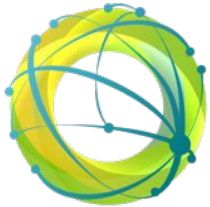
# FTS & XRootD Workshop

*Jozef Stefan Institute, Ljubljana, 27-31st March 2023*



FTS

File Transfer Service

# *FTS 2023: State of Affairs*

Mihai Patrascoiu
on behalf of the FTS team

CERN

# Open Source software for reliable and large-scale data transfers within WLCG

Features:

- TPC Orchestration
- Tape Operations (over multiple protocols)
- Certificate and token auth
- Multihop transfers
- Transfer Optimizer
- Cloud support
- Python bindings + CLI clients

…and many others

# FTS Team

- Mihai Patrascoiu (Project Leader) [CERN]
- Steven Murray (Service Manager) [CERN]
- João Lopes (C++ / Python developer) [CERN]
- Shubhangi Misra (C++ / Python developer) [CERN]

**External Contributors**

- Ed Dambik (C++ developer) [ATLAS / Indiana University Bloomington, USA]
- Eraldo Silva Junior (Python developer) [ATLAS / LHCb / CERN / CBPF, Brazil]

…and thanks to many other past and present contributors

# FTS Ecosystem

*Projects under FTS umbrella*

| | | |
|---|---|---|
| - FTS | (Server + QoS daemon) | [C++] |
| - FTS-REST | (Submission server) | [Python, Flask] |
| - FTS-clients | (Python & CLI) | [Python] |
| - FTS-Monitoring | (Django Web UI) | [Python, Django] |
| - webFTS | (Submission Web page) | [PHP] |

