Flexible file access control for iRODS

Alva L. Couch, Ph.D.
Tufts University
with
David Tarboton and Jeff Horsburgh: USU;
Ray Idaszak, Hong Yi, Michael Stealey: RENCI

HYDROSHARE

OCI-1148453
OCI-1148090
2012-2017

Tufts

Subtitle:
iRODS as a social collaboration environment(!)

HYDROSHARE

Tufts

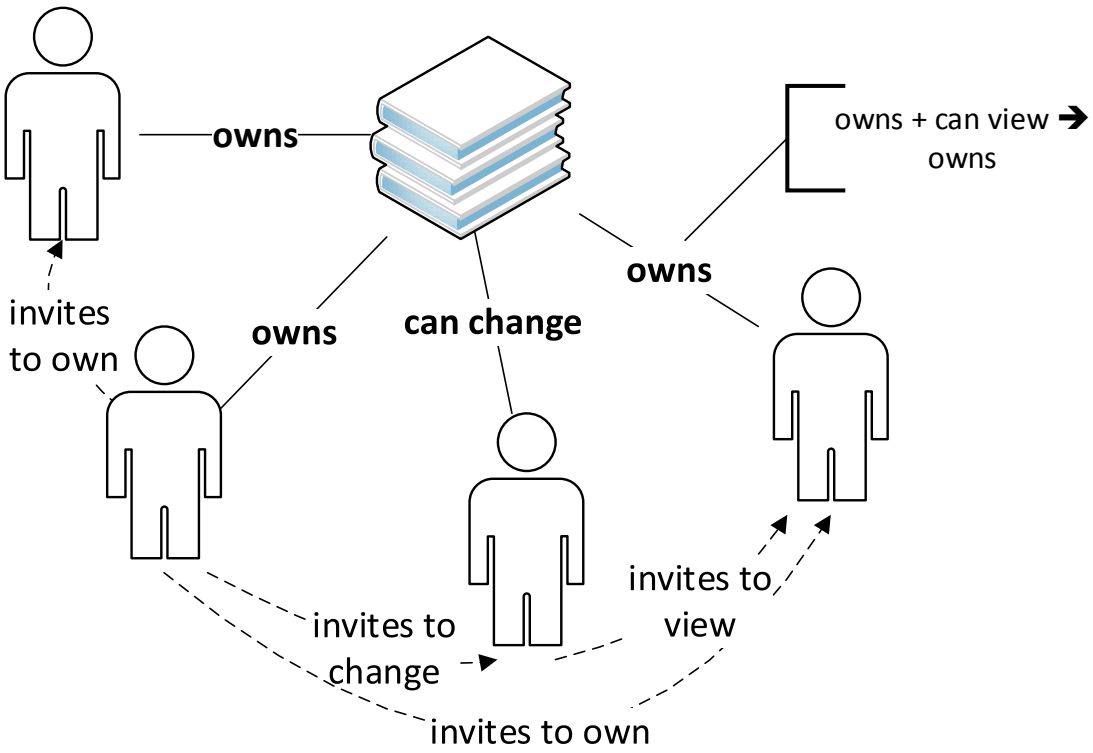
Data management versus collaboration

Data management	Social collaboration
Administrative control and management of user privilege over objects and user groups	**Unprivileged and transitive** user privilege and group management.
Only an owner can share an object.	**Anyone with a privilege** can grant that privilege to another.
Only an administrator can create a group.	**Anyone** can create and manage a group, including granting privilege to others.

The problem

- iRODS access control is designed around filesystem-like access.
  - Creating a user group requires administrative privilege.
  - Only owners can share files with others.
- We wanted a "social" access control in which:
  - Users can create and manage user groups at will.
  - Privileges can be delegated **without owning an object.**

Transitive privilege



HYDROSHARE

Tufts

Why not do this inside iRODS itself?

- Websites support this kind of sharing all the time now.
- Can we implement the same kind of sharing consistently at the iRODS level?

Am I crazy?

- Probably. Social networking does that to people.
- But iRODS can do this. The rule engine is that powerful.

