

# Data services in gLite

[www.eu-egee.org](http://www.eu-egee.org)

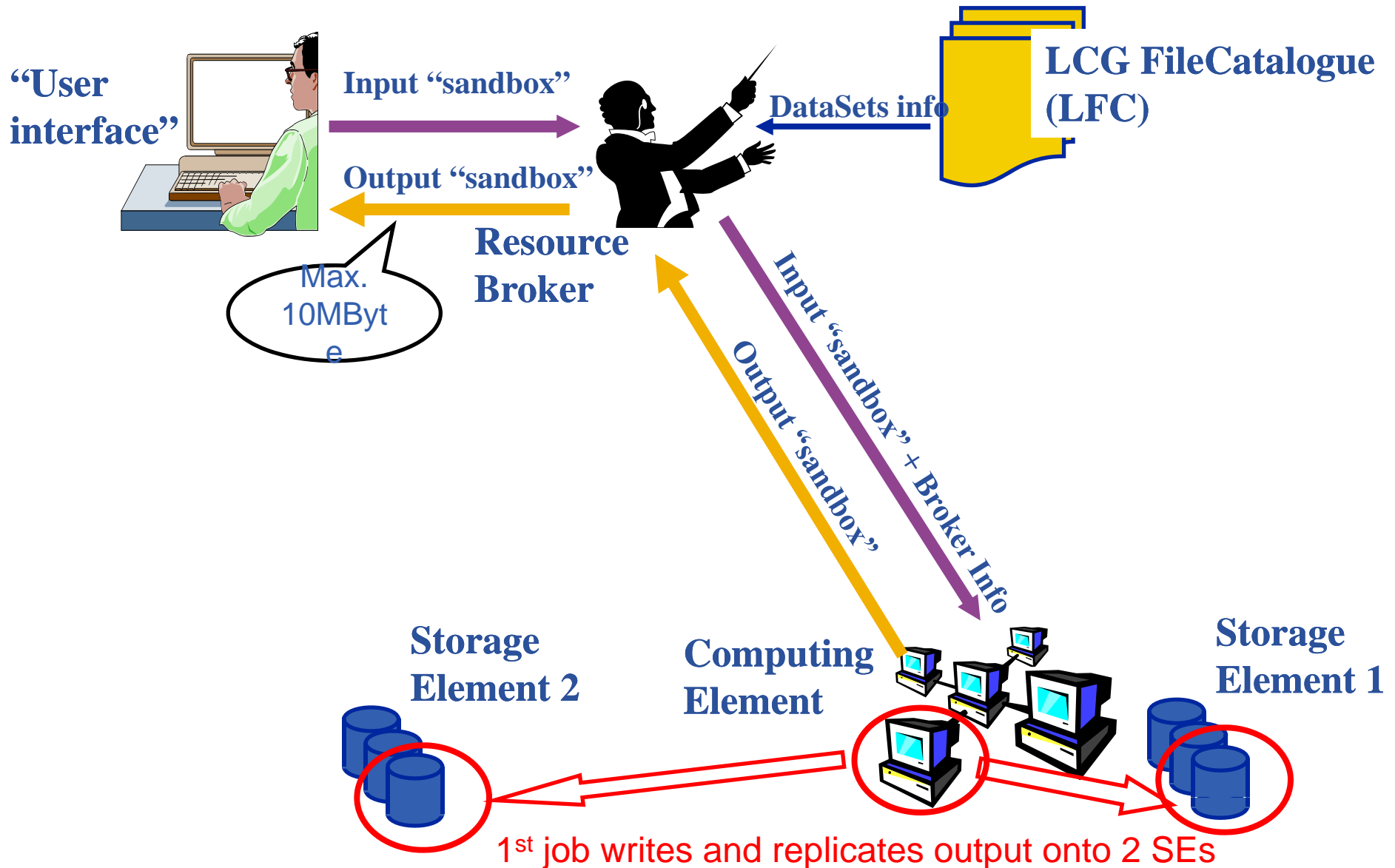
- **Simple data files on grid-specific storage**
- **Middleware supporting**
  - **Replica files**
    - to be close to where you want computation
    - For resilience
  - **Logical filenames**
  - **Catalogue:** maps logical name to physical storage device/file
  - **Virtual filesystems,** POSIX-like I/O
  - Services provided: storage, transfer, catalogue that maps logical filenames to replicas.
- **Solutions include**
  - **gLite data services**
  - **Globus: Data Replication Service**
  - **Storage Resource Broker**

