



Enabling Grids for E-science

# EGEE middleware

## *Data Management*

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



## Simple data files

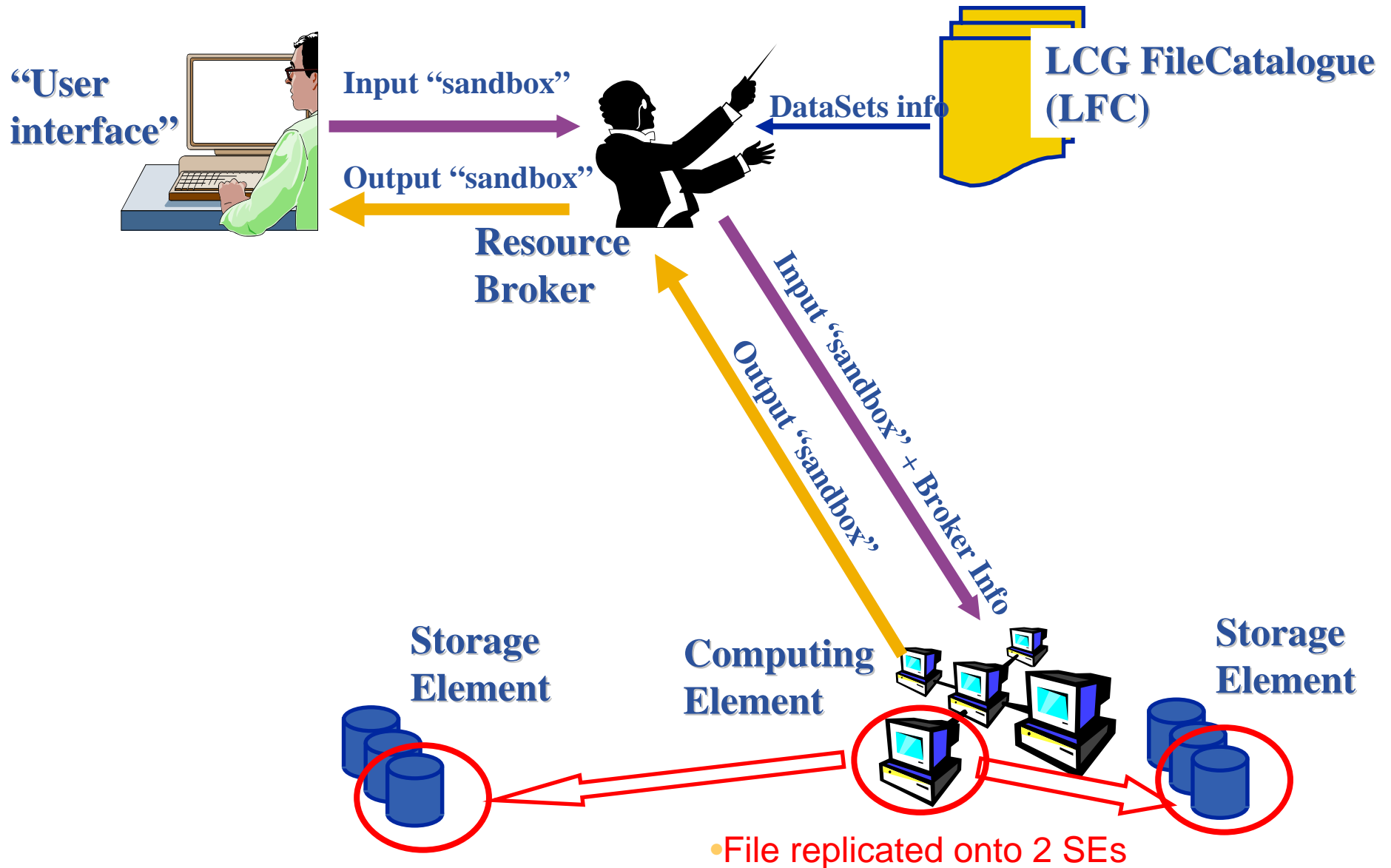
- **Middleware supporting**
  - **Replicate files**
    - to be close to where you want computation
    - For resilience
  - **Logical filenames**
  - **Catalogue:** maps logical name to physical storage device/file
  - **Virtual filesystems,** POSIX-like I/O
- **Several solutions**
  - **gLite data service**
  - **Globus**
  - **Storage Resource Broker**