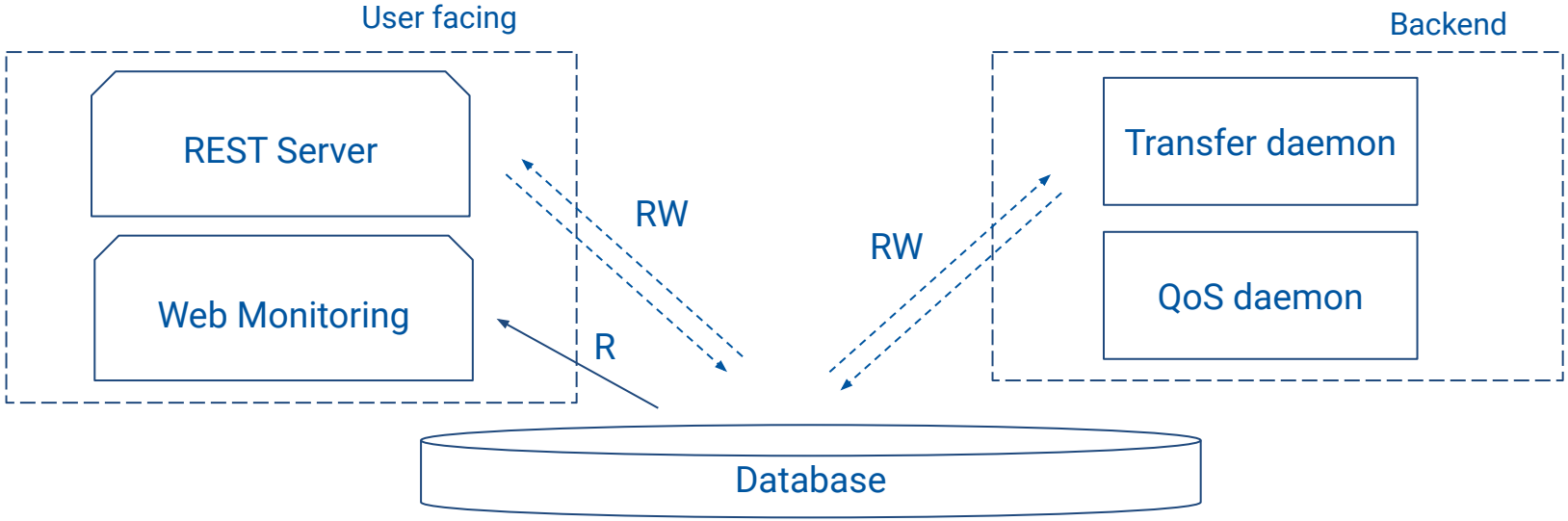
*Data Management Clients*

| | | |
|---|---|---|
| - Gfal2 | (Grid file access library) | [C++] |
| - Gfal2-python | (Python bindings) | [C++, BoostPy] |
| - Gfal2-util | (Python CLI) | [Python] |
| - Davix | (Grid HTTP client) | [C++] |
| - SRM-IFCE | (SRM interface for Gfal2) | [C, gsoap] |
| - CGSI-gSOAP | (gsi interface for Gfal2) | [C, gsoap] |

