FTS 2/3, GFAL 2 status update

Oliver Keeble
Zsolt Molnar
Michail Salichos
Michal Kamil Simon

FTS 2/3, GFAL 2

- FTS 2 status
- FTS 3 status and characteristics
- GFAL 2 status

An overview of these services will be provided and how they relate to one another

FTS 2.2.8

- Infrastructure is 100% running on EMI FTS
- FTS 2 development has now stopped
  - Is being ported to SL6
- All forthcoming features are now scheduled for FTS3
  - This is a strategic decision
    - Not technically inevitable in every case

FTS 2.2.8

- Features (wrt 2.2.4)
  - gsiftp endpoints
  - Transfer monitoring
  - Updated overwrite logic
  - Transfer resume
  - Tested with Oracle 11g
- A bugfix release may be made, depending on requirements

FTS/lcgutil common libraries

- GridFTP and IS access libraries
  - Used both in FTS and lcgutil/GFAL
  - high cohesion / low coupling
  - Common functionality
  - No code overlapping

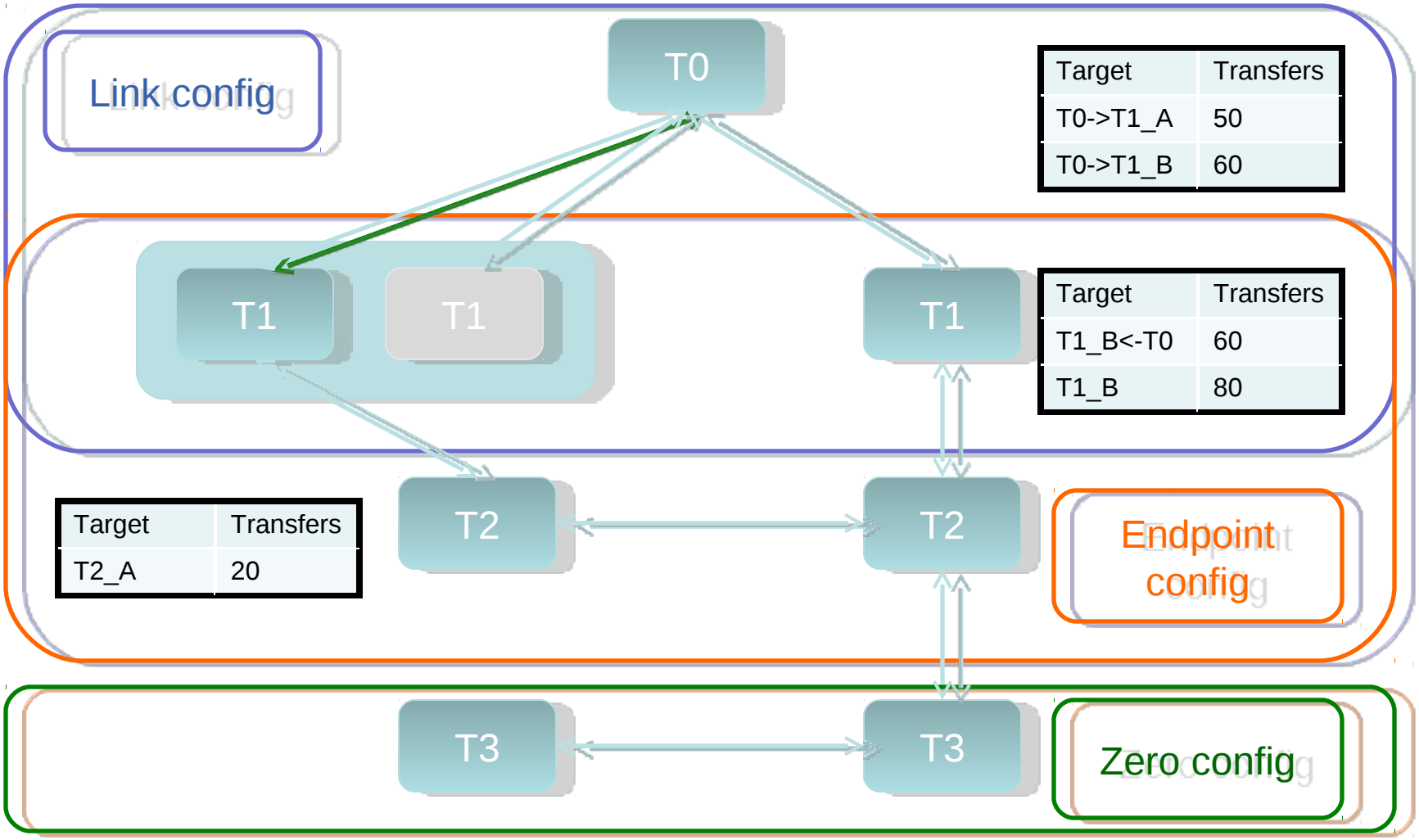
FTS 3 background

- FTS3 was conceived to address a particular set of FTS2 shortcomings
  - It was called FTS3 to emphasise continuity
- Main motivations wrt FTS2
  - Configuration model
    - Relax the requirement to configure channels
    - Instead use endpoints, good defaults and adaptation
  - Protocol support
  - Database support
    - Mysql
  - Simplified deployment
  - Code maintenance issues

Endpoint oriented config & scheduling

- The current channel model convolves information about endpoints and the intervening network
- The new model assigns characteristics to endpoints
- Solution
  - Optional config of parameters for a link (rather than endpoint)
  - Use of defaults
  - Adaptation

Configuration model



Protocol support

- FTS3 is built on top of gfal2
- gfal2
  - Provides protocol plugins
  - Offers a 3rd party copy extension
  - Thread safe posix library
  - Improved error reporting
  - Smaller, faster, caching
- Will come with srm and gridftp
- Planned
  - http
  - xroot (will be provided by ARC as gfal2 plug-in)

Database support

- FTS 2 is currently tied to Oracle
- FTS 3 is using a database independent interface
- It will be supplied with the oracle plugin
- Interest in mysql is still uncertain

Deployment of a server

