A Flexible File Sharing Mechanism for iRODS

Alva Couch
Tufts University
Medford, MA
couch@cs.tufts.edu

David Tarboton
Utah State University
Logan, UT
dtarb@usu.edu

Ray Idaszak
Rennaisance Computing
Institute (RENCI)
Chapel Hill, NC
rayi@renci.org

Jeff Horsburgh
Utah State University
Logan, UT
jeff.horsburgh@usu.edu

Hong Yi
Rennaisance Computing
Institute (RENCI)
Chapel Hill, NC
hongyi@renci.org

Michael Stealey
Rennaisance Computing
Institute (RENCI)
Chapel Hill, NC
stealey@renci.org

ABSTRACT

The traditional iRODS mechanisms for file sharing, including user groups, often require some form of iRODS administrative privilege. In the HydroShare project for enabling hydrology research, we perceived a need for more flexible file sharing, including unprivileged creation and management of user groups according to policies quite distinct from the Linux/Unix policies that initially motivated iRODS protections. This is enabled by a policy database in PostgreSQL and management API written in Python that are deployed in parallel to iCAT. Innovations in iRODS 4.1 allow access control based upon this PostgreSQL database rather than the default iCAT server, by interposing access control code before the access event using iRODS Policy Enforcement Points. The result is an access control mechanism that closely matches scientific needs for file sharing, and brings "dropbox-like" file sharing semantics to the network filesystem level.

Keywords

File sharing, authorization, access control, policy enforcement points

INTRODUCTION

The access control mechanisms implemented by iCAT for iRODS [1, 2, 3] – while suitable for a variety of types of file sharing – fell short of our requirements for file sharing in the data-centered social networking site "HydroShare" [4, 5, 6] (http://www.hydroshare.org). Based upon the Hydrologic Information System (HIS) [7] of the Consortium of Universities for the Advancement of Hydrologic Science, Inc. (CUAHSI), HydroShare brings social networking to data centered science, by enabling "object-centered" discussion in data-enabled science [8]. HydroShare enables posting of a large variety of data types as directories of files in the BAGIT [9] format, and stores metadata for each object in the object, using the Open Archives Initiative metadata guidelines [10] tailored to the needs of water sciences. The goal of HydroShare is to enable sharing of hydrologic data and models with the same ease with which one stores photos or videos on social networking sites, with social mechanisms including commenting, rating and sharing of data objects.

iRODS does not currently provide a sufficient protection model for objects in HydroShare, because of our desire to enable social features rather than just file access. Among other issues, iRODS normally requires administrative access in order to:

1. Change the owner of a file.
2. Create and manage a user group.

We desired a more flexible concept of object ownership, sharing, and mutability in which a user without administrative access can:

1. Reassign ownership and all privileges over an object to another user.
2. Create, manage, and destroy a user group and its privileges.

The result was the "IrodsShare" project, a sub-project of the "HydroShare" project, centered specifically upon creating access control mechanisms more suitable for the collaborative scientific data sharing desired in HydroShare than the default mechanisms in iRODS. We expect that this sharing model developed for HydroShare will be of broader interest for scientific data sharing beyond hydrology and water science.

HydroShare

The "HydroShare" project [4, 5, 6] aims to create social mechanisms for data sharing between researchers in the water sciences, that enhance the value of the data to other researchers. A HydroShare "resource" is a bag [9] of data that is accessible as one unit, and can contain many subfiles of potentially differing formats. It is best to think of a HydroShare "resource" as a directory of files rather than a single file, though single file resources are easily represented in bags. This use of the word "resource" is completely detached from the iRODS meaning of the word "resource"; HydroShare "resources" are directories adhering to a strict format and requirements for contents and metadata.

HydroShare is written in Python/Django and utilizes iRODS as a data management back end. Currently, HydroShare functions according to a Django access control model and allows or denies access via the HydroShare website. It is desirable to create a coherent access control mechanism that is homogeneous between the privileges granted at the Django website and all forms of access to HydroShare resources via iRODS, including REST, iCommands, etc. "IrodsShare" provides this desired mechanism.

Policies for file access

Our policies for file control are somewhat different than those in typical iRODS or Linux, and aligned around social and scientific needs and priorities rather than filesystem traditions. All objects in HydroShare are "HydroShare resources": bags of data containing perhaps multiple constituent files. The file protection model is very simple:

1. A user can possess "View", "Change", or "Owner" privileges over a specific resource.
2. "Owner"s can change any aspect of a resource, including its title, ownership, and access to others, and can delete the resource.
3. "Change" users are limited to changing the resource contents and/or metadata.
4. "View" users are limited to viewing the contents of the resource file.
5. Resources can have multiple "Owner"s.
6. The last "Owner" of a file cannot revoke personal ownership; ownership must be assigned to another user first.

