gLite Data Management System architecture

Emidio Giorgio

INFN

First gLite tutorial on GILDA, Catania, 13-15.06.2005

Outline

- gLite DMS overview
- gLite IO Server
- gLite IO Client
- Catalogs (FiReMan)
- Transfer and Replica Services

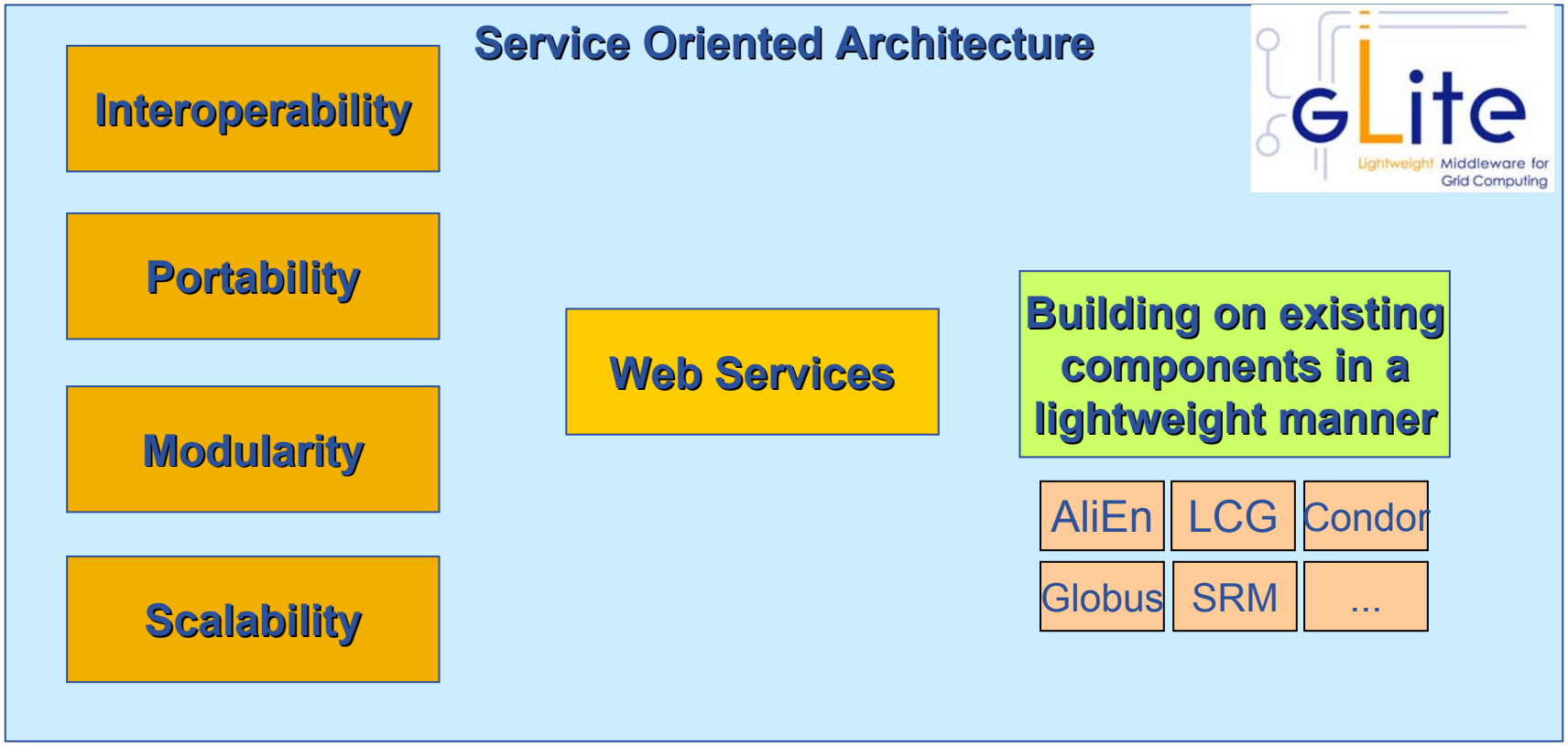
Data Management Tasks

- **File Management**
  - Storage
  - Access
  - Placement
  - Cataloguing
  - Security
- **Metadata Management**
  - Secure database access
  - Schema management
  - File-based metadata
  - Generic metadata