FTS and DMC clients published to PyPi, EPEL, Debian*

*Special thanks to Mattias Ellert

# FTS components (overly-simplified in 1 slide)




REST Server → Accepts submissions; queried for transfer status

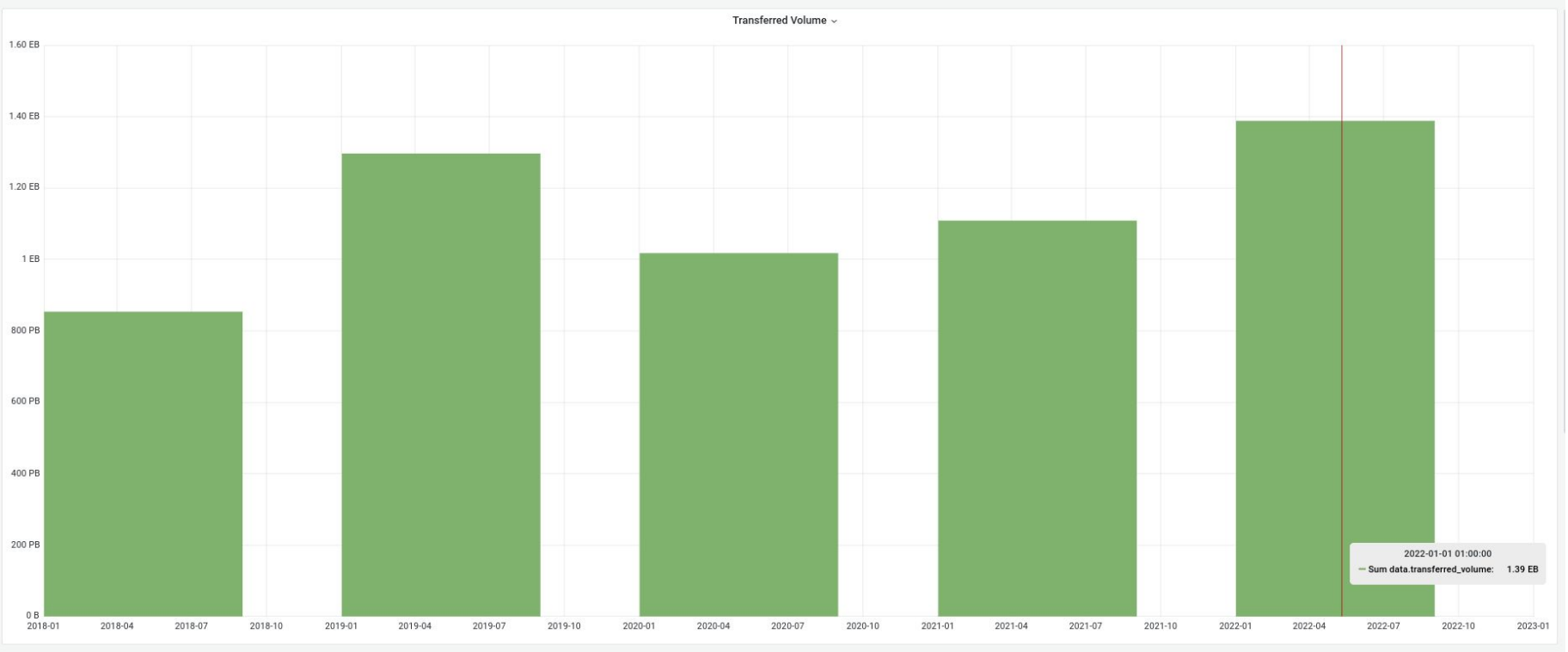
Web Monitoring → Visual webpage

Transfer daemon → Schedules transfers

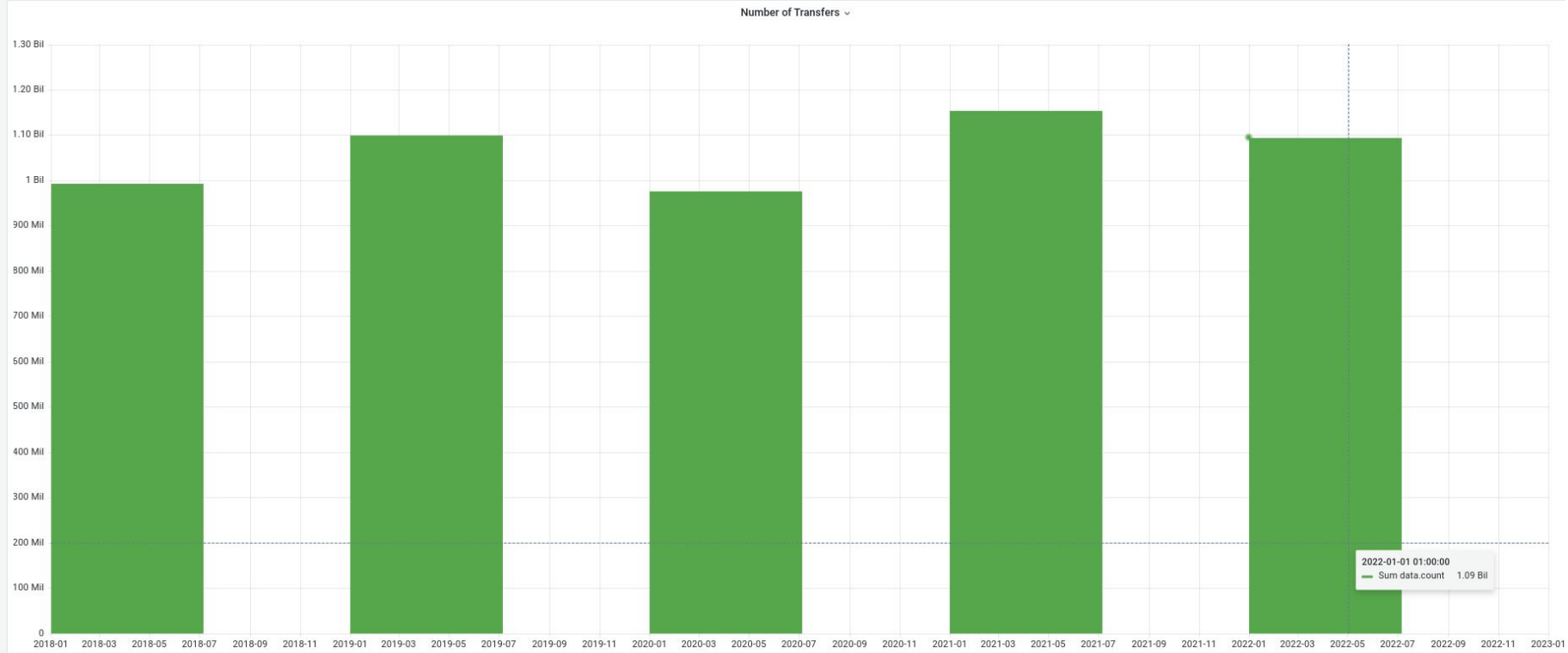
QoS daemon → Handles tape work

# *Service operations insights*

# FTS - Transferred Volume (2018-2022)



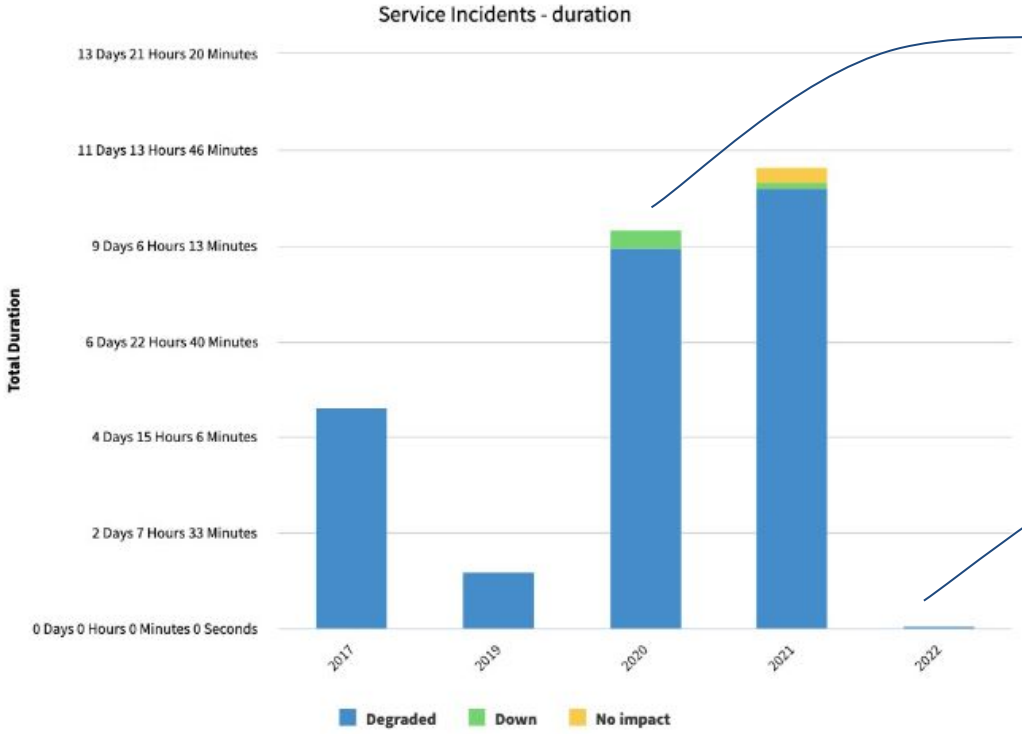
# FTS - Number of Transfers (2018-2022)



# FTS – Service changes at a glance

- Staging query investigation and change of algorithm (2021)
  → immediate effect on database performance
- Primary / replica database setup (2021)
  → increased robustness of the service deployment
- Alarming and health monitoring (2021/2022)
  → allows the team to be proactive wrt. service health
- MySQL8 deployment → allows for online schema upgrades
  and other service operations

# FTS – Service @ CERN




Started reporting
Web Monitoring
slowness / unavailability

Running new DB setup,
MySQL8 and FTS v3.12.0

# *Development insights*

## FTS v3.10 (2020)

- Addition of Archive Monitoring feature
- Appearance of FTS-QoS daemon (Bringonline daemon deprecated)
- First support for OIDC tokens introduced

## FTS v3.11 (2021)

- Destination file integrity report feature
- SE-issued tokens support built-in (Gfal2)
- Improvement of QoS staging database query (performance greatly improved after algorithm change)

## FTS v3.12 (2022)

- FTS-REST-Flask released (Python3 based)
- Movement to MySQL8 permitted (allowed by new FTS-REST)
- Schema updates w/o downtime
- Tape REST API

# Release strategy

Not *deliberately* following a 1-year major release! (despite looking that way)

**Convention:**

- FTS **3**.x.x: Generation version changes on complete software rewrite
- FTS x.**12**.x: Major version released on new features, behavior change or (large) schema changes. Often requires new Gfal2 & Davix major releases
- FTS x.x.**6**: Minor version changes on bugfixes, small adjustments, small schema change, etc