## Structured data

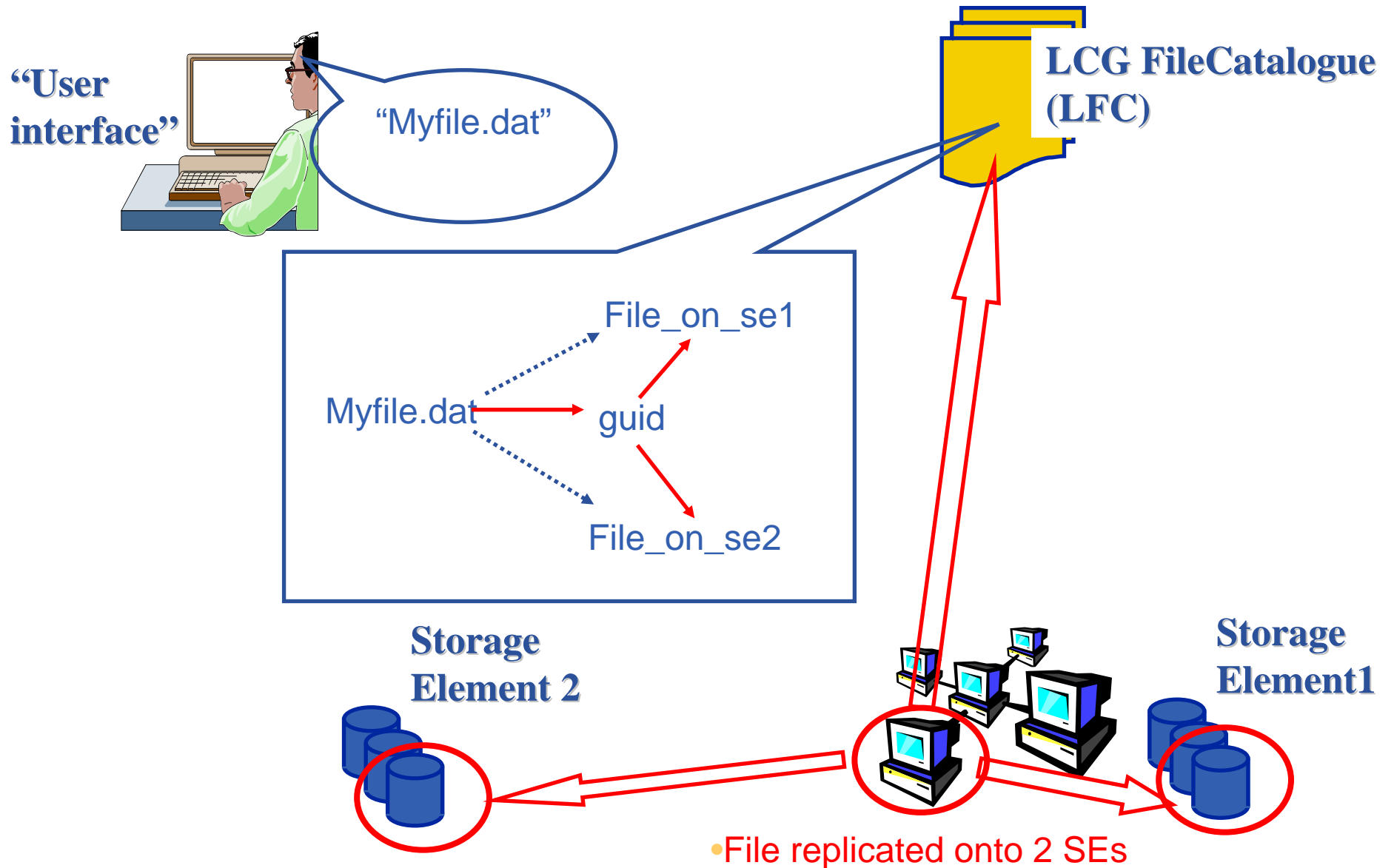
- RDBMS, XML databases
- Other data not created on/for grid services
- **Require extendable middleware tools to support**
  - Move computation near to database
  - easy access, controlled by AA
  - integration and federation
- **OGSA –DAI**
  - In Globus 4
  - Not (yet!...?) in gLite

