



| The European Synchrotron

What methods are used?

Inbuilt support for:

LDAP

Keycloak

What to do with custom methods?

Separate packages?

Authenticator can reference an absolute python module

<https://github.com/ispyb/py-ispyb/issues/161>

Historically roles/groups come from LDAP, etc

We have database tables to store this information

Makes py-ISPyB more consistent (single source of truth)

Can simplify authentication base class

Only deal with is this user valid

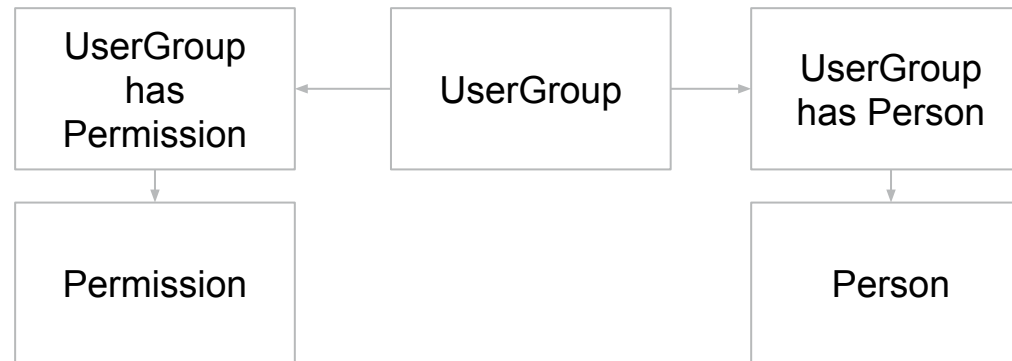
Not what can they do

-> Need to replicate group / permissions

Roles are quite restrictive

Groups / permissions table provide more fine grained access

py-ISPyB can provide routes to manage these



<https://github.com/ispyb/py-ispyb/issues/166>

### Proposal Login

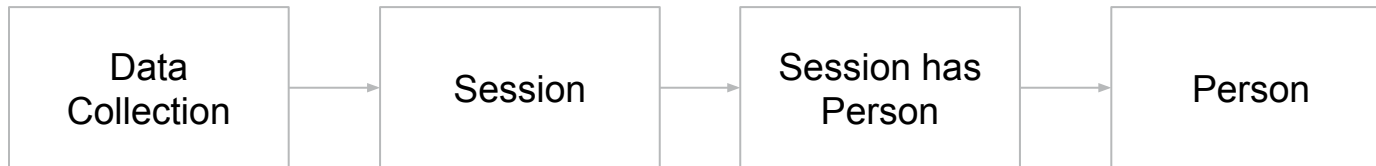
Anyone in BAG can see all data

Ideally should be deprecated.

Easy to limit users to only their own sessions

Sites need to populate Session has Person

With a specific permission certain users can access all sessions  
(or sessions within a group of beamlines)

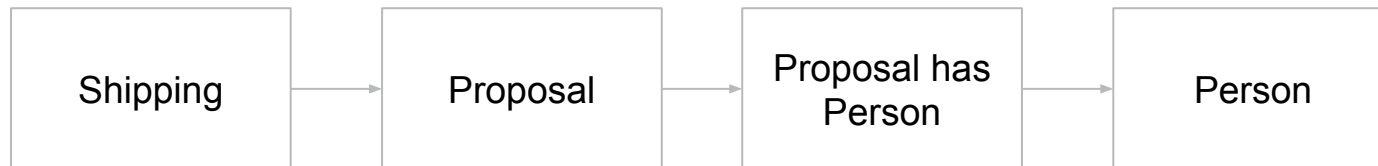


If no sessions exist, how do we allow users to create shipments?

Use case for more automated contexts

Populate Proposal has Person

Currently not implemented but table exists



**Longer term, do we need to restrict access to samples / proteins?**

**Samples / Proteins have no owner in the db at the moment**

**Containers do have an owner**

**Complicated with BAGs where a coordinator may register samples on behalf of PhDs / Postdocs**

Can we agree a common implementation?

If not, could be possible to make some things configurable  
sqlalchemy lets us abstract table joins, etc



Document what the existing permissions are and how they are used

`own\_sessions` can be deprecated (intrinsic with Session has Person)

`all\_sessions` can be made more flexible

<group>\_sessions

could allow staff to see all sessions / proposals from a group of beamlines rather than all