Data managements : general concepts

- **What does “Data Management” mean ?**
  - Users and applications produce and require data
  - Data may be stored in Grid files
  - Granularity is at the “file” level (no data “structures”)
  - Users and applications need to handle files on the Grid
- **Files are stored in appropriate permanent resources called “Storage Elements” (SE)**
  - Present almost at every site together with computing resources
  - We will treat a storage element as a “black box” where we can store data
    - Appropriate data management utilities/services hide internal structure of SE
    - Appropriate data management utilities/services hide details on transfer protocols

Guiding Principles



Data Management Services

- **Storage Element**
  - Storage Resource Manager — not provided by gLite
  - POSIX-I/O — gLite-I/O
  - Access protocols — gsiftp, https, rfio, …

rely on existing implementations

- **Catalogs**
  - File Catalog
  - Replica Catalog
  - File Authorization Service
  - Metadata Catalog

gLite FiReMan Catalog (MySQL and Oracle)

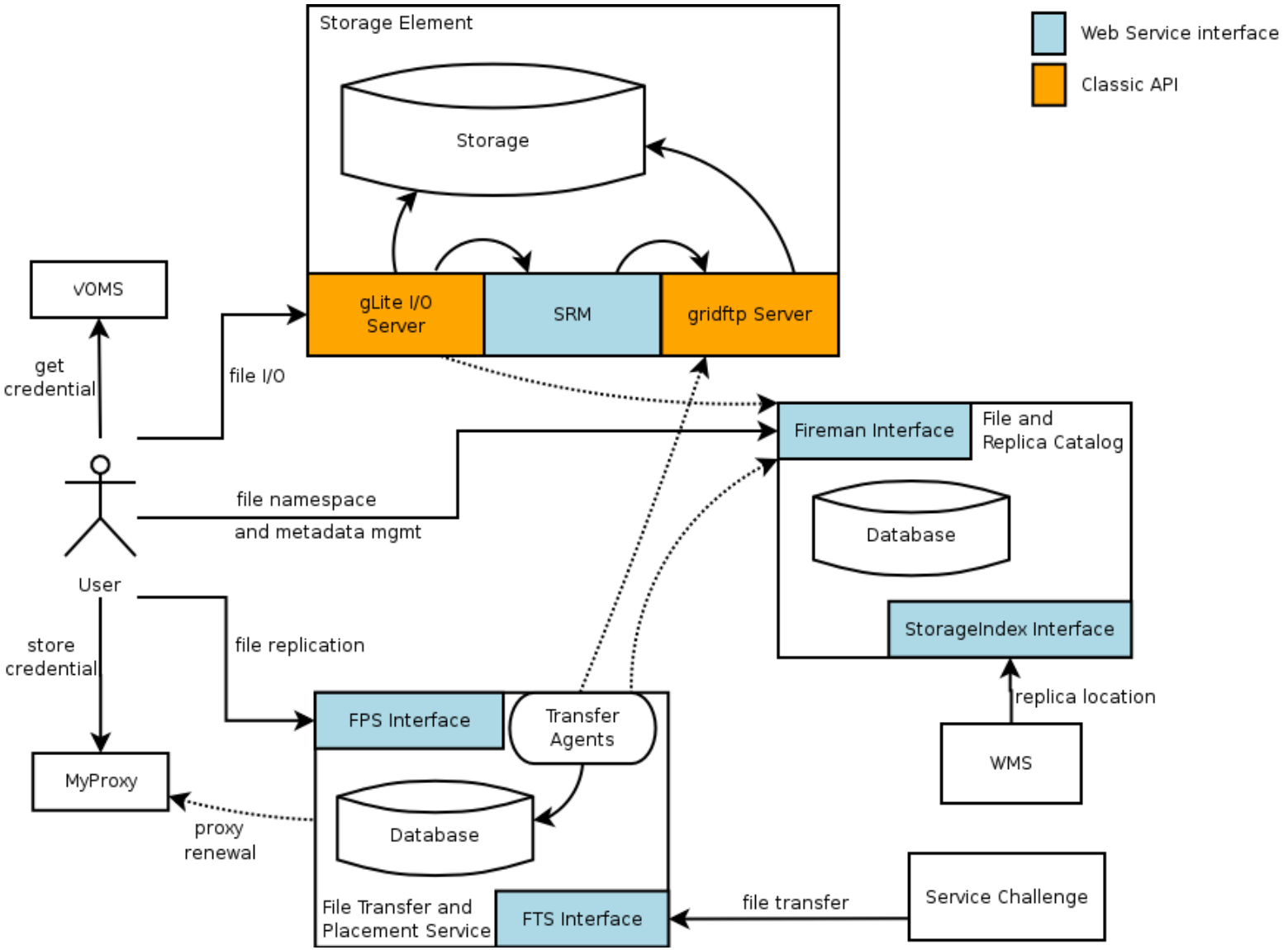
gLite Standalone Metadata Catalog

- **File Transfer**
  - Data Scheduler — planned for Release 2
  - File Transfer Service — gLite FTS and glite-url-copy
  - File Placement Service — gLite FPS

Product Overview

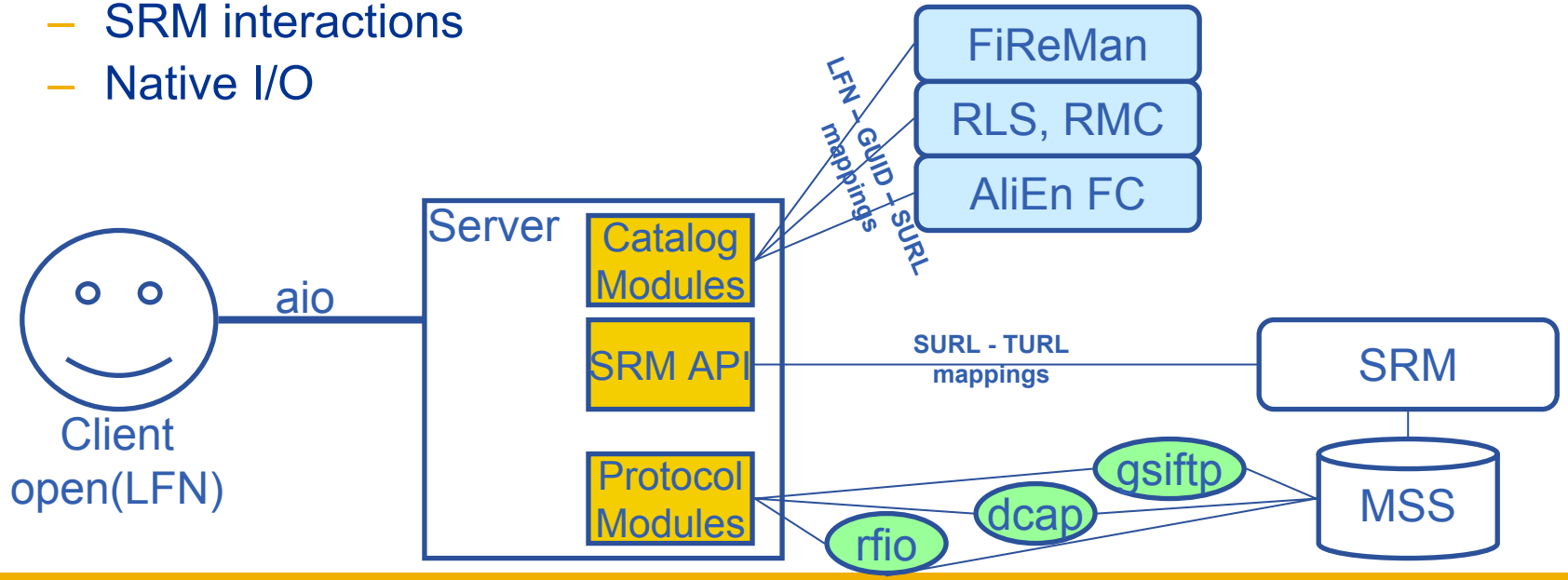
- File Storage
  - **Storage Elements** with **SRM** (Storage Resource Manager) interface
  - Posix I/O interface through **glite-io**
  - Supports transfer protocols (bbftp, https, ftp, **gsiftp**, **rfio**, **dcap**, …)
- Catalogs
  - **File and Replica Catalog**
  - **File Authorization Service**
  - **Metadata Catalog**
  - Distribution of catalogs, conflicts resolution (**messaging**)
- Transfer
  - Top-level **Data Scheduler** as global entry point (there may be many).
  - Site **File Placement Service** managing transfers and catalog interactions
  - Site **File Transfer Service** managing incoming transfers (the network resource)

