Rucio Token Workflow Evolution

Martin Barisits, <u>Dimitrios Christidis</u>
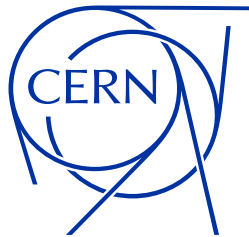
Martin Barisits, Dimitrios Christidis

Introduction

- The Rucio Tokens SIG has been launched

- Version 0.1 of the design document is now available

  - Expresses our current understanding and plans

  - Will continue to evolve with the development and together with input from the community

  - Some topics will be added as soon as they become clear

  - Some existing details might change

- New development may supersede existing functionality
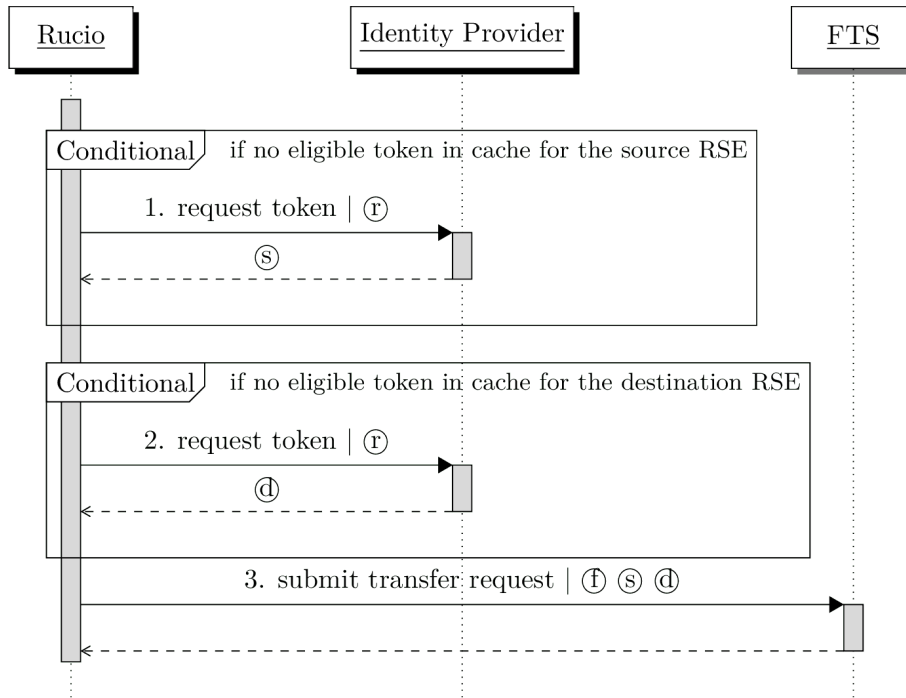
The Workflows

- Central
  - Third-Party-Copy Transfer
  - Deletion

- User
  - Authentication
  - Download
  - Upload

Third-Party-Copy Transfer (1)
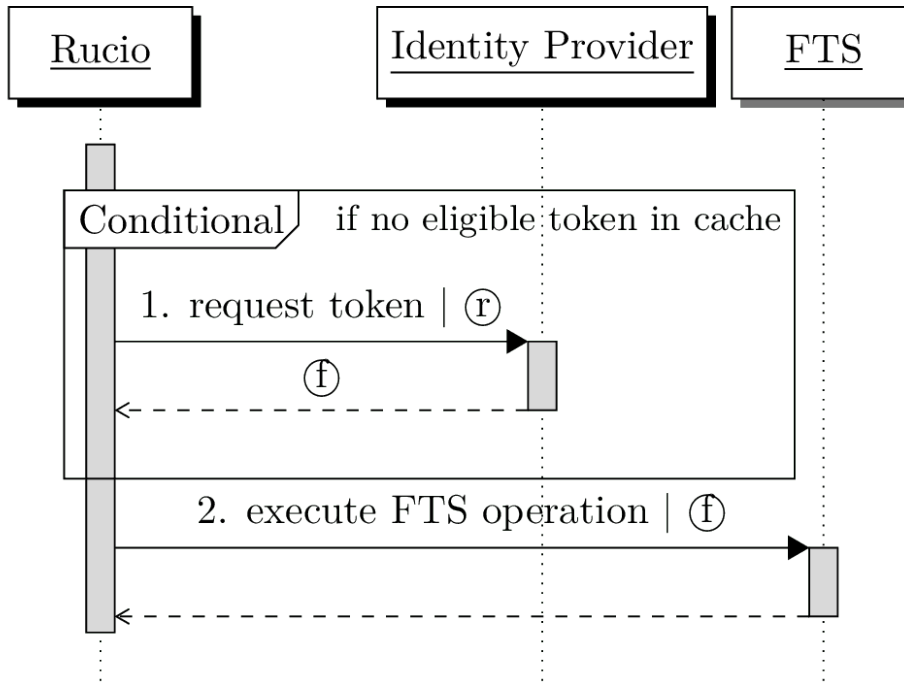
- First, obtain a token providing authentication with an FTS instance

