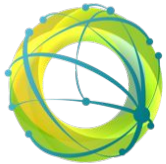
CHEP 2024

Kraków, Poland, 21 Oct - 25 Oct 2024

CHEP 2024



FTS

File Transfer Service

Token Support for a Proxy-less WLCG world

Mihai Patrascoiu
on behalf of the FTS team

CERN

mihai.patrascoiu@cern.ch (CERN IT-Storage)

FTS Team

- Mihai Patrascoiu (Project Leader) [CERN]
- Steven Murray (Service Manager) [CERN]
- João Lopes (C++ / Python developer) [CERN]
- Shubhangi Misra (Web developer) [CERN]
- Louis Regnier (C++ / Python developer) *(1st July 2024)* [CERN]

…and thanks to many other past and present contributors

FTS – At the heart of all things token

- Diverse ecosystem of experiment frameworks and technology providers (IAM, storages)
- FTS common element to link stakeholder requirements with provided technical solutions

Token approaches

I. FTS acts as “root” user
   - FTS can conveniently request tokens for any action (but no “end-user” traceability”)
   - Generally refused by the Grid community as “too powerful”

II. FTS does not interfere at all in token management
   - Clients have to delegate tokens (similar to what they do now with certificates)
   - Generally refused by the Grid community as “too complex” *(see later)*

III. FTS handles the token lifecycle
   - FTS is a superuser and can refresh other user’s tokens
   - Easy workflows for the clients, complexity in the FTS + IAM layer
   - Generally approved by the Grid community as the “desired approach”

FTS Token Plan

(presented @ DOMA-BDT, 21st June 2023)

- Standard OAuth2 token support
- Token submission: 1 token to identify with FTS

  \+ 2 tokens per transfer (source and destination)
- DC’24 is considered perfect time to test standard OAuth2 flow
- Client-facing developments will be released first
  - Other systems can follow-up on this early
- Tape plan decoupled from TPC → details to follow after DC’24

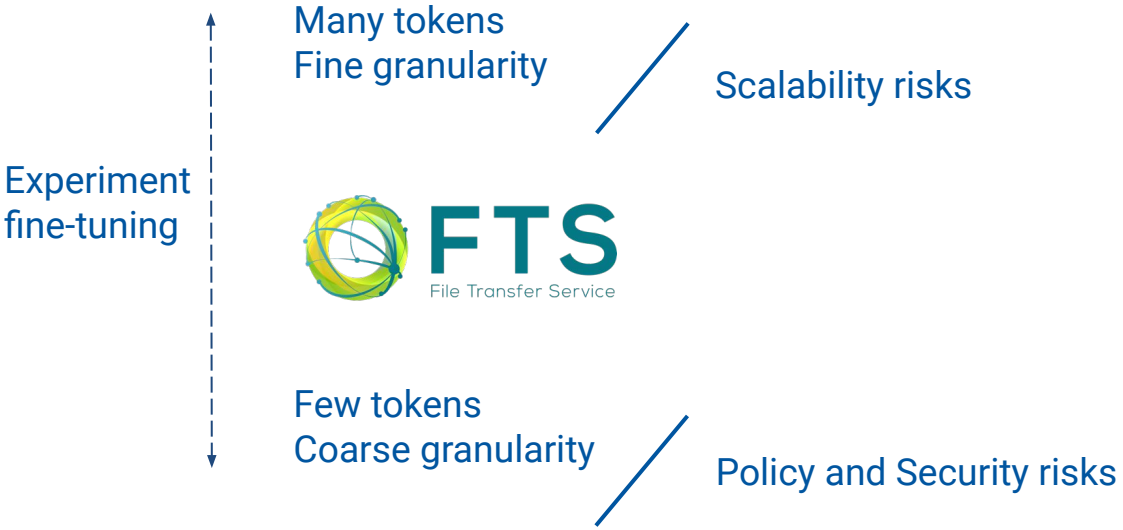
Token approach in practice

Decision

- Token to submit to FTS
- **2 tokens** for each transfer (source & destination)

Advantages

- FTS system is “agnostic” to transfer token contents
- Experiments can decide (and easily change) granularity



