



An overview

Carles Garcia Cabot  
FTS Team  
16/01/20

# Example

Get an access token and submit a job using fts-rest-cli (package in EPEL)

```
$ export tok=`oidc-token wlcg`
```

```
$ fts-rest-transfer-submit \  
-s https://fts3-xdc.cern.ch:8446 \  
--access-token $tok \  
https://prometheus.desy.de/Users/carles/source \  
https://prometheus.desy.de/Users/carles/destination
```

```
Job successfully submitted.
```

```
Job id: 2fb22734-360f-11ea-9524-fa163e362acc
```

# Token workflow



1a. Validate access token using cached keys from issuer

1b. Validate access token using Introspection (RFC 7662)

2a. If a credential already exists in the DB and has not expired, discard the new token and use the stored one.

2b. If a credential already exists in the DB but has expired, delete it.

3. If no credential exists, introspect the token to get additional information (if not done before). Then, exchange the access token with a refresh token. Store both tokens in the DB.

4. The transfer is submitted

5. A daemon refreshes the access tokens periodically

# Implementation details

We implement “Scenario 1” partially. Not implemented yet:

- audience
- scopes

OpenID Connect integration with fts-rest has been reimplemented with pyoidc

- <https://github.com/OpenIDC/pyoidc>
- certified by The OpenID Foundation

Starting soon: migrating fts-rest to Python3 and Flask