# FTS Releases – Change of (schema) approach

- Starting with FTS v3.12.0 + MySQL8: schema upgrades are no longer disrupting to the service operations

- **The FTS team will treat schema upgrades as non-disruptive to the service !**
  - No more need to bundle many schema upgrades into one big change
  - Allows us to release more often, including new features in patch releases

- We encourage all FTS sites to upgrade to v3.12.0 and MySQL8 (**or equivalent***)

** MySQL remains the CERN FTS deployment choice, but FTS will not break other implementations (e.g.: MariaDB) on purpose*

# FTS – Packages and Platforms

**Server**

- `fts-server / fts-rest-server / fts-monitoring`
- Only available on CC7 ! (Alma9 planned, no timeline established)
- Packages only available via the FTS repositories

**Clients**

- `fts-rest-client / ~~fts-client~~` (deprecated C++ client)
- `gfal2 / gfal2-python / gfal2-util / davix`
- Packages available via EPEL (covers EL7, EL8, EL9, active Fedora)
- EPEL packages: your best shot at FTS & DMC for other archs (e.g.: ARM)

# FTS – v3.12.x series

- Python2 → Python3 migration
- FTS-REST → FTS-REST-Flask rewrite
- Jump to MySQL8
- Tape REST API (**v3.12.2,** November 2022)
- WebMonit “linkinfo” feature (Eraldo contribution)
- HTTP-TPC IP reporting reworked
- Improved support for S3v4 transfers
- Staging & Archiving metadata over HTTP

- Eviction configurable per-StorageEndpoint
- Multihop “--overwrite-hop” feature
- Deprecation notice in C++ clients
- GDPR / OC11 compliance

  (VO name generation for certs w/o VOMS data)
- ActiveMQ format documentation up-to-date
- All DMC & FTS clients published to EPEL

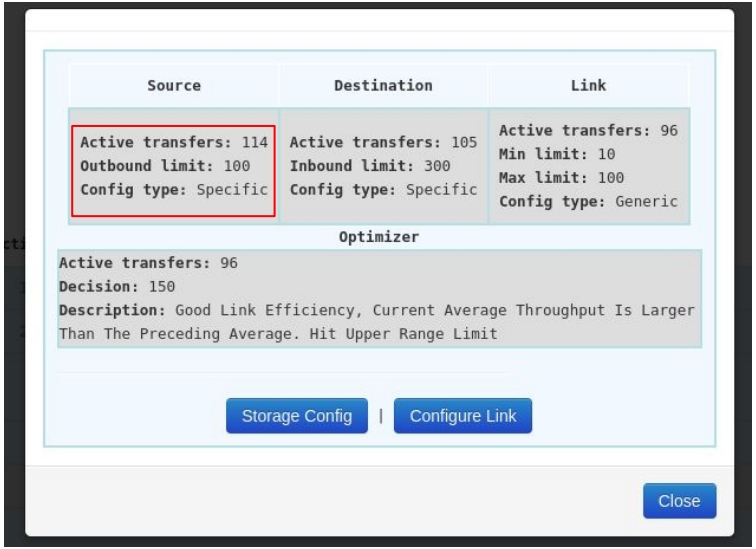
… and many others

# Web Monitoring “linkinfo” feature

- Allows transfer operators to quickly identify why a link is not progressing
  (e.g.: number of transfers for a <src> → <dst> link is “subpar”)
- A pain-point in a transfer operator’s day-to-day life, many times requiring assistance from the FTS team

- The answer lies in the storage and link limits, which have now been made fully transparent and easily accessible via the Web Monitoring
  (e.g.: fts3-atlas.cern.ch:8449)

- Successfully initiated Eraldo
  in the world of FTS development !



