



# GFAL 2.0 Evolutions & gfalFS

Devresse Adrien



# GFAL 2.0, complete rewrite of GFAL. Why ?

**GFAL 1.X does its job but....**

- **lots of hard coded dependencies** for deprecated technologies.
- **Error reporting** is quite poor ( errno and only errno )
- **Old source code**, very difficult to maintain
- **API difficult to use** and technology specific ( gfal\_Is for srm..., gfal\_handle,.....

# GFAL 2.0, The Grid File Access Library

- **Abstraction Layer** over the storage systems complexity.
- **Simple POSIX API** for the Grid interaction
  - `gfal_open`, `gfal_read`, `gfal_lseek`, `gfal_write`, `gfal_lstat`, `gfal_readdir`, `gfal_chmod`, `gfal_unlink`, `gfal_symlink`, `gfal_getxattr`
- **New advanced functions behind the POSIX calls** ( set/get replicas, guid resolution, turl resolution, set/get comments, link & rename )
- **basis** of the **EMI data consolidation** work.

→ **One** tool to access every storage system  
( LFC, DPM, Dcache, CASTOR, .. )

→ **One** tool to manage **any** protocol ( dcap,  
gsidcap, SRM, LFN, HTTP/Webdav, guid )

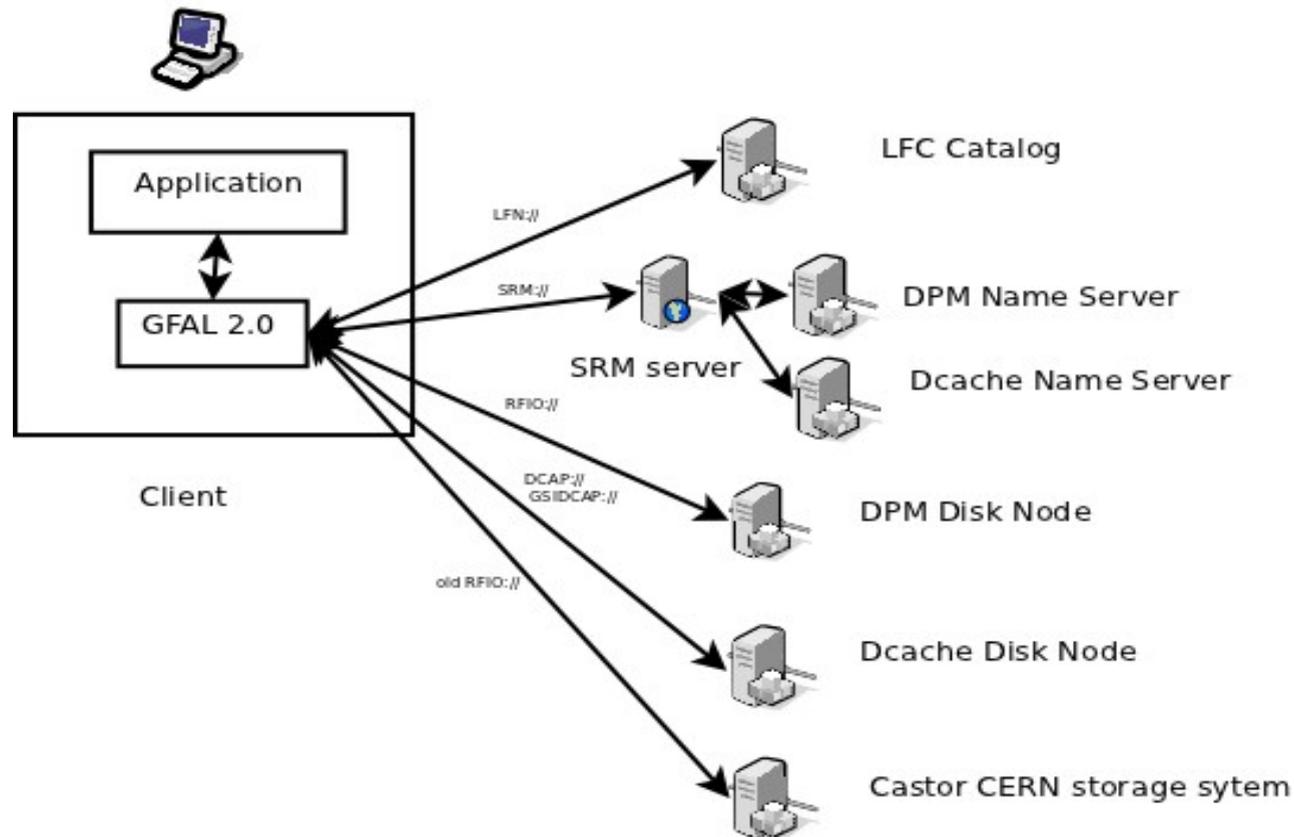
→ example :

```
gfal_open("lfn:/grid/dteam/test", O_RDONLY, )  
gfal_opendir("srm://gridurl.com:8446/something/")  
gfal_lstat("guid:b7ba213b-829d-[...]-7c9cc685fe2d")  
gfal_chmod("file:///tmp/test")  
gfal_unlink("rfio://myendpoint.org/something")  
[...]
```