- FTS3 is built around a multi-threaded server
  - Threads can be created for different roles
    - “FTS” functions
    - “FTA” functions
- This involved reimplementing the “FTS” part in C++
- Intention is to allow horizontal scalability
- A maximum of configuration is held in the db

FTS 3 status

- The development follows a scrum-like process
- This involves a demo of new functionality every 3 weeks
- Last demo showed
  - Demonstrating how to configure FTS3
  - Discussing different configuration scenarios

FTS3 features demoed

- GridFTP transfers on top of Gfal2
- FTS3 server config proposal
- FTS 3 server config file structure
- Running multi-threaded FTS 3 web server, in C++
- New transfer submit/poll commands
- Backward compatibility with the glite-transfer commands
- Configurable database backend type, working Oracle plugin

FTS3 next demo

- SE-centric scheduling (fifo) and simultaneous transfers
- When: 16th of May
- Reminder will be sent in fts3-steering mailing list

Between now and prototype 1

- What's left to do:
  - SE-centric configuration
  - SE-centric scheduling
  - Log and monitor transfer execution using GFAL2
  - Integrate monitoring messages from FTS2 to FTS3

Prototype 1 – expected features

- db independent interface
- Scalable deployment
- Transfers using gfal2
- SE-based job scheduling and configuration

Prototype 2 – expected features

- Http transfers
- Smart scheduling for ordering transfer requests
- Adjust transfer parameters based on historical data
- Various optimizations
  - gsiftp channels caching for small file transfers

FTS3 final release

- Expected in EMI3
- Will be Prototype 2 iterated into production quality

FTS3 features summary

- Remove channel model – SE centric
- More transfer protocols (HTTP) using plug-in mechanism
- Transfer optimization
  - Historical data from past transfers
  - Gsiftp channels caching for small files
- New retry logic
- Improved scheduling
- Multiple database back-ends support
- Simplification of initial configuration and parameterization

FTS3 Roadmap

- Main milestones
  - Prototype 1: 27th June, 2012
  - Prototype 2: December, 2012
  - FTS 3.0.0: April, 2013 (production candidate)

https://svnweb.cern.ch/trac/fts3/roadmap

Why GFAL 2.0 ?