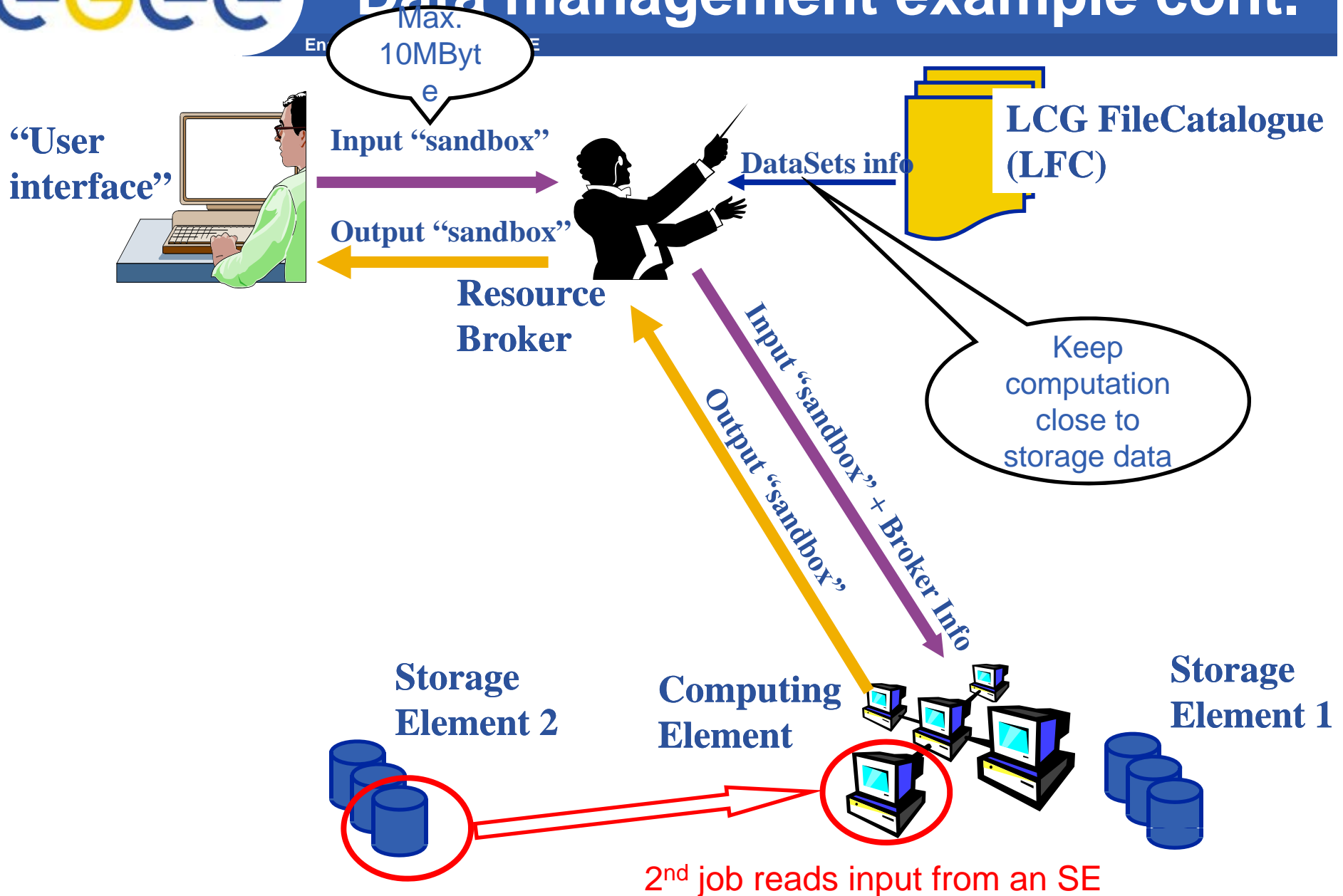
- **Other data! e.g. ....**
  - Structured data: RDBMS, XML databases,...
  - Files on project's filesystems
  - Data that may already have other user communities not using a Grid
- Require extendable middleware tools to support
  - Computation near to data
  - Controlled exposure of data *without replication*
- Basis for integration and federation
- **GReIC**
  - See tomorrow