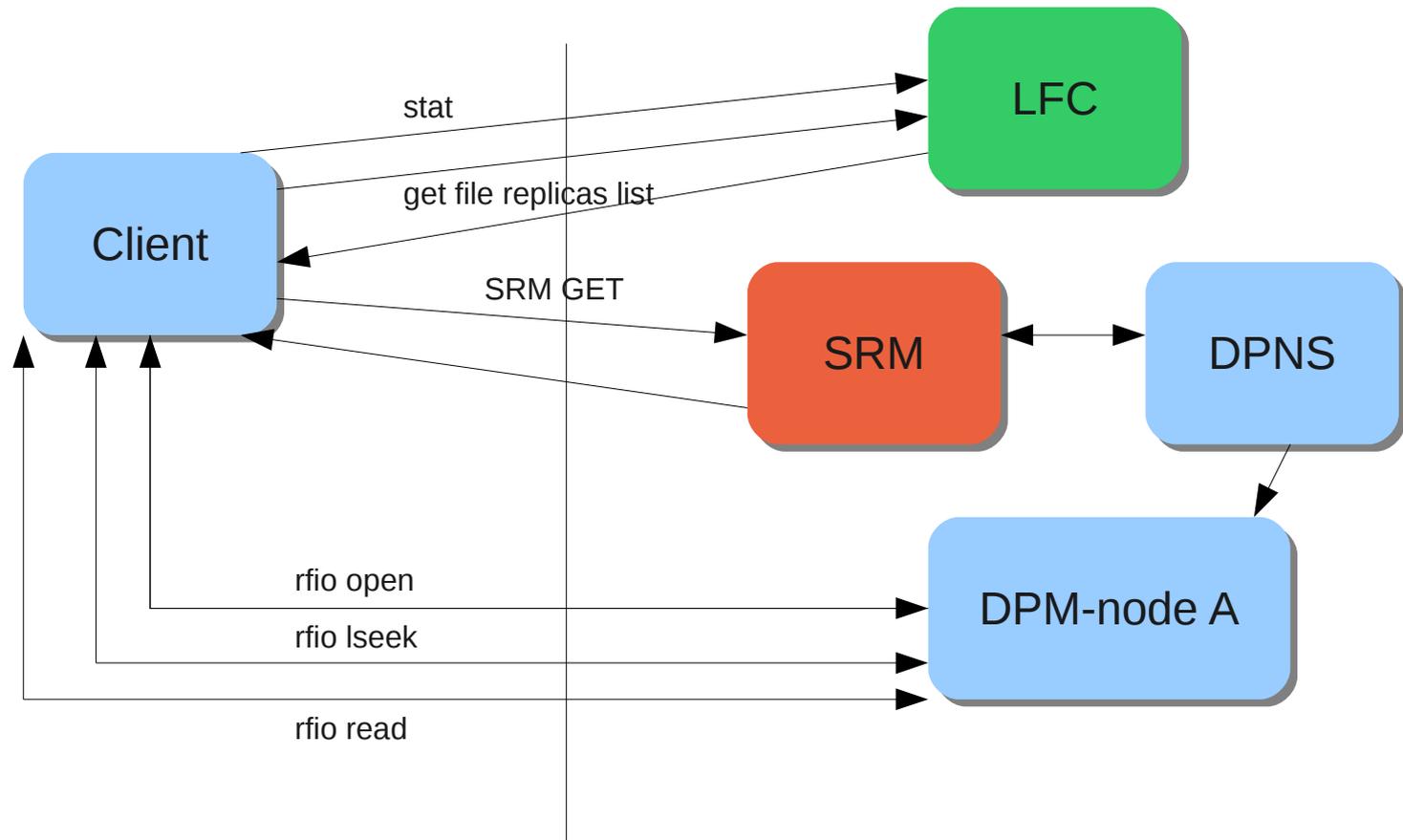
→ **API incompatible with GFAL 1.0**

→ GFAL 1.X released in parallel for some time

