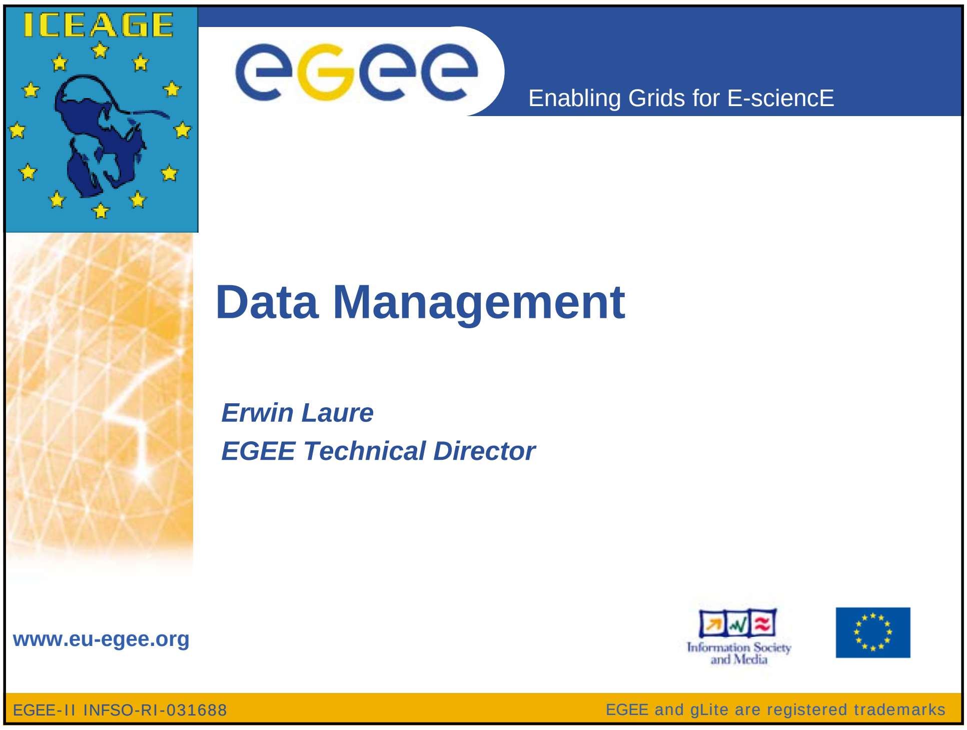
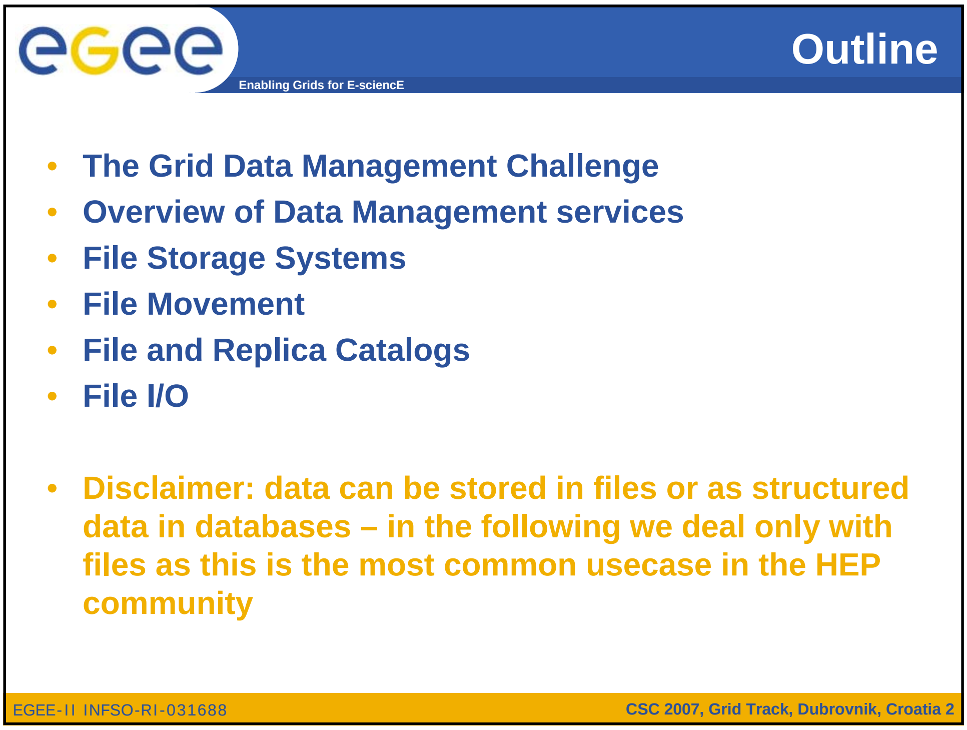


Enabling Grids for E-sciencE

Data Management

Erwin Laure
EGEE Technical Director

www.eu-egee.org

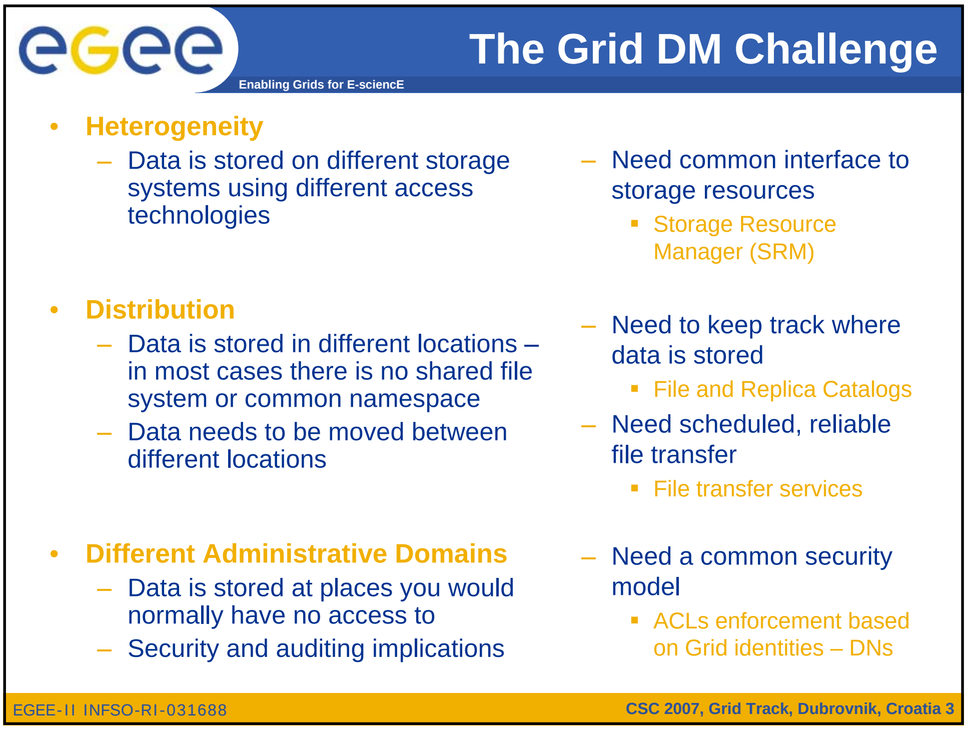


Enabling Grids for E-sciencE

Outline

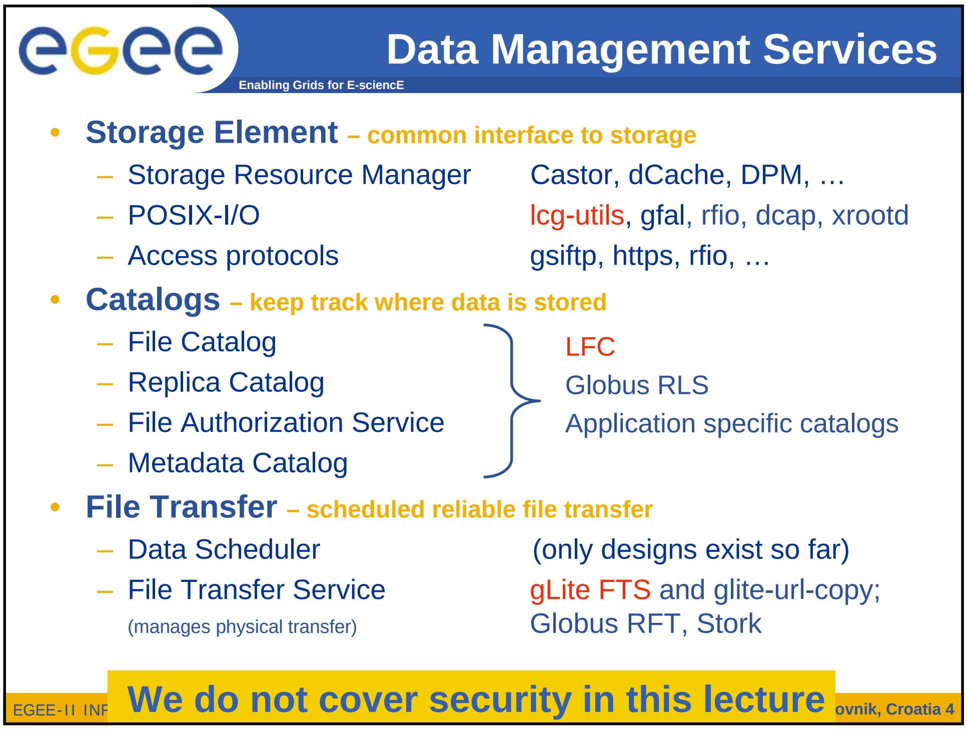
- The Grid Data Management Challenge
- Overview of Data Management services
- File Storage Systems
- File Movement
- File and Replica Catalogs
- File I/O

- Disclaimer: data can be stored in files or as structured data in databases – in the following we deal only with files as this is the most common usecase in the HEP community



The Grid DM Challenge

- **Heterogeneity**
  - Data is stored on different storage systems using different access technologies
  - Need common interface to storage resources
    - Storage Resource Manager (SRM)
- **Distribution**
  - Data is stored in different locations – in most cases there is no shared file system or common namespace
  - Data needs 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 services
- **Different Administrative Domains**
  - Data is stored at places you would normally have no access to
  - Security and auditing implications
  - Need a common security model
    - ACLs enforcement based on Grid identities – DNs



Data Management Services

- **Storage Element** – common interface to storage

– Storage Resource Manager	Castor, dCache, DPM, …
– POSIX-I/O	lcg-utils, gfal, rfio, dcap, xrootd
– Access protocols	gsiftp, https, rfio, …