- **OGSA –DAI**
  - In Globus 4
  - Not (yet...) in gLite

- **Files that are write-once, read-many**
  - If users edit files then
    - They manage the consequences!
    - Maybe just create a new filename!
  - No intention of providing a global file management system
  
- **3 service types for data**
  - Storage
  - Catalogs
  - Transfer

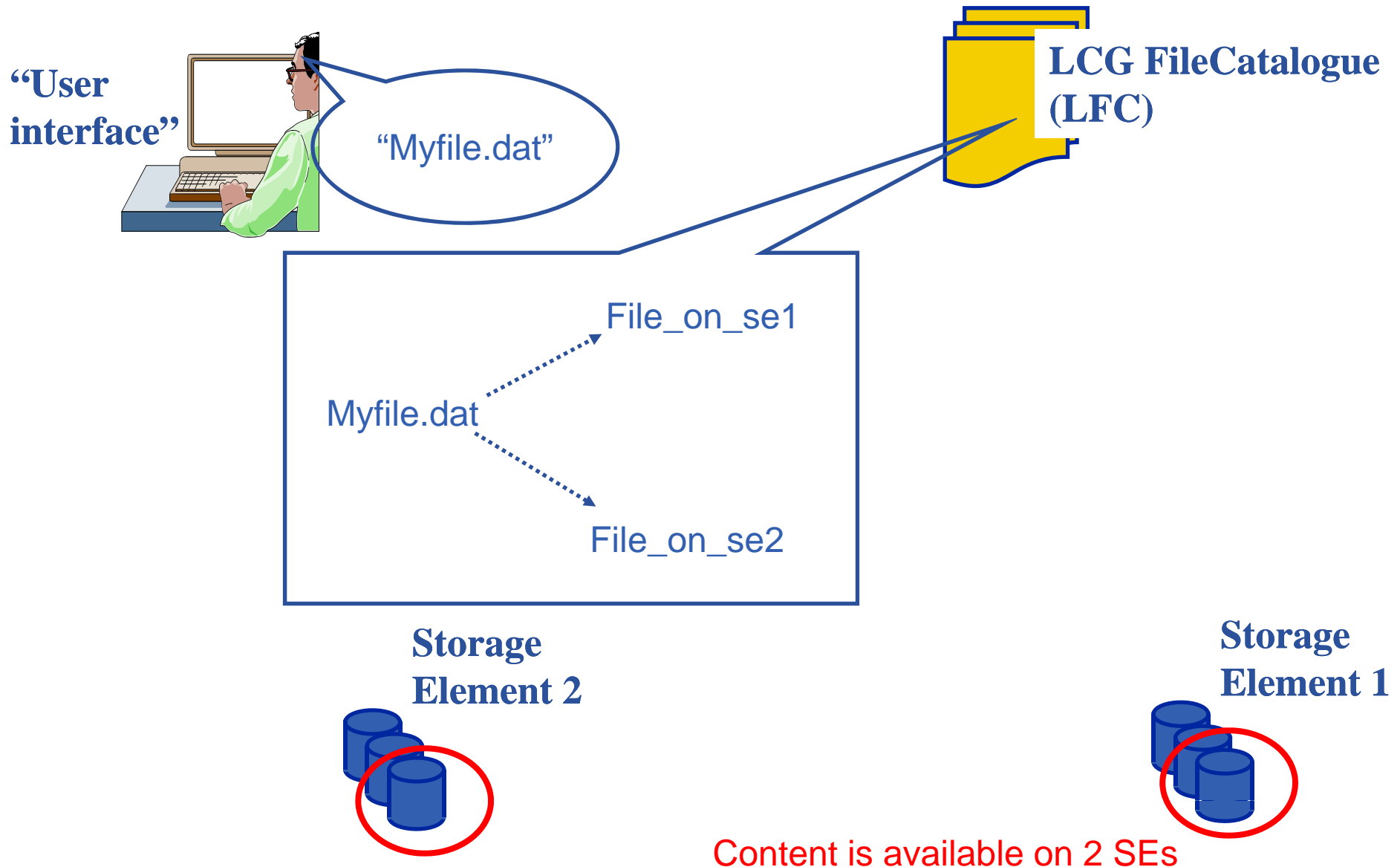
(+ Metadata catalog – AMGA)