- Erratum: authentication method must match what is used for the transfer

Third-Party-Copy Transfer (2)

- Second, obtain tokens for reading from the source RSE and writing to the destination RSE, then submit the transfer request

- Similar for more complex transfers (e.g. multi-source, multi-hop, tape storages)

Third-Party-Copy Transfer (3)

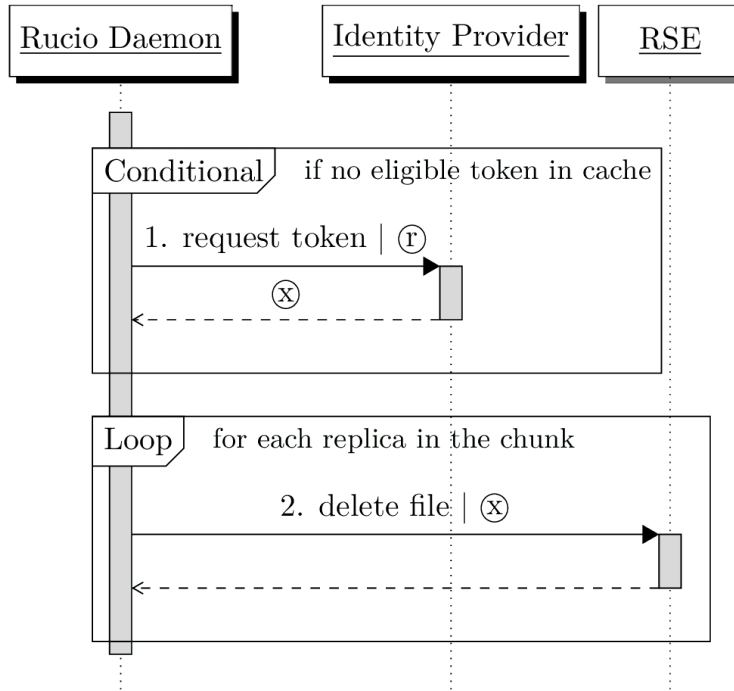- Third, FTS is responsible for refreshing the provided tokens until the transfer is activated
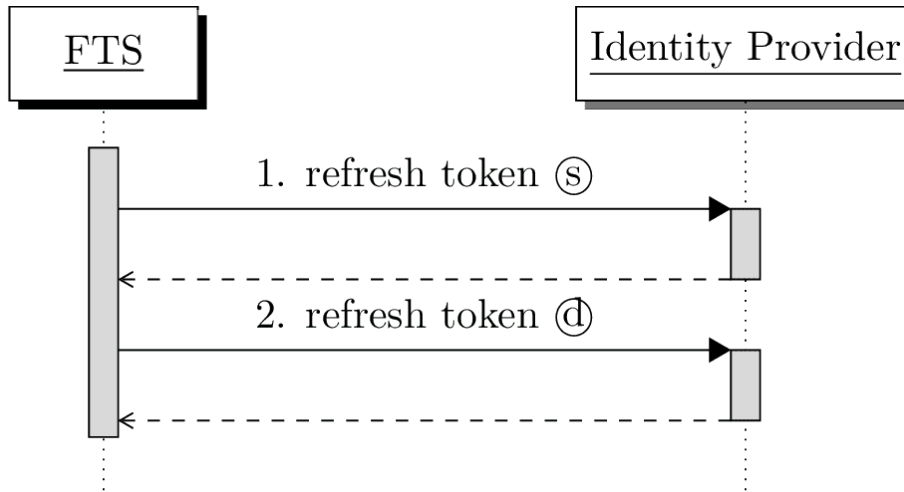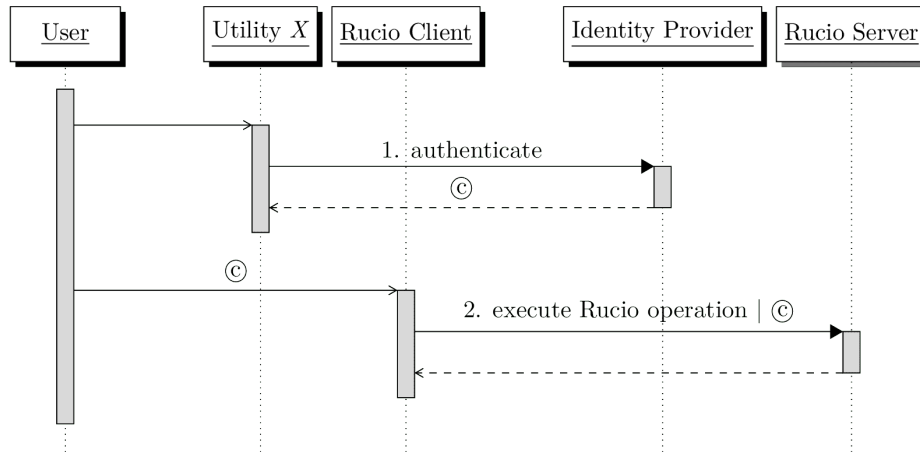
Third-Party-Copy Transfer (4)
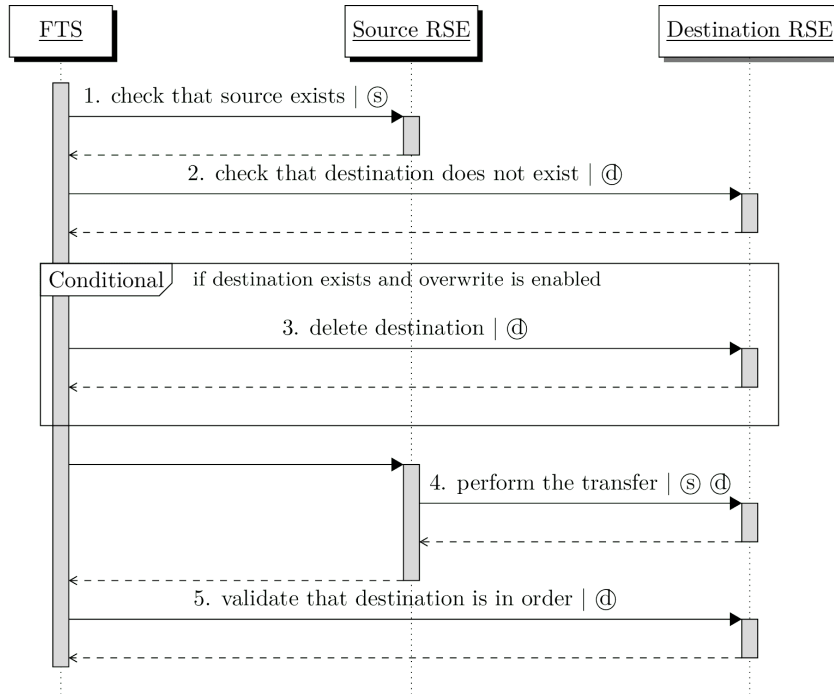
- Finally, FTS performs the actual transfer

