



FTS3 and GFAL2 status & description

IT-GT-DMS

Michail Salichos

Overview

- background
- current status
- design and features
- technical details
- status and roadmap
- summary
- references

FTS 3 background

- FTS is the service responsible for distributing the majority of LHC data across the WLCG infrastructure
- 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 (gsiftp,srm,http & xrootd soon)
 - _ Database support
 - Mysql
 - Oracle
 - _ Simplified deployment & configuration
 - _ Code maintenance issues

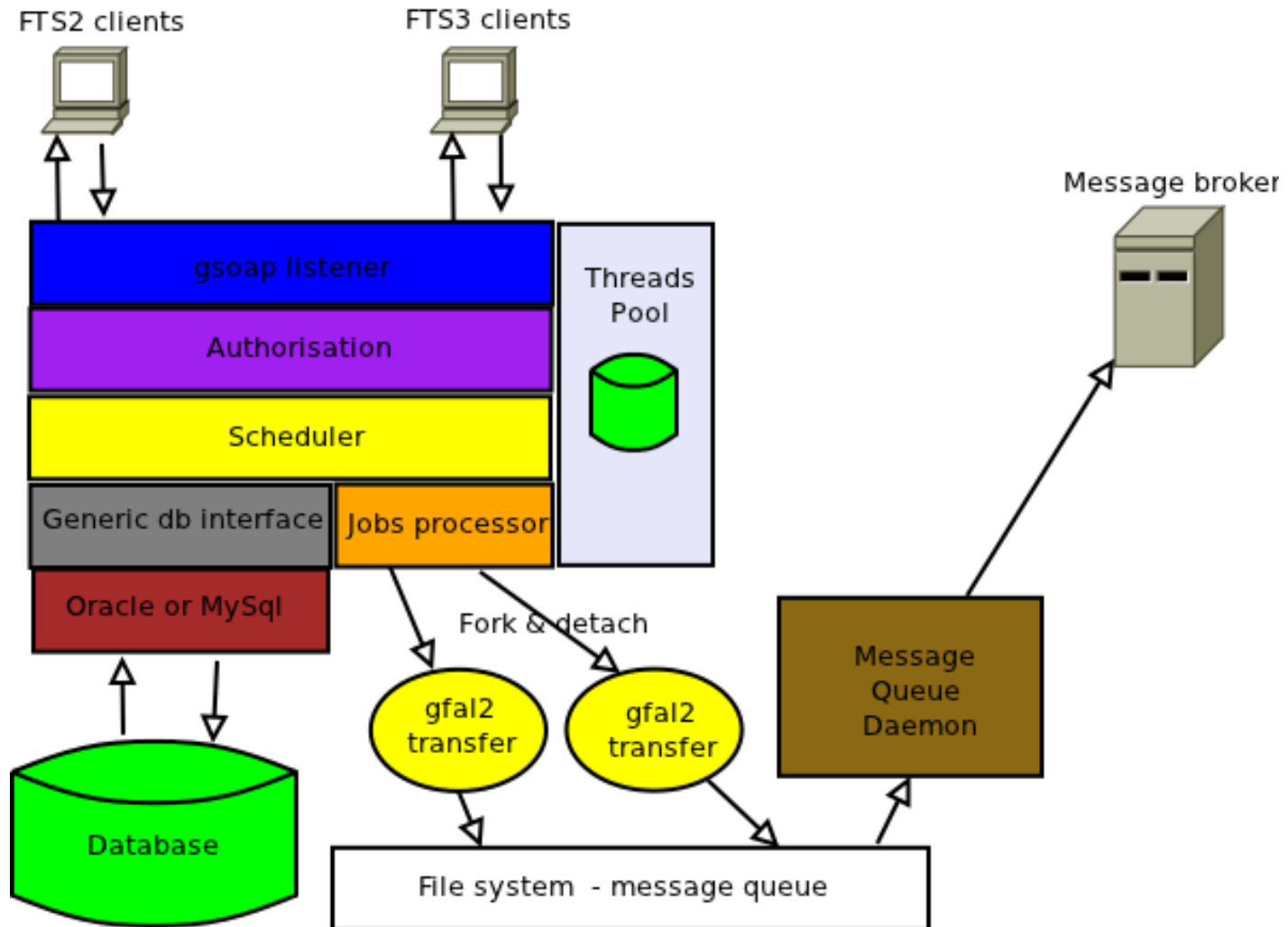
FTS3 Current status

- FTS3 is alive!
- prototype 1 has already been installed for functional testing at:
 - CERN, BNL, RAL, ASGC
 - Monitoring pages
 - <http://dashb-wlcg-transfers.cern.ch/ui/>
 - <http://www.gridpp.rl.ac.uk/fts3/>
 - <http://vt-092.grid.sinica.edu.tw/ftsmonitor>
- continuous development, debugging and keep adding new features