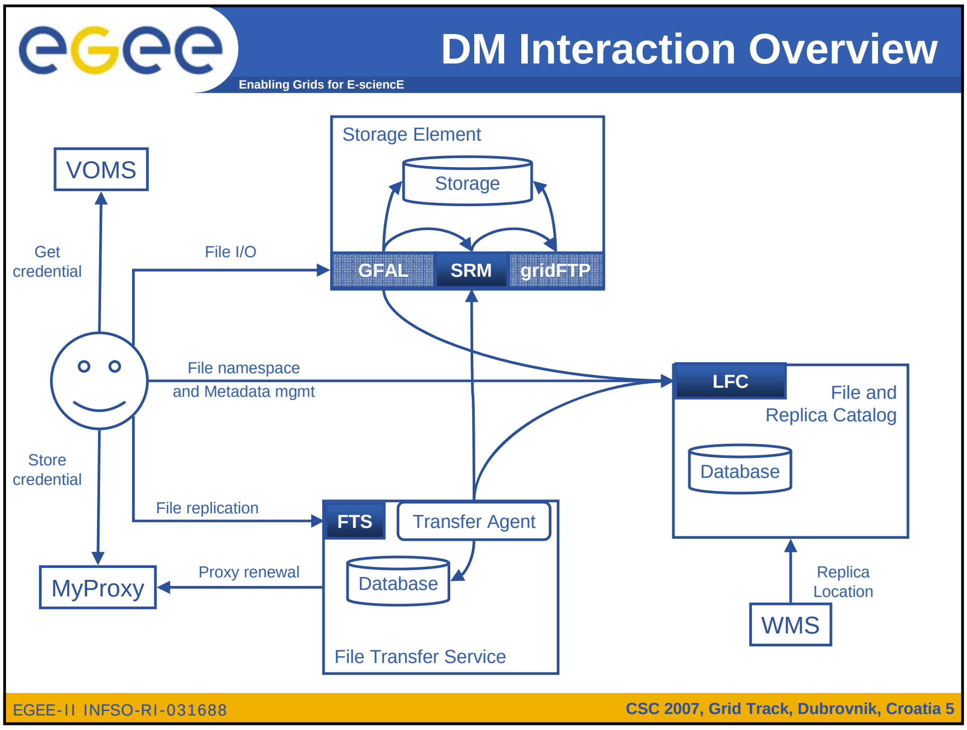
- **Catalogs** – keep track where data is stored
  - File Catalog
  - Replica Catalog
  - File Authorization Service
  - Metadata Catalog

LFC
Globus RLS
Application specific catalogs

- **File Transfer** – scheduled reliable file transfer

– Data Scheduler	(only designs exist so far)
– File Transfer Service (manages physical transfer)	gLite FTS and glite-url-copy; Globus RFT, Stork

We do not cover security in this lecture



Enabling Grids for E-sciencE

DM Interaction Overview

Storage Element

Storage

GFAL SRM gridFTP

VOMS

Get credential

File I/O

File namespace and Metadata mgmt

LFC

File and Replica Catalog

Database

Store credential

File replication

FTS

Transfer Agent

Database

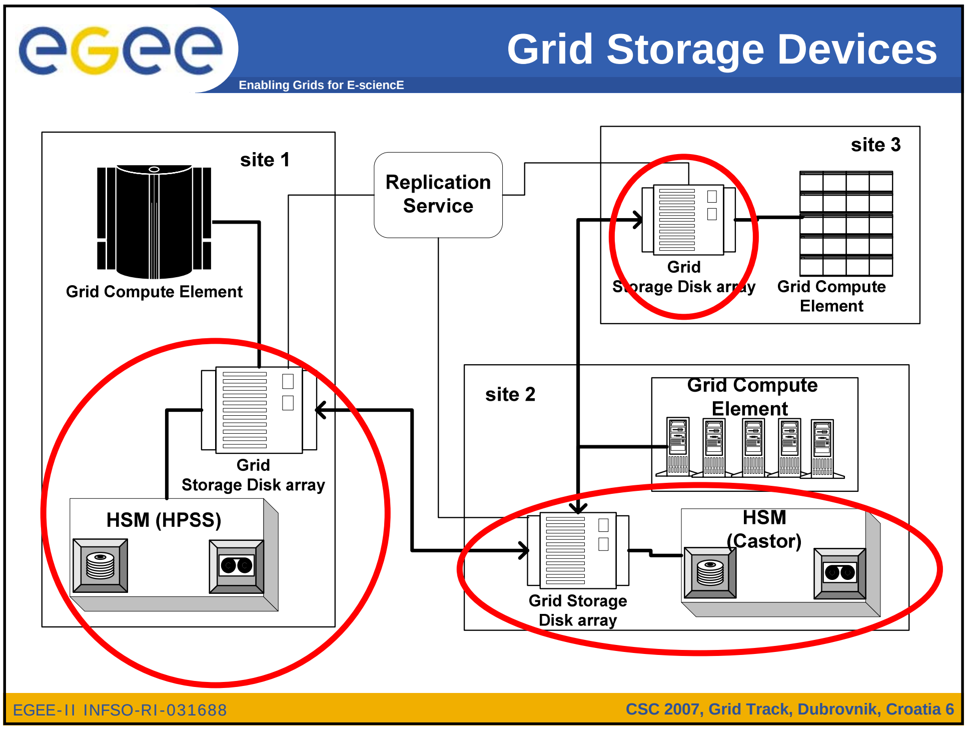
File Transfer Service

MyProxy

Proxy renewal

Replica Location

WMS



Enabling Grids for E-sciencE

Grid Storage Devices

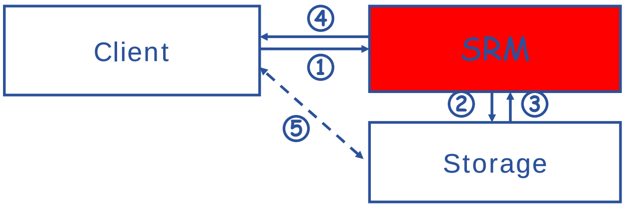
Grid Storage Requirements

- **Manage local storage and interface to Mass Storage Systems like**
  - HPSS, CASTOR, DiskeXtender (UNITREE), …
- **Provide a unique interface**
- **Support basic file transfer and access protocols**
  - GridFTP, FTP, POSIX-like…

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
  - 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 interaface to outside world
- Interactions with the SRM is typically hidden by higher level services – will not be exercised

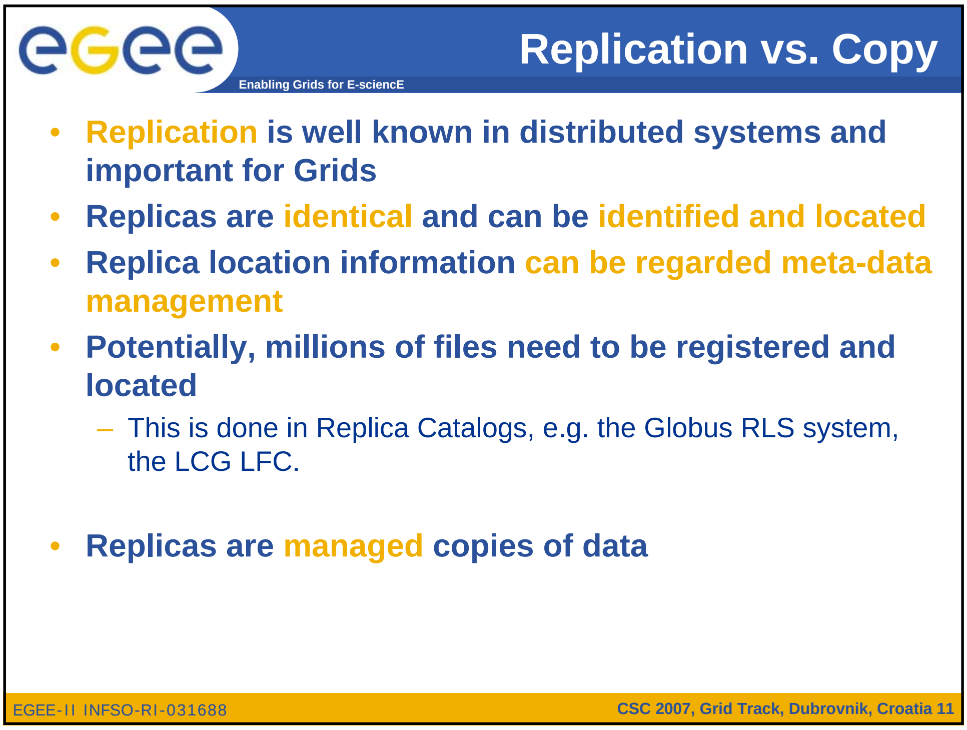
SRM Interactions



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

File Movement

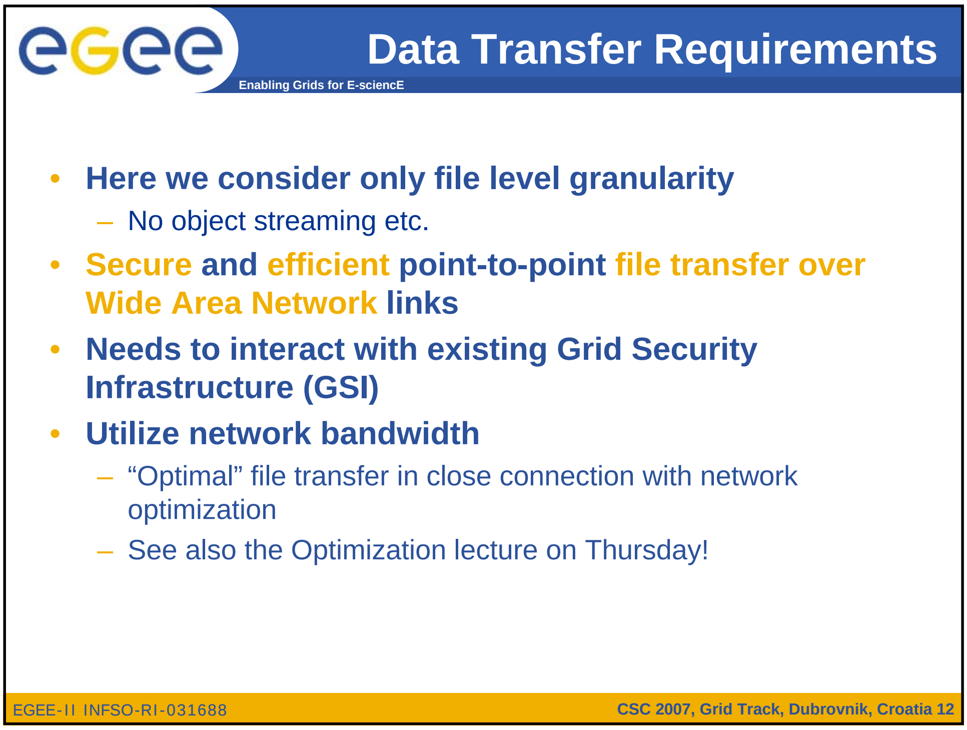
- The need for file movement
  - Data is produced at one place but needs to be analyzed at many places
    - E.g. LHC experiments
  - Data is produced at many places – needs to be combined for analysis
    - E.g. Astronomy, weather forecast, …
  - Not all computation can take place where data is originally stored
    - Better exploit available computational and storage resources
  - Having multiple copies of a file increases the availability of data and reduces the risk of data loss
    - In case of unavailability of one storage resource others may hold the data as well