# FTS – Tape REST API

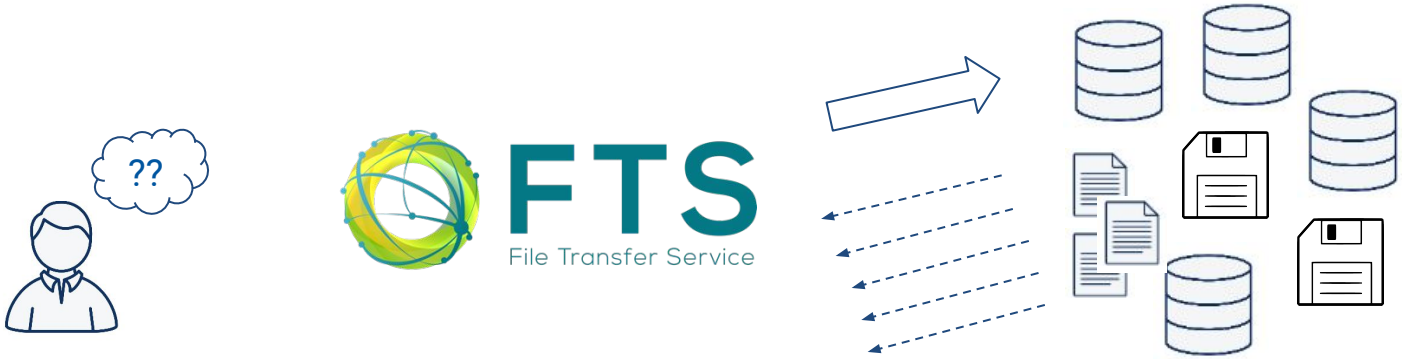
- The Tape REST API describes how to perform tape operations over HTTP
  - Designed and implemented by consortium of storage providers & FTS
  - Document link (CERNBox)
- Starting with FTS v3.12.2, full support for Tape REST API
- User-facing interaction remains almost unchanged

```
$ fts-rest-transfer-submit  --bring-online=<timeout>
                                    -s <fts-server>  davs://<src> <dst>

$ fts-rest-transfer-submit  --archive-timeout=<timeout>
                                    -s <fts-server> <src>  davs://<dst>
```

*more details in Joao's talk (Tape, REST API and more)

# FTS – The error reporting storm




`HTTP 400 / 500`
errors

Too many times, HTTP errors are not clear enough
→ FTS is in the middle of it, assisting sites and users in debugging

Need a better way of HTTP error reporting across the grid !
→ Stephan Lammel (CMS), DOMA-BDT & FTS collaboration to achieve this

# FTS – Improved TPC logs

- Involved hosts, performance markers and HTTP headers now always logged
- Reporting of involved TPC hosts done directly at the Gfal2 level
  - Gfal2 knows the TPC active party (following redirect headers)
  - Gfal2 knows the TPC passive party (from `RemoteConnections` marker)
  - Gfal2 reports this upstream to FTS (as in the example)

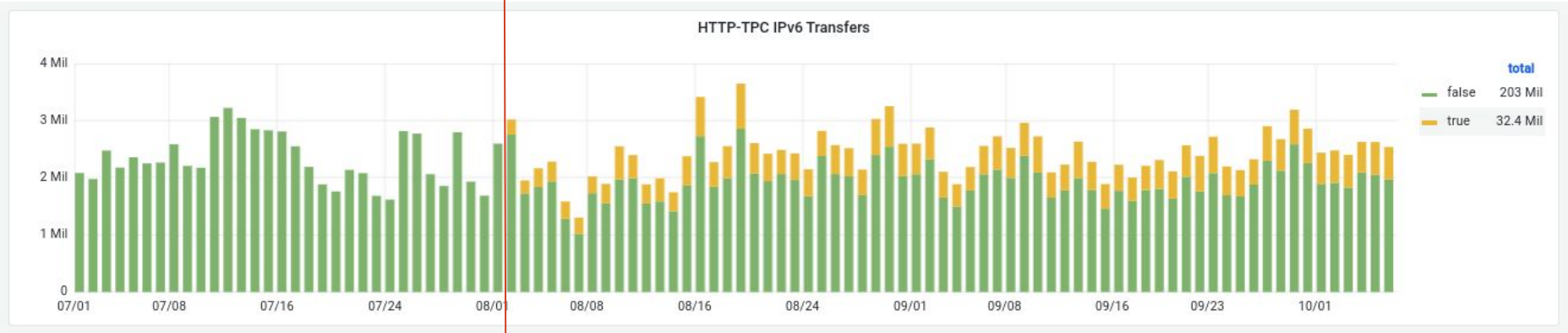
```
$ gfal-copy -v https://eospublic.cern.ch:443/eos/opstest/dteam/file.test
https://eospps.cern.ch:443/eos/opstest/dteam/file.test

event: [1670855013815] BOTH   http_plugin    TRANSFER:EXIT
https://eospublic.cern.ch:443/eos/opstest/dteam/file.test (p05151113281066.cern.ch) =>
https://eospps.cern.ch:443/eos/opstest/dteam/file.test ([2001:1458:301:17::100:e])
```

