

# ELFI

A Linux filesystem interface to EGEE Grid storage

Antonio Messina <antonio.messina@ictp.it>,  
**Riccardo Murri** <riccardo.murri@ictp.it>

The Abdus Salam International Centre for Theoretical Physics, Trieste

V INFNGrid Workshop  
Padova, December 18–20, 2006

## EGRID VO requirements

EGRID is a MIUR project to build a pilot Grid infrastructure for research in Economics and Financial Statistics.

Built on top of INFNGrid, but security requirements are *stricter than most HEP VOs!*

- Privacy of financial data
- Different access rights among users of the same VO

Custom add-ons to gLite middleware: fine-grained ACL-capable security architecture.

- StoRM, secure SRMv2 server (co-developed with INFN)
- ECAR, a SOAP interface to LFC ACLs
- ELFI, a filesystem interface to Grid storage

## What is ELFI?

ELFI is Linux filesystem interface to EGEE Logical Filename Catalog (LFC) and Storage Elements (SE).

Most usual Grid operations on the catalog or the SE have a local filesystem equivalent.

## What is ELFI?

ELFI is Linux filesystem interface to EGEE Logical Filename Catalog (LFC) and Storage Elements (SE).

Most usual Grid operations on the catalog or the SE have a local filesystem equivalent.

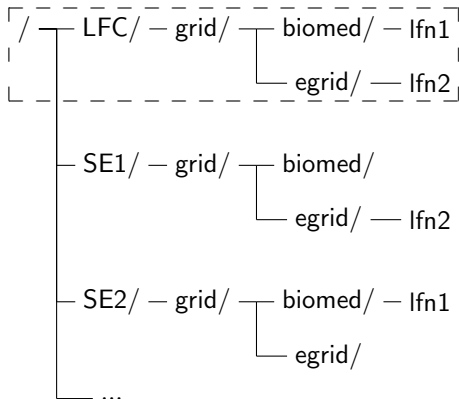
Built to meet EGRID requirements:

- User-friendly access to Grid data
- Transparent access to files for legacy applications
- Secure access to fine-grained inter-VO access control

## It's a real Linux filesystem

- compatible** Standard POSIX API: no need to recompile to access Grid-stored data. *Any Linux application can access Grid storage.*
- abstract** Shields client from knowing the details of Grid access; may change underlying protocols without affecting filesystem clients.
- user-friendly** Hides protocol details from users.

## Directory hierarchy

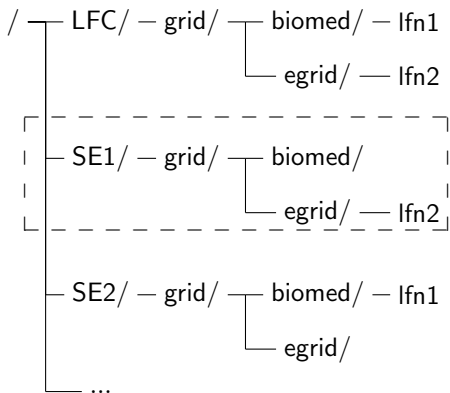


### LFC tree

The LFC tree mirrors the Logical File Name hierarchy. You refer to files by their LFN only.

- Metadata operations (ls, mv, chmod, get/set ACLs, ...) operate on the LFC catalog.
- Data read/write operations operate on an SE, which is set by user or randomly chosen by ELFI.

# Directory hierarchy



## SE tree

These directories are for operating on a specific replica.

- In a SE tree, you will only see LFNs of replicas that actually exist on that SE. You do not see the SE physical file layout.
- There is a directory for each SE that accepts your VO and whose access and control protocols ELFI can speak.

# Filesystem operations

- directory listing** Lists contents of corresponding LFC directory.
- read/write data** Reads/writes data to file replica via GSI-RFIO protocol. *Only downloads/uploads the portion of file data that is requested.*
- get/set ACL** Get/set ACL on corresponding LFC entry.
- remove file** *In LFC tree:* remove a LFN and *all* of its replicas.  
*In SE tree:* remove replica residing on that SE.
- mkdir/rmdir** Creates/removes directory on LFC.

## ELFI in EGRID's web portal

ELFI powers the EGRID web file management portlet.

- in VO production use since early 2006.
- Any logged in user has his/her own *separate* ELFI mount point.
- Minor modification of GridSphere's FileManagement standard portlet.
- Access Grid storage through *local* filesystem.
- Integrates GridSphere/P-GRADE into EGRID's security architecture.

