Meeting Minutes 2018-03-16

Attendees:

Meeting was postponed, due to minimal agenda. Discussion of agenda items took place in slack.

Agenda:

1. Roll call
2. Call for additional agenda items
3. Review previous action items from 2018-03-02 Meeting
    a. Meeting regarding non-standard entities took place on 3/13/2018. Jon Cameron - can you please add notes here if you took any?
    b. Followup re: status of filling out the spreadsheet
        i. Emory added some use cases noting custom entities, but still incomplete.
            1. Emily included screenshots from tools used at Emory to the user stories in the spreadsheet.
        ii. IU has not had time to add more since the non-standard entities meeting
4. Discuss next steps
    a. Jeremy Friesen schedule next meeting

Action Items:

1. Continue to work on filling out the spreadsheet.

Notes from 3/13 Meeting on Non-standard Entities

Playlists-
how permissions would work with that?
how would you apply it to hyrax in the first place

M: there is a lot of security in playlists
in playlists, you can only access items you have access to
in playlists, you see all the items, but you would not be able to access

L: Playlists would be a collection type, a new type of collection
A collection with ordered elements
We don't have that feature for collection types- no order

adding new features to collection type would be a change, but the permission on the collection type—"these are the permission" and it would check out the permission object for the collection types.
so it's pretty easy to do, and it would be a pretty easy mapping, as long as you get the features added for a playlist

there are a number of configurations that you can do with collection types

collections can contain a collection

participants: identifies the manager, editor, creator, viewer… who can add items to the collection?

do managers have edit access to items in the collection? yes, the manager gets edit access to the collection (larita believes they also are able to edit the items)

M: if you're a manager, it makes sense that …..
adminsets give you access if you're in a group, she thinks collections are different in that actual collections settings are used to test
LaRita thought the permissions looked at that role to determine whether someone could do things, though. But it might not affect the viewing of the object itself in Fedora.

M: do you see us be able to change things in adminsets?
L: Our view is that, what she was trying to do in her initial design, was that you create the functionality in the application that controls all of those pieces, and you have all of your people in groups over here, and there's an intermediate piece, you're not linking directly to something in the application, that the agent has the ability to use this thing, and the user has access to this agent

L: you can take an external thing, you can use whatever you use as an adapter

L: that should ideally be a piece of this work

every work has to be deposited into an admin set. you don't have to mess with multiple admin sets if you don't want to

L: adminsets are just a permission stamping feature

L: the permissions that are originally granted to a collection come from the collection type, and you can set up for each individual collection

only adminsets use workflows <——

when you nest a collection, it forces it to reindex (this could take a while)
in the time where it's reindexing, you can't do anything because it's in an incomplete state

manager/depositor/editor are the three things

workflows - to use them the way they're intended, you need to create roles to mesh with them, not hardcode roles

part of the work is going to be clearly identifying things that the application can do
by doing that, we'd be able to have collections be able to give access to everything in all of the works, but normally that's just done at the adminset level