Replication vs. Copy

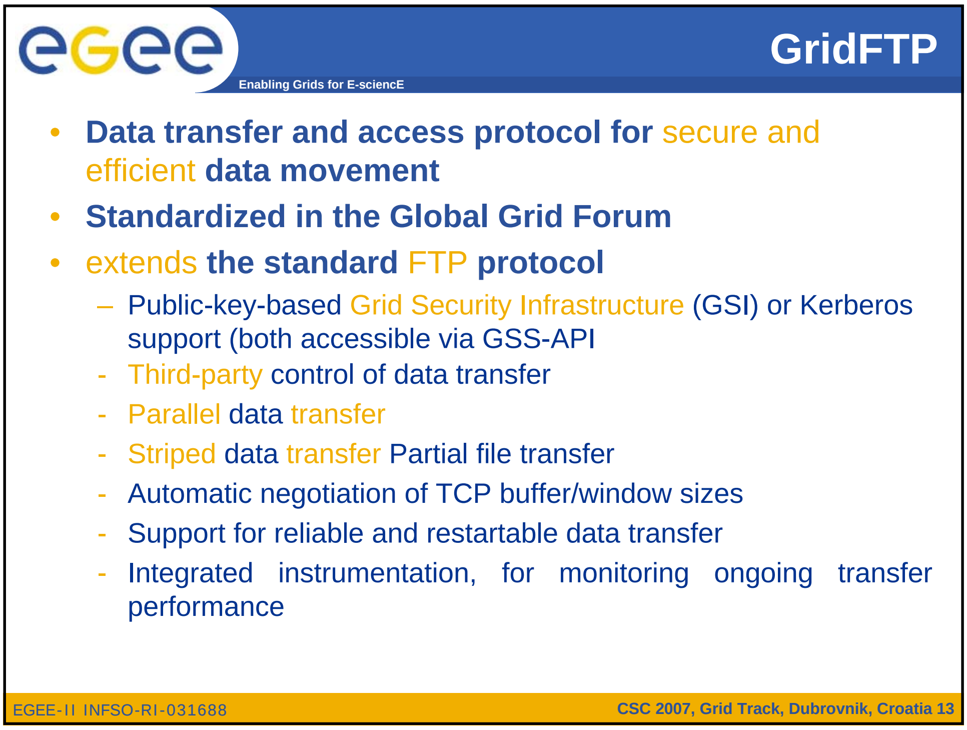
- Replication is well known in distributed systems and important for Grids
- Replicas are identical and can be identified and located
- Replica location information can be regarded meta-data management
- Potentially, millions of files need to be registered and located
  - This is done in Replica Catalogs, e.g. the Globus RLS system, the LCG LFC.

- Replicas are managed copies of data



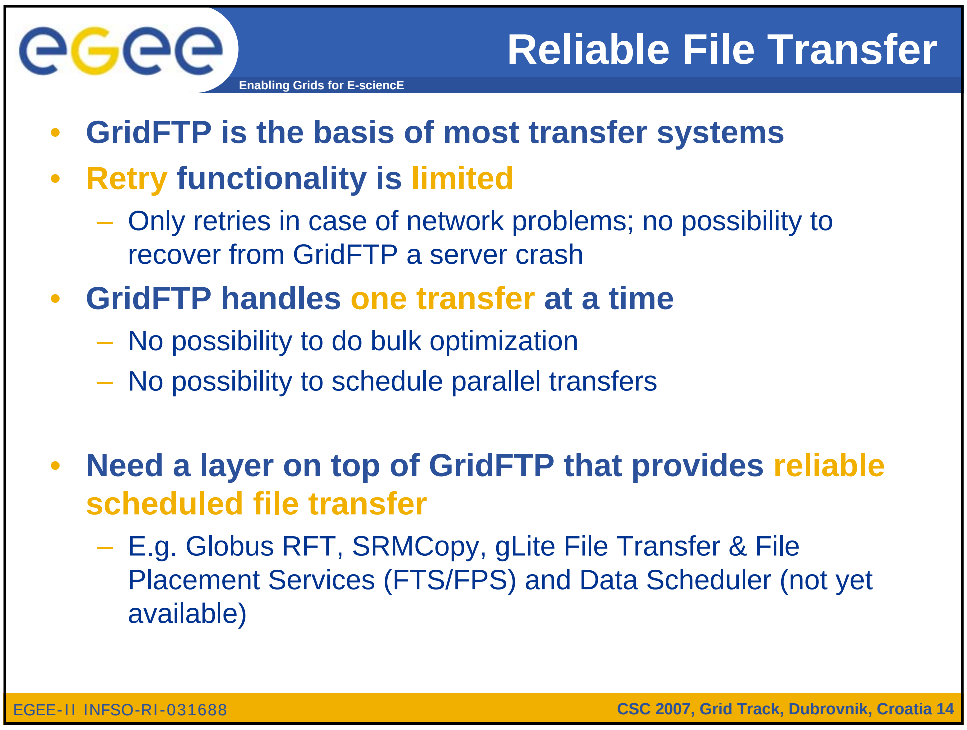
Data Transfer Requirements

- Here we consider only file level granularity
  - No object streaming etc.
- Secure and efficient point-to-point file transfer over Wide Area Network links
- Needs to interact with existing Grid Security Infrastructure (GSI)
- Utilize network bandwidth
  - “Optimal” file transfer in close connection with network optimization
  - See also the Optimization lecture on Thursday!



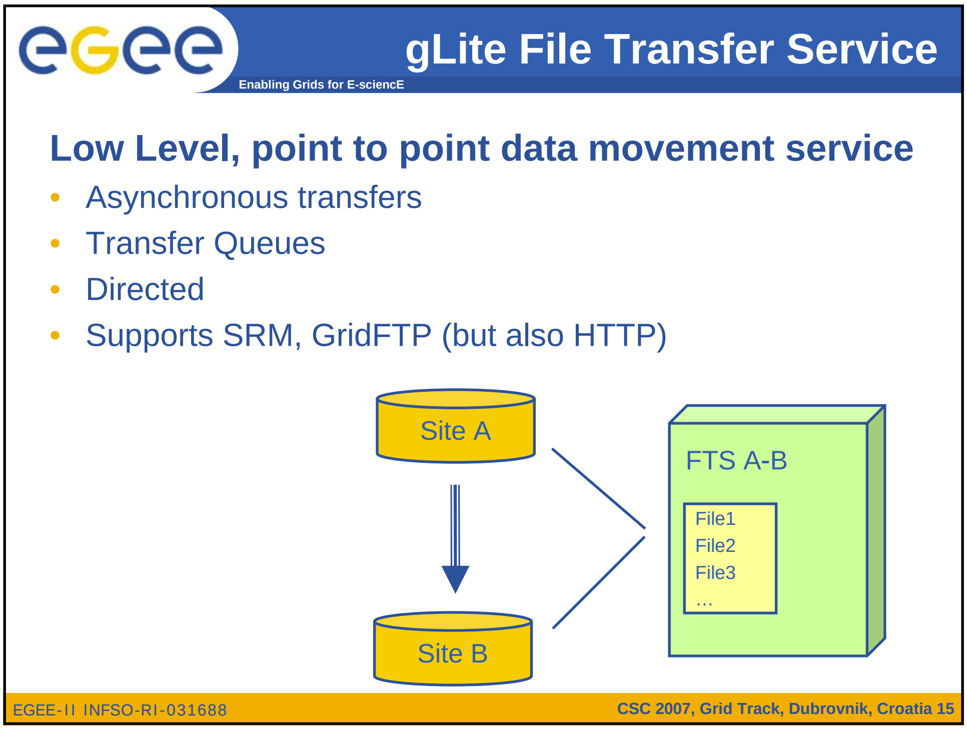
GridFTP

- 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
  - E.g. Globus RFT, SRMCopy, gLite File Transfer & File Placement Services (FTS/FPS) and Data Scheduler (not yet available)



gLite File Transfer Service

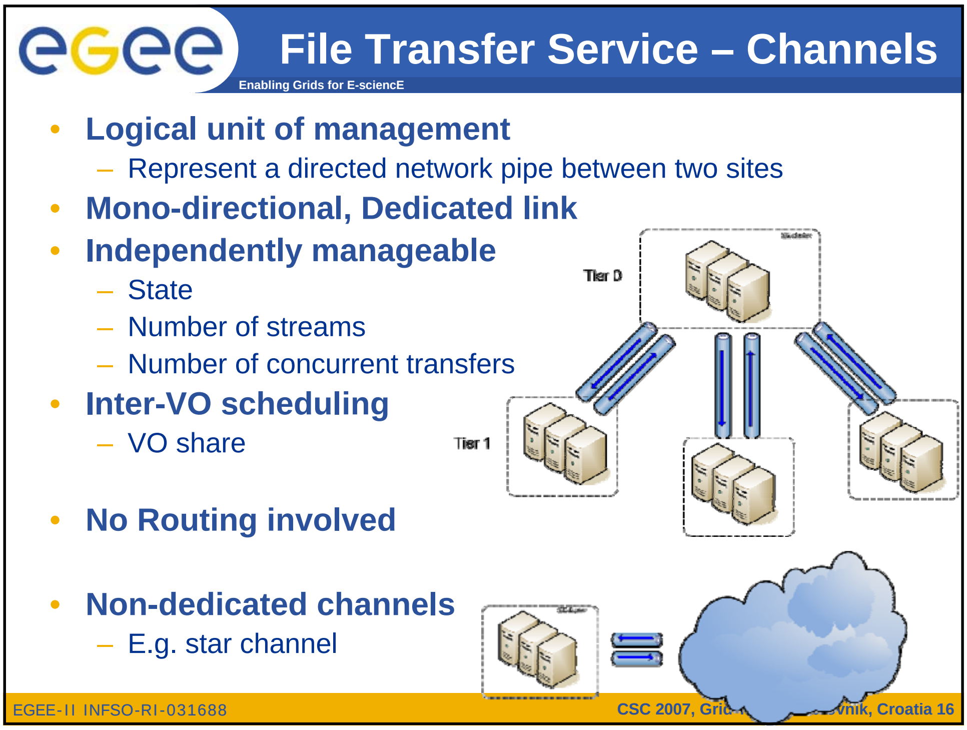
Enabling Grids for E-sciencE

Low Level, point to point data movement service

- Asynchronous transfers
- Transfer Queues
- Directed
- Supports SRM, GridFTP (but also HTTP)

