Revised Token Workflow

Mihai Patrascoiu for the FTS team
DOMA-BDT (21st June 2023)

Outline

I. How clients authenticate to FTS

II. How clients submit transfers to FTS

III. How FTS manages the token lifecycle

IV. Approach for DC24

V. Extended support beyond DC24

I. The “Identity token”

Clients must submit a token to FTS which will be used to:

- Identify the user within FTS (construct credential identity)
- Perform authorization on this credential identity
- Associate transfers to this credential identity

I. The “Identity token”

Constructing credential identity:

```
{
  "wlcg.ver": "1.0",
  "sub": <uuid>,
  "aud": "wlcg/any",
  "iss": "WLCG IAM",
  "scope": <scopes>,
  "wlcg.groups": <groups>,
  "exp": 1687269211,
  ...
}
```

Hashing function

```
credential_id
=
c1a58151d77cfa
8b
```

I. The “Identity token”

Parallels with proxy certificates:

Field name	Proxy certificates	OIDC Tokens
User ID	DN	Sub
VO	VO	Issuer
Roles	FQANs	WLCG Groups
Credential ID	Hash(DN, VO, FQANs)	Hash(Sub, Iss, Groups)

I. The “Identity token”

Example:

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

```
$ curl -s -H "Authorization: Bearer <token>"
                       https://fts3-pilot.cern.ch:8446/whoami
```

--identity-token? --auth-token? (Field name may still change)

II. Submitting to FTS

For token-based transfers, 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": { ... }
}
```

II. Submitting to FTS

Example:

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

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

**FTS may refuse submission if missing transfer AT*

III. FTS token lifecycle management

In order to successfully schedule transfers, FTS **must be allowed** to refresh tokens.

Prerequisites:

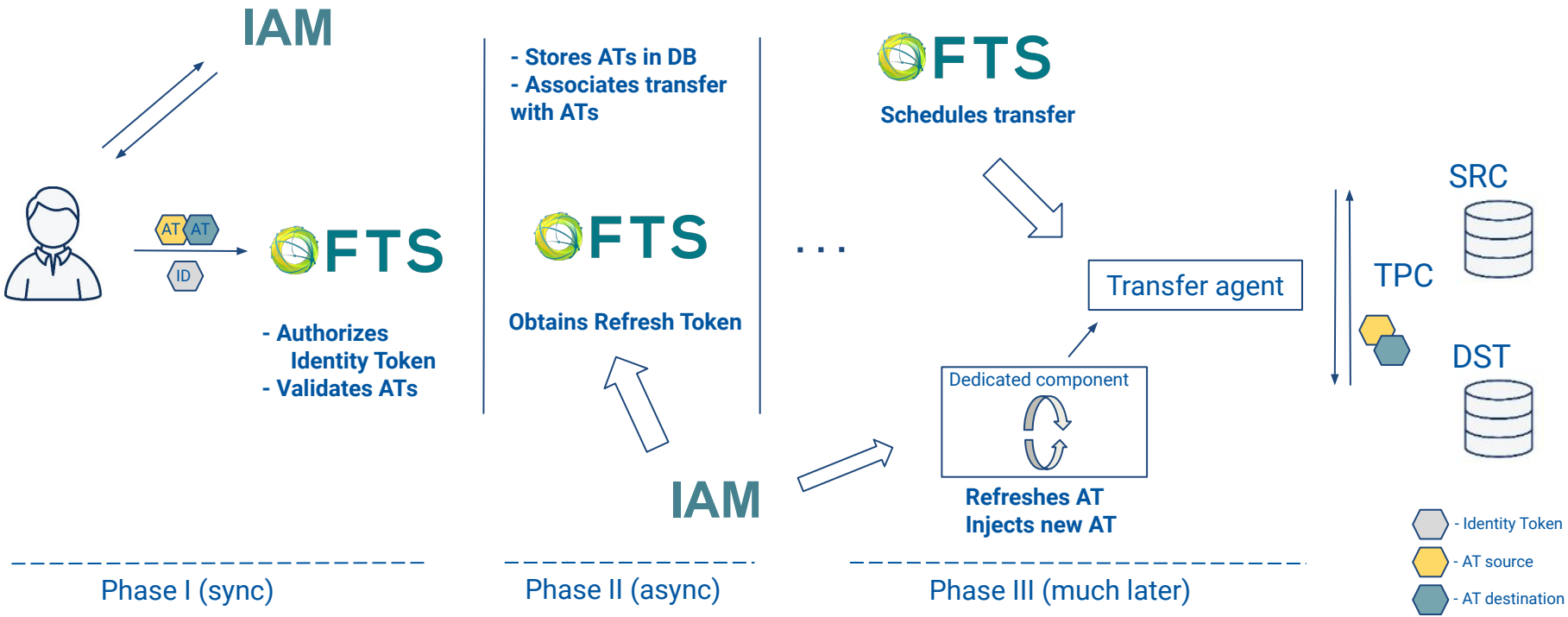
- FTS client ID must be allowed refresh capabilities by the Token Provider (config)
- The token issuer must be recognized and configured within FTS (runtime)
- The access tokens must contain the `"offline_access"` scope (runtime)
- The token must be valid via offline validation (runtime)

III. 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

III. FTS token lifecycle management



III. FTS token lifecycle management

FTS will generate unique token hashes.

Access tokens may be deduplicated in order to optimize database usage.