So far, these privileges should be familiar to most users. However, the system diverges from standard file access due to several extra boolean resource attributes, including:

1. "Public" – the content of this resource is made available to all users.
2. "Discoverable" – the existence and metadata for this resource file is available to all users, but not necessarily its content.

3. “Shareable” – this resource can be shared with other users and groups by a non-owner.

These settings are used to manage whether “IrodsShare”s flexible sharing mechanisms are enabled for a specific resource. A maximally protected resource in HydroShare would have “Shareable”, “Discoverable”, and “Public” set to FALSE. This is a resource that an owner can share with selected users, but that non-owners cannot in turn share with others. This is the typical setting for resources that contain pre-publication data.

1. Setting “Shareable” to TRUE allows people and groups with whom the resource has been shared to re-share it with others. This – in essence – enables limited distribution of the resource via transitive sharing. For example, a researcher can share the resource with that researcher’s research assistants, and those assistants – in turn – can share it with anyone they wish.
2. Setting “Discoverable” to TRUE allows researchers not associated with the owners to learn of the existence of the resource and request access from the owners. This enables the owners of the resource to reach a common understanding of the limits of use with the requester before sharing the resource. The use of this flag is to enable pre-publication access to data on a limited basis by trusted external researchers.
3. Setting “Public” to TRUE enables general access to the resource, by any user, and is normally enabled after results are published. When this flag is TRUE, any user anywhere can discover and download the data.

The default in HydroShare is to set “Shareable” to TRUE and the others to FALSE.

Sharing and groups

Perhaps the most unique aspect of IrodsShare – from the perspective of resourcesystem-like protections – is the concept of resource “Sharing”. Any user with a privilege over a “Shareable” resource can grant that privilege to others. As well, privileges previously granted by a specific user can be removed by that user. One can share a resource with either another user or a group of users.

In like manner, user groups can be created, managed, and destroyed at will by any user. A group of users is managed very much like a resource in IrodsShare:

1. Any user can create and thus own a group.
2. Users can have “View”, “Change”, or “Owner” privilege over a group.
3. “Owner”s can invite and remove group members (with any privilege) or destroy the group.
4. “Change” users can invite new members to a “Shareable” group, and can assign those members ”Change” or “View” privileges over the group.
5. “View” users cannot invite new members, even to a “Shareable” group.

As with resources, a group can be “Shareable” or not.

USE CASES

This model was conceived based on many conversations and experience working with water scientists interacting with the CUAHSI Hydrologic Information System implemented at the CUAHSI Water Data Center. It strives to balance the needs and concerns surrounding sharing and enabling access to scientific data while protecting the rights and desires of the researchers involved. These users are a set of water scientists with broad interests, from basic hydrology to water quality and various forms of modeling. We discussed what scientists really need, instead of limiting discussion to resource sharing mechanisms that currently exist.

The resulting design was documented in a working policy document [11] and motivated by several scientific use cases discussed below. As well, part of the design attempts to provide the basic infrastructure to become a DataOne node [12] and to serve the sharing needs of water science groups such as the Critical Zone Observatories, as documented in [13].

In the following, it is important to remember that HydroShare "resources" are actually – logically – directories of related files. Thus access to a resource means access to that directory and all files contained therein.

There are many non-conflicting interpretations of a HydroShare resource, including:

1. Files for one trial of a single "experiment".
2. Input, output, and software for a specific model run.
3. A set of related files that represents the published data output of a project.

and many other potential contents.

Like a UNIX group, the concept of an IrodsShare group has many non-conflicting interpretations, including:

1. Users associated with a specific project.
2. Users associated with a specific organization or laboratory.
3. Co-authors of a specific academic paper.
4. A loosely managed group of users sharing the same interests.

While the first and second interpretations apply to a regular user group in Linux or iRODS, the third and fourth interpretations are only really practical in IrodsShare, because of the administrative overhead involved in finding an iRODS administrator to create what is really an "ad hoc" notion of group.

The word "group" does not serve particularly well to describe the IrodsShare group concept because of its common association with the narrower definition used in iRODS and linux file systems. However, no better descriptive word has emerged.

Researcher plus graduate students and collaborators

To start a project, the researcher creates a group and assigns graduate students to the group with appropriate privilege (typically "View") (left side of Figure 1). As students enter and leave the project, their group access is managed by the researcher. A senior graduate student can be delegated the task of managing group membership by assigning group "Change" privilege to that student.