Site A
Site B
FTS A-B
File1
File2
File3
…

EGEE-II INFSO-RI-031688 CSC 2007, Grid Track, Dubrovnik, Croatia 15



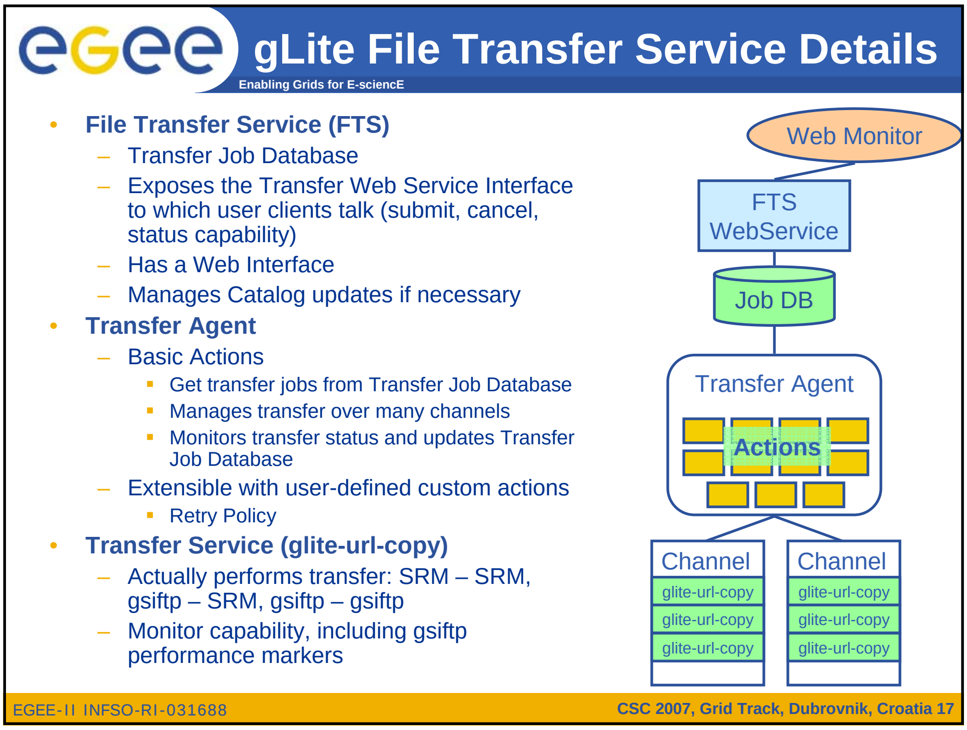
File Transfer Service – Channels

Enabling Grids for E-sciencE

- **Logical unit of management**
  - Represent a directed network pipe between two sites
- **Mono-directional, Dedicated link**
- **Independently manageable**
  - State
  - Number of streams
  - Number of concurrent transfers
- **Inter-VO scheduling**
  - VO share
- **No Routing involved**
- **Non-dedicated channels**
  - E.g. star channel

Tier 0
Tier 1

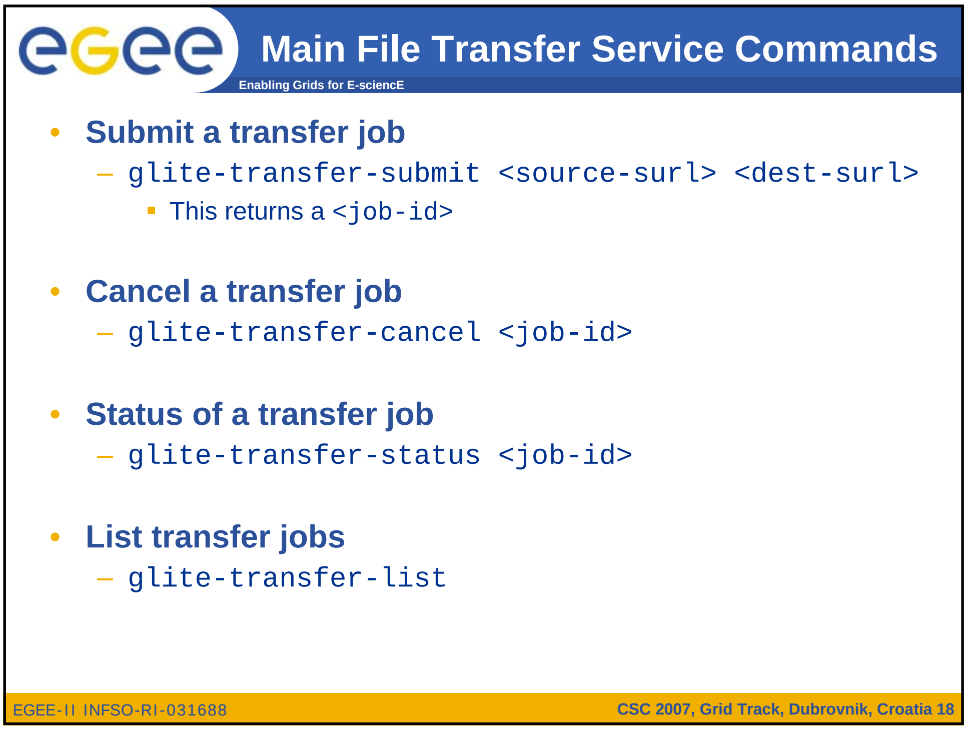
EGEE-II INFSO-RI-031688 CSC 2007, Gri vnik, Croatia 16



gLite File Transfer Service Details

Enabling Grids for E-sciencE

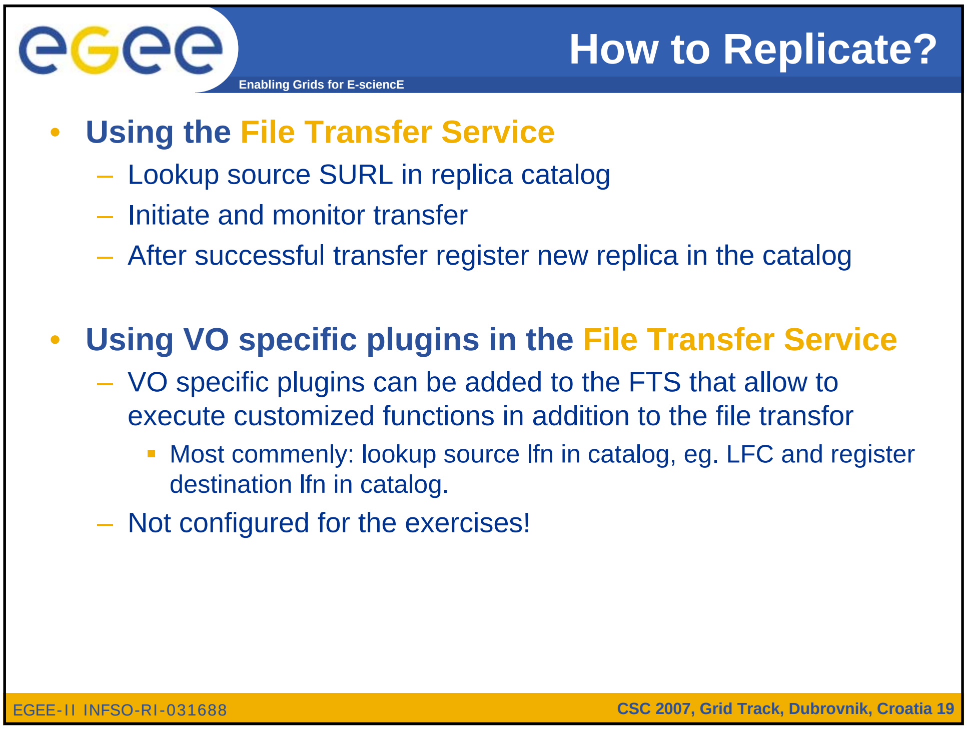
- **File Transfer Service (FTS)**
  - Transfer Job Database
  - Exposes the Transfer Web Service Interface to which user clients talk (submit, cancel, status capability)
  - Has a Web Interface
  - Manages Catalog updates if necessary
- **Transfer Agent**
  - Basic Actions
    - Get transfer jobs from Transfer Job Database
    - Manages transfer over many channels
    - Monitors transfer status and updates Transfer Job Database
  - Extensible with user-defined custom actions
    - Retry Policy
- **Transfer Service (glite-url-copy)**
  - Actually performs transfer: SRM – SRM, gsiftp – SRM, gsiftp – gsiftp
  - Monitor capability, including gsiftp performance markers