- GFAL 1.0 code is old styled : lots of dependencies and no consistent design
- GFAL 1.0 has a poor **Error reporting system (errno only)** , no log system...
- GFAL 1.0 is **not thread safe**, even the “pthr” version.
- GFAL 1.0 has a confusing API, lots of code duplication and overlapping

GFAL 2.0, The Grid File Access Library

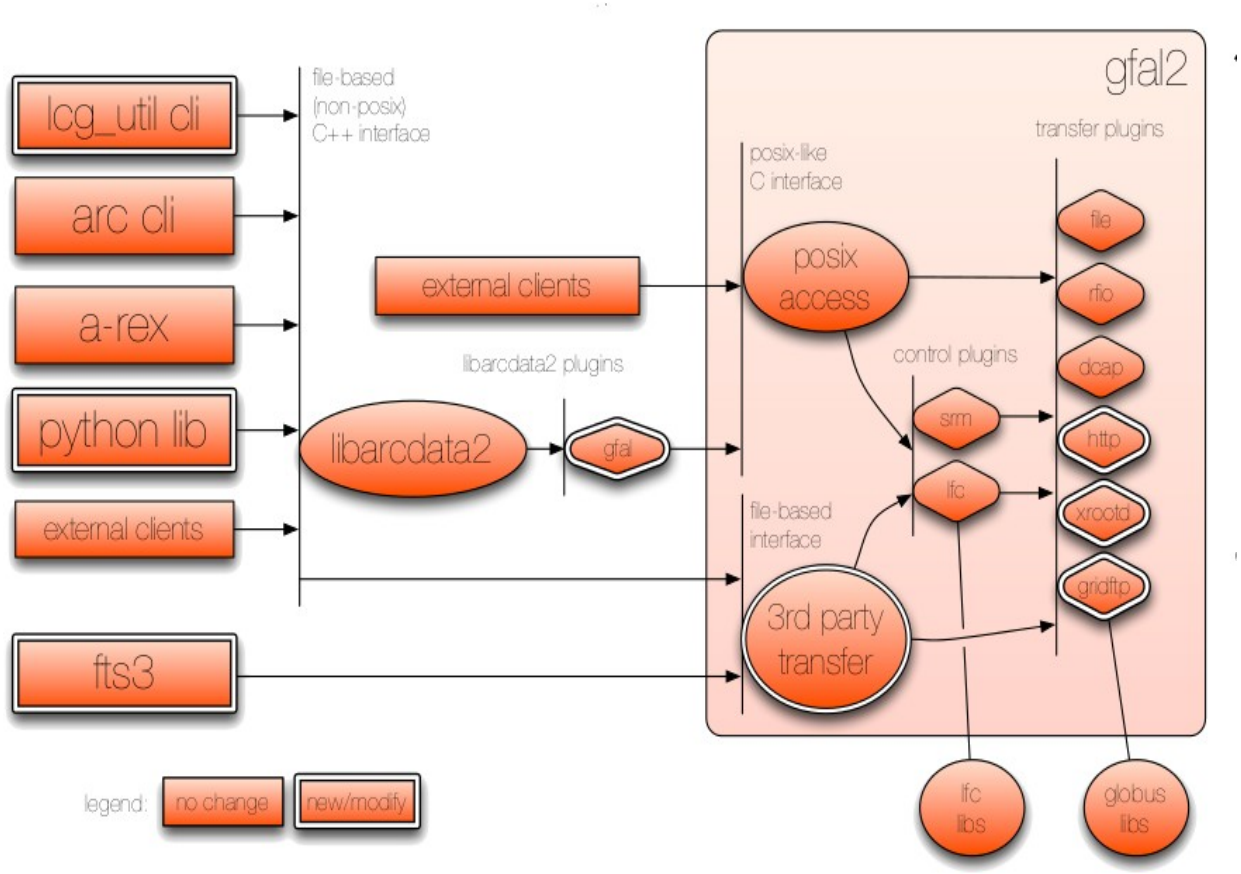
- **One modular Library for many protocols and storage systems :**