FTS Token Timeline

June 2023

FTS Token plan presented

“Pre-DC’24 Workshop” Development

- Token development
- Deployment on FTS Pilot/ATLAS/CMS/LHCb
- “Alpha” version of token support

Pre-DC’24

- Refined DC’24 token development

DC’24

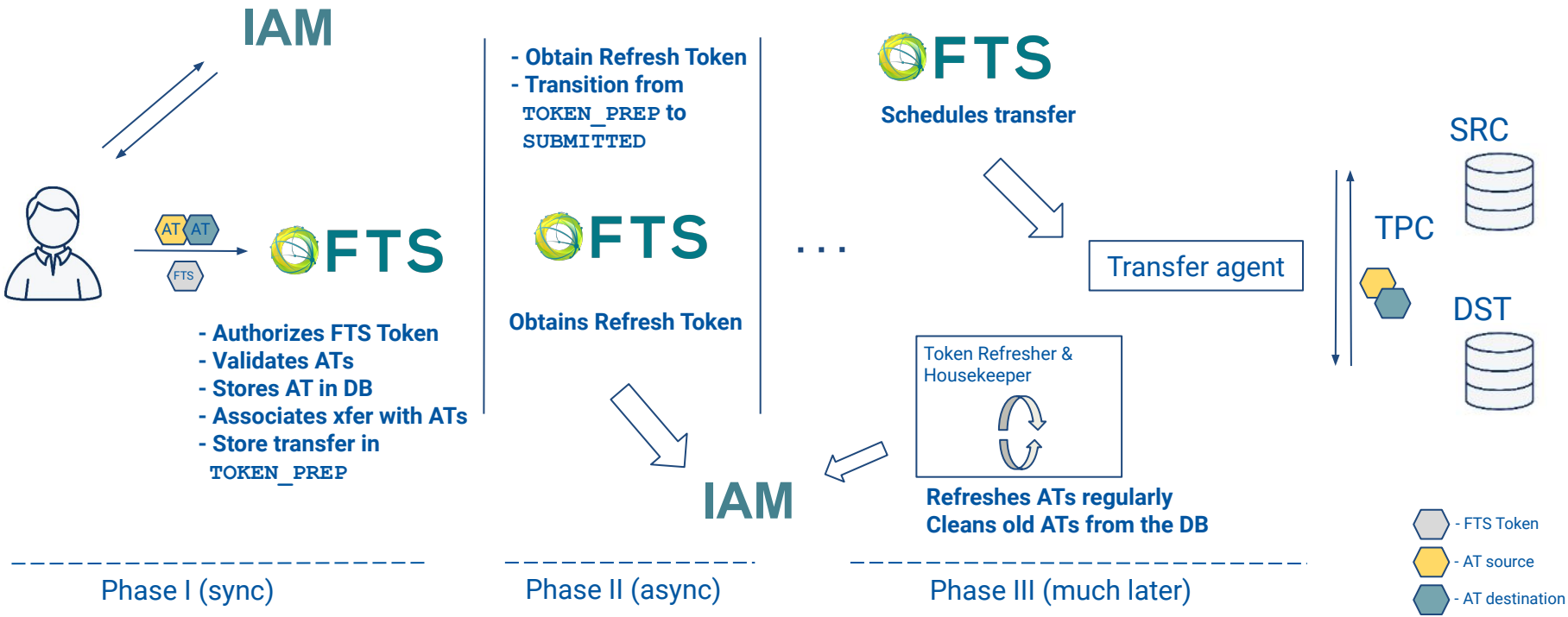
Post-DC’24 Era

- Improved congested SQL queries
- Token tests (Aug 2024)

Remaining 2024

- Just-in-time refresh?
- Tape considerations?

