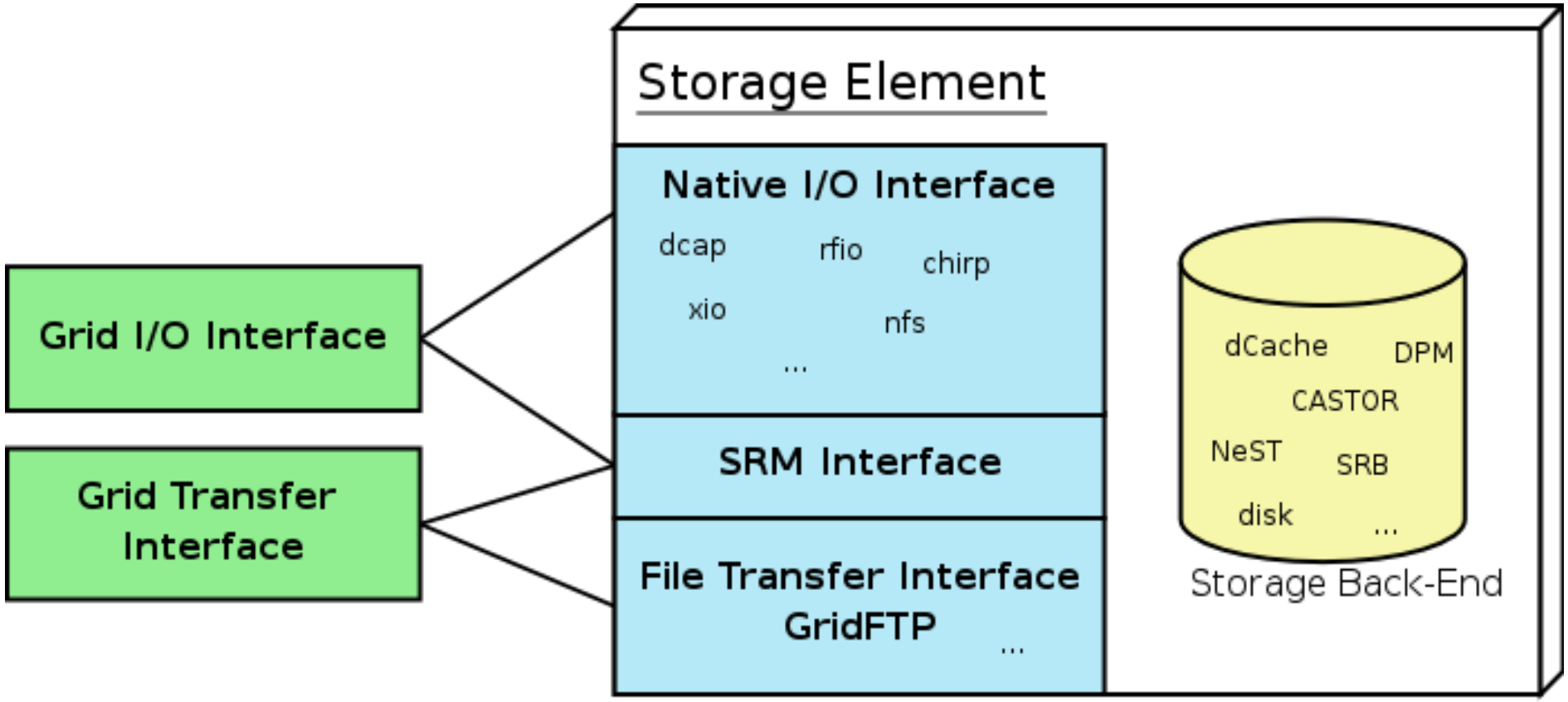
- Data are stored on **disk pool servers** or **Mass Storage Systems**
- storage resource management needs to take into account
  - Transparent access to files (migration to/from disk pool)
  - File pinning
  - Space reservation
  - File status notification
  - Life time management
- **SRM (Storage Resource 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

- In gLite, Interactions with the SRM is hidden by higher level services and API (gfal I/O)

Grid Storage Requirements

- **Manage local storage and interface to Mass Storage Systems like**
  - HPSS, CASTOR, DiskeXtender (UNITREE), …
- **Provide an SRM interface**
- **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

gLite Storage Element



File and Replica Catalogs

- **LCG-2 File and Replica Catalogs (LFC)**

LCG-2 File & Replica Catalog (I)

- 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.