→ **Compatible protocols :**
- **SRM**
- **LFN**
- **GRIDFTP**
- **DCAP/GSIDCAP**
- **RFIO/ RFIO secured**
- **HTTP/ Webdav**
- **XROOT (provided by ARC)**
- **S3(future, interest ?)**
- **Easy to extend**

→ **Compatible storages :**
- → **DPM**
- → **Dcache**
- → **Storm**
- → **Castor**
- → **LFC**
- → **EOS**
- → **Every GridFTP endpoint**
- → **Every Webdav endpoint**

→ **One single and consistent API** for everything.

EMI data consolidation effort is on top GFAL 2



→ Easy to Use

→ **Complexity is hidden behind simple POSIX calls**

→ gfal_open(gsiftp://myurl/myfile), read, write, close

→ gfal_lstat(srm://myurl/myfile), mkdir, rmdir, unlink

→ gfal_opendir(davs://myserver/myfile), readdir, etc

→ **Provides Transfer/ file level for easy transfers :**

→ third party transfer support

→ gfal2-transfer API

gfalt_copy_file(handle, NULL, “srm://src”, “srm://dest, &tmp_err) ;

→ Python bindings with a Pythonic design

→ **Python Exceptions** support

→ **Thread-safe**, **python GIL** support, **parallel read/write**

Designed to be easy to extend :

