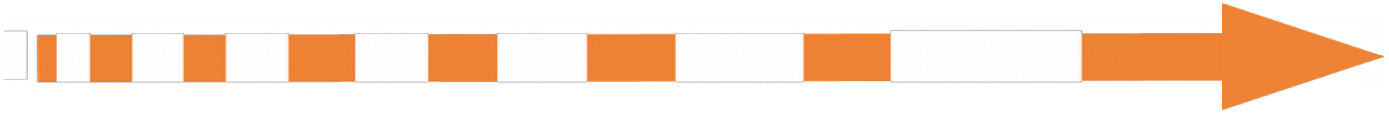
Task 4.1



Paul Millar

paul.millar@desy.de

eXtreme DataCloud

Data Management for extreme scale computing

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

OpenID-Connect: introduction

- User “logs in” to a service, using a login service somewhere else.
- Works (only) with a web-browser – at least, initially.
  - Can with without web-browser after an initial start (more in a bit…)
- 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: FTS

- We can demonstrate a transfer authorised with an access token
  - CLI client arrives with an access token
  - FTS validates token and authenticates the client
  - Internal concept of a *credential* generalised to accommodate tokens and proxies
  - Token used to authorise a transfer
    - gfal2 has been adapted appropriatly.

OpenID-Connect: dCacheView

- dCache provides frontend: a REST API that provides namespace QoS interactions
  - Indended to be a dCache propetary protocol for exposing dCache features.
  - OIDC support added during INDIGO
- dCacheView is the exemplary client, written in JavaScript
  - Providing a webbrowser based user- & admin- GUI for dCache.
  - File transfers (upload / download) use WebDAV door, NOT the frontend.
- During INDIGO, assumption was the client obtains the access-token
- We added support for obtaining an access token in dCacheView
  - Demonstrates a client obtaining an access token and interacting with multiple services.
  - Femi will demonstrate this.

OpenID-Connect: oidc-agent & co.

- INDIGO (and XDC) focus strongly on OpenID-Connect as AAI infrastructure.
  - Makes sense, industry standard…
- OIDC is (currently) a strongly web-based technology
  - Oidc as command-line is still in its infancy.
- Data management of (very often) involving command-line
- Introducing oidc-agent, a development from KIT (located in INDIGO github) We added support for obtaining an access token in dCacheView
  - Developed by KIT
  - Available from the INDIGO github “project”
  - Paul will demonstrate this.

RDA QoS group

- Have a framework ontology for expressing QoS classes
  - It's quite simple,
- Currently collecting use-cases
  - Currently mostly from storage / cloud-storage acquisition.
- Checking that the ontology is sufficient to describe those use-cases.
  - Manually creating ontology instances with information to describe what is desired.
- Currently adding a machanism to collect information from a CDMI endpoint and "publish" the data into the ontology.
  - Once completed, will aim to have a "life" web-page view of this data.

FTS: QoS

- gfal2 updated as basic CDMI client
  - Python binding done
- FTS can now accept a QoS job
  - QoS stuff passed in job metadata
  - FTS evaluates if the QoS criteria can be met with a simple transfer (i.e., QoS defaults are fine)
    - If so, transfer as usual
    - If not, check that the requested QoS is realisable with a transfer + transition [currently under development]
    - If so, schedule a QoS job which will manage the QoS transition after the transfer has completed [still under development]

dCache: QoS

- Deployed testbed dCache with simulated TAPE
  - Support TAPE-targeting directories as well as transitions
- Deployed CDMI server
  - Various small patches needed
- INDIGO work based on describing what already exists in dCache
- Started work on adding new QoS concept in dCache
  - Aim to add extra QoS types, making behaviour more admin configurable.
  - Prerequisite for federated storage, aggregated QoS.

Dynafed

- Integration of OIDC authentication
  - Apache-level integration
  - Allows browser based access to the Dynafed namespace
    - Browser is redirected to IAM in the usual way
  - Testbed currently configured to authorise all IAM identities
- Apache also offers a way to authorise base on Oauth2 access tokens
  - Not yet available on the testbed, trying to understand how to juggle OIDC, OAuth2, and X.509 credentials in one configuration.