ELFI could have Grid-enabled *any* web portal!

## ELFI usage example

<code>ls -l elfi/LFC/logical/file/name</code>	get info about an LFN
<code>vi elfi/LFC/logical/file/name</code>	directly access LFN contents for reading/writing
<code>cp local/file/name \</code> <code>elfi/LFC/logical/file/name</code>	copy a local file into Grid storage and register it under a given LFN
<code>mv elfi/LFC/logical/file/name \</code> <code>elfi/LFC/other/logical/file/name</code>	rename a LFN
<code>rm elfi/SE/logical/file/name</code>	delete replica residing on SE
<code>rm elfi/LFC/logical/file/name</code>	remove an LFN and all of its replicas

## ELFI is based on FUSE

**FUSE** is a **framework** for implementing Linux filesystems.

- **Well-tested code**, used by dozens of filesystems out there.
- **Standard**: Part of Linux kernel since 2.6.14.
  - Available as a separate module since 2.4.21.
- **Designed to be stable**: separates kernel code from filesystem implementation.
  - If filesystem implementation crashes, clients just get a ENOTCONN error.

### Included in Scientific Linux 4.4

- And most other recent Linux distros: Debian, Fedora 4, etc.
- EGRID provides RPMs for Scientific Linux 3
  - Ready to install on gLite UI and WN

**Secure**: only user that mounted filesystem can access it.

- By default, not even root can inspect other users' FUSE-mounted filesystem.

## SRMv2 support

ELFI supports using SRMv2.

- both v2.1.1 and v2.2 are supported

EGRID production environment uses ELFI together with the StoRM SRMv2 server.

SRM servers have interoperability problems.

- No two SRMv2.1.1 servers are really interoperable.
  - ELFI is committed to be compatible with StoRM SRMv2 server, may not talk with others.
  - But no-one is deploying SRM v2.1.1 in production environments anyway.
- It's getting better with SRMv2.2:
  - Interop issues are being ironed out.
  - Need to wait for interop tests to finish and final operational spec is released.

## classic SE + DPM-rfio

ELFI supports “classic” SEs where the DPM-rfio-server is installed. This is a non-standard setup in gLite 3.0, but...

- It's easy to install, anyway
- Standard CASTOR rfio should *never* be used outside local network.
  - It's a security hole!
- You can recompile ELFI to use the usual CASTOR-rfio-server.
  - To access the local SE from the site WNs.

A new RFIO version with ability to connect to both server flavors is upcoming.

## Future developments

Current focus on stability and bugfixing.

- Need a very stable version for EGRID production use.
- More users, more bugs, more bugfixes :-)
- Goal: run the *bonnie++* benchmark on ELFI!

Planned developments, when we have time:

- Support GFAL, GridFTP and DCAP
- Aggressive data caching
- Integration with EGEE Encrypted Data Storage
- Merger with parrot code base

## Summary of ELFI features

### Real Linux filesystem

- transparent access to Grid storage
- usual POSIX/Linux filesystem API
- based on Linux standard FUSE technology

### Entries in the LFC catalog are files in ELFI

- The directory tree mirrors the LFC namespace
- Access files by Logical File Name
- filesystem ACLs correspond to LFC ACLs

### File contents are taken from file replica on SE. Currently supports:

- “classic” SE with GSI-RFIO
- SRMv2 enabled (especially StoRM)

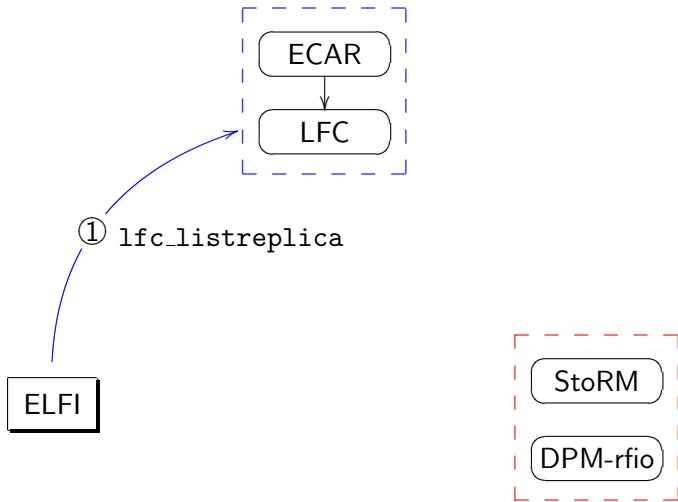
The End

Thank you!  
(Any questions?)