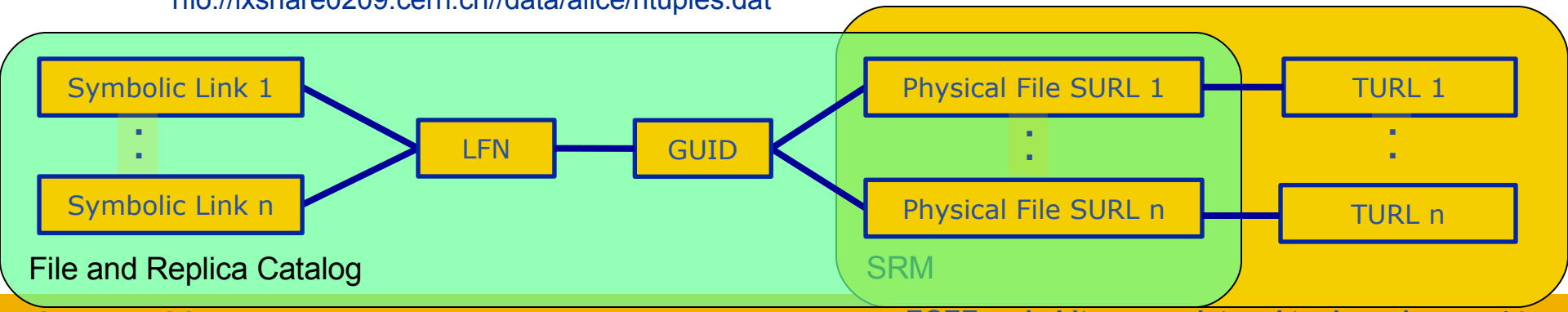
- In LCG-2, file cataloguing operations are provided by the LFC (LCG File Catalog); it is the best substitute of the oldest RLS (Replica Location Server).

LCG-2 File & Replica Catalog (II)

- 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.

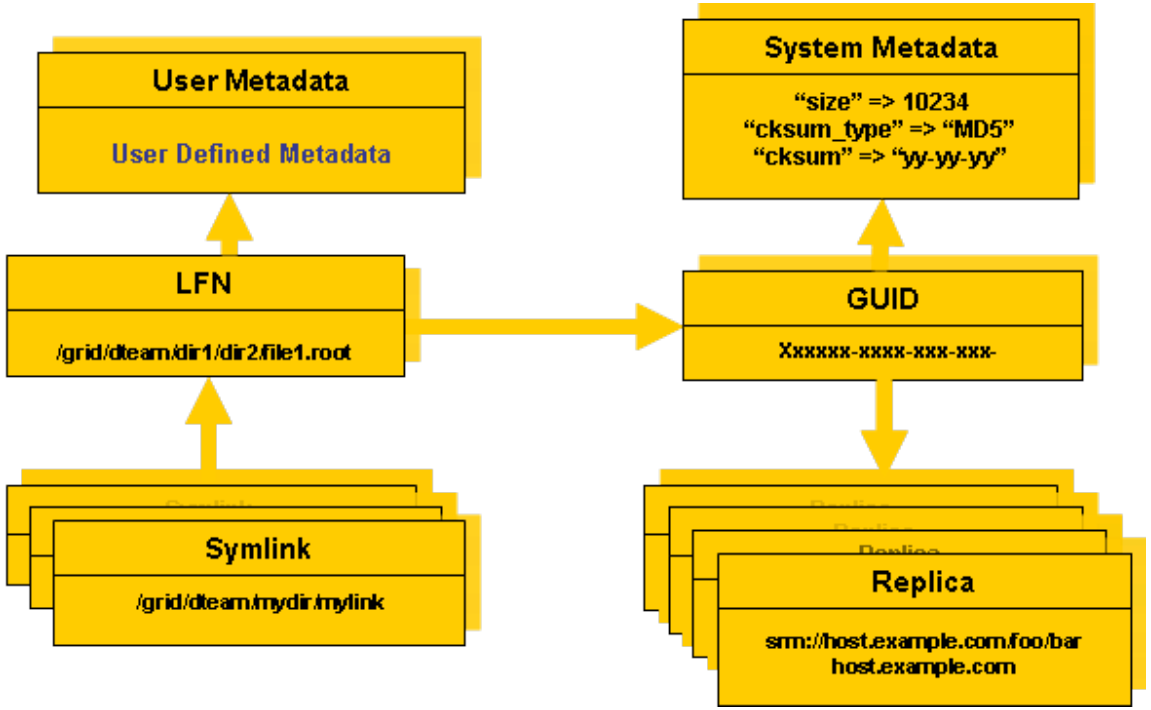
Files & replicas: Name Conventions (LFC)