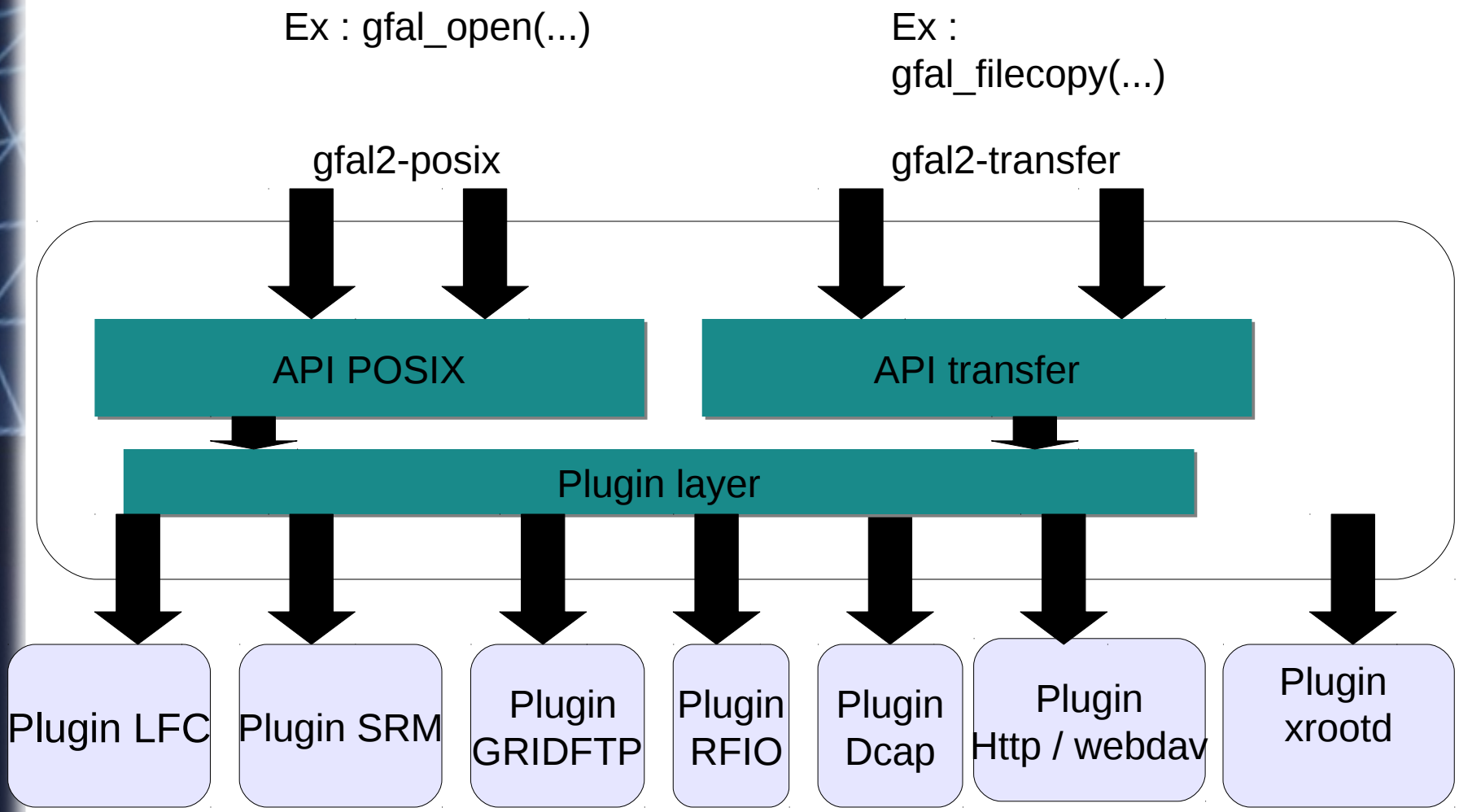
→ **GFAL 2.0 has a plugin API.**

→ **GFAL 2.0's plugins are standard shared libraries implementing an interface.**

→ **Any project can make its GFAL 2.0 plugin if needed.**

-> headers and tutorials are provided with the binaries.

GFAL 2.0 Design



gfalFS : a filesystem for GFAL 2.0

- **GOAL :**access remote files as easy as local files on the client side.

- **Example :**

```
# mount
gfalFS /tmp/mnt/ srm://myurl.org/myfolder
# and play !
# list
- ls -l /tmp/mnt
higgs_boson_found.txt mycat.jpg mydoc.pdf
# use
- cp /home/user/my_file /tmp/mnt/file_on_grid ; cat ;
sed ; grep ….
# and manage
rm -rf /tmp/mnt/useless ; mkdir ; chmod ; getfattr ; ….
```

GFAL 2.0 & gfalFS status

- **Already in EMI 2 (early update), EPEL and EPEL-testing**
  - → Easy to try, no configuration needed :
    - sudo yum install –-enablerepo=”epel-testing” gfal2-all gfal2-doc
    - sudo yum install –-enablerepo=”epel-testing” gfalFS

- **We** are looking for **testers and** feedback :
  - **https://svnweb.cern.ch/trac/lcgutil/wiki/gfal2**
  - **lcgutil-support@cern.ch**

Summary

- FTS 2 will run until replaced by FTS 3
- FTS 3 and GFAL 2 will evolve to support more and more features
- FTS 2 will be supported and potentially (if needed) enhanced
- Feedback and new feature requests are always welcome

Resources

- FTS2
  - https://svnweb.cern.ch/trac/glitefts/wiki/FTSRelease_2_2_8
- FTS3
  - https://svnweb.cern.ch/trac/fts3
  - fts3-steering@cern.ch
    - Subscribe - http://cern.ch/go/99Gg
- Gfal2
  - https://svnweb.cern.ch/trac/lcgutil/wiki/gfal2