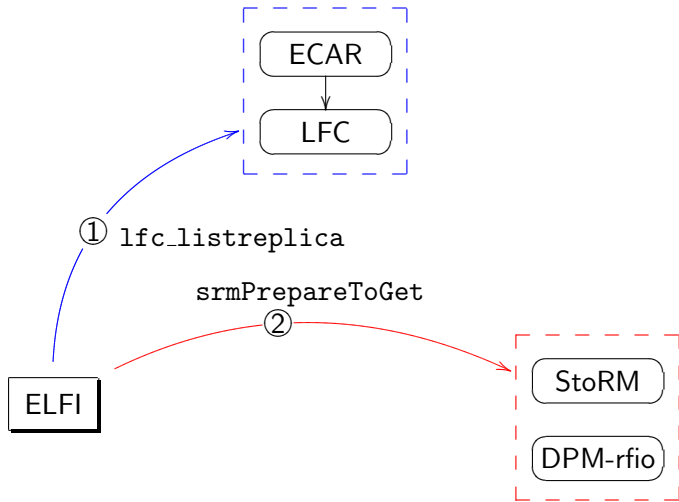
## References

- ELFI <http://www.egrid.it/elfi>
- EGRID <http://www.egrid.it>
- FUSE <http://fuse.sourceforge.net>

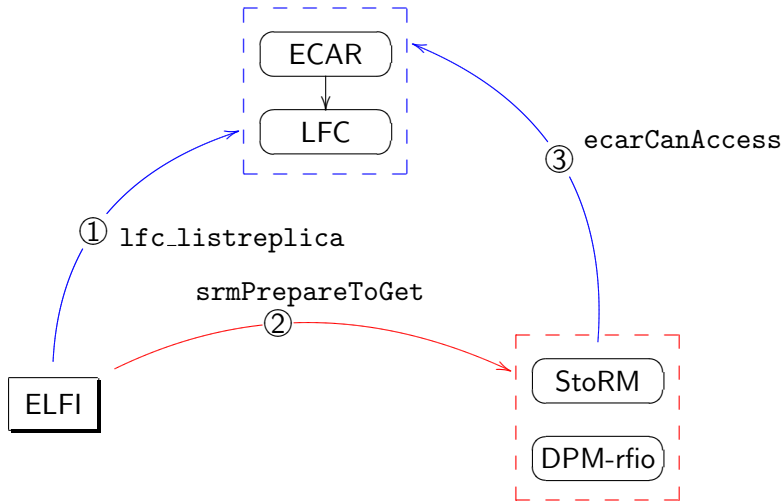
# EGRID security architecture



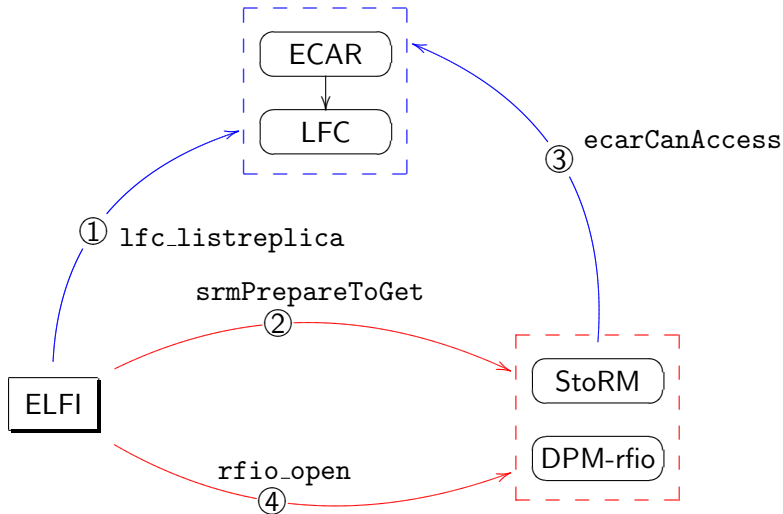
# EGRID security architecture



# EGRID security architecture



# EGRID security architecture



# Modular layered design

ELFI is built on three independent layers.

**Catalog** LFN to SURL lookup, list directories, get/set ACLs

**Control protocol** Get access to file, transform SURL to TURL

**Access protocol** Read/write file contents

Each layer implements a fairly generic API, so current code can be replaced and new functionality added.

# FUSE filesystems architecture