- **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

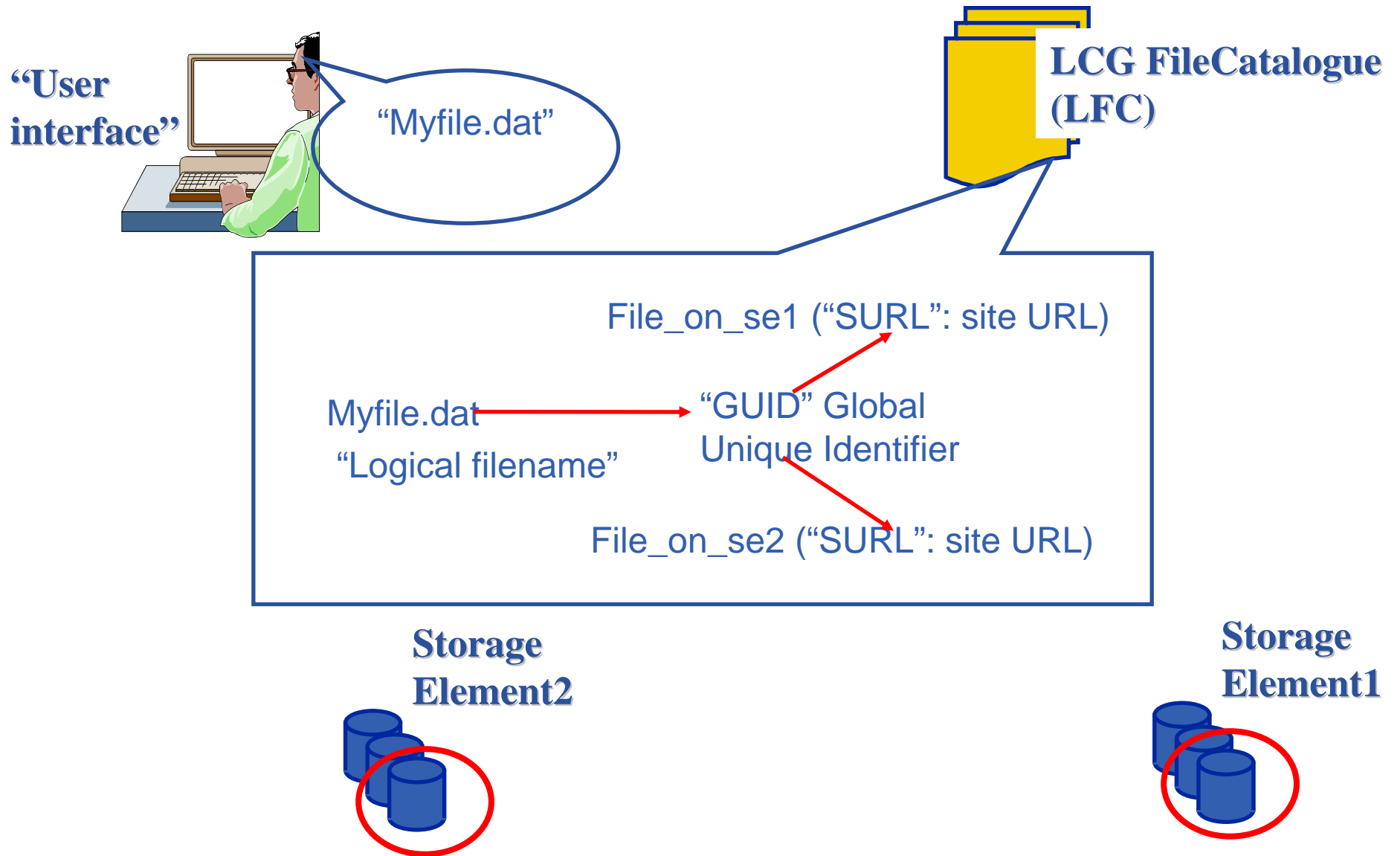
# Data management example



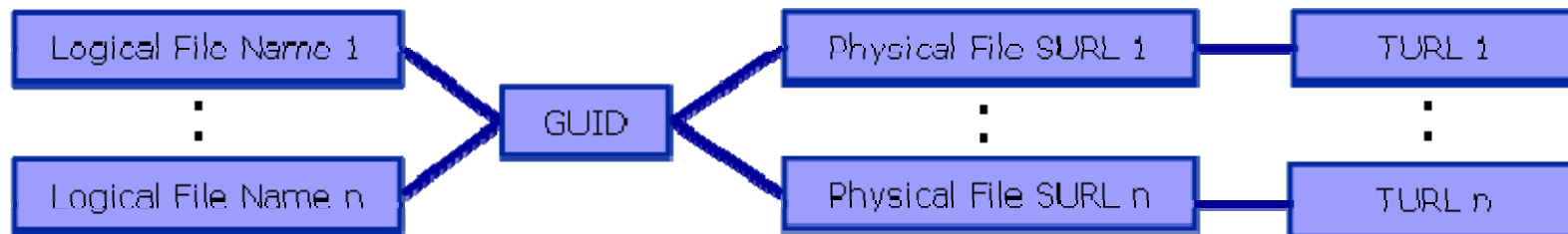
# Data management example



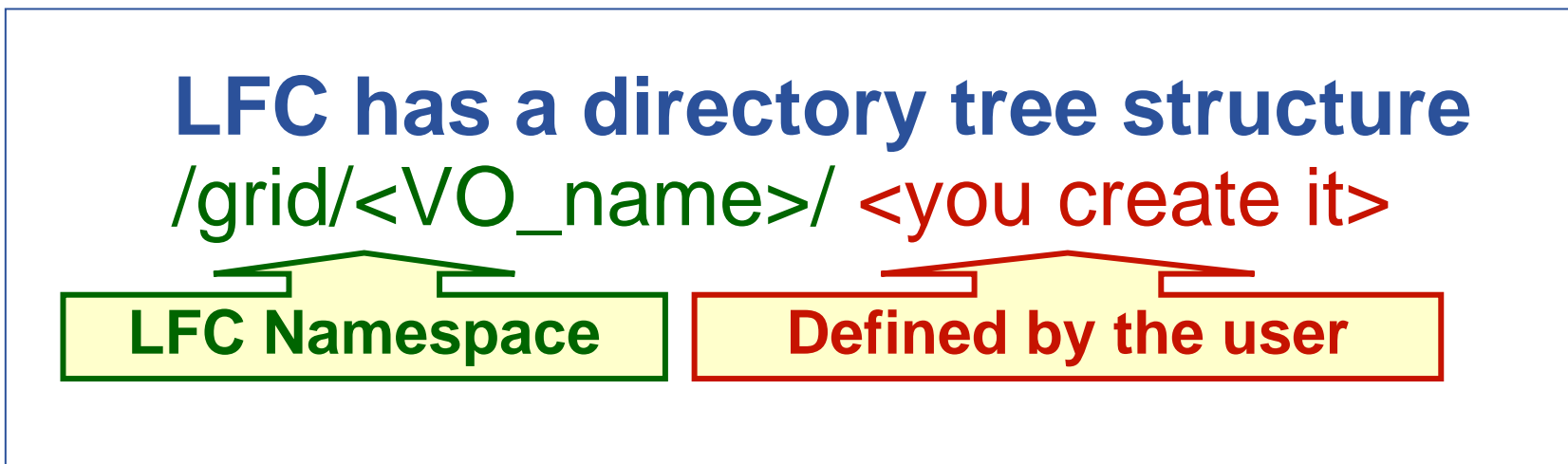
# Data management example



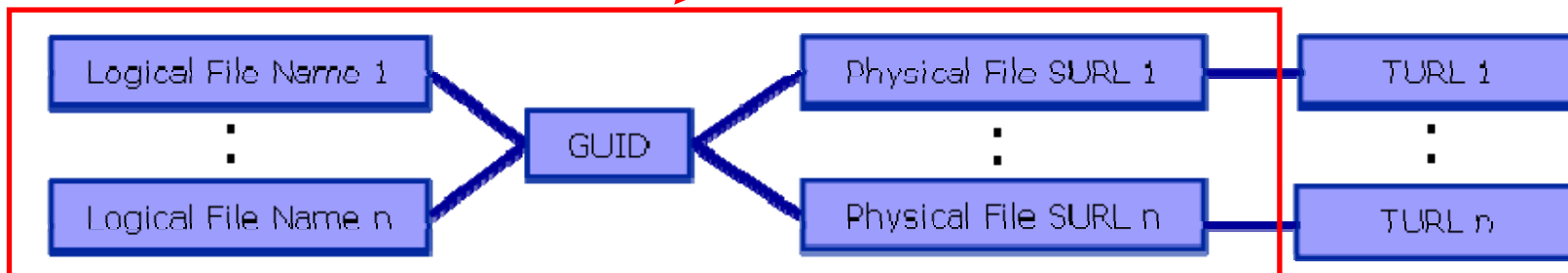
- **Logical File Name (LFN)**
  - An alias created by a user to refer to some item of data, e.g. “lfn:cms/20030203/run2/track1”
- **Globally Unique Identifier (GUID)**
  - A non-human-readable unique identifier for an item of data, e.g. “guid:f81d4fae-7dec-11d0-a765-00a0c91e6bf6”
- **Site URL (SURL) (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”



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



- Mapping by the “LFC” catalogue server