Next, the researcher shares resources at different privileges with group members. In this example (right side of Figure 1), the whole group has "View" privilege while "assistant 1" has "Change" privilege. The access control system automatically defaults to the highest level of privilege granted through any mechanism.

Thus, the researcher retains control and oversight over all project resources and can delegate responsibility to students very easily. Most notably, students with change privilege do not automatically assume ownership of a resource after they make changes. Each resource remains owned by the original creator – the researcher. This eliminates the very common plight that graduate students acquire too much privilege, and then graduate.

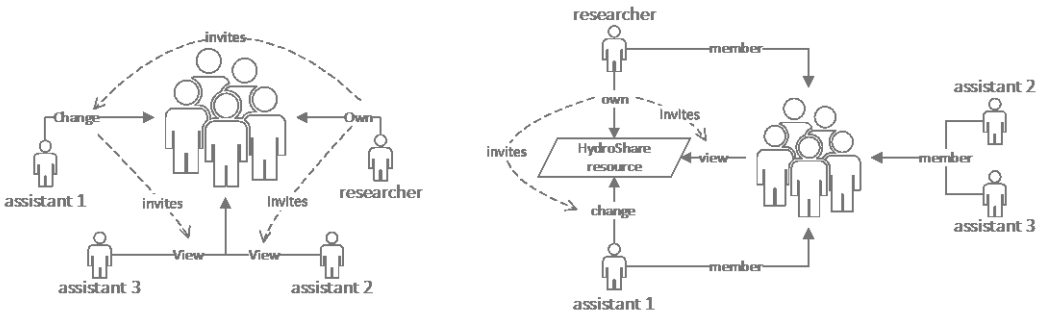

Figure 1. User groups and resource sharing in IrodsShare.

Pre-publication and post-publication data protection

The "Shareable" flag on each resource arose from the necessity of protecting resources from dissemination before results are published. The researcher can set this flag to FALSE so that no non-owner can share an accessible resource with another user or group.

For example, this is desirable if details of the original study that collected the data in question have not been published. Then the owner – e.g., an academic researcher – can delegate "Change" privilege to students without permitting students to disseminate the data to others. Likewise, the "Discoverable" flag arose from the desire to enable and encourage polite and acknowledged reuse of pre-publication data by permission of the owner. Setting this flag to TRUE allows all users to discover that the resource exists, but users must then request access from an owner, typically via email in this implementation. This allows the owner to set limits on resource use.

Finally, the "Public" flag arose from the desire to be able to publish results for general consumption after the results of the data have been published. Setting this flag to TRUE gives general access to the data to all users.

Miscellaneous use cases

A few other features enable other scientific needs, including:

1. "Published" – a resource flag that indicates that the resource has been assigned a Digital Object Identifier (DOI).
2. "Immutable" – a resource flag that indicates that the owner has frozen the contents of a resource, overriding "Change" privileges for all current users holding that privilege.

POTENTIAL DESIGN APPROACHES

To accomplish these requirements, we discussed and tried several design approaches before settling on the final approach. Ironically, while the project was to add non-admin functionality to iRODS, several approaches were discarded because the unprivileged user is *too privileged* and, at the present time, there is no formal access control on certain aspects of iCAT, including Attribute-Value Units (AVUs).

For example, we considered adding access control to iCAT metadata Attribute-Value Units (AVUs). Since there is no access control on AVUs, however, we would have had to create that access control. We considered writing several Policy Enforcement Point services to limit the changes one can make to access control AVUs. We abandoned this approach due to four main reasons:

1. The code for controlling AVU content was discovered to be very complex.

2. The code for utilizing AVU content for access control was predicted to be unacceptably slow.
3. Other projects' use of AVUs could potentially collide with our uses.
4. iRODS Policy Enforcement Points had not been used previously for this function and it was not clear at the time whether this would work properly.

Thus, after some prototyping and proof of concept, we concluded that this approach was more complex than warranted.

Other considered approaches included modifying the iCAT itself to include a different kind of metadata. This was abandoned because it was simpler for us to maintain a separate database just for access control than to branch the iCAT code.

THE CHOSEN DESIGN

After consultation with the iRODS team at RENCI, we decided to base our access control system upon a new feature of iRODS dynamic Policy Enforcement Points (PEPs) that was implemented for us and will be supported in the iRODS 4.1 release. If a function defined for a pre-execution PEP returns an error, then the action being protected by the PEP is canceled. Thus, we define PEP code for pre-execution before resource creation, update, and read requests, and this suitably enforces our access control by returning "Access Denied" errors as appropriate. This code reads an external PostgreSQL database to determine user privileges (Figure 2). In this figure, arrows represent directions of control during data access and policy enforcement. The API controls access policy, which in turn controls iRODS resource access.