# FTS – IPv6 reporting for HTTP-TPC

- HTTP-TPC IP version extracted via Davix, propagated to FTS
- Change implemented by Ed across the many layers: FTS <> Gfal2 <> Davix

*Change deployed in August 2022 (at CERN)*



# FTS – Evolved IPv6 reporting (FTS v3.12.4)

**Objective:**

A way to tell between “not ipv6” and “no info available”

**Approach:**

```
ipv6  = true | false (boolean type)
ipver = ipv4 | ipv6 | ipv4+6 | unknown
        (string type)
```

- Introduced a new co-existing field <`ipver`>
- Existing dashboards / scripts won’t be impacted by the addition of a new field
- Concerned dashboards / scripts **will have to adopt** the new field
- The `ipv6` is no longer guaranteed long-term longevity (e.g.: may disappear in 1y+)
- Already available and deployed at CERN

# FTS – Tape metadata over HTTP (FTS v3.12.6)

**Objective:**

Allow clients to pass tape metadata over HTTP for staging and archiving (per file)


```
--staging-metadata / {"staging_metadata"} = <metadata>
--archive-metadata / {"archive_metadata"} = <metadata>
```

*Staging metadata – Passed as specified in the Tape REST API*
*Archive metadata – Passed by Gfal2 via the "TransferMetadata" header (FTS v3.12.6)*

**available to all clients, not restricted to Rucio*

# On the horizon…

# FTS v3.13.0 – “Spring-Cleaning” release

- Next major FTS release: v3.13.0
- Dubbed “Spring-Cleaning” as it focuses on removing many parts old and forgotten code components
  - `fts_bringonline` daemon
  - `fts_bdii` reporting system
  - `fts-clients` (C++ based version)
- Couple of code consolidations long due (upgraded compiler support, changing JSON library, etc)
- Expected release in autumn 2023

# FTS - Token support for exascale transfers

*Native support for a scalable infrastructure based on tokens for the critical workflows of all the LHC experiments*

Shubhangi joined FTS team in February 2023

- Will liaison with WLCG AuthZ WG
- Active FTS participation in the next AuthZ workflow

**more details in her talk (FTS & Tokens)*

- Assess current FTS software stack w.r.t. tokens
- Discuss and review main workflows with all stakeholders
- Design and develop fully integrated token support for all supported protocols
- Design and develop automated token refresh mechanism
- Design and develop fully integrated token support for cold storage
- Test individual components and service scalability
- Develop and deploy a functional testing framework
- Deploy a prototype service with full token support
- Validate main experiments' workflows
- Highlight and address shortcomings and scalability issues

# FTS – ALTO Collaboration

Collaboration between Dr. Richard Yang's group (Yale University) and FTS team

**Goal**

- Incorporate networking usage information from ALTO
- Improve FTS Scheduling and Optimizer decision
- Full details explained in Richard Yang's talk

> ALTO/TCN: Rucio/FTS Control with Deeper Network Visibility (link)

# FTS – Cloud Storage improvements

- FTS + Cloud Storage transfers requires too much expert knowledge
  - Configuring a new Cloud Storage feels like an incantation
  - Part of the changes must be done via the Database
  - Other part must be done via Gfal2 custom StorageEndpoint config