# GFAL 2.0 Overview



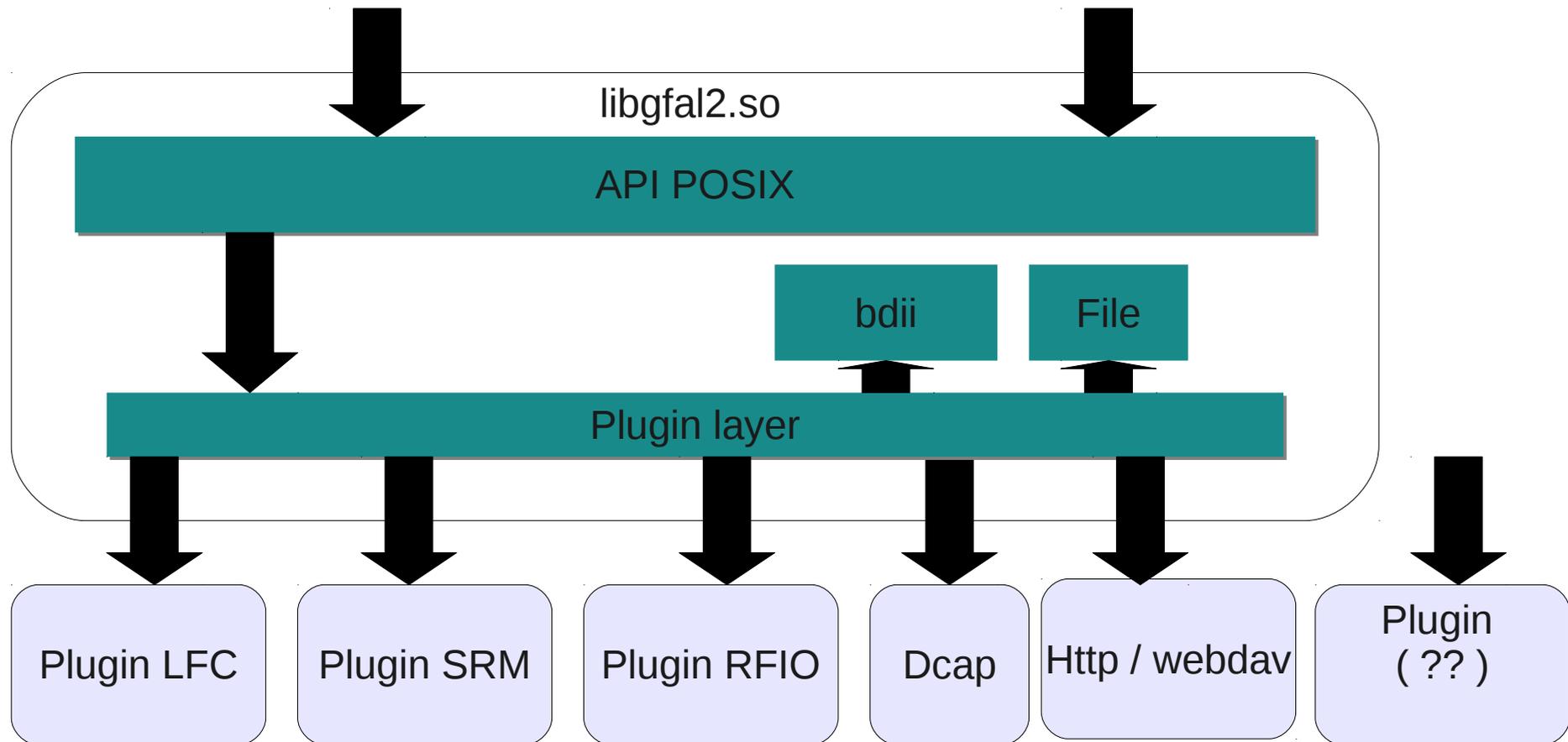
# The complexity of a random read file operation in the LFC