FTS Token Lifecycle management



DataChallenge’24 Reflections

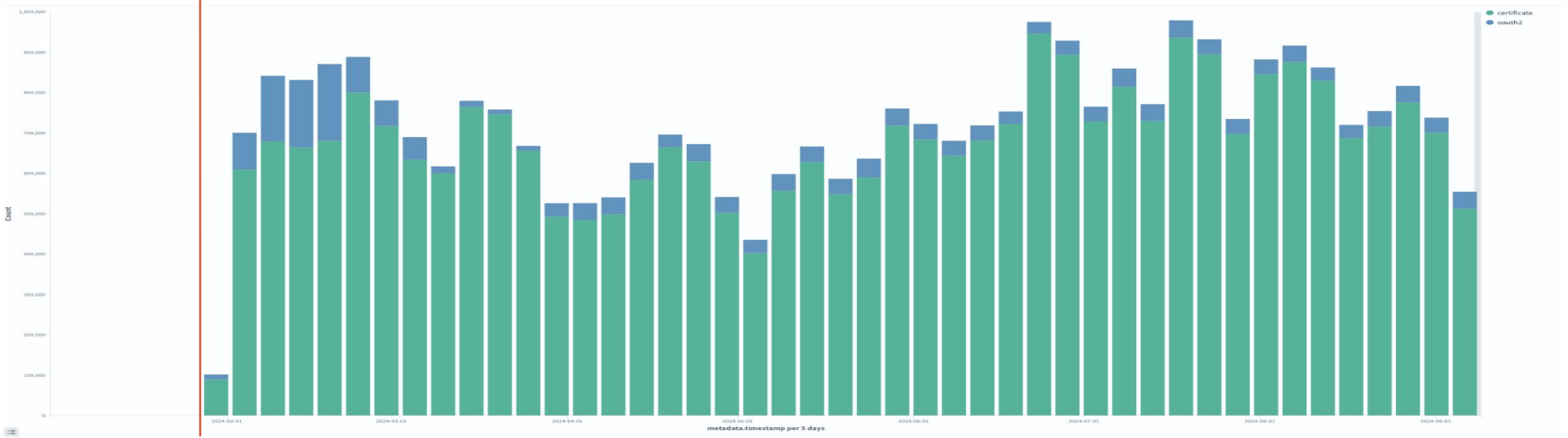
- FTS successfully pioneered WLCG transfers with token support in a high stress environment
- DataChallenge uncovered slow workflows in the FTS token management
  - Most FTS technical concerns have been addressed after the DataChallenge
  - “Just-in-time” token refresh: last missing development for disk-to-disk transfers
- Refreshing workflow is heavy, causing stress on both IAM and FTS systems
  - “Just-in-time” token refresh solution will optimize this workflow
- Monitoring update to show adoption of OAuth2 protocol

OAuth2 adoption monitoring

New field in MONIT data:

```
auth_method=<certificate|oauth2>
```

Change deployed in February 2024 (at CERN)



Current Status

- Tokens successfully deployed for DataChallenge'24, **in-use since**
- Token support deployed on all production FTS instances (available since v3.13.0)
- **"Just-in-time" refreshing**: the last development for disk-to-disk token transfers
- Different approaches within the experiments:
  - → CMS performing token tests using the standard token-refresh lifecycle
  - → ATLAS performing token tests using long-lived per-file unmanaged access tokens
  - → LHCb plans to use per-file tokens
  - → What token approach will DUNE have? (expected in 2025)
- **Missing Tape REST API + Token plan**

Just-in-time refresh?

Replace the `TokenRefresher` daemon with just-in-time refresh

- Just-in-time: refresh in the transfer agent, right before using the token
- Would ensure FTS refreshes **only the tokens it needs**
- Last piece missing to call token development “complete”

Current reliance on Gfal2 prevents this (too many operations are hidden under `gfal2.copy(..)`)