# Data management example



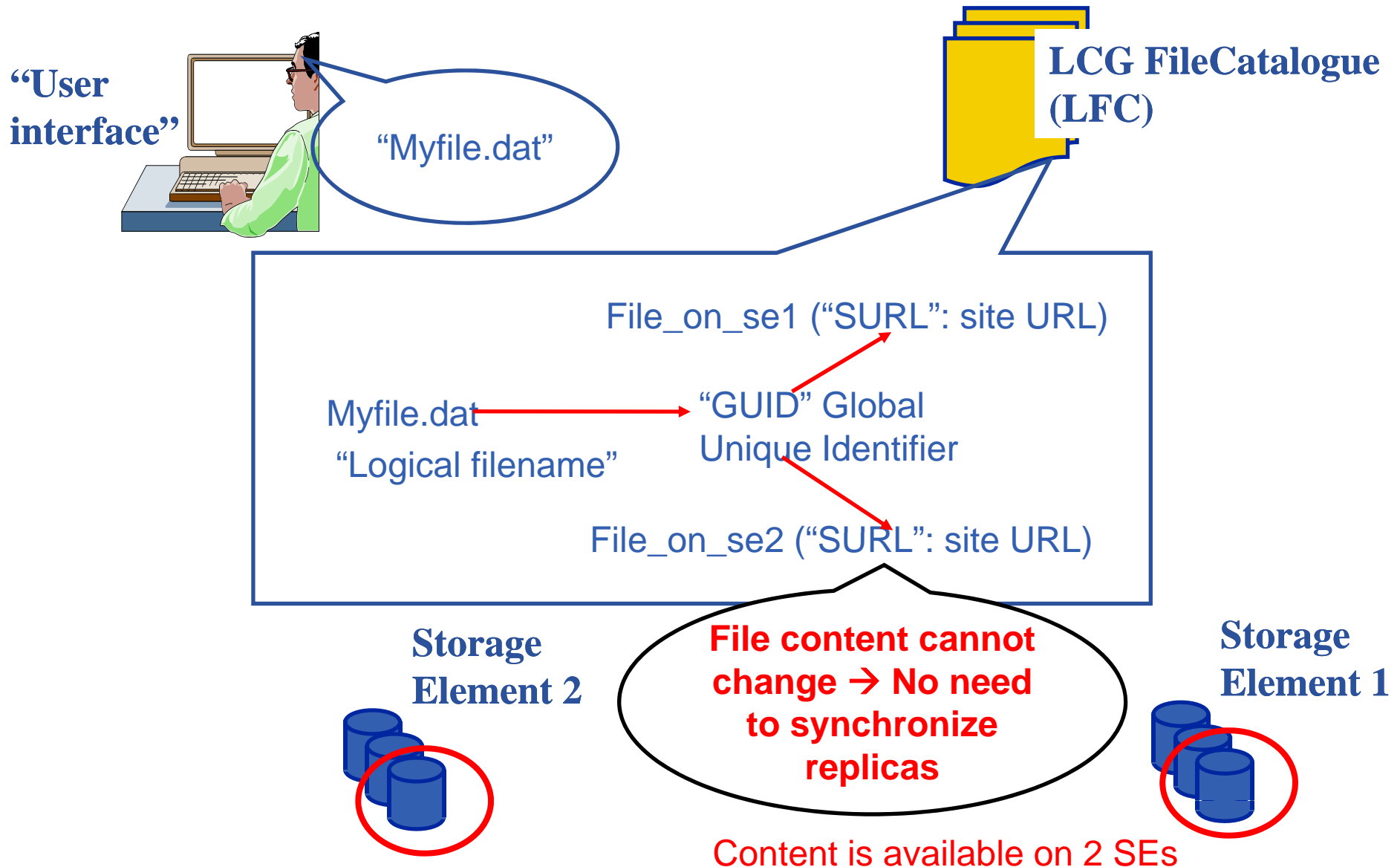


# Resolving logical file name

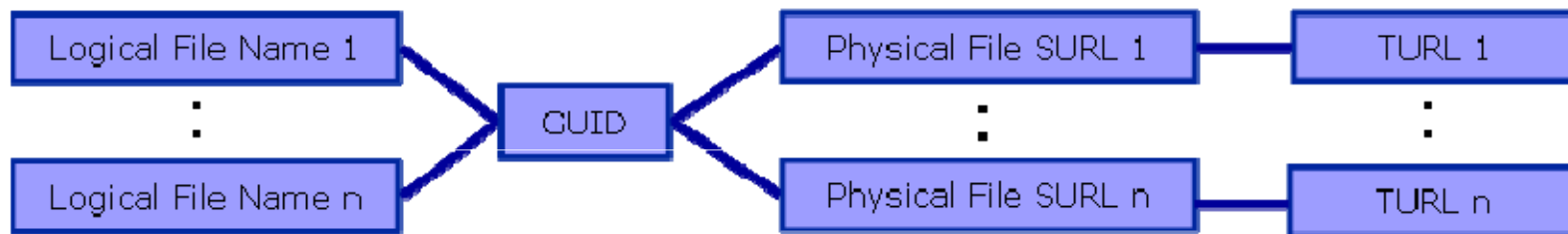


Content is available on 2 SEs

# Resolving logical file name

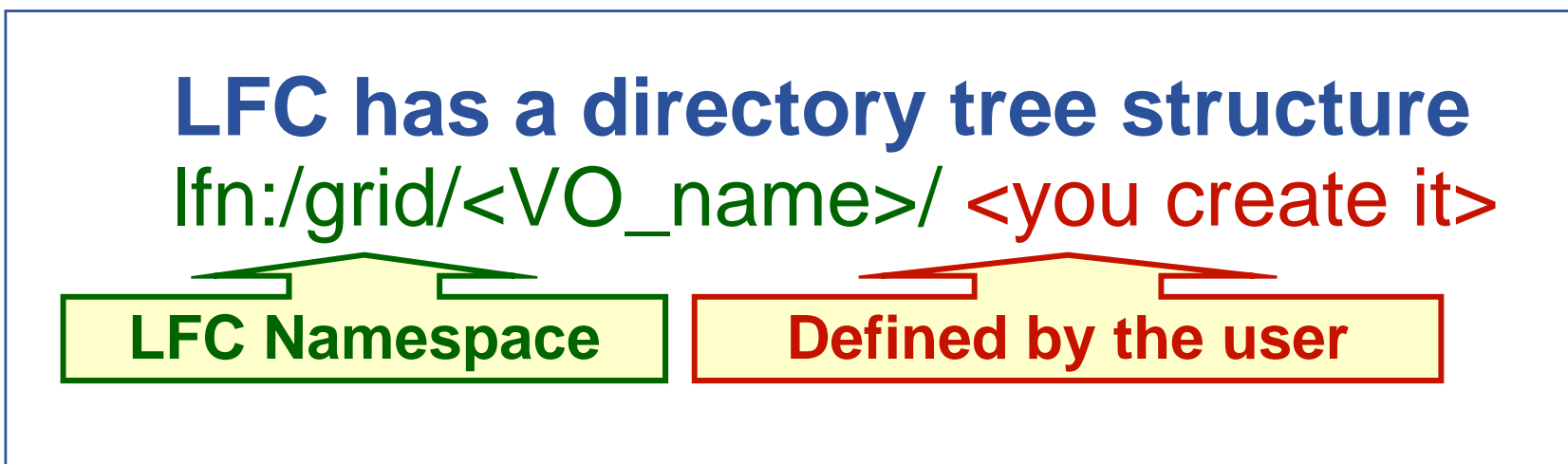


- **Logical File Name (LFN)**
  - An alias created by a user to refer to some item of data, e.g.  
`lfn:/grid/gilda/budapest23/run2/track1`
- **Globally Unique Identifier (GUID)**
  - A non-human-readable unique identifier for an item of data, e.g.  
`guid:f81d4fae-7dec-11d0-a765-00a0c91e6bf6`
- **Site 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`