# GFAL 2.0 abstraction:

- `int fd = gfal_open("lfn:/grid/dteam/my_file", O_RDONLY) ;`
- `gfal_lseek(fd, pos, SEEK_SET) ;`
- `gfal_read(fd, buffer, size) ;`

# GFAL 2.0 Design



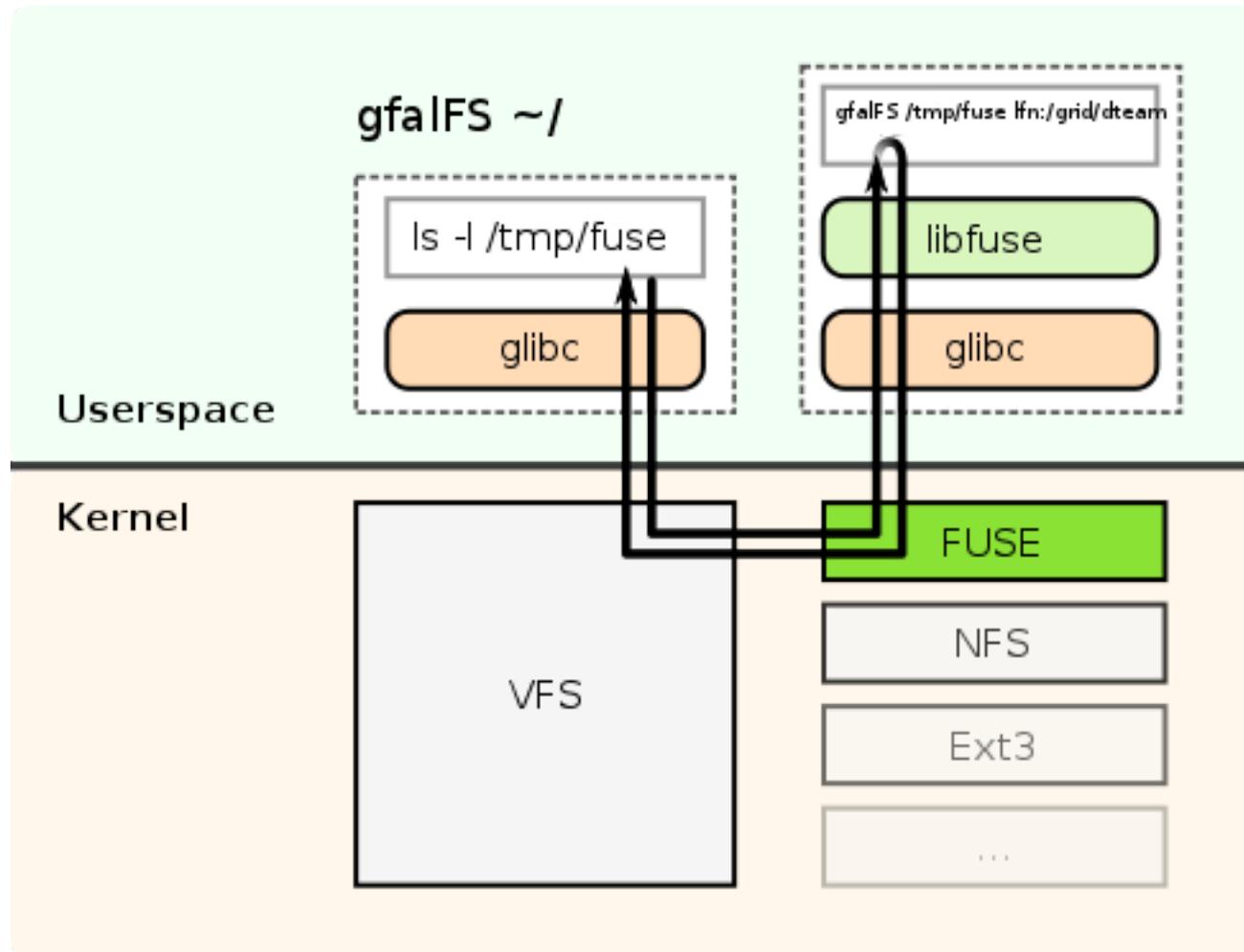
# GFAL 2.0 Plugin system :