Main File Transfer Service Commands

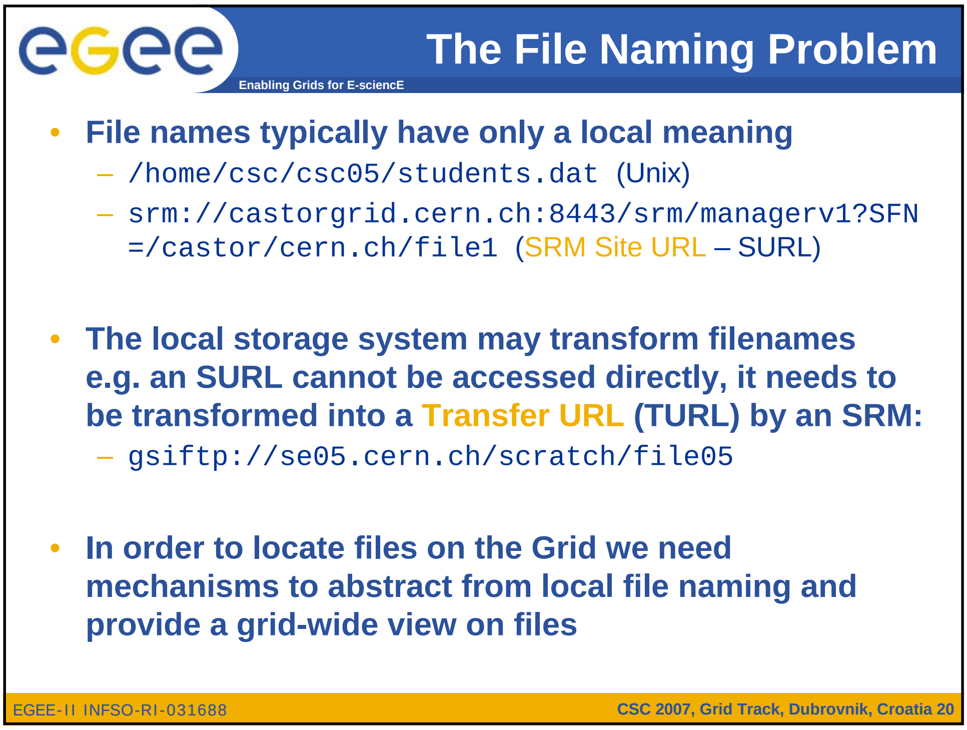
Enabling Grids for E-sciencE

- **Submit a transfer job**
  - `glite-transfer-submit <source-surl> <dest-surl>`
    - This returns a `<job-id>`
- **Cancel a transfer job**
  - `glite-transfer-cancel <job-id>`
- **Status of a transfer job**
  - `glite-transfer-status <job-id>`
- **List transfer jobs**
  - `glite-transfer-list`



How to Replicate?

- **Using the File Transfer Service**
  - Lookup source SURL in replica catalog
  - Initiate and monitor transfer
  - After successful transfer register new replica in the catalog
- **Using VO specific plugins in the File Transfer Service**
  - VO specific plugins can be added to the FTS that allow to execute customized functions in addition to the file transfor
    - Most commenly: lookup source lfn in catalog, eg. LFC and register destination lfn in catalog.
  - Not configured for the exercises!



The File Naming Problem

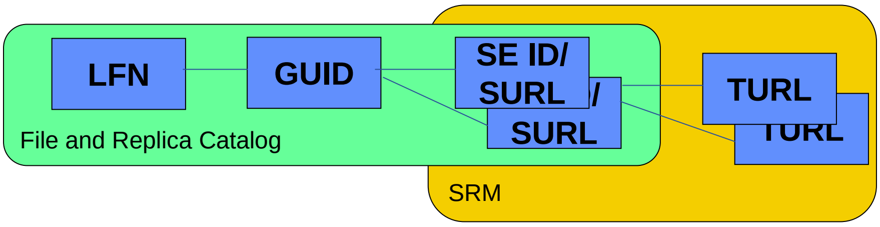
- **File names typically have only a local meaning**
  - `/home/csc/csc05/students.dat` (Unix)
  - `srm://castorgrid.cern.ch:8443/srm/managerv1?SFN=/castor/cern.ch/file1` (SRM Site URL – SURL)
- **The local storage system may transform filenames e.g. an SURL cannot be accessed directly, it needs to be transformed into a Transfer URL (TURL) by an SRM:**
  - `gsiftp://se05.cern.ch/scratch/file05`
- **In order to locate files on the Grid we need mechanisms to abstract from local file naming and provide a grid-wide view on files**

The File Naming Problem II

- Logical File Names (LFN)
  - Provide a human readable identifier for files on the Grid level
  - Can be arbitrary URIs
  - Need to be unique
  - `lfn:/grid/myVO.org/production/run/07/123456/calibration/cal/cal-table100`
- Global Unique Identifiers (GUID)
  - LFNs may be created in a distributed fashion – hard to guarantee uniqueness
  - Assigning a GUID to each file when it is created allows to always uniquely identify it and thus conflict resolution in LFNs
  - Drawback: not human readable
  - `004c3326-0daf-126d-87f9-898a04b4beef`

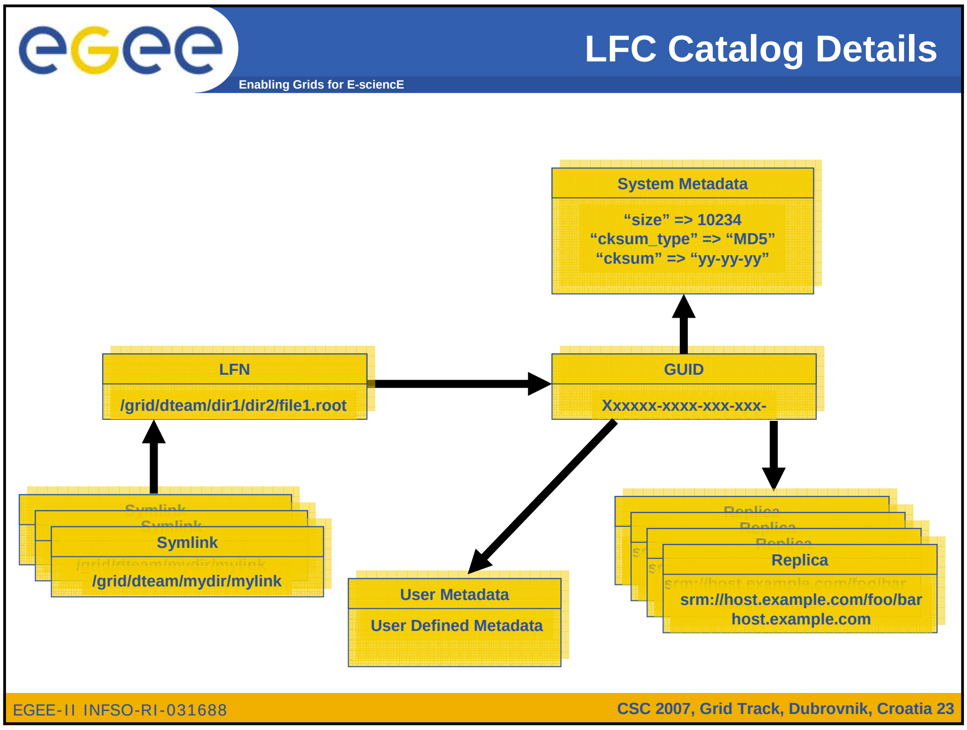
File and Replica Catalogs

- **Allow to find where files are stored on the Grid**
- **May implement additional semantics in the LFN namespace (directories, ACLs, …)**
- **Allow to locate the location of replicas (i.e. identical copies of files)**



LFN – Logical File Name
GUID – Global Unique Identifier

SURL – Site URL
TURL – Transfer URL



LFC Catalog Details

Enabling Grids for E-sciencE

System Metadata

"size" => 10234
"cksum_type" => "MD5"
"cksum" => "yy-yy-yy"

LFN

/grid/dteam/dir1/dir2/file1.root

GUID

Xxxxxx-xxxx-xxx-xxx-

Symlink

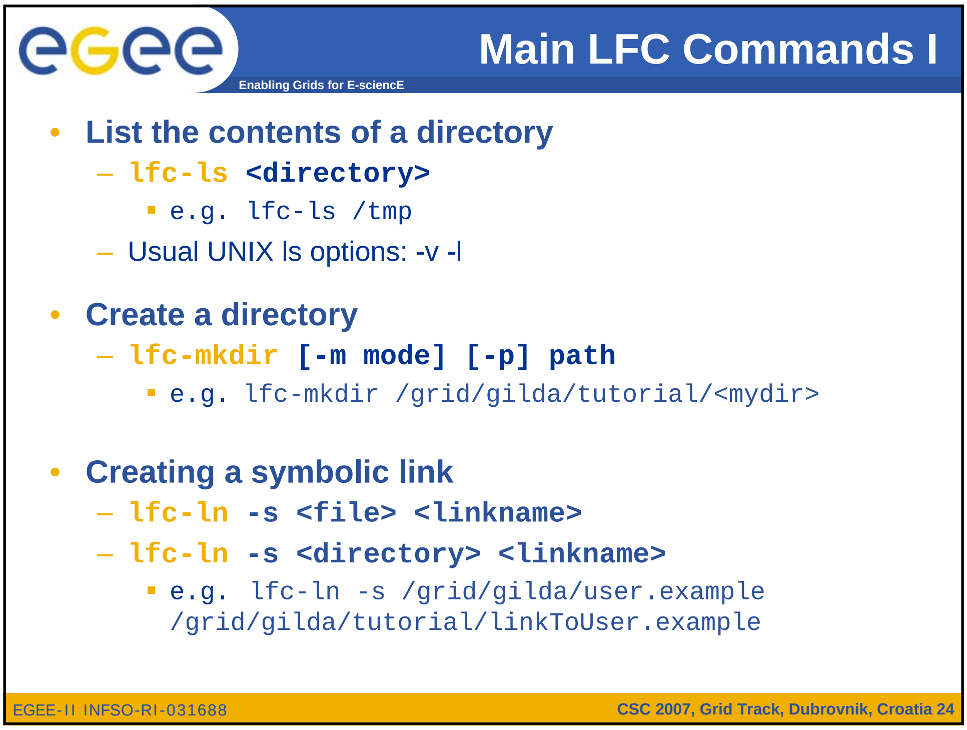
/grid/dteam/mydir/mylink

User Metadata

User Defined Metadata

Replica

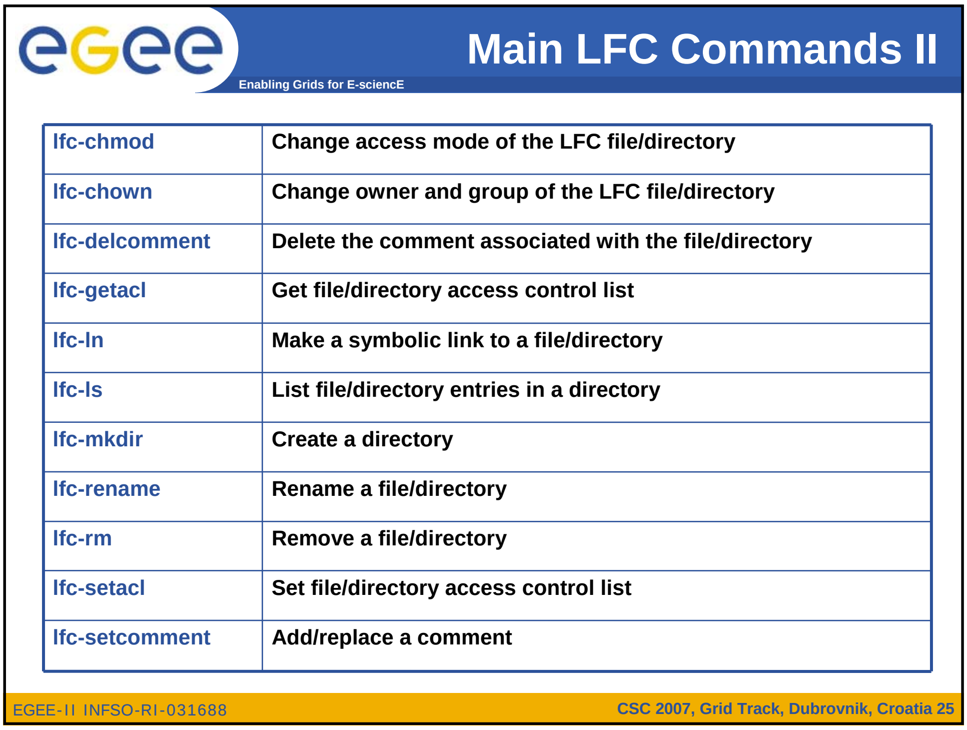
srm://host.example.com/foo/bar
host.example.com