FTS3 - Prototype 1

- Simplified configuration and administration of the service
 - Zero configuration supported
- Auto-tune transfers (experimental)
- Functionally comparable to FTS2
- Channel-less
- Generic db interface – multiple back-ends support (oracle only for now)
- Generic transfer protocol interface
- Job execution and status (FTS2 backward compatible clients)
- Improved fair-share and transfer optimization
- GridFTP session reuse (FTS3 clients only)

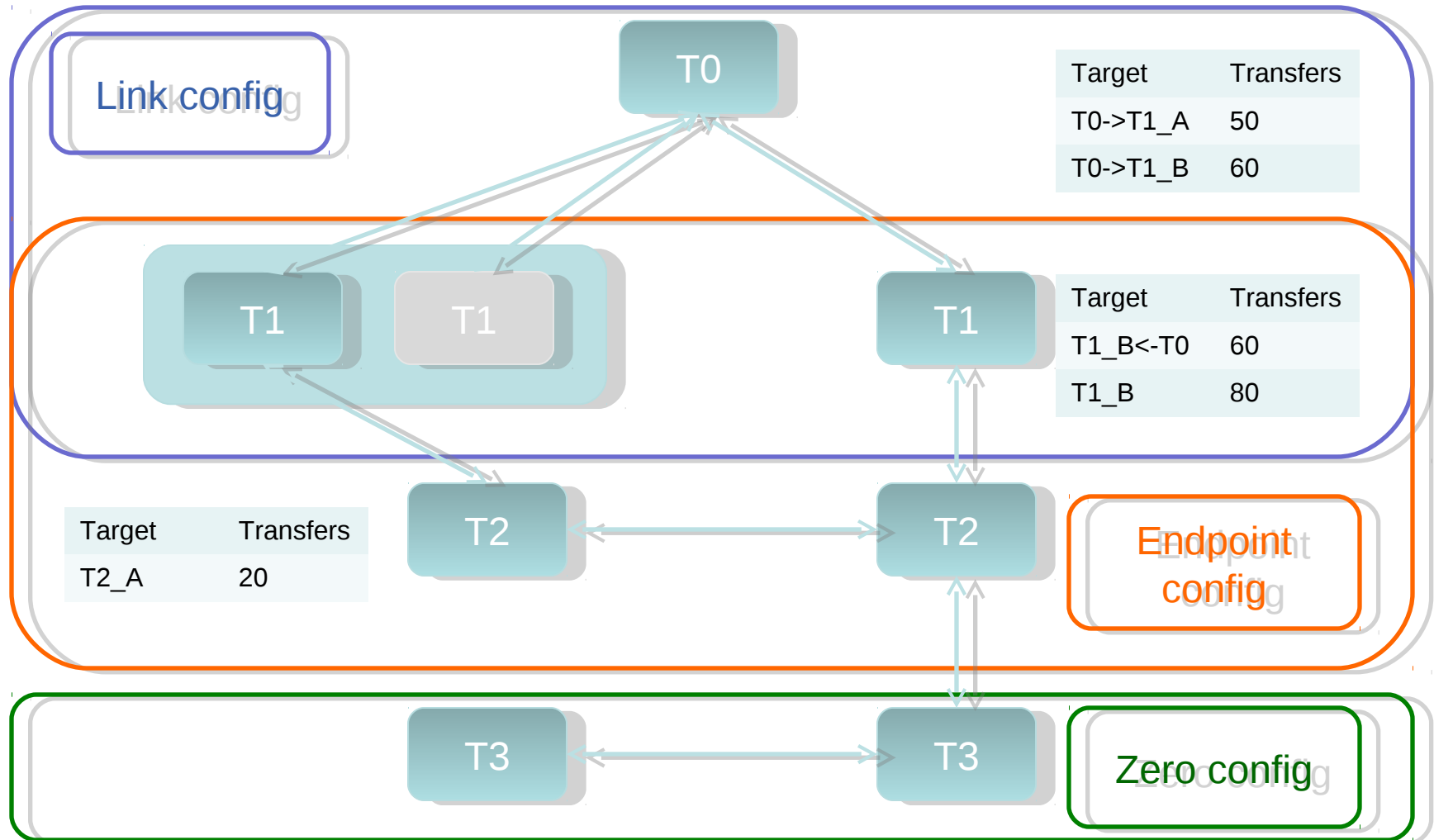
FTS3 architecture



FTS3 features

- No channel model / endpoint centric
- More protocols (gsiftp, srm, http & xroot soon) using gfal 2.0 plug-in mechanism
- Multiple database back-ends support (Oracle,MySql)
- Transfers auto-tuning
- Improved error verbosity
- Client's Python API
- Simplification of configuration & parameterization
 - Zero configuration – good defaults
 - Manual, JSON-formatted
 - Let auto-tuner decide (heuristics based on performance of last transfers)

FTS3 configuration model



FTS3 status and roadmap

- Main milestones
 - FTS 3 Prototype 2: December, 2012
 - FTS 3.0.0: April, 2013 (production candidate)

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

FTS3 - Prototype 2

- More protocols (http, xroot)
- More backends (MySQL)
- Authorization
- Staging files
- Multiple replicas fallback mechanism
- Optimization

FTS3 - Long term goals

- integration with perfsonar (e.g. to retrieve RTT between src-dest)
- inter-FTS3 messaging to establish common view of activity between servers (for load balancing ?)
- embedded FTS3? (sqlite backend)
- SE's back pressure

FTS3 summary

- FTS3 is an evolution from FTS2 designed to address a number of problems in its design assumptions and internal architecture
- We are demoing functionality along the way
 - Next demo:
<http://indico.cern.ch/event/fts3>

