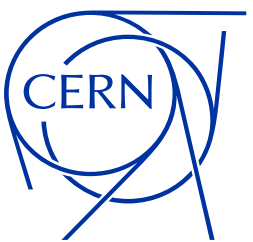
FTS & Rucio

Dimitrios Christidis
for the Rucio Team



Introduction

- Rucio is becoming the distributed-data management software of choice for many scientific communities
- Rucio doesn’t implement site-to-site transfers; instead, it relies on external transfer tools
  - FTS is ubiquitous
  - Globus is becoming important for some HPCs
  - BitTorrent is intended for small-scale evaluations of Rucio
- Very productive collaboration with the FTS team

How Rucio uses FTS (abbreviated)

- Mainly two operations:
  - Submit new transfers to FTS
  - React when existing transfers reach a terminal state
- Do those as quickly as possible
- Allows Rucio to offload an enormous amount of responsibility onto FTS
  - Crucial that FTS is sufficiently supported by CERN IT

Tokens

- FTS and Rucio have had limited token support since 2020
- Replace X.509 with a ‘fat’ token
  - Common for both source and destination
  - No audience restriction
  - No capability-based restrictions
- Before DC24, FTS and Rucio worked closely together on a new implementation for third-party-copy transfers

Token payload

- Tokens are an ‘industry standard’, but TPC transfers are not
- Too many options, too many questions, little prior experience
- What can we control?
  - Subject
  - Audience
  - Scope (including resource path)
  - Lifetime (limited)

The new TPC workflow implementation

- One token for the source, one for the destination
- Rucio is entirely responsible for the token payloads
- FTS must refresh the tokens until no longer necessary
- Rucio must cache and reuse tokens as much as possible
  - In fact, most of our concerns at the time were about the scalability of the token provider

Token subject

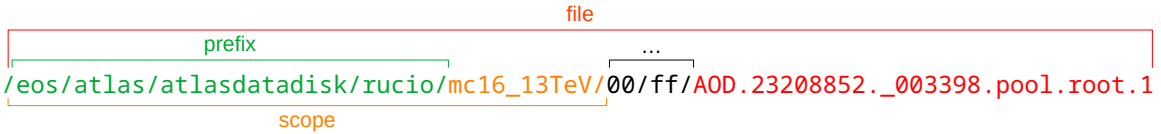
- Only one; all tokens ‘belong’ to Rucio itself
- No tangible benefit in doing anything else
- An easy choice to make

Token audience

- Restrict the token to a specific storage
- Originally, lack of consensus on how to populate it
  - Storages do a simple string comparison
  - Rucio encouraged to use the URL host
- The cheapest way to improve security
  - Amount of tokens scales with the number of storages

Token scope (source)

- Only the `storage.read` scope to choose from
- Risk assessment is low
- Can be the the prefix of the Rucio Storage Element (RSE)
  - The amount of tokens scales with the amount of RSEs



Token scope (destination)

- Need to make a choice between the `storage.create` and `storage.modify` scopes
- Risk assessment is low for the former but high for the latter
- FTS needs to be able to delete, under certain conditions
- Use an RSE-wide `storage.modify` for the needs of DC24
  - But review afterwards

Token lifetime

- Sadly, cannot be controlled by Rucio
  - The configuration is done in the token provider, per client
- Shorter lifetimes may be preferable for security, but increase the amount of tokens
- Decided on six hours, as described in the WLCG profile
  - That recommendation proved to be a source of confusion

Starting small

- The prototype for the new implementation must be proved to work before additional capabilities can be added
- This means:
  - Only INDIGO IAM
  - Only the WLCG token profile
  - Only disk storages
  - Only WebDAV protocol
  - RSE-wide tokens
  - No configurability of any kind

Leading up to DC24

- Deployments of FTS and Rucio that supported the new implementation became available in late 2023
  - Both projects advertised it as a technology preview
  - Barely two months before the commencement of DC24
- Major effort by the experiments to enable tokens at as many sites as possible
- A limited file-specific token test validated our concerns

During DC24

- Lack of prior experience put a major strain on the operations teams
- The token refresh was the source of some problems
- The choice of token lifetimes was unfortunate

Aftermath of DC24

- Was it a success? Absolutely! However:
  - Came at a non-insignificant cost
  - The goal was far from ambitious
- Following DC24, ATLAS and CMS reduced or disabled the use of tokens due to security concerns
- After a hiatus, FTS and Rucio returned to the drawing board

On-going experimentation

- ATLAS:
  - No token refresh workflow at all
  - Greatly increased token lifetime (multiple days)
  - File-specific destination tokens
- CMS:
  - See next talk by R. Chauhan

Near-term goals

- Refine the TPC workflow and start adding configurability
  - Rucio must be able to confidently offer a recommended configuration to its communities
- Design and implement token support for tape storages
- Must also commence work on the client workflows
  - The WLCG token transition timeline expects this in Q1 2025
- Reminder: support for CILogon is frequently requested

Profile compliance is a challenge

- Paraphrasing P. Vokac: ‘everything is described in the standards, so the behaviour is implementation dependant’
- Two kinds of divergence:
  - Not authorising tokens that should be (annoying)
  - Authorising tokens that shouldn’t be (scary)
- This has to be a collective effort

Developments unrelated to tokens

- Source selection strategies
- Finer control of overwrites for tape storage
- Tape metadata (scheduling and co-location hints)
- Improved support for commercial clouds

Plans for GFAL

- Rucio is making use of GFAL for central deletions and client workflows (i.e. upload and download)
- Was very necessary for SRM and GridFTP, but we would prefer to move away from it in the future
  - By offering bespoke implementations for WebDAV and XRootD protocols
  - Must now take into consideration Monday's discussion
- Paraphrasing C. Haen: 'the GFAL CLI utilities are the basis of easily reproducible manual tests'

Questions?