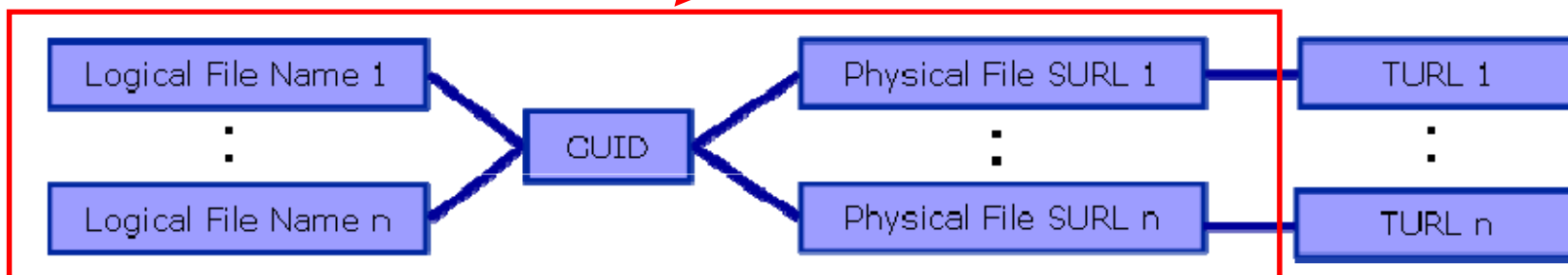




- Users primarily access and manage files through “logical filenames” - LFC



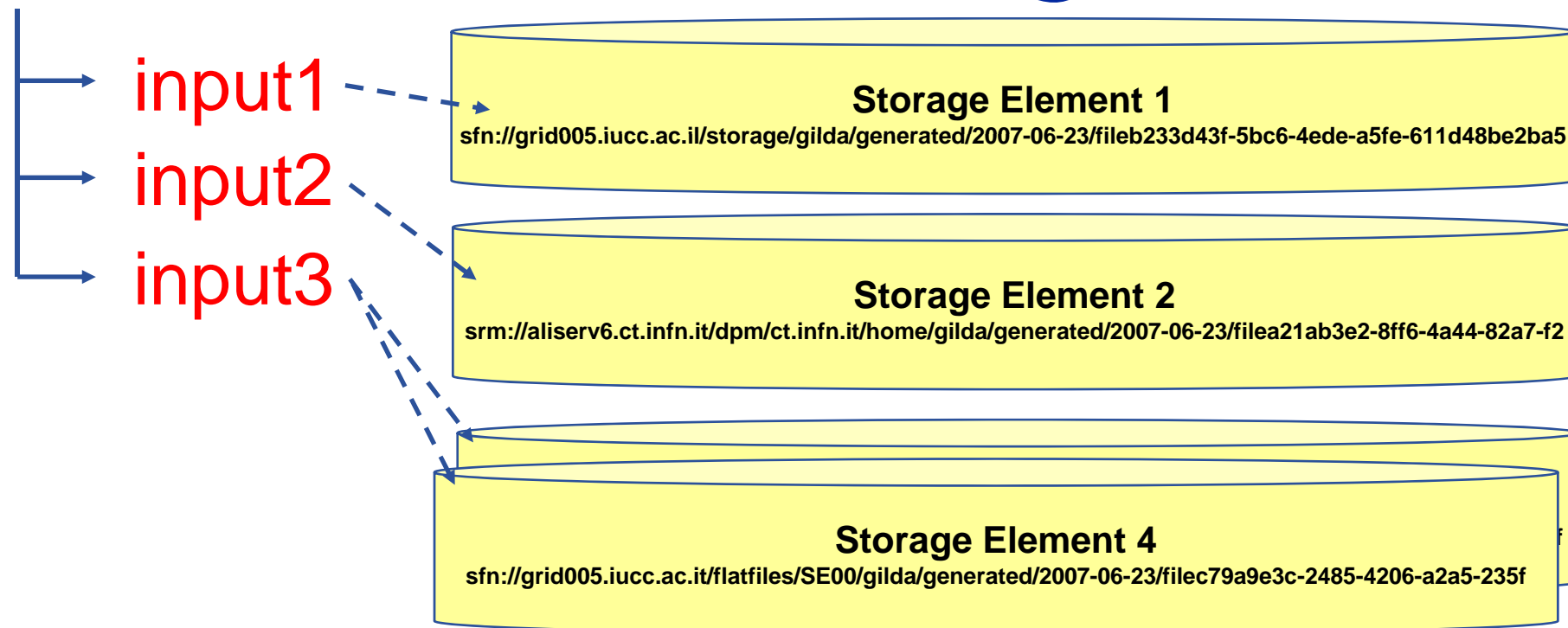
- Mapping by the “LFC” catalogue server



lfn:/grid/gilda/sipos/run2/



LCG FileCatalogue (LFC)



- **LFC directories = virtual directories**
  - Each entry in the directory may be stored on different SEs

- **lfc-\***
  - LFC = LCG File Catalogue
    - LCG = LHC Compute Grid
    - LHC = Large Hadron Collider
  - Use LFC commands to interact with the catalogue only
    - To create catalogue directory
    - List files
  - Used by you, your application and by lcg-utils (see below)
  
- **lcg-\***
  - Couples catalogue operations with file management
    - Keeps SEs and catalogue in step!
  - Copy files to/from/between SEs
  - Replicated