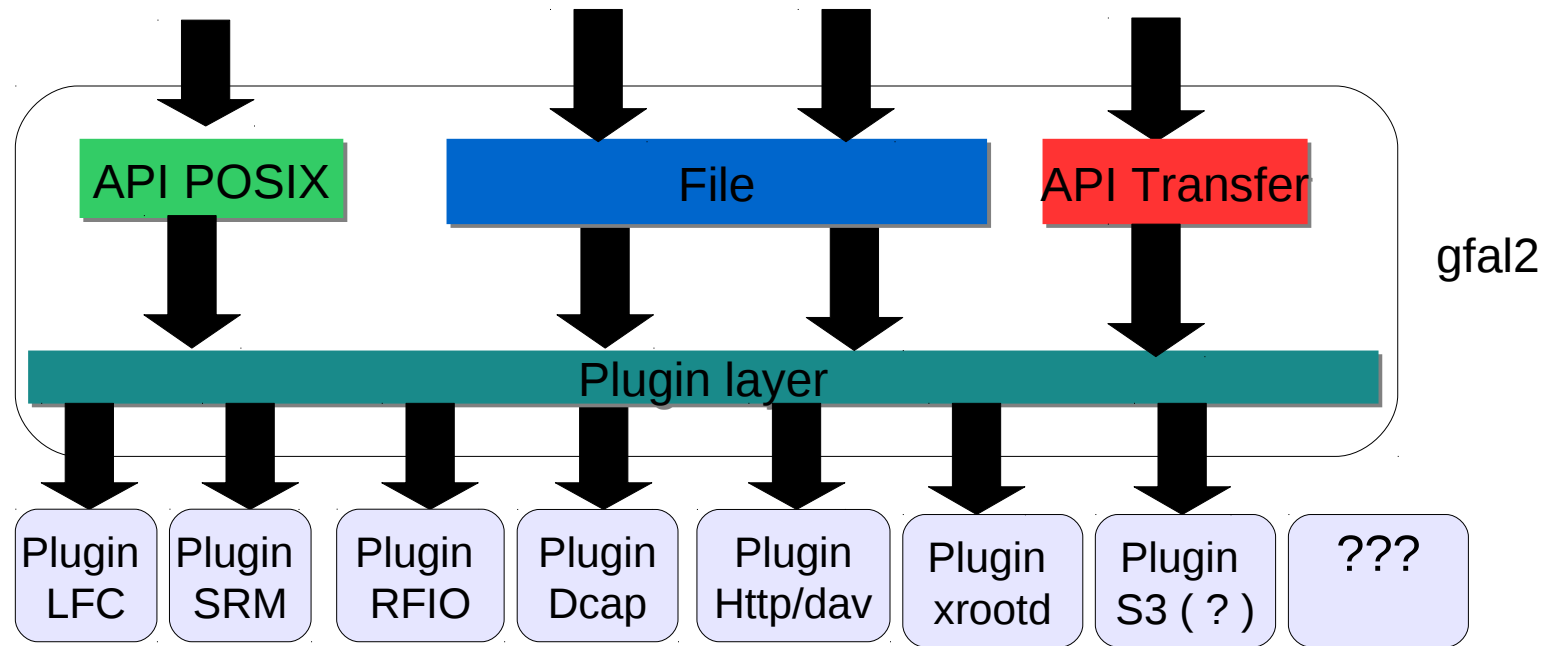
GFAL2 background

- One library for all Grid and Cloud data access and management
- One library for all data access and transfer protocols
 - Supported protocols
 - GridFTP, DCAP/GSIDCAP, RFIO/ RFIO secured
 - LFN, SRM, HTTP/ Webdav, (S3 ?)
 - XROOTD (developed by NorduGrid)
 - easy to extend : plugins mechanism
- Fully thread-safe, better error logging
- **It's the transfer workhorse of FTS3!**

GFAL2 Current status

- Already released in EMI2 and EPEL
 - **yum install gfal2-all gfal2-doc gfalFS**
- Will be packaged for Debian support
- continuous development, debugging and keep adding new features
- **Ready to used!**

GFAL2 architecture



Any Cloud
with Http/Dav

Any xrootD
point

Extensible to any storage
system :
Amazon S3, HDFS, etc...

GFAL2 evolution vs GFAL1

- better error messages, logging system, trace mode
- fully thread safe
- T-party file transfer support
- Plugin system
 - Reduce dependencies
 - Install only what you use
 - Easily extensible

GFAL2 how to use

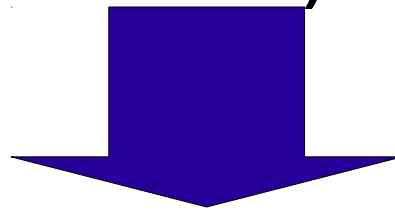
- **libgfal2**
 - C Library
 - set of independent plugins
- **gfal2-python**
 - simple and pythonic python bindings
- **gfalFS**
 - fuse module for gfal 2
- **gfal-tools**
 - experimental command line tools
 - gfal-ls, gfal-copy, gfal-mkdir, gfal-cat, ...

What GFAI2 can do

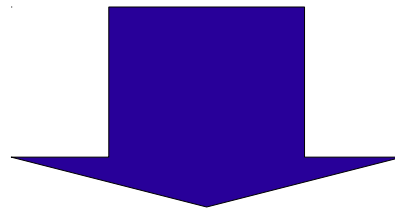
- **Meta-data operations**