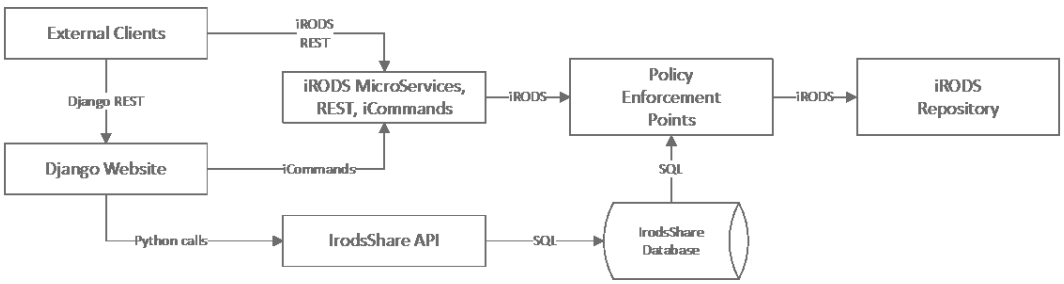

Figure 2. The access control mechanism interposes policy enforcement points between iRODS resources and a user website, thus providing consistent access regardless of the access mechanism in use.

In order for this to work, all resources being protected in this fashion must be world-writeable according to each user's iRODS protections. We accomplish this by making all resources writeable to a designated "HydroShare" iRODS group, in which all users are members. Thus, the access control scheme is subtractive; it denies access to resources to which the iRODS user would otherwise have access according to normal iRODS protections. Because of this implementation, it is unimportant whether the actual resources are owned by specific iRODS users or not; ownership in iRODS is orthogonal to ownership in IrodsShare, and we are currently using a proxy user as the iRODS owner.

The IrodsShare API

The contents of the access control database are managed by a Python API (to interact with the Django/Python website, and to suit the maintenance requirements of the HydroShare team). The API requires authentication of a HydroShare user and acts on behalf of that user. All policies regarding access control are implemented via the API, which modifies the PostgreSQL access control database directly. In turn, this database is shared with iRODS microservices to control the PEPs as discussed above.

While – ideally – we would like this to be accessible via iRODS REST microservices, that feature has not yet been implemented.

Sharing Policies

In any social sharing situation, there is a balance between enabling sharing and minimizing "SPAM" or other unwanted information. This concern complicates operating policies somewhat, because sharing can in principle be a source of SPAM. For example,

1. A user must give permission to be made an owner of a resource. This prevents users being inadvertently made responsible for resources without their knowledge.

2. A user must give permission to join a group, after being invited by an authorized member. This prevents users from placing everyone in a group, for the purpose of broadcasting information to everyone.

Sharing with individual users is otherwise unmoderated, and anything shared with a group is immediately distributed to its members.

PROJECT STATUS

At present, only the database and Python policy engine are complete and tested, with documentation in the Sphinx documentation system. Deployment to HydroShare has been deferred due to the need to deal with issues in Django access control first. PEP enforcement has been designed but not implemented.

CONCLUSIONS

We presented a novel model of access control for scientific data that mimics file sharing in social networks, at the filesystem level. This mechanism strikes a balance between capability at the filesystem level and ease of implementation, by strategic use of iRODS dynamic Policy Access Points. By setting iRODS to "allow anything" and then constructing selective denials, we change as little of iRODS as possible to accomplish this.

This is just a first step toward moving social networking to the filesystem level. While critics of IrodsShare believe that it creates a two-phase commit problem with Django, this criticism is only valid as long as Django and iRODS are competing to be ground truth for access control.

As for the actual usability and utility of the project, only time and experience will tell. We believe we have struck a balance between utility, ease of understanding, and limitation of abuse potential.

Availability

The project is freely available from http://github.com/hydroshare/IrodsShare.

ACKNOWLEDGMENTS

This work was supported by the National Science Foundation under collaborative grants ACI-1148453 and ACI-1148090 for the development of HydroShare (http://www.hydroshare.org). Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

We are also particularly indebted to the scientists of the HydroShare project, who vetted our access control design carefully at repeated meetings over more than two years. The iRODS team at RENCI – and particularly Jason Coposky and Antoine de Torcy – provided crucial help in evaluating options for implementing this software, and implemented features of the Python library for iRODS specific to this project.

REFERENCES

[1] Reagan Moore, "Toward a Theory of Digital Preservation", *International Journal of Digital Curation* 3(1), June 2008.

