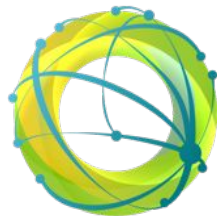




DIRAC & Rucio Workshop 2023



FTS

File Transfer Service

FTS 2023: State of (Token) Affairs

Mihai Patrascoiu
on behalf of the FTS team



Outline

- I. FTS in numbers
- II. FTS Development: Old & New
- III. FTS & DataChallenge '24

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 (Python developer) [CERN]

Collaborators

- Dr. Richard Yang [started 2023] [Yale University, USA]

Notable Past Contributors

- Ed Dambik [ended 2022] (C++ developer) [ATLAS / Indiana University Bloomington, USA]
- Eraldo Silva Junior [ended 2023] (Python developer) [ATLAS / LHCb / CERN / CBPF, Brazil]

...and thanks to many other past and present contributors



FTS Community



- FTS instances @ CERN: 5 WLCG, 1 general + EGI instance
- Outside CERN: 4 WLCG instances, 18 non-WLCG instances
 - 2 new ones: SKA @ RAL + INFN, Italy
- **First FTS + XRootd Workshop (Ljubljana, March 2023) !**
 - Next workshop? Est. September 2024
- Started collaboration with ATLAS + Yale University (Dr. Richard Yang)
- Looking towards Discourse forum

FTS 2023 in numbers

	# Files Transferred	Volume Transferred	# Files Staged	Tape Volume Staged
FTS3-ATLAS	373 M	702 PB	26 M	94.6 PB
FTS3-CMS	169 M	207 PB	5.3 M	20 PB
FTS3-LHCb	8.1 M	22.3 PB	215 K	2.21 PB
FTS3-DAQ	5.4 M	6.7 PB	12 K	0.05 PB
FTS3-Public	11.9 M	14.0 PB	5.6 M	7.8 PB
FTS3-Pilot	22.2 M	35.4 PB	434 K	0.9 PB
ALL CERN FTS3	590 M (-10%)	987 PB (+29%)	37.5 M (+23%)	126 PB (+36%)
ALL FTS3 Instances	726 M (-9%)	1.21 EB (+29%)	48.7 M (+25%)	162 PB (+36%)

* Reference period: 1st Jan – 1st Oct 2023 (+ comparison with last year)

* Trend: slightly less files, larger volume, more tape staging

FTS Development: Old & New

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 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

FTS v3.13 (2023 ?)

- Code consolidation
- OAuth2 Tokens Rework
- Dropping legacy components

FTS – Recent updates (still v3.12.x)

- **Battle-tested Tape REST API**
- HTTP-TPC IP reporting reworked
- Staging & Archiving metadata over HTTP
- Always-unpin (needed for Tape REST API)
- Configure Storage with “Full” or “Partial” TPC capability
- **SciTags**
- Improved TPC logs
 - IPs of source & dest storages
 - HTTP header always present
- Optimized database queries
- DMC & FTS clients published to EPEL

... and many others

FTS – v3.13.x series (ongoing)

- **OAuth2 tokens rework**
- Code consolidation
 - Moved codebase to C++17 features
 - Addressed all compiler warnings
 - Dropping “unmaintained” library dependencies
- Migration to Alma9: some perils ahead
 - OpenSSL v1 → OpenSSL v3
 - Python3.6 → Python3.9
 - SQLAlchemy v1 → SQLAlchemy v2
- Optimizer & Scheduler improvements
 - Optimizer decision must consider Storage Limits → less starvation between links
 - Prevent scheduler favoring certain activities → “deficit-based round-robin”
(Collaboration w/ Yale group)
- Cloud support: enhanced config pages
(Merge Eraldo’s work)

~~Ending~~ remarks

Mid-presentation

- Current year brings less feature-rich FTS
 - ... but much needed technical debt pay-off
- Alma9 will trigger many dependency re-evaluations
- More community involvement with workshop (+ Discourse to follow)
- Tokens are a hot topic

FTS & DataChallenge '24

FTS & DataChallenge '24

Two items identified as critical for the DataChallenge '24

- SciTags (Packet Marking)
- OAuth2 Tokens (WLCG)

FTS & DC'24 – SciTags

FTS must propagate SciTags (passed by the client) to the involved TPC storages

- Requires work in full FTS stack:

FTS submission > FTS server > Transfer Agent > Gfal2

- [FTS-1829](#): FTS3 SciTag work → (fts3 3.12.11)
- [DMC-1344](#): Gfal2 Scitag work → (gfal2 2.22.0)

SciTags: Development

Development in accordance with

Flow and Packet Marking Technical Specification ([link](#))

Summary:

- FTS support only for HTTP protocol
 - o XRoot must use opaque info: ?scitag=<scitag>
- Header sent always to **active** party
- Validation: $2^{16} < \text{scitag} < 2^{32}$

```
HTTP COPY /<path>
....
Source: <source_url>
SciTag: <scitag>
....
```


SciTags: Submission

Example:

```
$ gfal-copy --scitag <scitag> <src> <dst>
```

```
$ fts-rest-transfer-submit -s https://fts3-pilot.cern.ch:8446/  
  --scitag <scitag> <src> <dst>
```

JSON
submission

```
{  
  "files": [  
    { ...  
      "scitag": <scitag>,  
      ...  
    }  
  ]  
}
```

FTS & DC'24 – Tokens

Token workflow & development plan presented at [DOMA-BDT](#) (21st June 2023)

- FTS Token Document: requirements, specifications & development guide
- [FTS-1925](#): JIRA Epic with breakdown of tasks
- Client side + transfer submission adaptations first
 - o Development well on its way

FTS & DC'24 – Tokens Approach

The FTS team wants to test the standard OAuth2 as much as possible during DC24! (general use-case for most communities; IAM for scalability)

- FTS will support the standard OAuth2 flow for DC24 (token-exchange + refresh)
- Each transfer submission requires: 1 token for source, 1 token for destination
- The client has to ensure FTS can perform overwrite / clean-up on destination

The FTS Tokens Document

FTS team sends appreciation for the Rucio Token Design Document ([link](#))

FTS Token Document in the works:

- Will cover token-base submission & FTS interaction
- Target release date: DataChallenge Workshop (November 2023)
- As implementation takes place, document will evolve (FTS token-lifecycle)

Tokens: At a glance

FTS interaction will require a token, called: **<fts-token>**

- Used by FTS to identify and authorize the client request (sub, VO, [wlcg.groups])
- **May require** `aud` (FTS service) and FTS-specific `scopes` (names not decided yet)

FTS transfers will require tokens for src/dst: **<src-token>** & **<dst-token>**

- Every transfer (submission) will have at least two tokens

Tokens: Client submission

```
$ fts-rest-whoami --fts-access-token <fts-token>  
-s https://fts3-pilot.cern.ch:8446/
```

```
$ fts-rest-transfer-submit --fts-access-token <fts-token>  
-s https://fts3-pilot.cern.ch:8446/  
--src-access-token <src-token>  
--dst-access-token <dst-token>  
<src> <dst>
```

```
$ curl -s -X POST -H "Authorization: Bearer <fts-token>"  
--data @submission.json  
https://fts3-pilot.cern.ch:8446/jobs
```

Tokens: Client submission

A token for each transfer must be provided!

```
{
  "files": {
    "sources": [URL1, URL2, ...],
    "destinations": [URL3, URL4, ...],
    "checksum": <xsum>,
    "filesize": <size>,
    "metadata": <metadata>
  },
  "params": { ... }
}
```



```
{
  "files": {
    "sources": [URL1, URL2, ...],
    "destinations": [URL3, URL4, ...],
    "source_tokens": [AT1, AT2, ...],
    "destination_tokens": [AT3, AT4, ...],
    "checksum": <xsum>,
    "filesize": <size>,
    "metadata": <metadata>
  },
  "params": { ... }
}
```

DC'24 Timelines

Weeks leading to DC'24 Workshop (9-10th November 2023)

- Finish SciTags and the new Token submission interface
- Advertise publicly (fts3-steering and Rucio lists)
- Deploy to FTS instance: Pilot or token-dedicated instance

Weeks leading to DC'24

- Implementation continues: full token lifecycle management

Post DC'24

- Review of DC'24 performance
- Move discussion towards tape + tokens

Conclusion

- Short-term plan for DC'24 Workshop
 - Submission API done by then (Rucio needs something to test against!)
 - Publish simpler, use-case focused FTS Token document
- DC24 is considered perfect time to test **standard** OAuth2 flow
- Re-evaluate post-DC24: further Rucio ↔ FTS optimizations as needed

Thank you!