- 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"



The LFC

- **It keeps track of the location of copies (replicas) of Grid files**
- **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



LFC Features

- Cursors for large queries
- Timeouts and retries from the client
- User exposed transactional API (+ auto rollback on failure)
- Hierarchical namespace and namespace operations (for LFNs)
- Integrated GSI Authentication + Authorization
- Access Control Lists (Unix Permissions and POSIX ACLs)
- Checksums
- Integration with VOMS

Data Management CLIs & APIs

- **lcg_utils: lcg-* commands + lcg_* API calls**
  - Provide (all) the functionality needed by the LCG user
  - Transparent interaction with file catalogs and storage interfaces when needed
  - Abstraction from technology of specific implementations
- **Grid File Access Library (GFAL): API**
  - Adds file I/O and explicit catalog interaction functionality
  - Still provides the abstraction and transparency of lcg_utils
- **edg-gridftp tools: CLI**
  - Complete the lcg_utils with low level GridFTP operations
  - Functionality available as API in GFAL
  - May be generalized as lcg-* commands

lcg-utils commands

Replica Management

lcg-cp	Copies a grid file to a local destination
lcg-cr	Copies a file to a SE and registers the file in the catalog
lcg-del	Delete one file
lcg-rep	Replication between SEs and registration of the replica
lcg-gt	Gets the TURL for a given SURL and transfer protocol
lcg-sd	Sets file status to “Done” for a given SURL in a SRM request

File Catalog Interaction

lcg-aa	Add an alias in LFC for a given GUID
lcg-ra	Remove an alias in LFC for a given GUID
lcg-rf	Registers in LFC a file placed in a SE
lcg-uf	Unregisters in LFC a file placed in a SE
lcg-la	Lists the alias for a given SURL, GUID or LFN
lcg-lg	Get the GUID for a given LFN or SURL
lcg-lr	Lists the replicas for a given GUID, SURL or LFN

LFC C API

Low level methods (many POSIX-like):

lfc_access
lfc_aborttrans
lfc_addreplica
lfc_apiinit
lfc_chclass
lfc_chdir
lfc_chmod
lfc_chown
lfc_closedir
lfc_creat
lfc_delcomment
lfc_delete
lfc_deleteclass
lfc_delreplica
lfc_endtrans
lfc_enterclass
lfc_errmsg
lfc_getacl
lfc_getcomment
lfc_getcwd
lfc_getpath
lfc_lchown
lfc_listclass
lfc_listlinks
lfc_listreplica
lfc_lstat
lfc_mkdir
lfc_modifyclass
lfc_opendir
lfc_queryclass
lfc_readdir
lfc_readlink
lfc_rename
lfc_rewind
lfc_rmdir
lfc_selectsrvr
lfc_setacl
lfc_setatime
lfc_setcomment
lfc_seterrbuf
lfc_setfsize
lfc_starttrans
lfc_stat
lfc_symlink
lfc_umask
lfc_undelete
lfc_unlink
lfc_utime
send2lfc