- **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 and by lcg-utils
- 
- **lcg-utils**
    - 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**

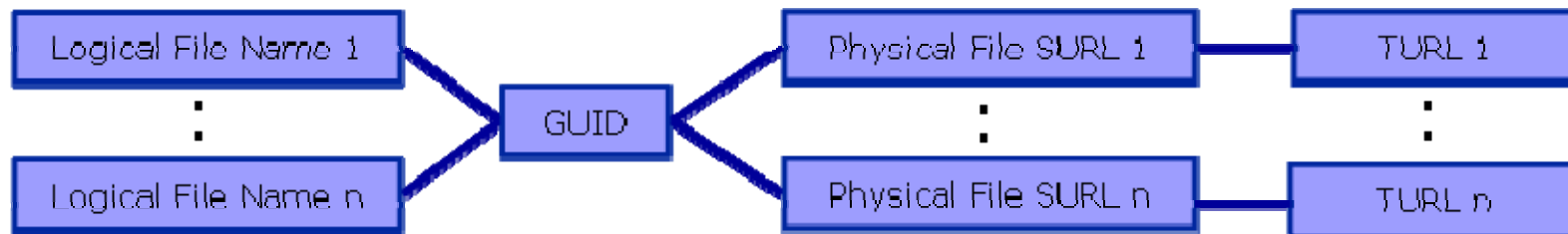
**Defined by the user**

- All members of a given VO have read-write permissions in their directory
- Commands look like UNIX with “lfc-” in front (often)
- We will use /grid/gilda/training/sofia/...

- **Provides**
  - Storage for files : massive storage system - disk or tape based
  - Transfer protocol (gsiFTP) ~ GSI based FTP server
  - POSIX-like file access
    - Grid File Access Layer (**GFAL**)
      - *API interface*
      - *To read parts of files too big to copy*
- **Two types**
  - “Classic” SE
    - Not implementing SRM
  - “SRM” SE
    - Storage Resource Manager
    - SEs are virtualised by common interface

- **FTS offer an important advance on client managed file transfers**
  - Support for third party transfer
  - Creation of channels set
- **FTS channel architecture offers very useful features to control transfers between sites or into a single site, though it may become overly complex in a grid without clear data flow patterns.**
  - The ability to control VO shares and transfer parameters on a channel is important for sites.
  - Improved reliability for transfers
  - Asynchronous file transfer mode → support to batch mode
- **FTS agent architecture allows VOs to connect the transfer service closely with their own data management stacks, a useful feature for HEP experiments.**
- **No catalogue interactions yet ☹ → users have to handle SURL**