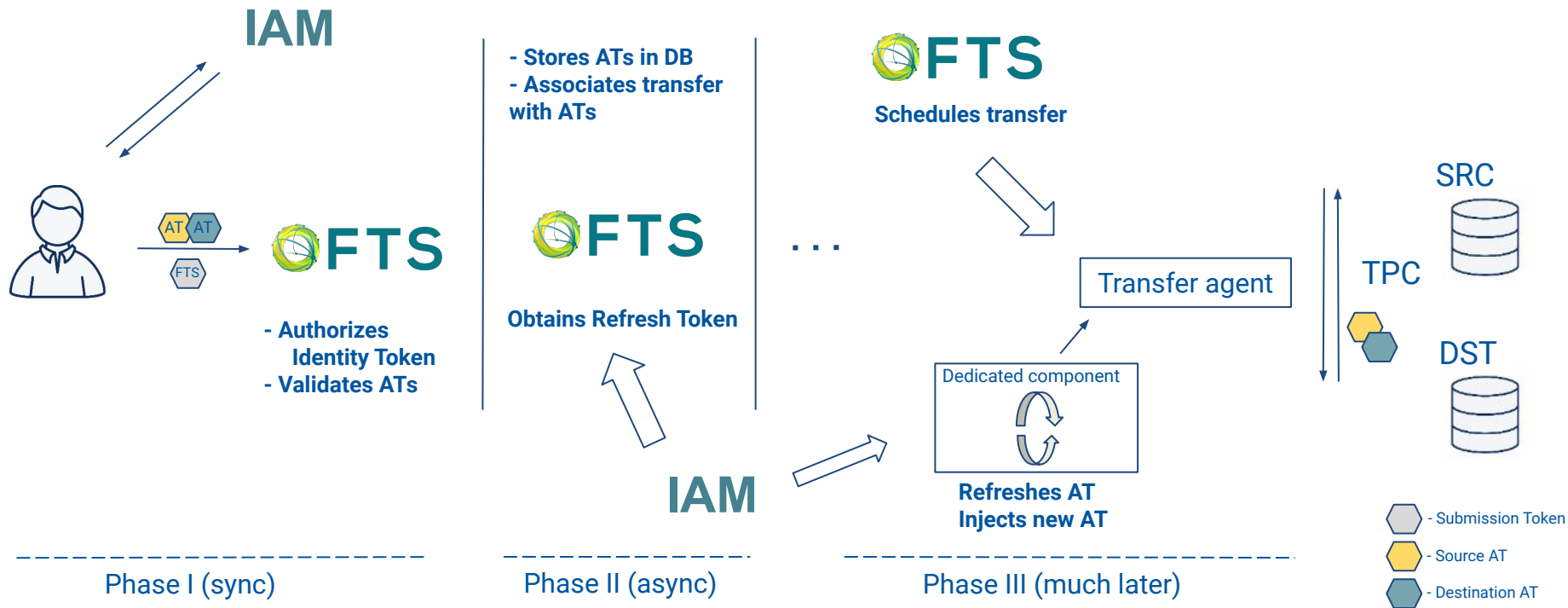
Backup

FTS token lifecycle management

Token lifecycle management steps:

1. FTS receives AT from the submitting client
2. Token is validated and associated with transfer in FTS database
3. A refresh token is obtained for the AT as soon as possible (async from submission)
4. Transfer is scheduled with token loaded from the database
 - a. Dedicated component refreshes ATs past expiration threshold via **token-exchange**
 - b. Dedicated component can re-inject token to transfer when needed

FTS token lifecycle management



FTS token lifecycle management

New developments needed:

1. Handling of identity token
2. New Submission API (two ATs per transfer)

3. (Async) "Token-exchange" component
4. (Just-in-Time) "Token-refresh" component

5. Injecting token credentials to Transfer Agent
6. Gfal2 loading tokens from credential file (build on [cern-fts/gfal2#13](https://cern-fts.github.io/gfal2/#13))

Client facing
(first to arrive)

Impacts IAM

Anatomy of the transfer process

