EGI-Engage GOCDB Plans

david.meredith@stfc.ac.uk

Wiki: https://wiki.egi.eu/wiki/GOCDB
EGI Production Instance: https://goc.egi.eu
Src: https://github.com/GOCDB
Info Doc / Executive Summary:
https://wiki.egi.eu/w/images/d/d3/GOCDB5_Grid_Topology_Information_System.pdf

Extend AAI + Account Linking

GOCDB AAI layer:

- AuthTokens: PreAuthenticating (x509,BASIC) + Manual (un/pw)
- SecurityContext: Stateless + Stateful
  - REST endpoints (no HTTP session)
  - Portal pages (create HTTP session)

In same webapp

- Currently supports x509 + SAML2 (EGI SSO)
  - To login via SSO, must associate cert DN with EGI account (one time setup)
  - Allows login via un/pw from browser without certificate

Extend above - Likely requirement to:

- Support more AAI Federations / SecurityRealms
- Consider different Levels Of Assurance (LoA)
- Link accounts from ‘n’ AuthProviders to one Goc user account (Perun)
- Q. will AccountLinking be responsibility of Central EGI IdP or each SP?

Account Linking – Responsibility of SP (GOCDB) or Central IdP?

```
<gocdb_user>
  <AuthProvider>
    <Principle>
        /C=UK/O=eScience/OU=CLRC/L=DL/CN=
    </Principle>
    <Realm>IGTF</Realm>
    <Scheme>x509</Scheme>
    <LevelOfAssurance>4</LevelOfAssurance
  </AuthProvider>

  <AuthProvider>
    <Principle>davidismeredith@gmail.com<
    <Realm>Google</Realm>
    <Scheme>usernamePassword</Scheme>
    <LevelOfAssurance>1</LevelOfAssurance>
  </AuthProvider>

  <AuthProvider>
    <Principle>davidm</Principle>
    <Realm>EGI_SSO</Realm>
    <Scheme>usernamePassword</Scheme>
    <LevelOfAssurance>2</LevelOfAssurance
  </AuthProvider>

   <Roles>
     ...list user's gocdb roles...
   </Roles>
</gocdb_user>
```

Do something like above if no central Idp

AccountLink in SP (GOCDB)

- Provide API for other tools? 'get_user'
  - Principle (userId)
  - Security Realm (Federation)
  - Scheme
  - Assigned LoA
- Different actions in GOCDB would require different LoA :
  - Site Admin: LoA == 4
  - Service Group Admin: LoA => 2

Account Link in Central EGI IdP

- Need to communicate LoA, Principle, Realm attributes back to the SP (EUDAT's plan) Virtual IdP/IdpProxy

....

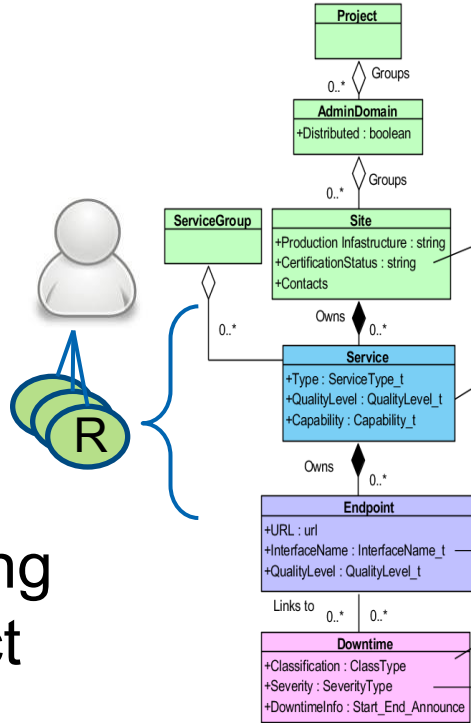
Different Levels of Assurance Provided by Single Realm

- IGTF (IOTA, SLCS, MICS, Classic)
- https://www.incommon.org/ (Bronze, Silver, Gold)
- https://kantarainitiative.org/
- UK Access Federation distinguishes between IdPs signed up to section 6 (or not)
- https://discovery.refeds.org/guide/

Roles/Rules Abstraction + Customisation

- Users have Roles over objects in the Domain model
- Abstract the logic/rules so different projects hosted in same Gocdb instance can customise rules:
  - Roles,
  - RoleTypes,
  - Business Rules,
  - Diff LoA for diff AuthProvider?

Allow customising per-project



- Investigate BRMS

Audit Logging

Expand current audit logging (who did what/when)

An EUDAT requirement for CMDB purposes

1. Role Action Log
   - Role Approve/Denial/Revoke history
   - Ready for testing - v5.4
2. On editing an object, record an object diff
   - Pre/post change

Extend Data Model

- Support changing infrastructure/marketplace requirements
  - Which Goc services should be included ?
  - Marketplace could aggregate service summaries
  - But don't duplicate data, stay DRY + consider stale data (marketplace may need to frequently poll)

- Are more data model attributes/objects required?
  - Pay4Use/marketplace price list, rely on extension props?
  - GLUE 2.1 cloud?
    - XSEDE may not use GLUE2.1 ? (~think they intend to stay with v2?)

Summary Roadmap (subject to change as stuff evolves…)

- Auditing extensions
  - Action logging / Object diffs
- Role/Rule abstractions + customisation
- Extend AAI
  - +/- Account Linking
- Support changing infrastructure requirements + marketplace
  - Extend Data Model where necessary
- MVC + GUI refactoring
  - Finer grained content rendering (PermitAll + Protected pages)
  - Replace proprietary MVC with Symfony2 (lower priority)

- Add a suggestion box / textArea in Gocdb

Writable REST API?

- CRUD API, insert dynamic service status
- EUDAT have already implemented a writable PI interface to their GOCDB instance to POST extension properties (potential for re-use).
- May not be needed for GOCDB

Summary (in order of decreasing priority):

1. Extend AAI + Account Linking
2. Roles, Rules and Auditing extensions
3. Extend Data Model
4. MVC + GUI refactoring
5. CRUD API