HYDROSHARE

Tufts

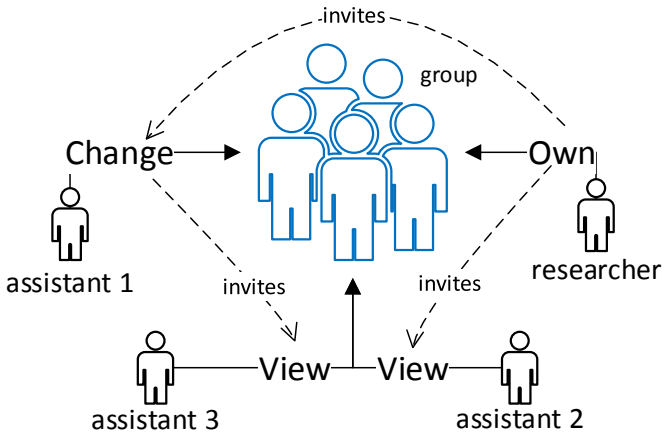
Two Use Cases

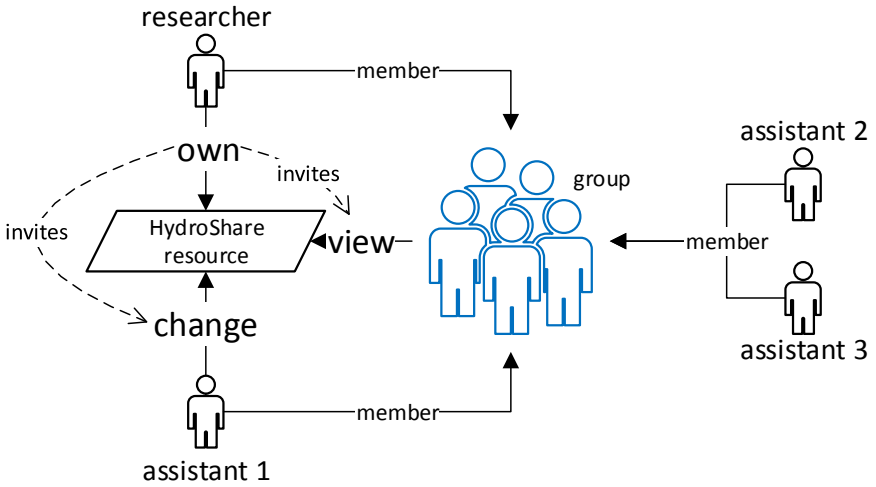
1. Researcher plus graduate students.
2. Researcher plus peer researchers at other institutions.

Researcher plus graduate students

- Researcher should "own" all files.
- Researcher "invites" graduate students to receive "view" or "change" permission for directories. Files in directory inherit directory privilege.
- When graduate student graduates, researcher retains ownership and control.
- Graduate students can delegate privileges they have to others, as needed.

Researcher plus graduate students

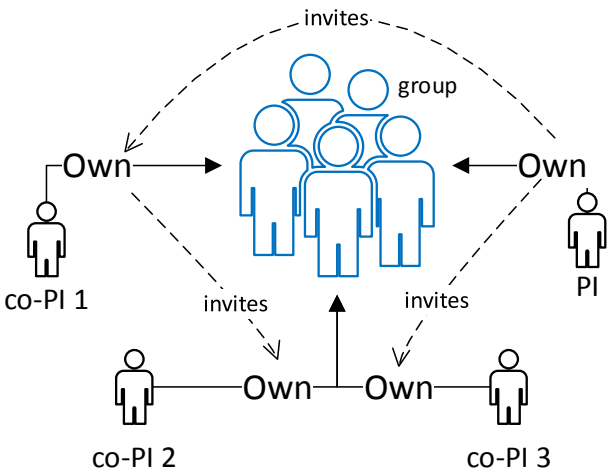


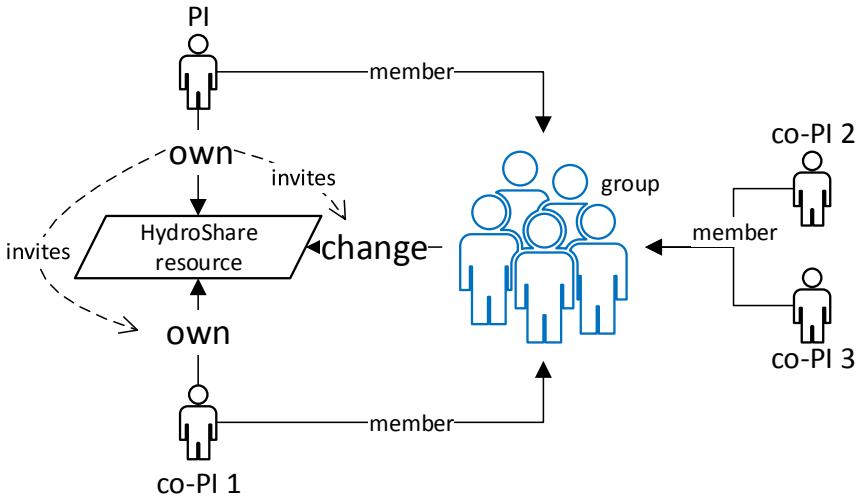


Researcher plus peers

- Each researcher can assign joint ownership to others for
  - iRODS objects.
  - IrodsShare user groups.
- Those, in turn, can delegate privilege to their assistants.