LFC commands

Summary of the LFC Catalog commands

lfc-chmod	Change access mode of the LFC file/directory
lfc-chown	Change owner and group of the LFC file-directory
lfc-delcomment	Delete the comment associated with the file/directory
lfc-getacl	Get file/directory access control lists
lfc-ln	Make a symbolic link to a file/directory
lfc-ls	List file/directory entries in a directory
lfc-mkdir	Create a directory
lfc-rename	Rename a file/directory
lfc-rm	Remove a file/directory
lfc-setacl	Set file/directory access control lists
lfc-setcomment	Add/replace a comment

LFC other commands

Managing ownership and permissions:

- lfc-chmod
- lfc-chown

Managing ACLs:

- lfc-getacl
- lfc-setacl

Remember that per user mapping can change in every session.

The default is for LFNs and directories to be VO-wide readable.

Consistent user mapping will be added soon.

Renaming:

- lfc-rename

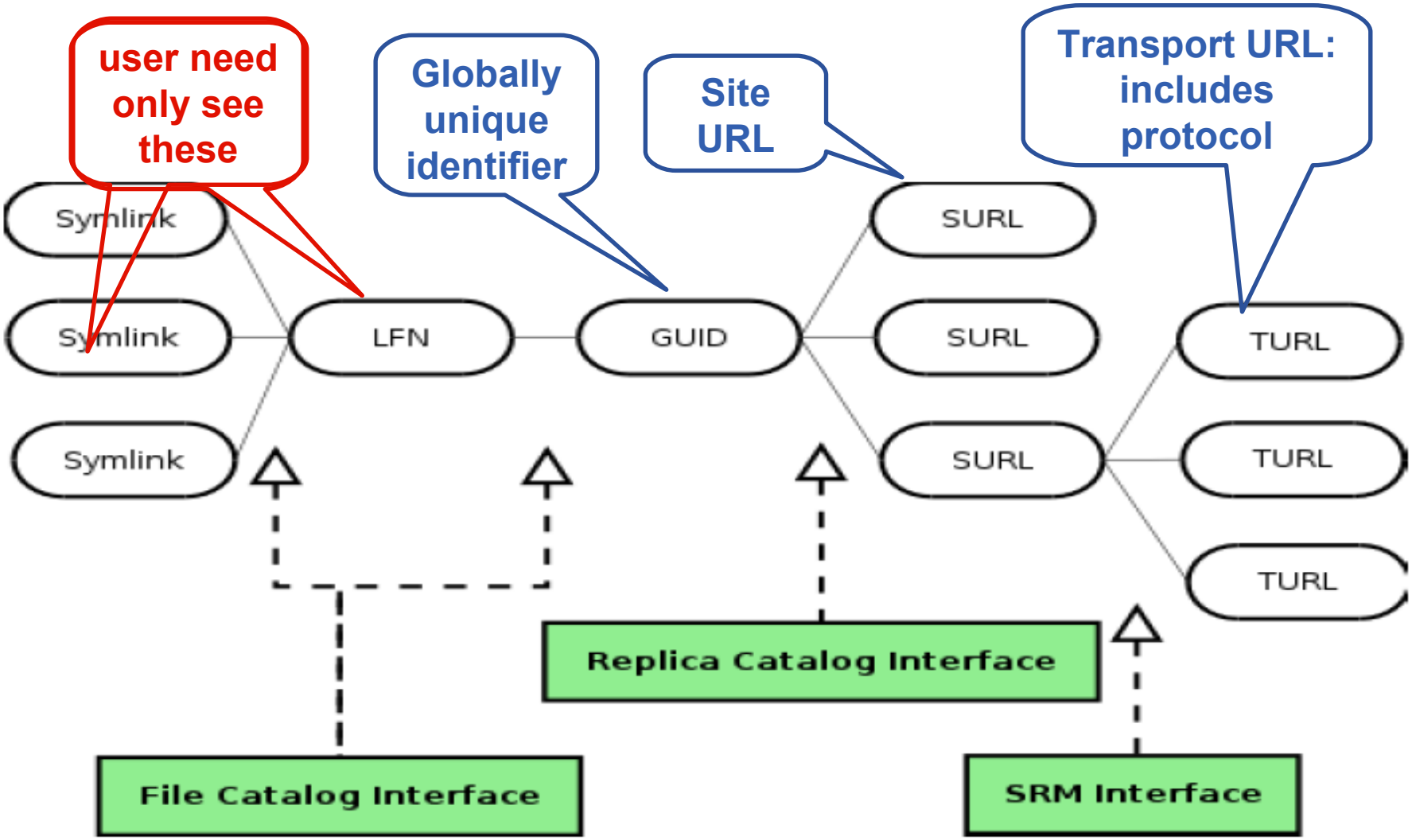
Removing:

- lfc-rm

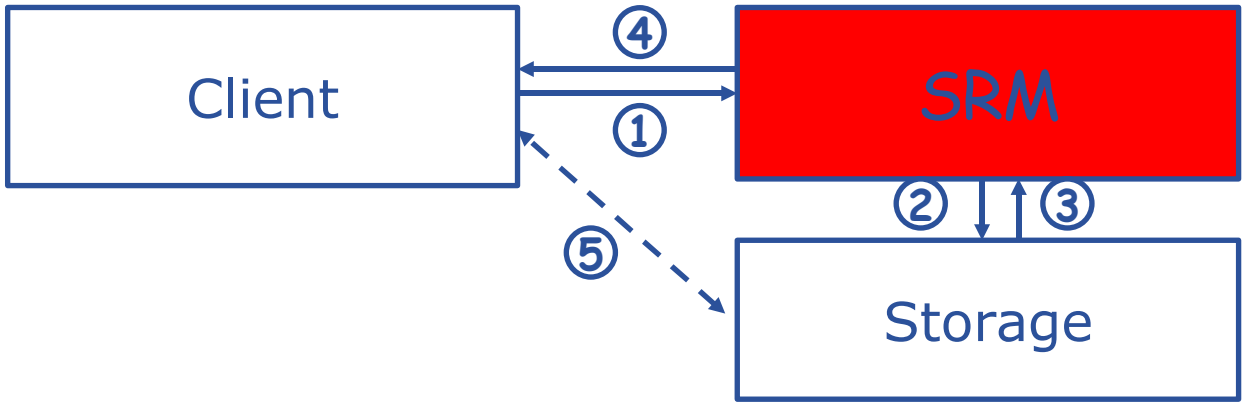
An LFN can only be removed if it has no SURLs associated.

LFNs should be removed by lcg-del, rather than lfc-rm.

File names and identifiers in gLite



SRM Interactions



1. The client asks the SRM for the file providing an 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

GFAL I/O

Data Movement Service

- gLite FTS
- gLite FPS

Data Movement Service (1)

- **Many Grid applications will distribute a LOT of data across the Grid sites**
- **Need efficient and easy way to manage data movement service**

- **gLite File Transfer Service FTS**
  - Manage the network and the storage at both ends
  - Define the concept of a CHANNEL: a link between two SEs
  - Channels can be managed by the channel administrators, i.e. the people responsible for the network link and storage systems
  - These are potentially different people for different channels
  - Optimize channel bandwidth usage – lots of parameters that can be tuned by the administrator
  - VOs using the channel can apply their own internal policies for queue ordering (i.e. professor's transfer jobs are more important than student's)
- **gLite File Placement Service**
  - It **IS** an FTS with the additional catalog lookup and registration steps, i.e. LFNs and GUIDs can be used to perform replication. Could've been called File Replication Service. (**replica = managed/catalogued copy**)

Data Movement Service (2)

- **File movement is asynchronous – submit a job**
  - Held in file transfer queue
- **Data scheduler**
  - Single service per VO – can be distributed
  - VO can apply policies (priorities, preferred sites, recovery modes..)
- **Client interfaces:**
  - Browser
  - APIs
  - Web service
- **“File transfer”**
  - Uses SURL
- **“File placement”**
  - Uses LFN or GUID, accesses Catalogues to resolve them

Data Movement Service (3)