Interaction Overview

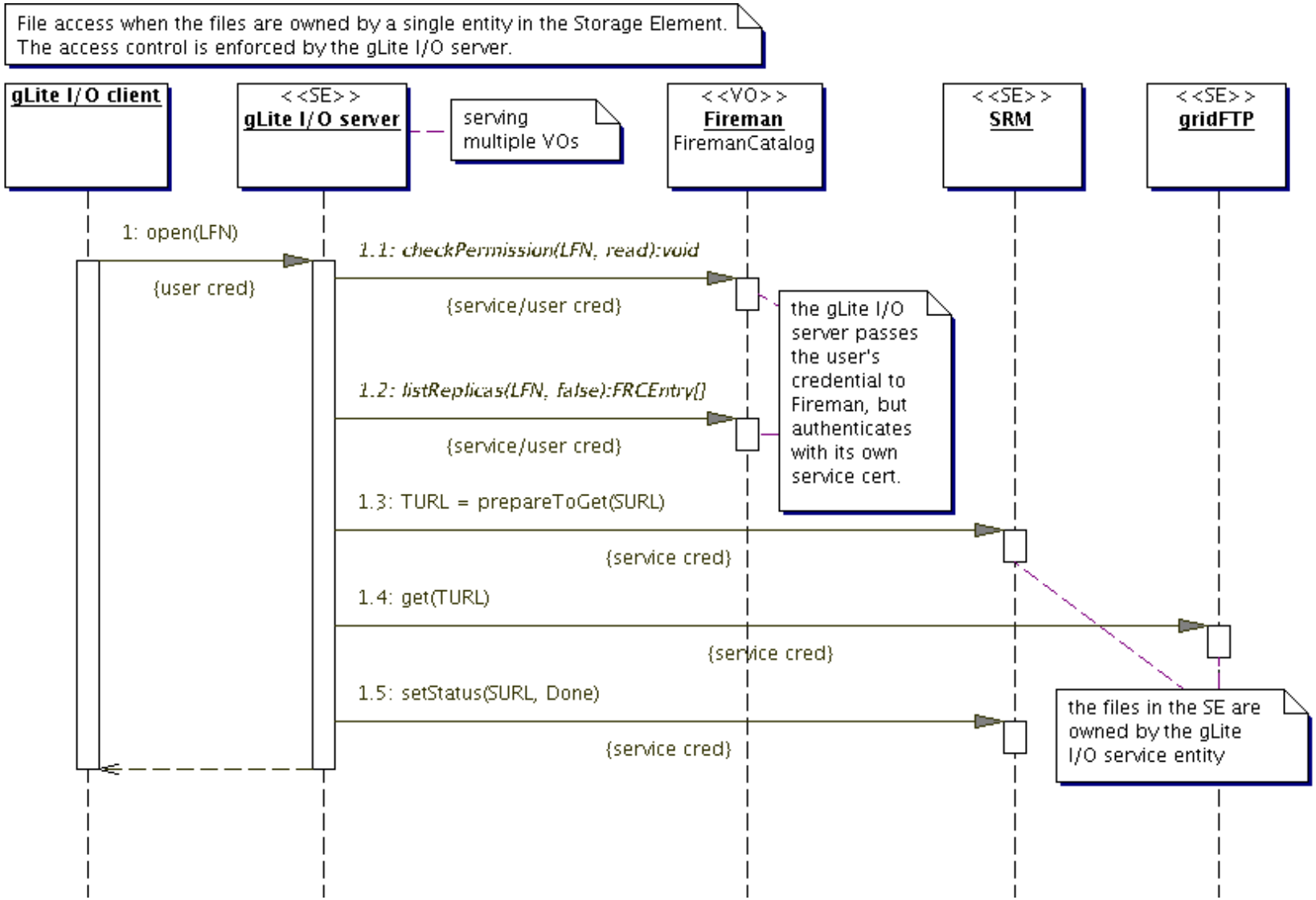


File Access Overview

- **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



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>>
gridFTP
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

