Group Membership

- define groups of users to support teamwork & collaboration

- group identifier used to authorize access to resources

  `ivo://cadc.nrc.ca/gms#NGVS`

  – resources controlled by users

  – remove fragment, lookup URI in registry to find a GMS service, verify membership

- primary function: assert membership

  `GET <baseURL>/groups/<groupID>/<userID>`

  - userID is an x509 Distinguished Name (SSO)

- primary function: assert membership

```
GET <baseURL>/groups/<groupID>/<userID>
```

 – userID is an x509 Distinguished Name (SSO)

- standardise management of groups

```
GET <baseURL>/groups/<groupID>
POST <baseURL>/groups/<groupID> (modify)
POST <baseURL>/groups (create)
PUT <baseURL>/groups/<groupID> (create)
DELETE <baseURL>/groups/<groupID> (delete)
```

- read-only reverse mapping

```
GET <baseURL>/memberships/<userID>
```

  - retrieves a list of groups the user belongs to
  - it will not be a complete list (globally), just this service
  - useful for users
  - useful as an optimisation when checking multiple groups

- authorization within GMS (optional)
  - – users can assert their own membership
  - – users can get list of all groups they belong to
  - – group owner can perform all operations on group
  - – group membership can be very private and it still work (requires CDP to assert!)

- why standardise group membership?

  - support authorization in other services

  - enable re-use of groups across services at multiple sites: teams are spread out geographically and so are the resources they want to share

  - users who control resources assign permissions to groups

  - services delegate membership checks & authority responsible maintains member list

- how does it work?

  ivo://cadc.nrc.ca/gms#NGVS (group ID)

  ivo://cadc.nrc.ca/gms (service ID)

  https://www.cadc.hia.nrc.gc.ca/gms
  (service URL)

  GET <service URL>/groups/NGVS/<userID>
  (assert membership)

  200 = member

  404 = not a member

  401 = not authorized to assert

- primary function: assert membership

  `GET <baseURL>/groups/<groupID>/<userID>`

  - userID is an x509 Distinguished Name (SSO)

- standardise management of groups

  `GET <baseURL>/groups/<groupID>`

  `POST <baseURL>/groups/<groupID>` (modify)

  `POST <baseURL>/groups` (create)

  `PUT <baseURL>/groups/<groupID>` (create)

  `DELETE <baseURL>/groups/<groupID>` (delete)