- List directory
- Upload a file to an SE and register a logical name (lfn) in the catalog
- 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
  
- **Please go to the web page for this practical**



- **Go to the agenda page for this event as follows:**
  - Go to <http://www.egee.nesc.ac.uk/schedreg/>
  - Click on this event and bookmark it
- **The practical is led by a web page**
  - data management “more information”



- **Practical – from agenda page**



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

## 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>

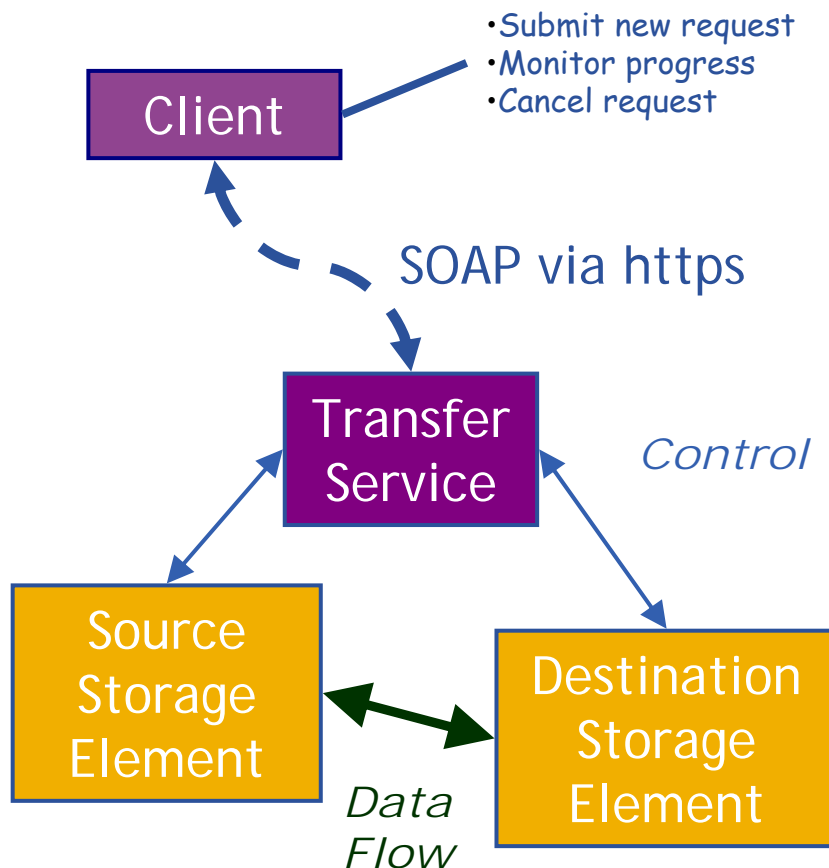
## FTS client

<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>

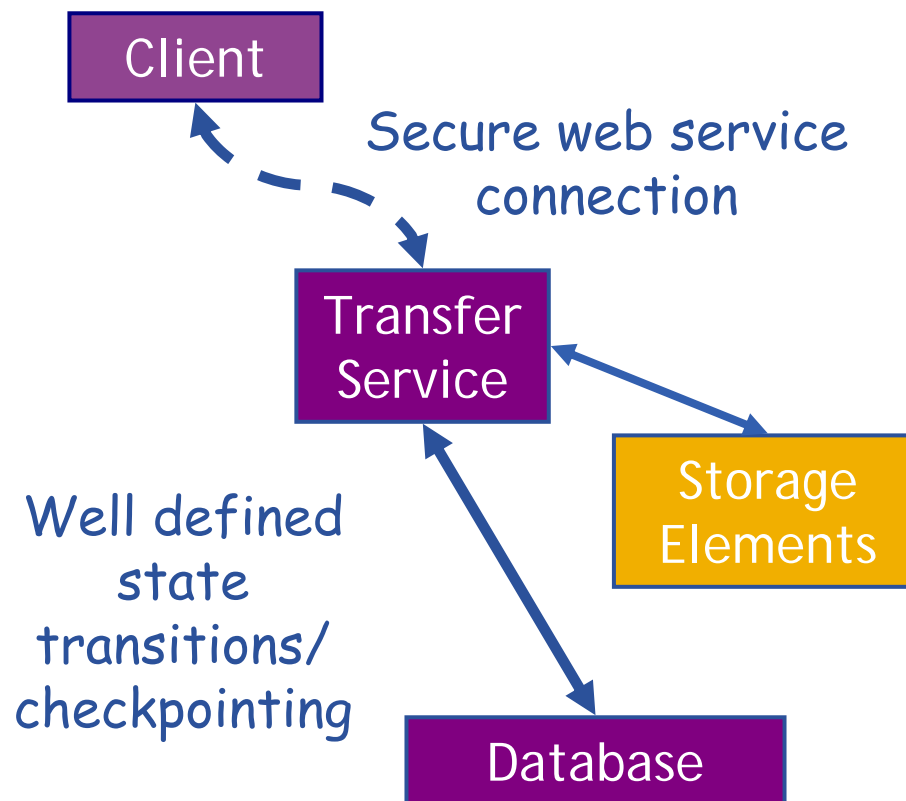
- FTS slides taken from EUChinagrid presentation given by *Yaodong Cheng*
- *IHEP, Chinese Academy of Sciences*
  