gLite IO Server Installation

- **gLite IO Server supposes a MSS with SRM interface**
- **Download and execute script installer *glite-io-server.sh***
- **Basic configuration by specifyng**
  - Srm endpoint (e.g. httpg://<MSS-FQDN>:8443/**srm/managerv1**)
  - Root path to the VO dedicated directory in MSS (e.g. /pnfs/gilda)
  - Protocol (rfio → Castor, dcap → dCache)
  - Maybe necessary add support to a protocol by installing a plugin
  - Catalog Type (*supported catalog….*)
  - Catalog endpoint
  - Fas endpoint (with FiReMan it's equal to the Catalog endpoint)
- **Configure other parameters/services (global, R-GMA, VOMSes served)**
- **Run post-configuration script *glite-io-server.py***

MSS and SRM

- gLite IO server relies against a Mass Storage System implementing SRM interface
- gLite IO server comunicates with MSS through SRM
- SRM is not provided by gLite !
- Tested MSS are, till now, CASTOR and dCache
- Full support to functionalities depending also from MSS
- Installing and configuring MSS is apart from gLite issues
- How to and guides to do so

http://egee-na4.ct.infn.it/wiki/out_pages/dCache-SRM.html
http://storage.esc.rl.ac.uk/documentation/html/D-Cache-Howto

IO server and catalogs

- The “official” gLite catalog is FiReMan
- Other catalogs types are supported
  - File and Replica Catalog (AliEn)→ `fr`
  - EDG RLS & RMC → `catalogs`
- Value to be set is **init.CatalogType**

- If, for any reason, IO Server cannot contact any catalog, won’t be able to run
- Need to configure only parameters needed by the supported catalog (typically its endpoints)

gLite IO client

- **IO client installation comes with UI and WN's ones**
- **Xml file to be edited is `/opt/glite/etc/config/glite-io-client.cfg.xml`**
- **Needs only to have specified**
  - IO server hostname and listening port
  - VO served by the instance
  - Catalogs type and endpoints
    - Several catalogs can be specified, default is the first one
    - User switchs them through –s <catalog Name> option
- **Configuration is effective when is run *glite-io-client.py***
- **Supported catalog on the UI are the ones listed under**

  `/opt/glite/etc/services.xml`

Basic IO commands

Copy a local file to Storage Element

- glite-put *local-file lfn:///lfn-name*

Copy a file from Storage element

- glite-get *lfn:///lfn-name localfile-path*

Remove a file from Storage element

- glite-rm *lfn:///lfn-name*

  if the lfn is the last replica, file entry is removed from the catalog

Before of executing glite-put or glite-rm, Fas checks that user has rights to perform requested operation.

Data transfer and replication

- **Data movements capability (should be…) provided by**
  - Data scheduler (DS) (top-level)
  - File Placements Services (FPS) (local)
  - Transfer Agent (FTA) (local)
  - File Transfer Library (low lewel, called by applications)
- **DS keeps track of data movement request submitted by clients**
- **FPS pools DS fetching transfers with local site as destination, updating catalog**
- **FTA mantains state of transfers and manages FTA**
- **Data scheduler has not been released with gLite 1.x**
- **So actually no replica can be performed with gLite DMS**

Distribution Mechanism 1

- **Data Scheduler (global and local schedulers)**
  - Global scheduler (VO-specific) takes requests like
    - Copy set of files from A to B
    - Make set of files available at C
    - Upload files from GSIFTP server to D
    - Delete files
    - *Maybe also metadata operations*
  - Local scheduler fetches tasks from known global schedulers
    - Coupled tightly to a local transfer service
    - Manage transfer where the local site is a target
    - Assure atomicity of transfer and catalog operations
- **Transfer Service**
  - Queue data transfers to/from a given Storage Element (SRM)
  - Receives jobs from local scheduler
  - Manages transfers through a set of states

Questions…