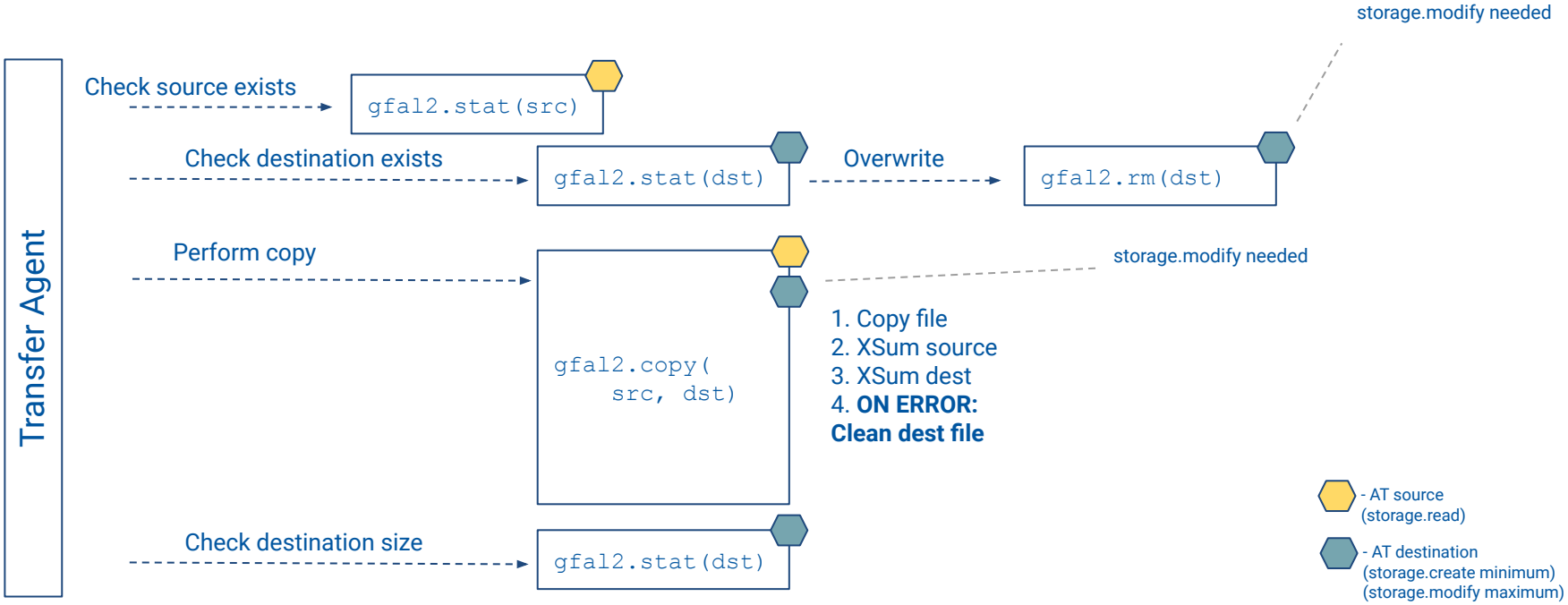
→ Break-down the large `gfal2.copy(..)` into individual sub-calls [thanks Louis!]