- **The plugins** are separated **Shared library** loaded at runtime.
- **Low coupling** between GFAL and the protocols / systems supported.
- **Modular** system able to support new protocols.
- **API** and example provided to develop plugins

# gfaFS, an userspace filesystem for GFAL 2.0

- **Make** the grid storage system **transparent** to the user
  - as easy as a local file to use
- **Allow mounting** of any remote point managed by **GFAL 2.0** ( LFC, DPM, dCache, Webdav )
  - Allow all the **system calls** on it
  - Allow the user to use **any application** on the grid directly ( **seek files** .... )

# gfaFS overview



# gfalFS example of use cases

- **gfalFS /tmp/mnt/ srm://myurl.org/myfolder**
- **ls -l /tmp/mnt**  
**# copy file**
- **cp /home/user/my\_file /tmp/mnt/file\_on\_grid**
- # read file and work on it**
- **cat /tmp/mnt/file\_on\_grid**
- **sed 's/foo/bar/g' /tmp/mnt/file\_on\_grid > /tmp/mnt/file\_on\_grid2**
- **grep bar /tmp/mnt/file\_on\_grid2**

# gfaFS easy file management

**# change right, remove, create folder on grid**

- **chmod 777 /tmp/mnt/file\_on\_grid**
- **rm /tmp/mnt/file\_on\_grid**
- **mkdir /tmp/mnt/my\_folder\_on\_grid**

**# get turl of a file**

- **getfattr -n user.replicas /tmp/mnt/file\_on\_grid2**

**# move file**

- **mv /tmp/mnt/file\_on\_grid2 /tmp/mnt/file\_on\_grid**

**# do what you want, almost everything is possible !**

# gfalFS, advanced operations

**gfalFS /tmp/mnt/ lfn:/grid/dteam**

## **# get / set replicas**

- **getfattr -n user.replicas /tmp/mnt/testfile**
- **setfattr -n user.replicas -v srm://myurl/file/tmp/mnt/testfile**

## **# get/ set comments**

- **getfattr -n user.comment /tmp/mnt/testfile**
- **setfattr -n user.comment -v "my msg" /tmp/mnt/testfile**

## **# get guid of a file**

- **getfattr -n user.guid /tmp/mnt/testfile**

# GFAL 2.0 & gfaFS status

- **PREVIEW stage**, last modifications before validation.
- **We are researching testers and feedback :**
  - **<https://svnweb.cern.ch/trac/lcgutil/wiki/gfal2>**
  - Easy install and test with **yum repositories**.
  - Do not hesitate to send your comments or ideas.
- **lcgutil-devel@cern.ch, adrien.devresse@cern.ch**

# END