**LFC has a directory tree structure**

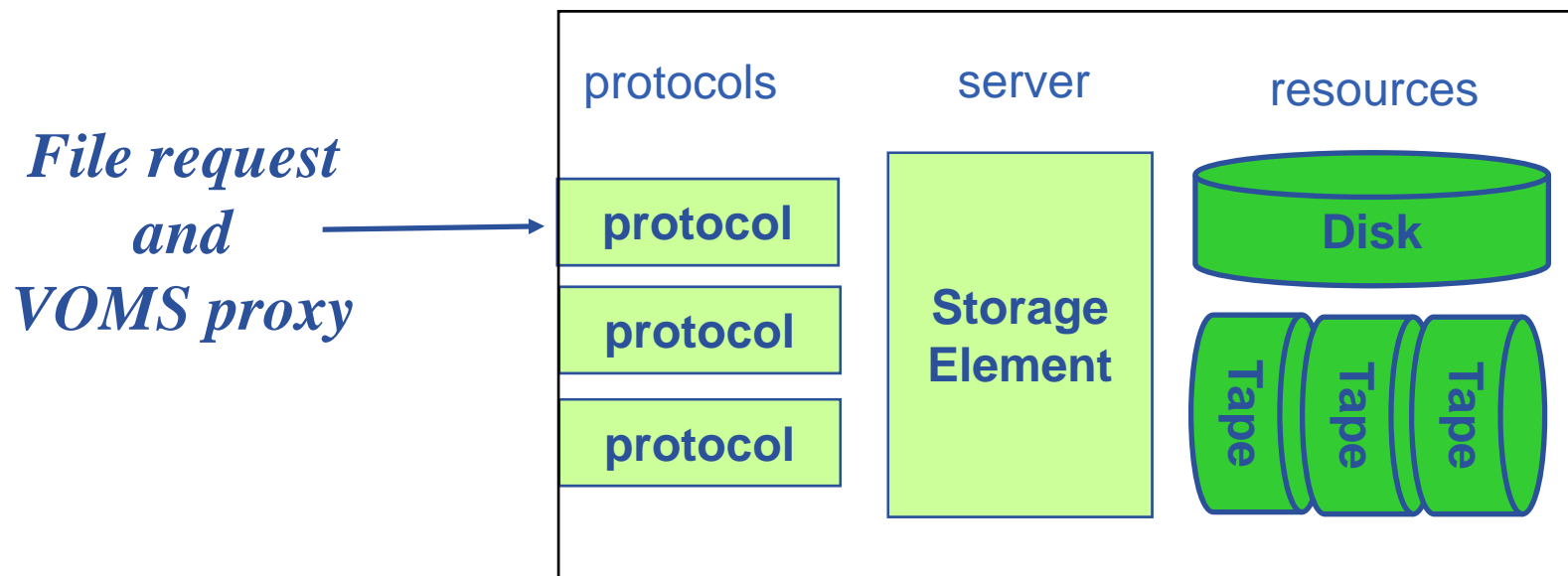
**/grid/<VO\_name>/ <you create it>**

**LFC Namespace**

**Defined by the user**

- All members of a given VO have read-write permissions in their directory
- Commands look like UNIX with “lfc-” in front (often)

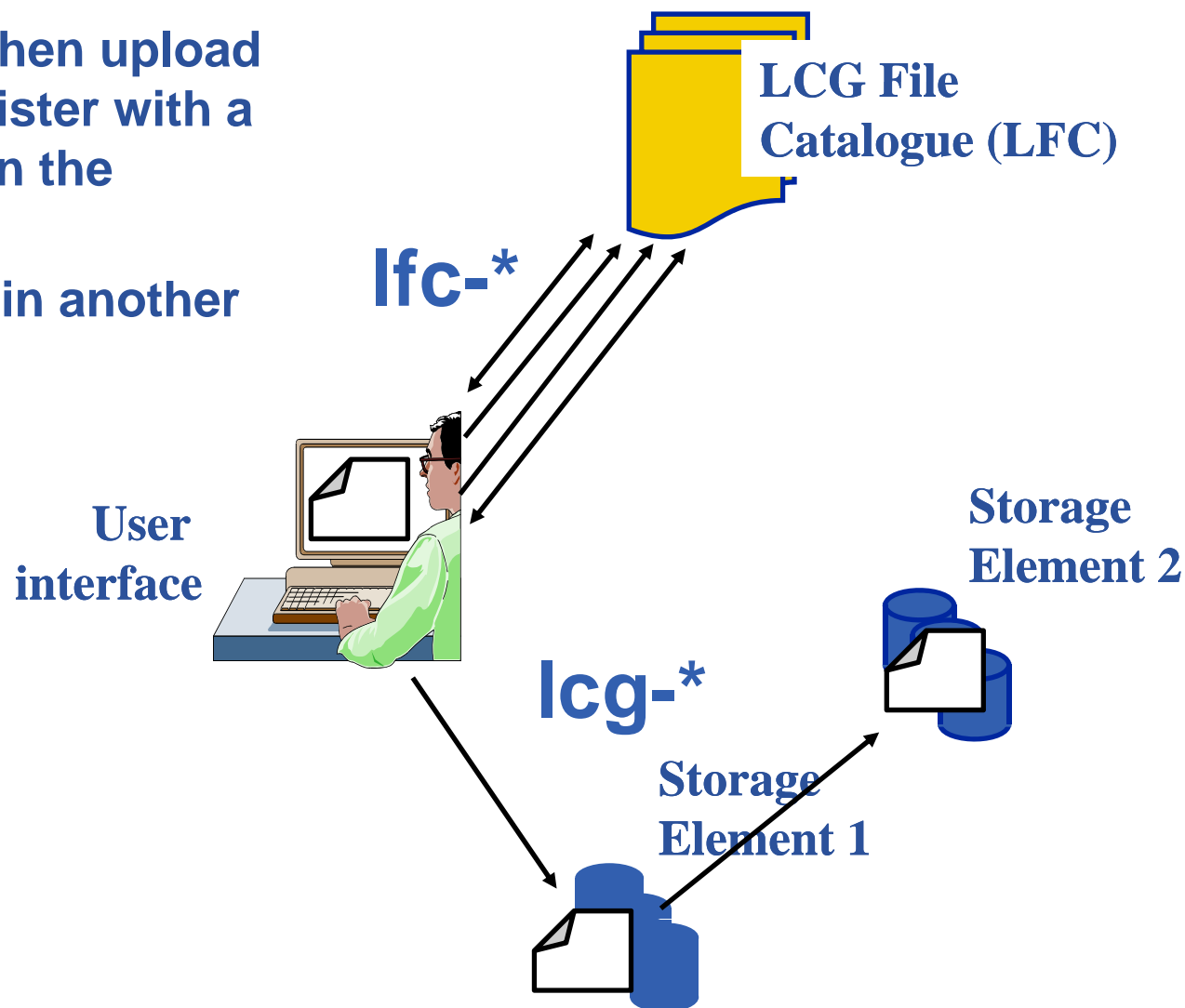
- **Provides**
  - Storage for files : massive storage system - disk or tape based
  - Transfer protocol (gsiFTP) ~ GSI based FTP server
    - Striped file transfer – cluster as back-end



