2022-05-17: py-ISPyB Framework Meeting

Attendees:

- SF, AM, MB, MG (ESRF)
- CB (DESY)
- AN (MAXIV)
- SI (SOLEIL)

Agenda:

- Authentication
- Authorisation

Authentication

Current survey of authentication methods
https://docs.google.com/spreadsheets/d/1lhR0O0Gp8grZ6JJLO9KmRfQvnRnuLUlltErFXq748V8/edit#gid=0

Agreed to keep common authentication methods in the core repository, for now that comprises

- LDAP (and modified version with support for active directory at MAXIV)
- Keycloak

Specific authentication mechanisms can be kept in separate repositories.

The authentication mechanisms currently mix in authorisation by retrieving the groups from the authentication source (i.e. retrieve the roles from LDAP). Discussed how it would be better to decouple these two processes so that the authentication mechanism only returns whether a user is who they say they are or not. This will aid in simplifying the base authentication class.

Authorisation

Two different authorisation cases:

- What data can the user access (SessionHasPerson, ProposalHasPerson)
- What special privileges does the user have (Permissions, historically roles)

If we decouple authentication and authorisation as above we can then specify that Permissions can be read directly from ISPyB. This makes permissions consistent for all sites.

This requires that facilities replicate their user groups and permissions into the database. This could be handled by the user portal sync.

We can provide resources in the API to create and manage permissions and groups

Data Access

Currently many sites use proposal login, this allows anyone who logs in as the proposal to view all sessions in the proposal. BAGs comprise multiple establishments, it would be better to limit what sessions / data each user can use.

There are three current data access policies based on individual logins:

- Limit data to sessions which the user was part of
    - Needs an entry to be replicated into SessionHasPerson
    - An additional permission can grant access to all sessions for admins
    - Can maintain proposal login by adding SessionHasPerson entry for each proposal
    - Can add individual users to a session by adding an entry in SessionHasPerson
        - Can create a resource to manage this in the UI
- Access proposals without a session (i.e. create a shipment)
    - Needs an entry in ProposalHasPerson (table agreed, not yet implemented)
- Possibly limit access to samples / proteins (long term)?
    - Protein has personId, but this limits to a single person
        - Would need a linker table
    - Sample has no owner
        - Would need a linker table
    - Container does have owner, but person registering container may be different to person needing access (BAG coordinator vs students, postdocs)
    - Needs discussion with staff and collaboration

Permissions

Agreed to document the existing rows in the Permissions table for both developers and users,

- Implement fine grained permissions based on individual features. I.e. a specific permission to use the user portal sync.
- Do we want features like ICAT where access can be granted to an individual protein, sample, and/or data collection?

Discussed refactoring `own_sessions` (can be removed with schema above based on SessionHasPerson) and `all_sessions` (could be made more flexible)

Actions

Action: We need to create a base authentication interface that can be inherited from, and document how this can be used by developers

Action: Request sites to provide requirements from developers and scientific staff in terms of how data is accessed. Would sites like to be able to show for example:

- Beamline activity for example (all data collections on a beamline)
    - Currently these resources are tied to a session (agreed to untie these for new endpoint development)
- Limit access to particular samples?
- What other features would be useful?