    …which often requires Puppet
- Eraldo's work will redesign the Cloud Storage config interface (REST /config/cloud_storage)
- Finishing end to a successful collaboration

# FTS - Through the user's lens


**Overview**

**Showing 1 to 50 out of 1631 from the last 1 hour**

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 ... Next Last

| Source | Destination | VO | Submitted | Active | Staging | S.Active | Archiving | Finished | Failed | Cancel | Rate (last 1h) | Thr. |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| davs://fal-pygrid-30.lancs.ac.uk | davs://xgate.hec.lancs.ac.uk | atlas | 86054 | 32 | - | - | - | 431 | - | - | 100.00 % | 194.59 MiB/s |
| davs://gdav1.phys...uni-mainz.de | davs://atlaswebdav-kit.gridka.de | atlas | 14443 | 10 | - | - | - | 12 | 33 | - | 26.67 % | 4.07 MiB/s |
| davs://gdav1...k.uni-... | davs://tech-gftp.hep.technion.ac.il | atlas | 10939 | 12 | - | - | - | 20 | 29 | - | 40.82 % | 5.17 MiB/s |
| davs://gdav1.physik.uni-mainz.de | davs://se-goegrid.gwdg.de | atlas | 9087 | 10 | - | - | - | 15 | 12 | - | 55.56 % | 5.20 MiB/s |
| davs://gdav1.physik.uni-mainz.de | davs://golias100.farm.particle.c | atlas | 8320 | 9 | - | - | - | 11 | 27 | - | 28.95 % | 3.74 MiB/s |
| davs://gdav1.physik.uni-mainz.de | davs://eosatlas.cern.ch | atlas | 7008 | 10 | - | - | - | 12 | 33 | - | 26.67 % | 5.41 MiB/s |
| davs://gdav1.physik.uni-mainz.de | davs://bohr3226.tier2.hep.manche | atlas | 6612 | 10 | - | - | - | 7 | 29 | - | 19.44 % | 2.17 MiB/s |
| davs://gdav1.physik.uni-mainz.de | davs://clrlcgse01.in2p3.fr | atlas | 5723 | 11 | - | - | - | 16 | 12 | - | 57.14 % | 4.66 MiB/s |

Our Documentation and Web Monitoring is too developer oriented

→ This causes confusion for our end users

Will need volunteers to "walk us" through the pages and highlight weak points !

# FTS - Planning session


**Service operations**

- True microservice model
- Reduce components coupling
- Add built-in service protection
- Add built-in service health monitoring

**Community requests**

- FTS global config
- Aggressive Optimizer
- User-friendly Cloud configuration
- Improved HTTP(TPC) error reporting

**Project evolution**

- High Luminosity LHC
- Modernise codebase
- Deterministic + Global view scheduler
- DMC clients evolution
- Transfers for non-WLCG environments
- Activity & Priority discussion

- Need a way to prioritise between service and stakeholder requests

**Lists not exhaustive*

# Conclusions

- Warm welcome to our FTS newcomer
- FTS at the heart of WLCG topics such as HTTP-TPC evolution, token transition and HTTP + Tape activity
- Many features delivered throughout the years

  → need some time for code consolidation
- Many items to do → sync with stakeholders on prioritization

  (upcoming FTS Planning session)
- The time for High Luminosity planning is here

# Thank you!

- Issue tracking: JIRA FTS / DMC
- Code: Gitlab (CERN) → mirrored on Github

https://gitlab.cern.ch/fts/fts3
https://gitlab.cern.ch/dmc/gfal2
https://gitlab.cern.ch/dmc/davix

- **Non-formal communication:** ~IT-FTS (Mattermost, CERN)
- **E-mail:** fts-devel@cern.ch / dmc-devel@cern.ch
- **Announcements:** fts3-steering@cern.ch
- **FTS Service Managers:** fts-ops-forum@cern.ch
- **User support:** fts-support@cern.ch / Service Now (CERN) / GGUS (experiments, WLCG)

- Documentation: cern.ch/fts3-docs, cern.ch/dmc-docs, cern.ch/fts

FTS
File Transfer Service

# *Questions?*