→ Paves the way for FTS to decouple from Gfal2

→ Allows FTS transfer agent to refresh token before any SE contact

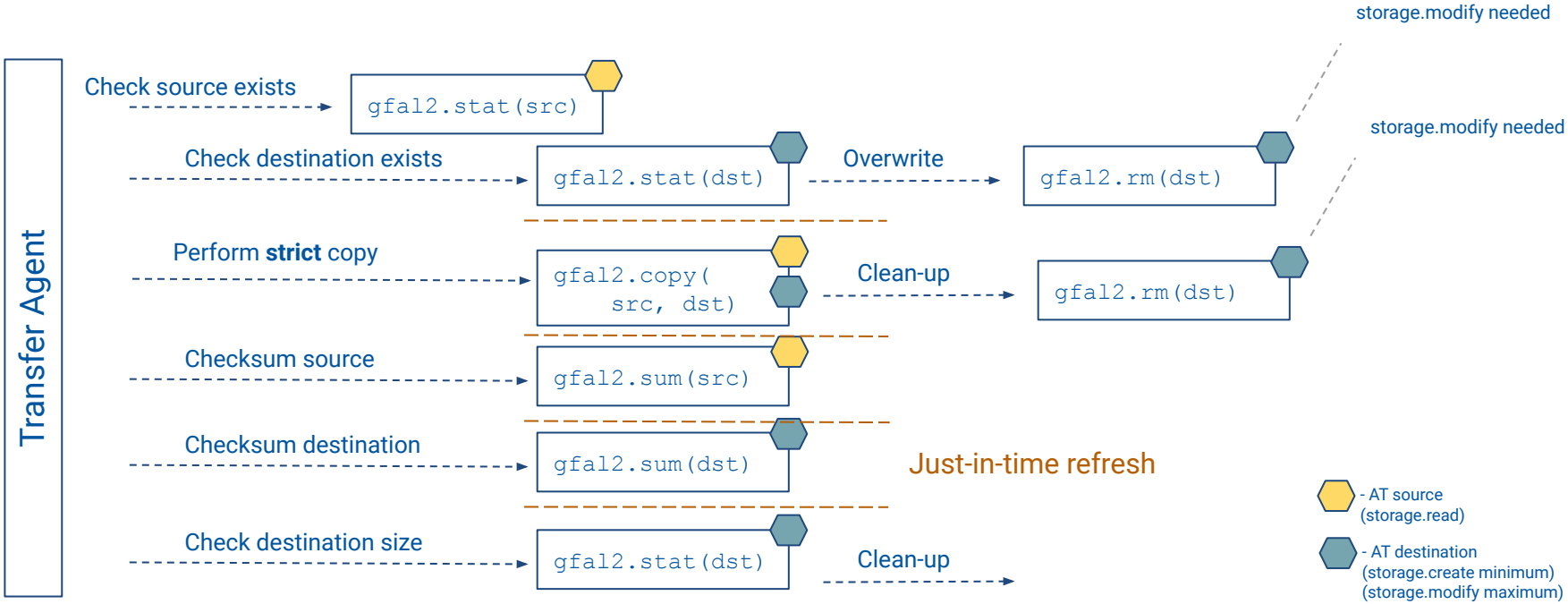
Anatomy of the transfer agent

(existing implementation)



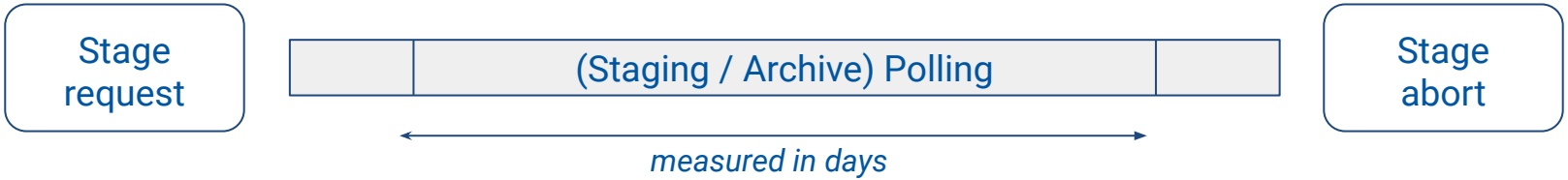
Anatomy of the transfer agent

(with Gfal2 operations breakdown + just-in-time refreshing)



Tokens + Tape?

- Tape interaction presents 3 distinct actions:



- Staging + stage abort will require FTS to manage the token lifecycle
- Is it worth it to refresh for the lifespan of the Polling operation?
- FTS creates staging batches based on (SE, credID) pairs.
  How will this look like with tokens / scopes involved?

- Tokens + Tape discussed for first time @ FTS & XRootd Workshop 2024
- Proposal for Tape REST API + Tokens work-in-progress *(ongoing Tape REST API v0.2 draft! Stay tuned!)*

Token Reflections

- Large token usage during DC’24, shy adoption afterwards
- A fragile system of many levers: Rucio, FTS, IAM
  - Can we ensure transfers continue even in the case of downtime?
- Higher service administration costs:
  - FTS has to be registered with each community TokenProvider
- The OAuth2 RFC is very broad:
  - **FTS will need to eventually enforce a token profile (likely WLCG)**
  - **Would want to see more alignment between experiments and WLCG coordination**

Token Reflections (cont'd)

- Likely, FTS will likely have to support two token solutions:
  - Targeted solution for large frameworks, such as Rucio & DIRAC
  - Full token lifecycle management for smaller communities or clients

- **2025 should bring Tokens + Tape prototype**

- **2025 should bring larger token adoption for WLCG transfers**

Thank you!