Researcher plus peers





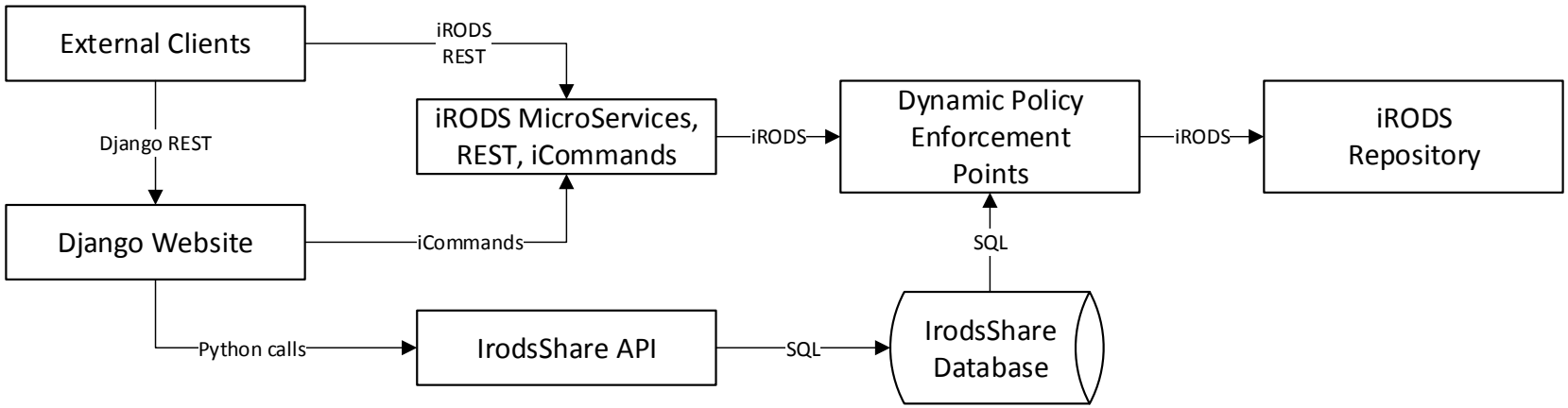
HYDROSHARE

Tufts

Options for implementation

- Strategy: use **dynamic policy enforcement points** to deny access by raising an error condition if user does not have appropriate privileges. Available in 4.1.
- Several options for authorization information
  - In iRODS AVUs: insufficiently secure at this time.
  - In iRODS ACLs: relatively difficult to manage without a provenance database to implement "undo" policies.
  - In the iCAT catalog as separate tables: too brittle and subject to disruption as iCAT is expanded.
  - **In a separate database parallel to iCAT.**

Present IrodsShare architecture



HYDROSHARE

Tufts

Subtractive Access Control

- A user of IrodsShare is a member of the *HydroShare* iRODS group and by default is granted full change permission over all HydroShare group resources.
- Policy enforcement points deny access according to IrodsShare policies.
- Thus "all that is not denied is permitted".

Access control policies

- Any user who has a privilege can **grant that privilege or less** to another user.
- **Grantors can also revoke** privileges that they granted.
- **Privileges are additive:** if two users grant a third different privileges on the same object, the strongest privilege is granted.
- **Object owners can revoke any privilege,** and can set an object as unshareable, which turns off the main features of the sharing mechanism.
- This applies to folders, their contents, and groups.

Status and availability

- Access control logic and policy engine implemented, http://github.com/hydroshare/IrodsShare
- Policy enforcement not yet implemented.
- Plans for immediate future
  - Move API inside iRODS via Python Microservices.
  - Expose as REST.
- Considering several options for the future:
  - Reflecting policies into ACLs.
  - Using limited privilege policies to allow regular users to create and manage regular iRODS groups.

Conclusions

- iRODS is not just for data management.
- iRODS policies are powerful enough to allow very fluid data sharing and collaboration.
- These policies bring collaboration facilities normally thought of as UI features to the filesystem level.