[2] Terrell Russell, Jason Coposky, and Michael Stealey "Hardening iRODS for an Initial Enterprise Release (E-iRODS)" *Proc. 2012 iRODS User Group Meeting,* Renaissance Computing Institute (RENCI), Chapel Hill, NC.

[3] Terrell Russell, Jason Coposky, Harry Johnson, Ray Idaszak, and Charles Schmitt, "E-iRODS Composable Resources", *Proc. 2013 iRODS User Group Meeting,* Renaissance Computing Institute (RENCI), Chapel Hill, NC.

[4] Tarboton, D. G., R. Idaszak, J. S. Horsburgh, J. Heard, D. Ames, J. L. Goodall, L. E. Band, V. Merwade, A. Couch, J. Arrigo, R. Hooper, D. Valentine and D. Maidment, "A Resource Centric Approach for Advancing Collaboration Through Hydrologic Data and Model Sharing," *Proceedings of the 11th International Conference on Hydroinformatics (HIC 2014)*, New York City, USA, http://www.hic2014.org/proceedings/handle/123456789/1539.

[5] Tarboton, D. G., R. Idaszak, J. S. Horsburgh, J. Heard, D. Ames, J. L. Goodall, L. Band, V. Merwade, A. Couch, J. Arrigo, R. Hooper, D. Valentine and D. Maidment, (2014), "HydroShare: Advancing Collaboration through Hydrologic Data and Model Sharing," in D. P. Ames, N. W. T. Quinn and A. E. Rizzoli (eds), *Proceedings of the 7th International Congress on Environmental Modelling and Software,* San Diego, California, USA, International Environmental Modelling and Software Society (iEMSs), ISBN: 978-88-9035-744-2, http://www.iemss.org/sites/iemss2014/papers/iemss2014_submission_243.pdf.

[6] Heard, J., D. Tarboon, R. Idaszak, J. Horsburgh, D. Ames, A. Bedig, A. M. Castronova, A. Couch, P. Dash, C. Frisby, T. Gan, J. Goodall, S. Jackson, S. Livingston, D. Maidment, N. Martin, B. Miles, S. Mills, J. Sadler, D. Valentine and L. Zhao, (2014), "An Architectural Overview of Hydroshare, A Next-Generation Hydrologic Information System," *Proc. 11th International Conference on Hydroinformatics (HIC 2014)*, New York City, USA, http://www.hic2014.org/proceedings/handle/123456789/1536.

[7] Tarboton, D.G., Horsburgh, J.S., Maidment, D.R., Whiteaker, T., Zaslavsky, I., Piasecki, M., Goodall, J., Valentine, D., Whitenack, T. (2009). "Development of a community Hydrologic Information System." In: Anderssen, R. S., R. D. Braddock, and L.T.H. Newham (eds.) *18th World IMACS Congress and MODSIM09 International Congress on Modelling and Simulation,* Modelling and Simulation Society of Australia and New Zealand and International Association for Mathematics and Computers in Simulation, July 2009, pp. 988-994, ISBN: 978-0-9758400-7-8.

[8] Engestrm, J. (2005). *Why some social network services work and others dont Or: the case for object-centered sociality,* http://www.zengestrom.com/blog/2005/04/why-some-social-network-services-work-and-others-dont-or-the-case-for-object-centered-sociality.html, last accessed 5/9/2015.

[9] Boyko, A., J. Kunze, J. Littman, L. Madden, B. Vargas (2012). *The BagIt File Packaging Format (v0.97),* (Network Working Group Internet Draft), available at http://tools.ietf.org/html/draft-kunze-bagit-10, last accessed 2/20/2015.

[10] Lagoze, C., Van de Sompel, H., Johnston, P., Nelson, M., Sanderson, R., Warner, S. (2008a). *Open Archives Initiative Object Reuse and Exchange: ORE Specification Abstract Data Model,* available at http://www.openarchives.org/ore/1.0/datamodel.html, last accessed 3/3/2015.

[11] Alva Couch and David Tarboton, *Access Control for HydroShare*, internal project document, verson as of May 1, 2015.

[12] DataONE (2015). *DataONE Architecture, Version 1.2,* available at http://releases.dataone.org/online/api-documentation-v1.2.0/, last accessed 3/7/2015.

[13] Zaslavsky, I., Whitenack, T., Williams, M., Tarboton, D.G., Schreuders, K., Aufdenkampe, A. (2011). "The initial design of data sharing infrastructure for the Critical Zone Observatory," *Proceedings of the Environmental Information Management Conference,* Santa Barbara, CA, 28-29 September, EIM 2011, http://dx.doi.org/10.5060/D2NC5Z4X.