OpenID Connect in FTS

eXtreme DataCloud

Data Management for extreme scale computing

eXtreme DataCloud is co-funded by the Horizon2020 Framework Program – Grant Agreement 777367

XDC Overview

- The eXtreme DataCloud's aim: Develop scalable technologies for federating storage resources and managing data in highly distributed scientific computing environments

- XDC is a 2 year, 3M€, EU-funded software development and integration project
  - Started active work 1st Feb 2018

- The targeted platforms are the current and next generation e-Infrastructures deployed in Europe
  - European Open Science Cloud (EOSC)
  - The e-infrastructures used by the represented communities
  - WLCG

CERN participation to XDC project

- CERN IT is participating to XDC with FTS, EOS and Dynafed

- FTS development Tasks
  - OpenId Connect integration in FTS
  - Storage QoS exploitation
    - Through the integration of the CDMI interface

FTS

- File Transfer Service developed at CERN
- Multiprotocol support (GridFTP, Webdav/https, xroot etc)
- Transfers from/to different storages (EOS, DPM, dCache, Storm, etc)
- Transfer scheduler, transfer optimizer, Real Time monitoring

XDC & OpenID Connect (OIDC)

- This is a standardised part of the “token based auth” landscape
  - Tracking WLCG policy direction
- XDC uses the Indigo IAM as the IdP
  - Others should work too – it’s standardised
- User “logs in” with a browser, using a login service somewhere else.
  - Can work without web-browser subsequently
- Primary an “access-token” – a bearer token that lets whoever holds it obtain identity information. Usually short-lived.
  - The access token may be passed around, but has a finite lifetime.
- Also a “refresh token” – allows an agent to fetch a fresh access-token once it runs out.
  - The refresh token is bound to the client’s identity, it cannot be passed around.
- A process called “delegation” allows an agent that receives an “access token” to obtain a fresh access token and refresh token
  - Typical use-case: a long-running job that is acting on behalf of a user.

OpenID Connect in FTS (1/2)

- FTS Auth/Authz currently done only with X509 proxy certificates and VOMS groups/Roles
  - not user-friendly
  - X509 delegation needed
- 2 types of OIDC integrations implemented:
  - Directly accepti access tokens from users via CLI/REST API (FTS is the Protected Resource)
    - https://fts3-xdc.cern.ch:8446
  - Redirect WebFTS users to IAM in order to acquire a token and using it via the FTS REST API (WebFTS is the Relying Party)
    - https://webfts.data.kit.edu -> WebFTS extension implemented by KIT
- Tokens are used both to authenticate to FTS and to the storages
  - Only dCache is supporting OIDC for now
  - **X509 delegation is not needed anymore**! (both to FTS and to storages)

OpenID Connect in FTS (2/2)

- Python Flask App has been written to easily acquire an IAM access token
  - Repo: https://gitlab.cern.ch/fts/openIdConnectPOC

- FTS-REST component has been modified in order to accept an access token and refresh it when needed
  - Access tokens are verified via introspect endpoint of IAM
  - A refresh token related to the access token is acquired (grant-type:token-exchange)
  - Valid access and refresh tokens are saved to the FTS DB
  - A daemon refreshes the access tokens that are about to expire through the token endpoint of IAM by using the refresh tokens
  - Repo: https://gitlab.cern.ch/fts/fts-rest/tree/fts-oidc-integration

- FTS Server has been modified and can use access tokens for transfers
  - Access tokens are retrieved from the DB and set to gfal2 API as BEARER credentials
  - Repo: https://gitlab.cern.ch/fts/fts3/tree/fts-oidc-integration

Next Steps

- First XDC release by the end of the year
  - FTS 3.9.0

- Implement Offline validation of the access tokens

- Understand how to handle groups/roles for certain REST operations
  - With X509 they are based on VOMS groups/roles

- Extend REST operations to non-X509 identities
  - User banning now is based on the X509 User DNs

Next Steps

- Integration of a Token Translation Service
  - Present a token – get an X509 certificate
  - Needed for EOS in XDC, but of course for all the other storages which do not support OIDC yet
    - Needed also to use other protocols than HTTP
  - First tests with Watts
    - Developed in the context of the Indigo DataCloud project
    - https://watts.data.kit.edu/ (configured with XDC IAM)
    - https://indigo-dc.gitbooks.io/token-translation-service/content/config.html

Questions?