- This is an initial design
  - FTS expects the destination token ⓓ to be capable of deletion
  - A second design in the works
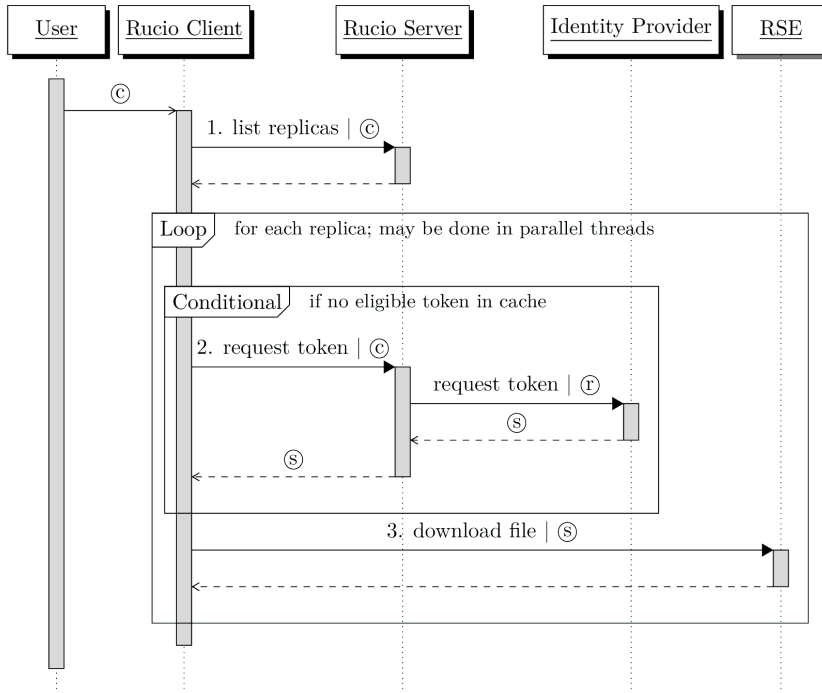
Deletion