Main LFC Commands I

Enabling Grids for E-sciencE

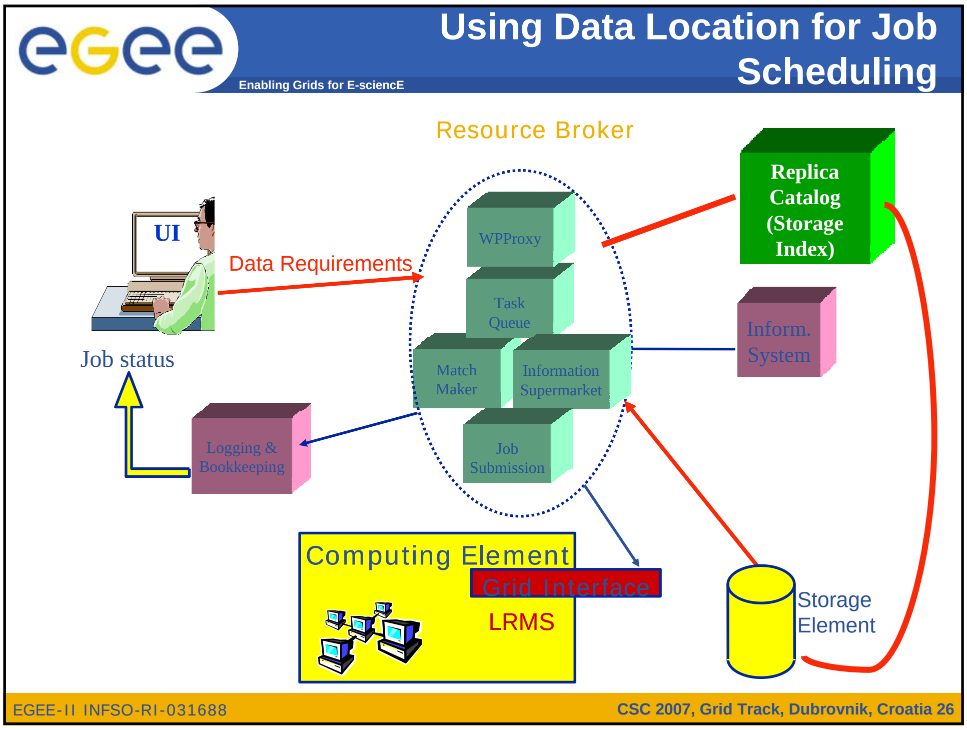
- List the contents of a directory
  - `lfc-ls <directory>`
    - e.g. `lfc-ls /tmp`
  - Usual UNIX ls options: -v -l
- Create a directory
  - `lfc-mkdir [-m mode] [-p] path`
    - e.g. `lfc-mkdir /grid/gilda/tutorial/<mydir>`
- Creating a symbolic link
  - `lfc-ln -s <file> <linkname>`
  - `lfc-ln -s <directory> <linkname>`
    - e.g. `lfc-ln -s /grid/gilda/user.example /grid/gilda/tutorial/linkToUser.example`



Enabling Grids for E-sciencE

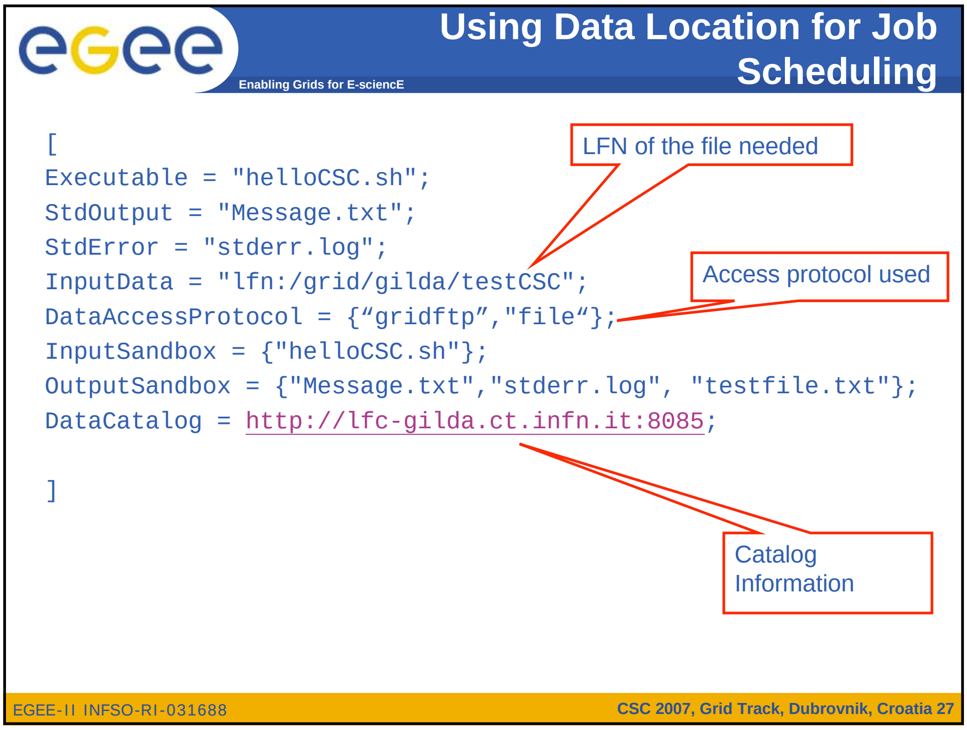
Main LFC Commands II

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 list
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 list
lfc-setcomment	Add/replace a comment



Enabling Grids for E-sciencE

Using Data Location for Job Scheduling



Using Data Location for Job Scheduling

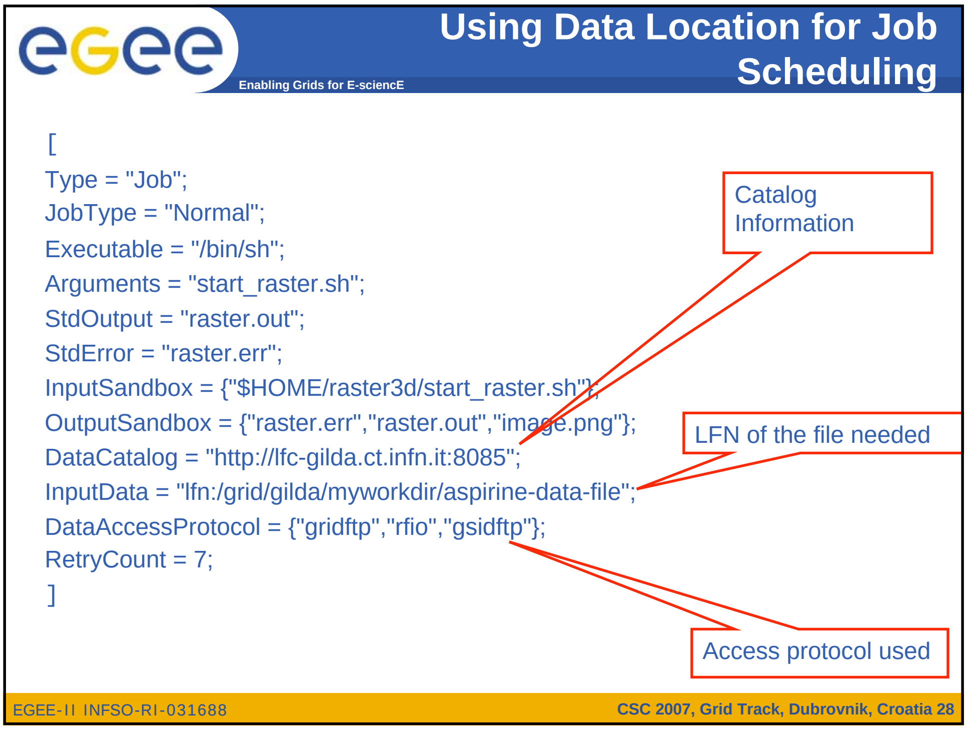
```
[
Executable = "helloCSC.sh";
StdOutput = "Message.txt";
StdError = "stderr.log";
InputData = "lfn:/grid/gilda/testCSC";
DataAccessProtocol = {“gridftp”,"file“};
InputSandbox = {"helloCSC.sh"};
OutputSandbox = {"Message.txt","stderr.log", "testfile.txt"};
DataCatalog = http://lfc-gilda.ct.infn.it:8085;

]
```