```
{
  "wlcg.ver": "1.0",
  "sub": <uuid>,
  "aud": "wlcg/any",
  "iss": "WLCG IAM",
  "scope": <scopes>,
  "wlcg.groups": <groups>,
  "exp": 1687269211,
  ...
}
```

Hashing function

```
token_id
=
b8afc77d15185a
1c
```

III. FTS token lifecycle management

Associating a transfer with an access token:

t_file	File ID	Job ID	...	Token ID
	5375461456	cb72af2e...6e	...	**b8afc77d15185a1c**

t_token	Token ID	Token	Issuer	...	Scopes
	b8afc77d15185a1c	eyJraWQ...so	WLCG IAM	...	"storage.read:/"

III. FTS token lifecycle management

New developments needed:

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

Client facing
(first to arrive)

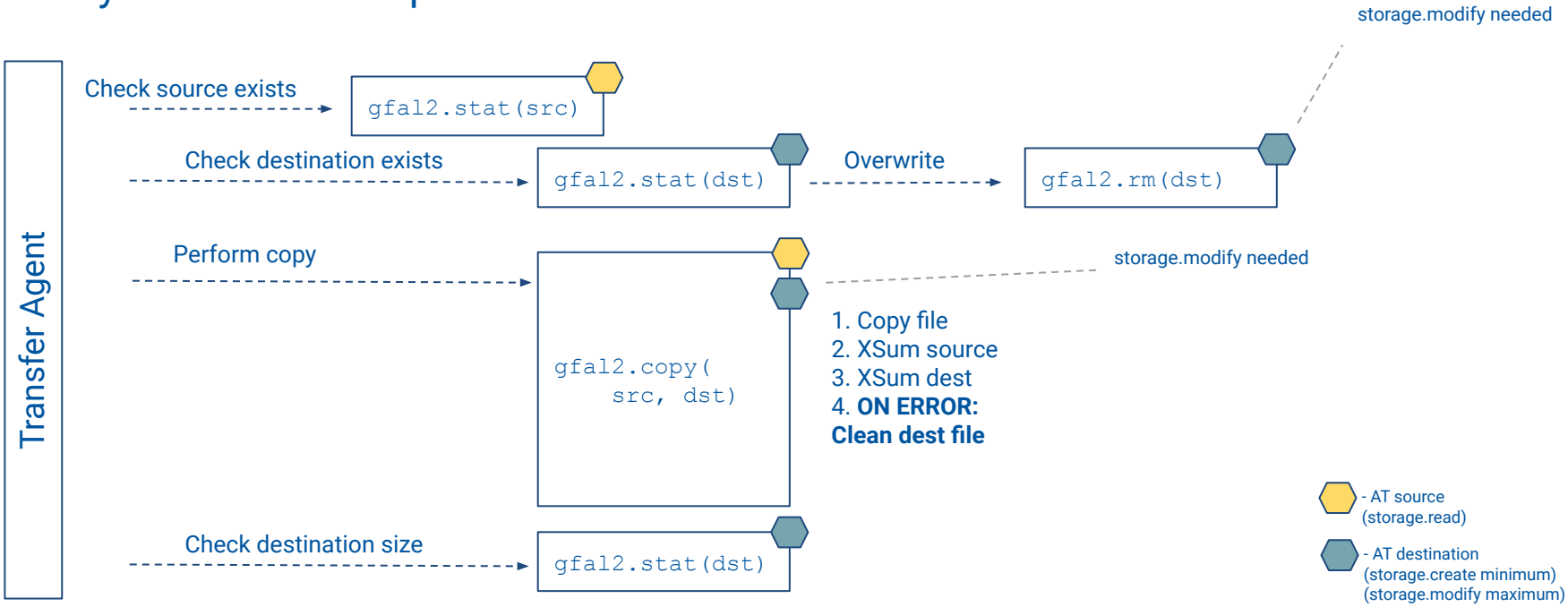
3. (Async) "Token-exchange" component
4. (Just-in-Time) "Token-refresh" component