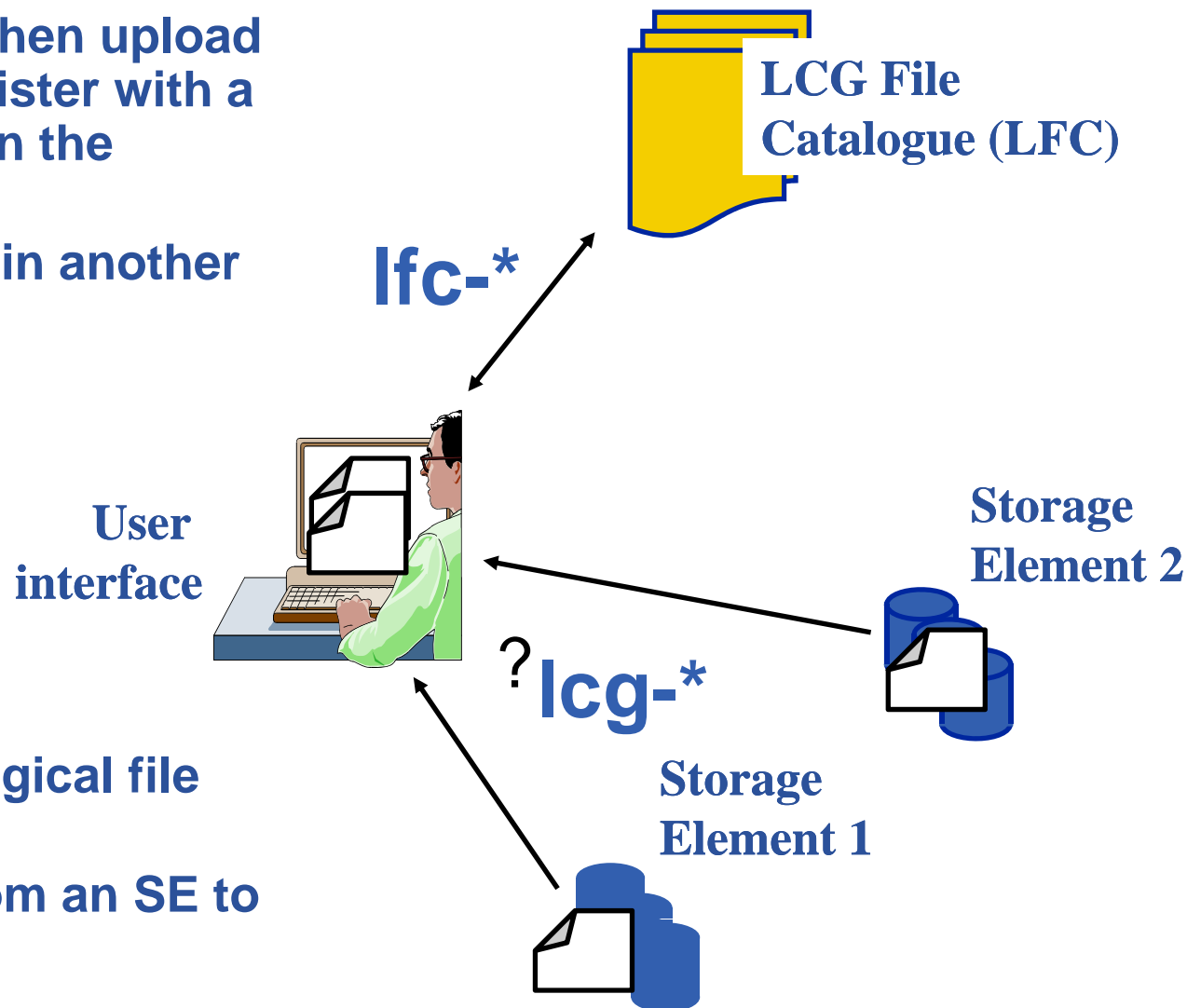
*Authentication,  
authorization*

Type	Resources	File transfer	File I/O	SRM
<b>Classic SE</b>	Disk server	GSIFTP	insecure RFIO	No
<b>DPM</b>	Disk pool	GSIFTP	secure RFIO	Yes
<b>dCache</b>	Disk pool/MSS	GSIFTP	gsidcap	Yes
<b>CASTOR</b>	MSS	GSIFTP	insecure RFIO	Yes

- List directory
- Create a local file then upload it to an SE and register with a logical name (lfn) in the catalogue
- Create a duplicate in another SE
- List the replicas