- *EUChinaGRID tutorial*
- *Beijing, 15-16 June 2006*
  
- <http://agenda.euchinagrid.org/fullAgenda.php?ida=a0621>

- **Clear need for a *service* for data transfer**

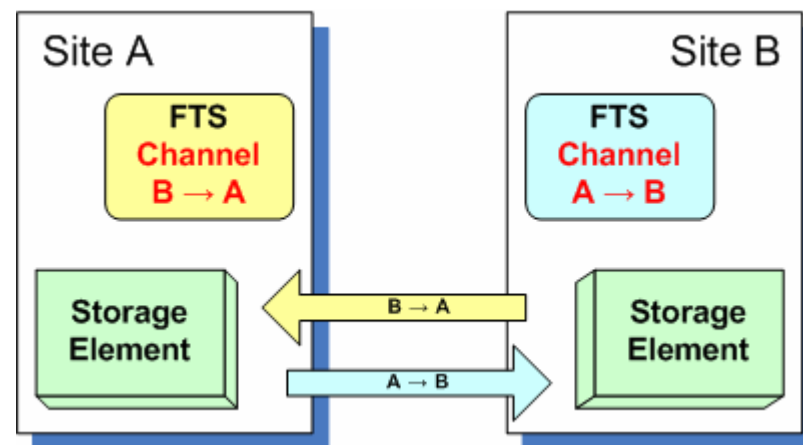
- Client connects to service to submit request
- Service maintains state about transfer
- Client can periodically reconnect to check status or cancel request
- Service can have knowledge of global state, not just a single request
  - Load balancing
  - Scheduling



- **Clear need of a *service* for (massive) data transfer**
  - Client connects to service to submit request
  - Service maintains state about transfer
  - Client can periodically reconnect to check status or cancel request
- **Jobs are lists of URLs in srm:// format. Some transfer parameters can be specified (streams, buffer sizes).**
- **Clients cannot subscribe for status changes, but can poll.**
- **C command line clients. C, Java and Perl APIs available.**
- **Web service runs in Tomcat5 container, agents runs as normal daemons.**



- FTS Service has a concept of *channels*
- A channel is a *unidirectional* connection between two sites
- Transfer requests between these two sites are assigned to that channel
- Channels usually correspond to a dedicated network pipe associated with production
- But channels can also take wildcards:
  - \* to MY\_SITE : All incoming
  - MY SITE to \* : All outgoing
  - \* to \* : Catch all



- Channels control certain transfer properties: transfer concurrency, gridftp streams.
- Channels can be controlled independently: started, stopped, drained.



- VO Agents
- Any job submitted to FTS is first handled by the VO agent
- VO agent authorises job and changes its state to “Pending”
- VO agents can perform other tasks – naturally these can be VO specific:
  - Scheduling
  - File catalog interaction

## Channel Agents

- Transfers on channel are managed by the channel agent
- Channel agents can perform inter-VO scheduling