Impacts IAM

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

IV. Considerations for DC24

Anatomy of the transfer process:



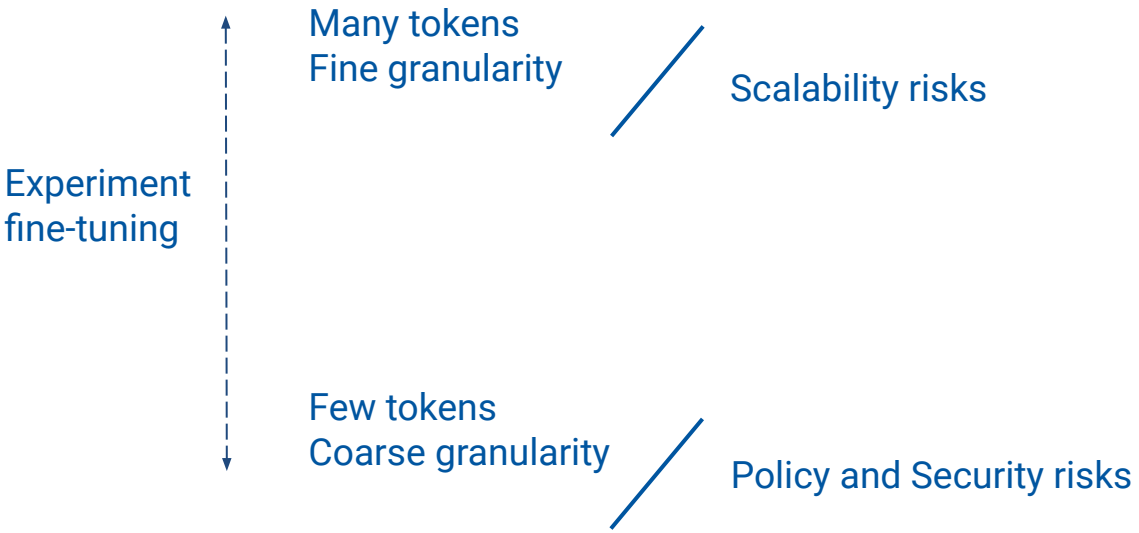
IV. Approach for DC24

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

IV. Approach for DC24

FTS approach will be able to accommodate both scenarios:



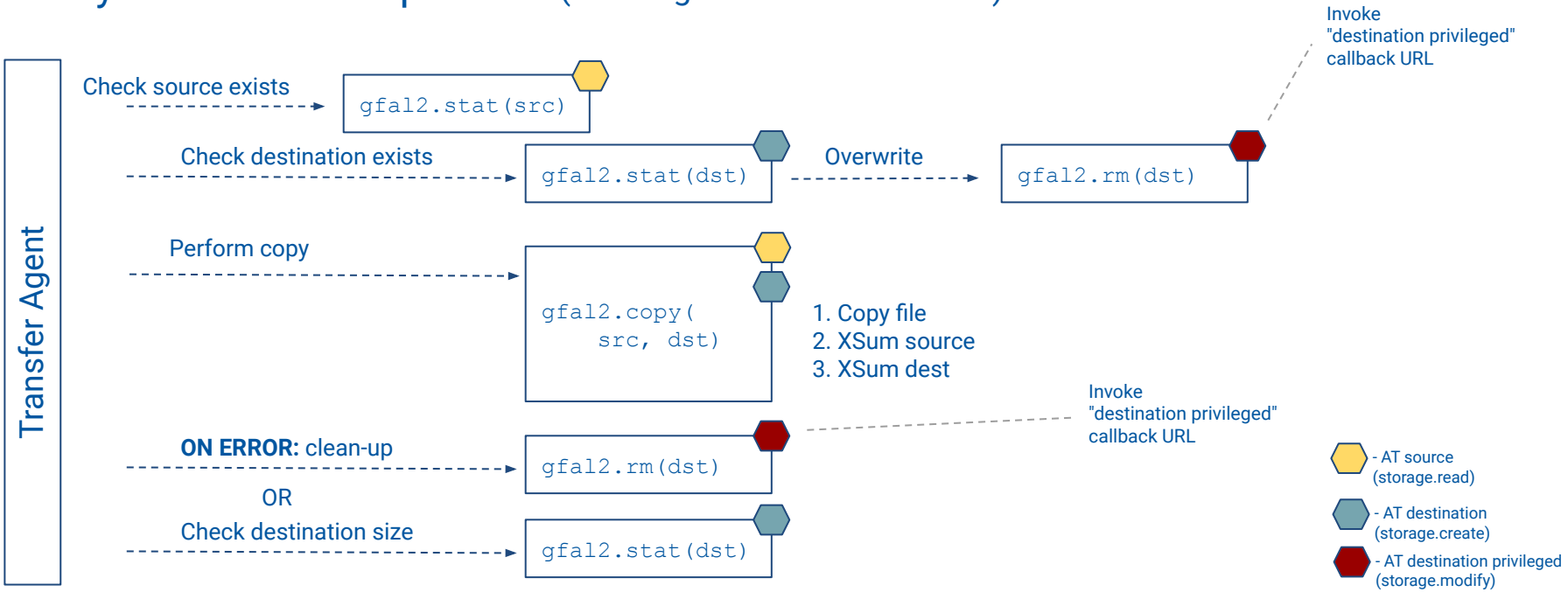
V. Extended support beyond DC24

Together with the Rucio team, several scenarios discussed in order to keep fine granularity, but reduce the number of tokens. Proposed solution:

- FTS **does not** do token-exchange and token-refresh
- FTS receives a callback URL (from Rucio) for the token resource
- When FTS needs a new token, it invokes the callback URL
- Rucio provides the token to FTS
- Trust relationship established by pre-signing the callback URL

V. Callback URL approach (at a glance)

Anatomy of the transfer process (invoking token callback URLs):



V. Callback URL approach

Additional developments needed:

1. Extend Submission API:
   - 3 callback URLs per transfer: `read_src, create_dst, modify_dst`
2. Bypass "token-exchange" and "token-refresh" in this scenario
3. Implement callback invocation mechanism in Transfer Agent

Conclusion

- Standard OAuth2 token support priority for the FTS team
- DC24 is considered perfect time to test standard OAuth2 flow
- Client-facing developments will be released first
  - Other systems can follow-up on this early
- Further scenarios envisioned for post-DC24
- Tape plan decoupled from TPC → details to follow after DC24