- **File movement is asynchronous – submit a job**
  - Held in file transfer queue
- **FPS fetches job transfer requests, contact File Catalogue obtaining** source / destination **SURLs**
- **Task execution is demanded to FTS**
- **User can monitor job status through jobID**
- **FTS maintains state of job transfers**
- **When job is done, FPS updates file entry in the catalogue adding the new replica**

Baseline: GridFTP

- **Data transfer and access protocol for** secure and efficient **data movement**
- **Standardized in the Global Grid Forum**
- extends **the standard** FTP **protocol**
  - Public-key-based Grid Security Infrastructure (GSI) or Kerberos support (both accessible via GSS-API
  - Third-party control of data transfer
  - Parallel data transfer
  - Striped data transfer Partial file transfer
  - Automatic negotiation of TCP buffer/window sizes
  - Support for reliable and restartable data transfer
  - Integrated instrumentation, for monitoring ongoing transfer performance

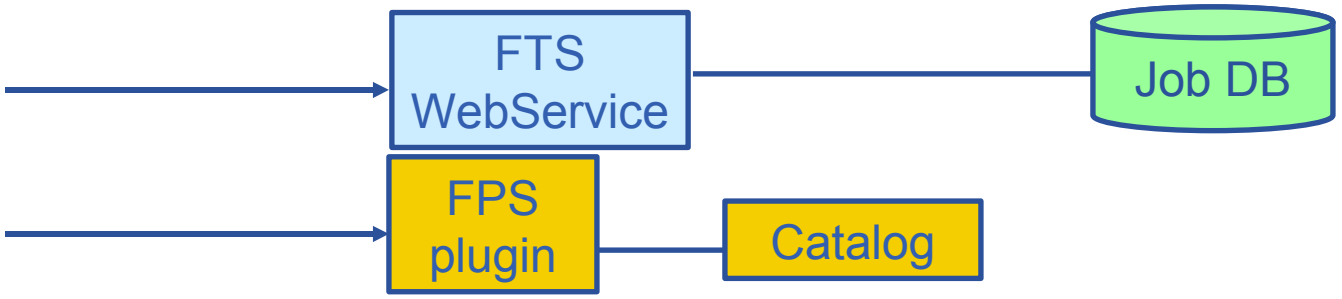
Reliable File Transfer

- **GridFTP is the basis of most transfer systems**
- **Retry functionality is limited**
  - Only retries in case of network problems; no possibility to recover from GridFTP a server crash
- **GridFTP handles one transfer at a time**
  - No possibility to do bulk optimization
  - No possibility to schedule parallel transfers

- **Need a layer on top of GridFTP that provides reliable scheduled file transfer**
  - FTS/FPS
  - Globus RFT (layer on top of single gridftp server)
  - Condor Stork

FTS vs FPS (1)

- **File Transfer Service (FTS)**
  - Acts only on SRM SURLs or gsiftp URLs
  - `submit(source-SURL, destination-SURL)`
- **File Placement Service (FPS)**
  - A plug-in into the File Transfer that allows to act on logical file names (LFNs)
  - Interacts with replica catalogs (similar to gLite-I/O)
  - Registers replicas in the catalog
  - `submit(transferJobs) (transferJob = sourceLFN, destinationSE)`



FTS vs FPS (2)