- Very little change compared to X.509
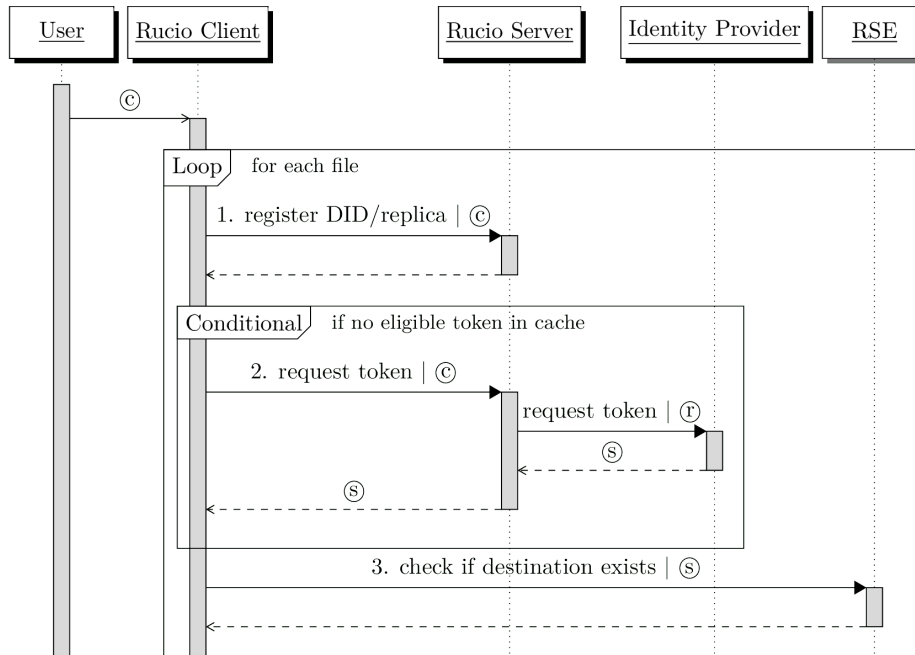
Authentication

- We expect some form of external utility like `voms-proxy-init`
  - For now, the placeholder name 'utility X' is used

- Not an actual requirement for upload/download using tokens

Download

- Requires a new endpoint in the REST API

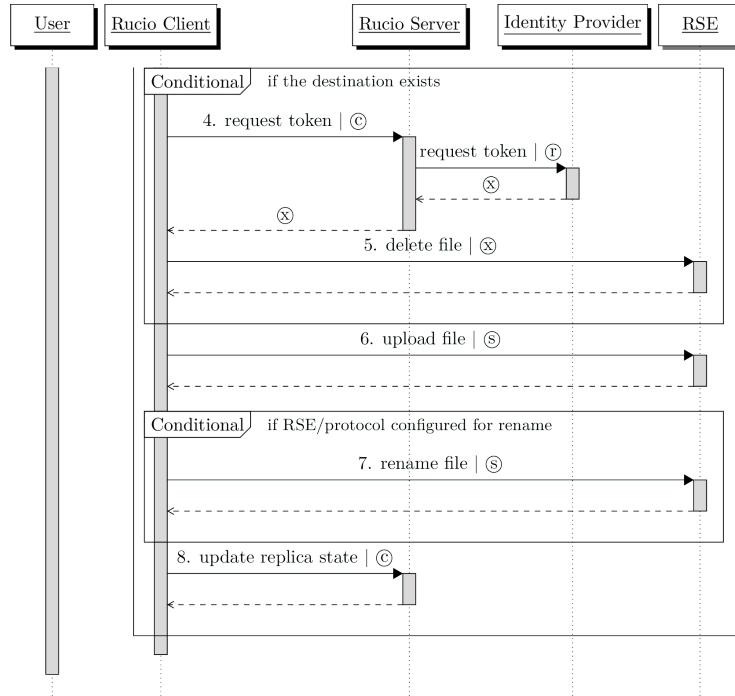- Might require some form of token introspection in the client

Upload (1)

- Similar to the download workflow

Upload (2)

- Added complexity due to the client potentially needing to delete

Plugin-Based Approach

- No single solution that satisfies all

- Envisioned areas:

  - Identity provider (INDIGO IAM, CILogon)

  - Token profile (WLCG)

  - Audience restrictions (RSE hostnames, WLCG 'any')

  - Scope restrictions

  - Namespace access

`/eos/atlas/atlasdatadisk/rucio/` `mc16_13TeV/` `00/ff/` `AOD.23208852._003398.pool.root.1`

prefix — scope — file

Hybrid Functionality

- For communities depending on X.509, the transition must be allowed to happen progressively

- An RSE attribute for most workflows
  - For TPC, must be all RSEs related to a transfer job
  - Won't support a situation where an RSE supports only tokens while others use X.509

- Account identities for user authentication

Questions?