LFN of the file needed

Access protocol used

Catalog Information



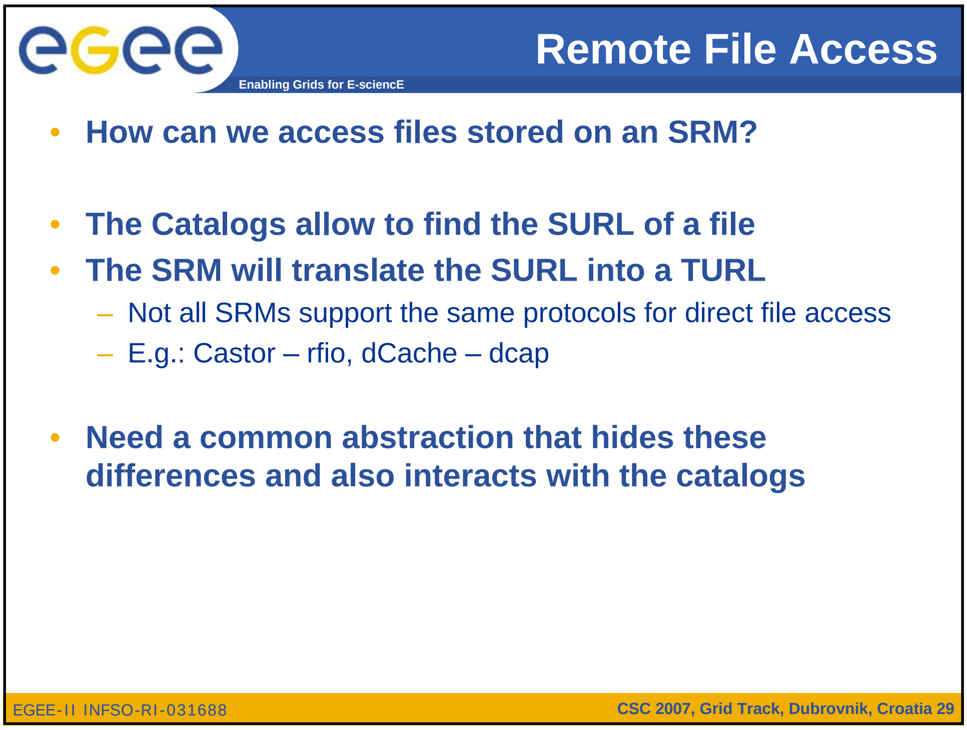
Using Data Location for Job Scheduling

```
[
Type = "Job";
JobType = "Normal";
Executable = "/bin/sh";
Arguments = "start_raster.sh";
StdOutput = "raster.out";
StdError = "raster.err";
InputSandbox = {"$HOME/raster3d/start_raster.sh"};
OutputSandbox = {"raster.err","raster.out","image.png"};
DataCatalog = "http://lfc-gilda.ct.infn.it:8085";
InputData = "lfn:/grid/gilda/myworkdir/aspirine-data-file";
DataAccessProtocol = {"gridftp","rfio","gsidftp"};
RetryCount = 7;
]
```

Catalog Information

LFN of the file needed

Access protocol used

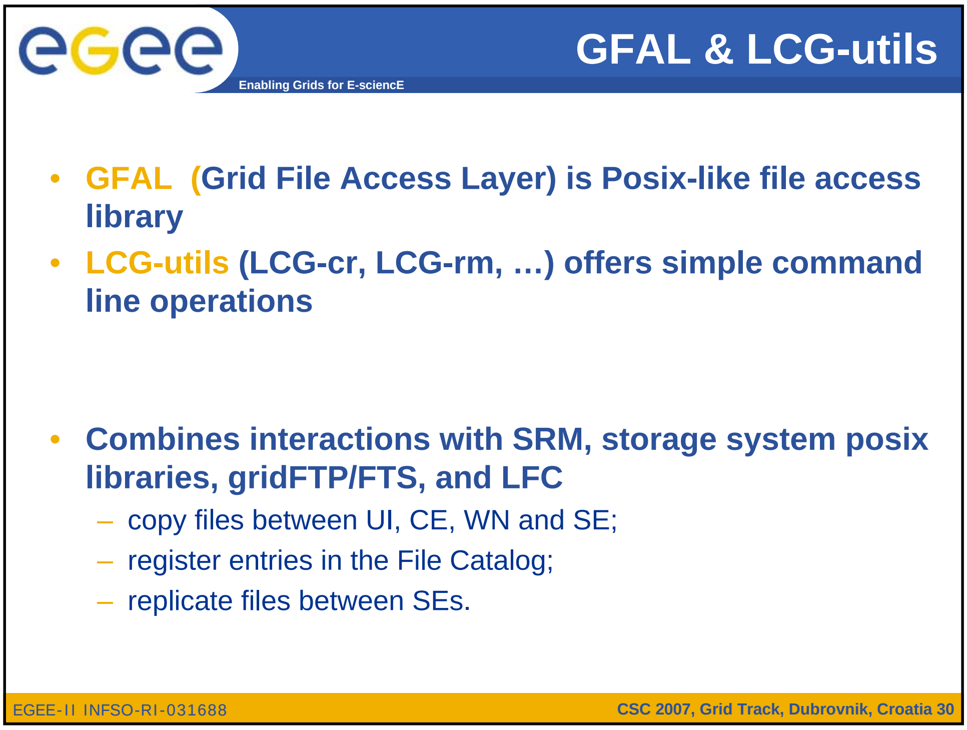


Remote File Access

- **How can we access files stored on an SRM?**

- **The Catalogs allow to find the SURL of a file**
- **The SRM will translate the SURL into a TURL**
  - Not all SRMs support the same protocols for direct file access
  - E.g.: Castor – rfio, dCache – dcap

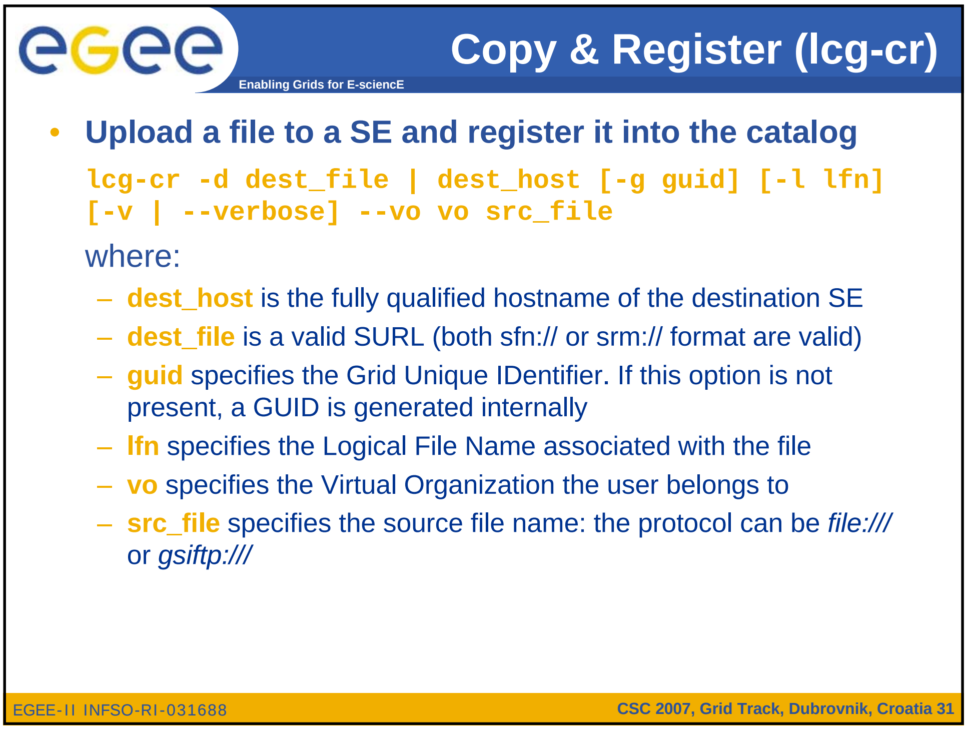
- **Need a common abstraction that hides these differences and also interacts with the catalogs**



GFAL & LCG-utils

- **GFAL (Grid File Access Layer) is Posix-like file access library**
- **LCG-utils (LCG-cr, LCG-rm, …) offers simple command line operations**

- **Combines interactions with SRM, storage system posix libraries, gridFTP/FTS, and LFC**
  - copy files between UI, CE, WN and SE;
  - register entries in the File Catalog;
  - replicate files between SEs.



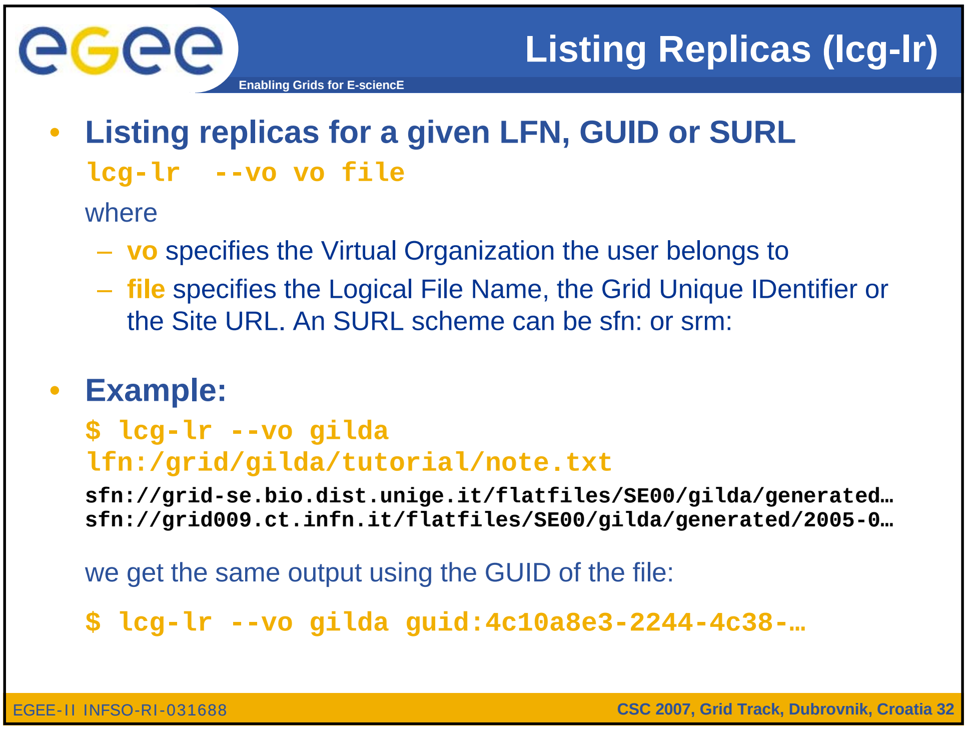
Copy & Register (lcg-cr)

- **Upload a file to a SE and register it into the catalog**

```
lcg-cr -d dest_file | dest_host [-g guid] [-l lfn]
[-v | --verbose] --vo vo src_file
```

where:

- **dest_host** is the fully qualified hostname of the destination SE
- **dest_file** is a valid SURL (both sfn:// or srm:// format are valid)
- **guid** specifies the Grid Unique IDentifier. If this option is not present, a GUID is generated internally
- **lfn** specifies the Logical File Name associated with the file
- **vo** specifies the Virtual Organization the user belongs to
- **src_file** specifies the source file name: the protocol can be *file:///* or *gsiftp:///*