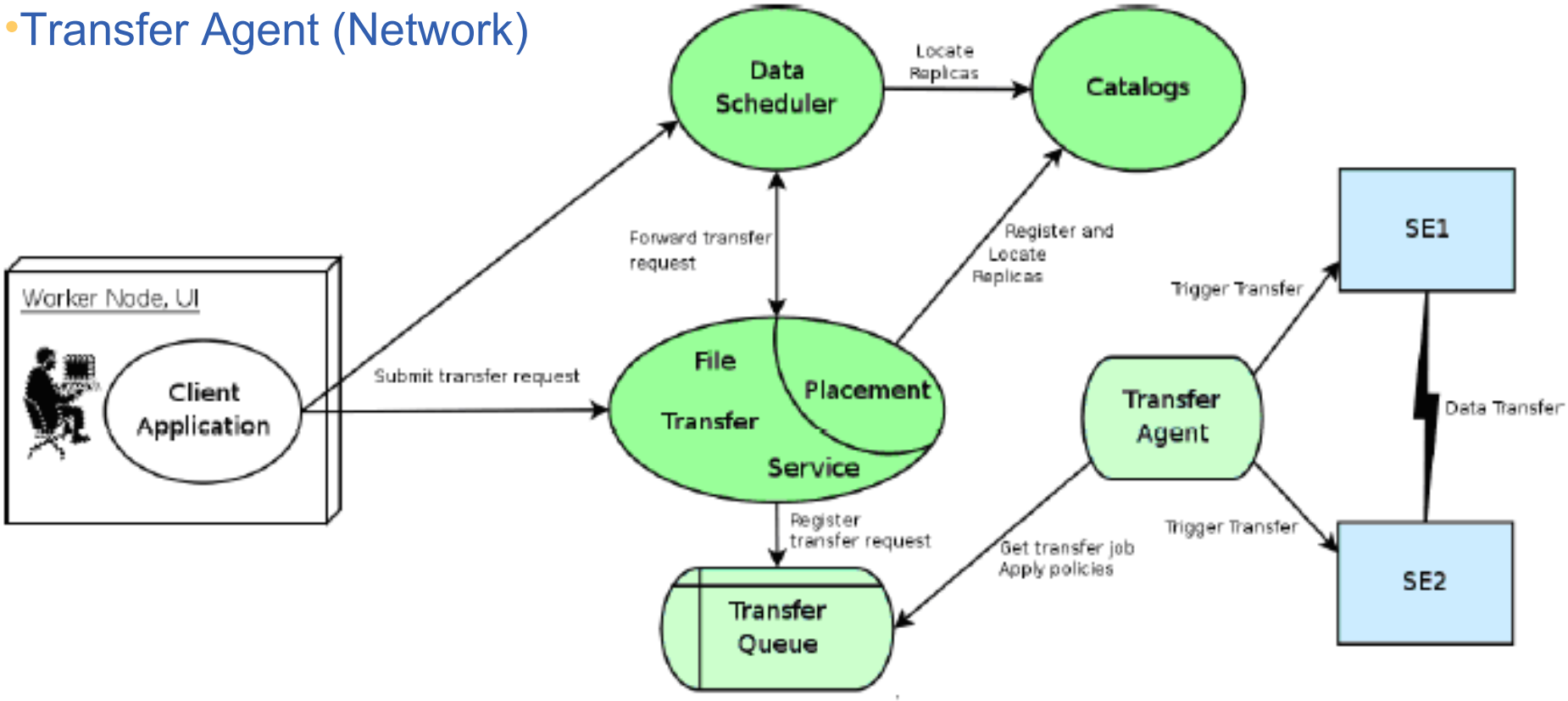
- **Using the File Transfer Service (FTS)**
  - Initiate and monitor transfer
  - Plugin takes care of catalog interactions
- **Using the File Placement Service (FPS)**
  - Lookup source SURL in replica catalog
  - Initiate and monitor transfer
  - After successful transfer register new replica in the catalog
- **FTS and FPS offer the same interface**
  - Difference only in input parameters to the submit command
  - Different configuration
    - SURLs vs. LFNs
    - FPS requires catalog endpoint

Data Movement Stack

	LFN	SURL	Manipulates	Notes
File Catalog	Yes	Yes	Nothing	Only valid data should get here
File Placement Service	Yes	Yes	Catlog entries, FTS transfers	Will make new catalog entries
File Transport Service	No	Yes	Channels, Data transfers	Will retry failed transfers
Grid FTP	No	Yes	Low level data transport	Can fail disgracefully!

Overview of Data Movement Services

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



References

- **gLite homepage**
  - http://www.glite.org
- **DM subsystem documentation**
  - http://egee-jra1-dm.web.cern.ch/egee-jra1-dm/doc.htm
- **gLite-I/O user guide**
  - https://edms.cern.ch/file/570771/1.1/EGEE-TECH-570771-v1.1.pdf
- **FTS/FPS user guide**
  - https://edms.cern.ch/file/591792/1/EGEE-TECH-591792-Transfer-C

Questions…