 - _ stat, rm, mkdir, mv, rmdir, etc...
 - _ list directory, xattr, etc
- **Remote I/O in any protocol**
 - _ open/read/write/lseek/close
 - _ GET/PUT
 - _ pread/pwrite
- **High level file transfer in any protocol**
 - _ session-reuse, spacetoken, parallel streams, ...

GFAL2 is protocol generic

a, b in
guid, lfn, srm, rfiio, dcap , gsidcap, gsiftp, file, xrootd, ... (http,
dav)



`gfalt_copyfile(context, NULL, « a://src/file », «b://dst/file»)`



FINISHED !

lcg-util/gfal/gfal2 status & roadmap

- lcg-util 1.13.9 quick bug fix release is on its way
- lcg-util/gfal are in maintenance mode
 - we are still fixing bugs
 - new features are pushed to **gfal2**
- Migration from gfal1 to **gfal2**
 - We are working on a step-by-step migration guide
 - API is not 100% compatible
 - but posix API is the same
 - some SRM low level operations are not exposed, do you need them ?
- We indent to implement a subset of useful/in-use lcgutil clients on top of **gfal2**
 - which ones ?

GFAL2 summary

- Hides the Grid complexity from the client side
- Simplifies the data access world:
One API
- Reduces the number of dependencies
- Makes application independent of the technology / protocols

References

- 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>
 - _ lcgutil-support@cern.ch

IT-GT-DMS

Oliver Keeble

Michail Salichos

Michal Kamil Simon

Adrien Devresse

Alejandro Alvarez Ayllon

+ our students and short term visitors