Listing Replicas (lcg-lr)

- **Listing replicas for a given LFN, GUID or SURL**

```
lcg-lr  --vo vo file
```

where

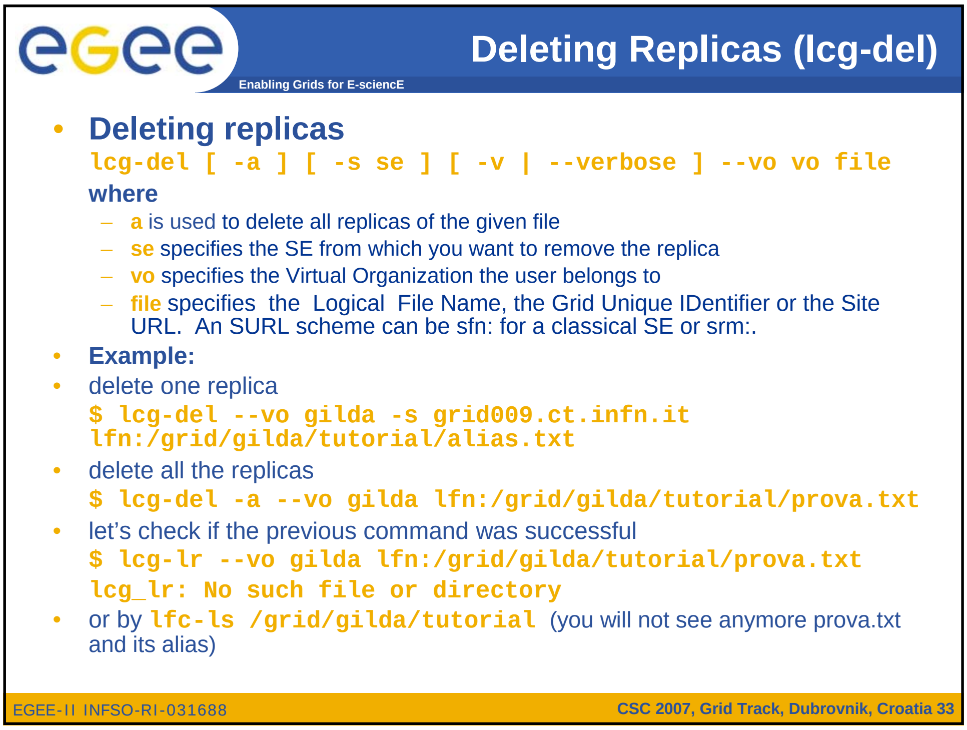
- **vo** specifies the Virtual Organization the user belongs to
- **file** specifies the Logical File Name, the Grid Unique IDentifier or the Site URL. An SURL scheme can be sfn: or srm:

- **Example:**

```
$ lcg-lr --vo gilda
lfn:/grid/gilda/tutorial/note.txt
sfn://grid-se.bio.dist.unige.it/flatfiles/SE00/gilda/generated…
sfn://grid009.ct.infn.it/flatfiles/SE00/gilda/generated/2005-0…
```

we get the same output using the GUID of the file:

```
$ lcg-lr --vo gilda guid:4c10a8e3-2244-4c38-…
```



Deleting Replicas (lcg-del)

- **Deleting replicas**

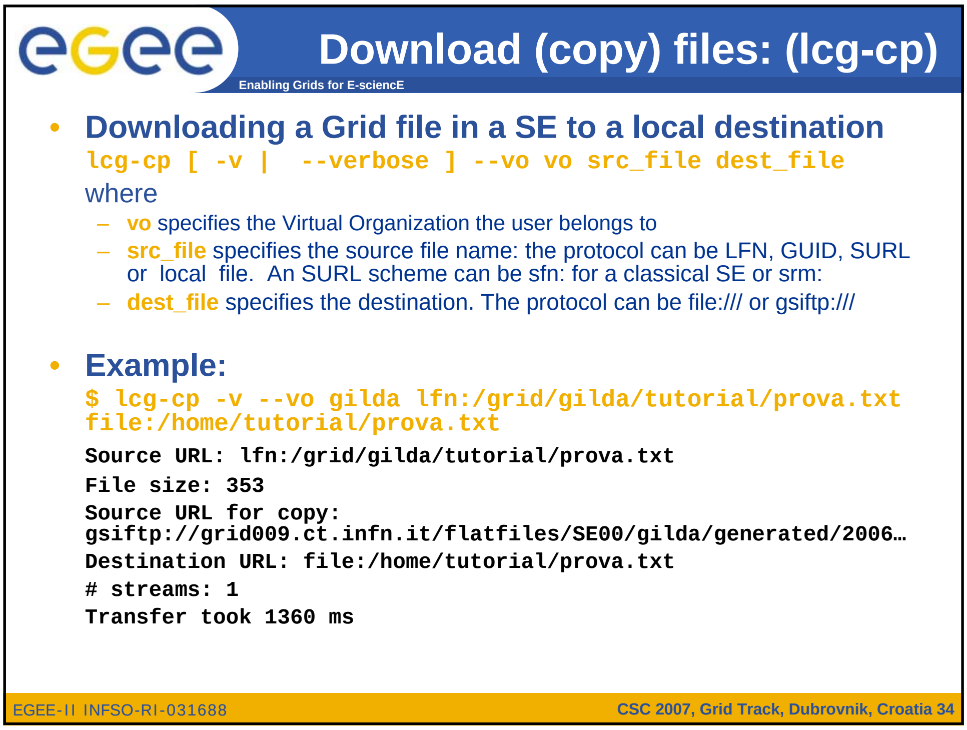
  ```
  lcg-del [ -a ] [ -s se ] [ -v | --verbose ] --vo vo file
  ```

  where
  - **a** is used to delete all replicas of the given file
  - **se** specifies the SE from which you want to remove the replica
  - **vo** specifies the Virtual Organization the user belongs to
  - **file** specifies the Logical File Name, the Grid Unique IDentifier or the Site URL. An SURL scheme can be sfn: for a classical SE or srm:.
- **Example:**
- delete one replica

  ```
  $ lcg-del --vo gilda -s grid009.ct.infn.it
  lfn:/grid/gilda/tutorial/alias.txt
  ```
- delete all the replicas

  ```
  $ lcg-del -a --vo gilda lfn:/grid/gilda/tutorial/prova.txt
  ```
- let's check if the previous command was successful

  ```
  $ lcg-lr --vo gilda lfn:/grid/gilda/tutorial/prova.txt
  lcg_lr: No such file or directory
  ```
- or by `lfc-ls /grid/gilda/tutorial` (you will not see anymore prova.txt and its alias)



Download (copy) files: (lcg-cp)

- **Downloading a Grid file in a SE to a local destination**

  ```
  lcg-cp [ -v |  --verbose ] --vo vo src_file dest_file
  ```

  where
  - **vo** specifies the Virtual Organization the user belongs to
  - **src_file** specifies the source file name: the protocol can be LFN, GUID, SURL or local file. An SURL scheme can be sfn: for a classical SE or srm:
  - **dest_file** specifies the destination. The protocol can be file:/// or gsiftp:///

- **Example:**

  ```
  $ lcg-cp -v --vo gilda lfn:/grid/gilda/tutorial/prova.txt
  file:/home/tutorial/prova.txt
  ```

  ```
  Source URL: lfn:/grid/gilda/tutorial/prova.txt
  File size: 353
  Source URL for copy:
  gsiftp://grid009.ct.infn.it/flatfiles/SE00/gilda/generated/2006…
  Destination URL: file:/home/tutorial/prova.txt
  # streams: 1
  Transfer took 1360 ms
  ```

Alias management (lcg-aa, lcg-la)

- **Adding an alias for a given GUID**

  ```
  lcg-aa --vo vo guid lfn
  ```

  where
  - vo specifies the Virtual Organization the user belongs to
  - guid specifies the Grid Unique Identifier of the file you want to add the alias to
  - lfn specifies the new alias
- Example:

  ```
  $ lcg-aa --vo gilda guid:bf95f82e-de21-4452-a4b5-
  b9d40a94ee2c lfn:/grid/gilda/tutorial/alias.txt
  ```

- To check if the previous command was successful, you can use lcg-la command to list the aliases for a given LFN, GUID or SURL:

  ```
  $ lcg-la --vo gilda lfn:/grid/gilda/tutorial/alias.txt
  lfn:/grid/gilda/tutorial/prova.txt
  lfn:/grid/gilda/tutorial/alias.txt
  ```

Summary of lcg-utils commands

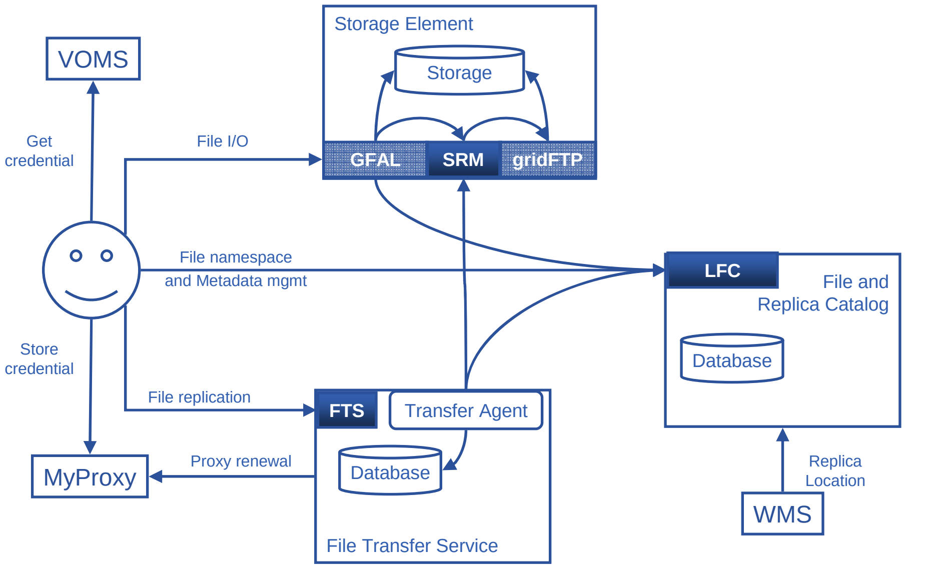
Replica Management

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

Data Management Summary



More Information

- **gLite homepage**
  - http://www.glite.org
- **gLite documentation**
  - http://www.glite.org/documentation