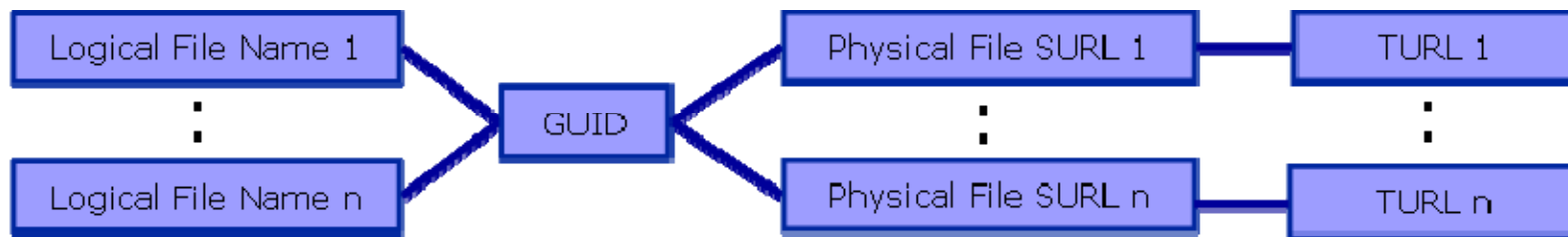


- List directory
  - Create a local file then upload it to an SE and register with a logical name (lfn) in the catalogue
  - Create a duplicate in another SE
  - List the replicas
- 
- Create a second logical file name for a file
  - Download a file from an SE to the UI





# Open the practical from the agenda



- **Spare slides follow – could be used after the practical**

- **GFAL (Grid File Access Library) is a POSIX interface for operation on file on Storage Element**
- **Enable remote handling of files**
  - When the file is large to transfer and only part of it is needed
  - When you generate so many files that they would fill the CE
- **Libraries are in C and can be included in C/C++ sources**
- **GFAL Java API – wrapper around the C code**
- **The most common of I/O operations are available, just prefix `gfal_` to the function name (`open()`, `read()`...)**
- **`man gfal` for further details**
- **The destination SE must provide secure rfio (*classic SEs don't*)**
- **GFAL API Description**
  - [http://grid-deployment.web.cern.ch/grid-deployment/documentation/LFC\\_DPM/gfal/html](http://grid-deployment.web.cern.ch/grid-deployment/documentation/LFC_DPM/gfal/html)

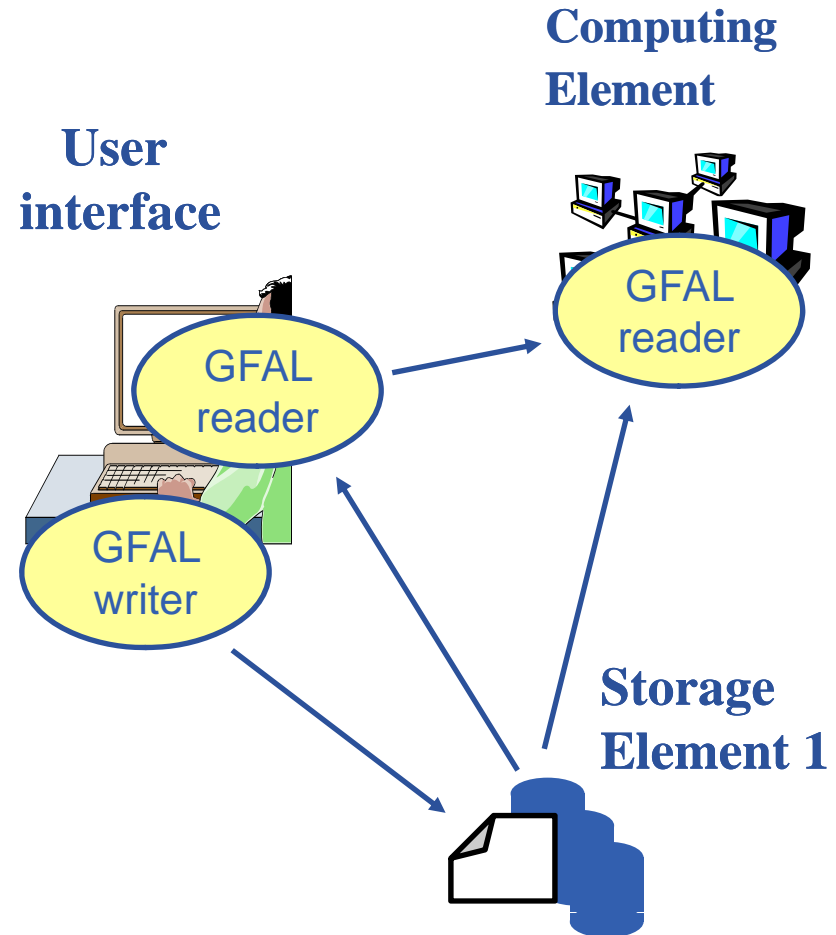
## Examples in gLite3 User Guide (Appendix F)

– <https://edms.cern.ch/file/722398//gLite-3-UserGuide.pdf>

```
int fd;
struct stat remote_file_stat;

fd = gfal_open(file_ref, O_RDONLY, 0644);
cod_ex = gfal_stat(file_ref, &file_stat)
...
cod_ex = gfal_read(fd, buffer, file_stat.st_size);
...
cod_ex = gfal_close(fd);
```

- Write a file to an SE
- Read a file from an SE
- Submit the reader code as a job into the GILDA, read the file remotely



## 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

## Replica Management

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

## FTS client

<b>glite-transfer-submit</b>	<b>Submit a transfer job : needs at least source and destination SURL</b>
<b>glite-transfer-status</b>	<b>Given one or more job ID, query about their status</b>
<b>glite-transfer-cancel</b>	<b>Delete the transfer with the give Job ID</b>
<b>glite-transfer-list</b>	<b>Query about status of all user's jobs; support options for query restrictions</b>
<b>glite-transfer-channel-list</b>	<b>Show all available channel; detailed info only if user has admin privileges</b>



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 the handling of database backups

- Minimum requirements (First scenario)
  - 2Ghz processor with 1GB of memory (not a hard requirement)
  - Dual power supply
  - Mirrored system disk