IDS – the INCF Dataspace

Raphael Ritz[1], Sina Khaknezhad[1], Chris Smith[2], and Sean Hill[1]

[1]INCF Secretariat, Stockholm, Sweden, and [2]Distributed Bio, San Francisco, CA, USA

Abstract

This contribution describes the architecture and design of the INCF Dataspace based on iRODS. The core concept of the INCF Dataspace is to enable collaboration for the individuals, research groups, and organizations that make up the membership of the INCF, whether collaboration is between individuals, or INCF-wide. To that end, the architecture of the system is designed to enable data discovery through data organization and meta-data querying, and to enable organizations to easily control the level of sharing of their data. This document will describe the capabilities needed to support this core concept, and how they map to iRODS components and features.

Index Keyword Terms— Data Federation, Data Sharing, Data Publication, Neuroinformatics

1. Introduction to the INCF Dataspace

The INCF Dataspace is a platform that provides capabilities for storing, sharing and annotating data amongst the researchers that form the membership of the International Neuroinformatics Coordinating Facility (INCF; www.incf.org), and potentially to share with the research community at large. The purpose of the INCF Dataspace is to enable collaboration between researchers through the sharing of experimental and analysis data, whether the collaboration is between individual researchers, or intended to be with all member organizations of INCF. Some of the use cases that the INCF Dataspace is intended to support include:

- Allowing researchers to disseminate data to other scientists in a globally distributed fashion, as stipulated by some grants, whether or not the other scientists are members of INCF. The INCF Dataspace must make this as easy as possible for the researcher. The researcher in this case might be part of an organization who already has a data resource being shared, where they can place their data to be published, but the Dataspace also needs to be suitable for researchers who might need to join an existing INCF regional node and provide their own storage, all the way to the case where the researcher needs to integrate with the Dataspace by creating their own zone.

- Providing capabilities so that researchers and research groups share data, where the level of sharing of the data is limited in scope to a small subset of the INCF membership, and where controlled access to data is mandatory. In addition to being able to protect works in progress before publication, this also allows data protection in cases where the data itself is restricted from dissemination by regulations. The sharing should also be as de-centralized as possible. Researchers should be able to leverage existing user and groups that INCF defines, but must also be able to create their own associations and collaborations on the fly, and use them in iRODS ACLs.

- Providing a secure, reliable and searchable archive for experimental and analysis data. Not only can the INCF Dataspace be used as a "backup" for important data, but data can also be tracked through its lifecycle, even though individual researchers might come and go from a particular project, research group or organization. No longer does a PI need to worry that the only copy of an important dataset disappears on the laptop of a leaving postdoc, or is a casualty of a corrupted hard drive, since there are many options for data storage and protection within the Dataspace.

- Browsing of very large datasets in place, regardless of whether they are located within a researcher's organization, or across the world in the data repository of another INCF collaborator. One can imagine a researcher browsing thumbnails of very large files remotely, so that they only need to retrieve the subset of data actually needed. For some types of files, in place browsing could also include retrieving single slices of data (e.g. parts of a BAM file representing genes of interest).

Having a mechanism for annotating data with meta-data, perhaps using automated extraction, and allowing users to search the entire INCF Dataspace using these meta-data terms. The Dataspace could also allow researchers to apply well-known and/or standardized ontologies as they become more widely defined and used.

The INCF Dataspace is implemented using the iRODS software (https://www.irods.org). iRODS provides various features to support the use cases described above, with federation for global data sharing, ACLs for fine-grained control of data sharing, a data catalog for meta-data annotation, and a host of other features for "information lifecycle management" (http://en.wikipedia.org/wiki/Information_Lifecycle_Management), allowing the INCF Dataspace to implement policies for data management.

2. Overall Architecture of IDS

The virtual and distributed nature of INCF gives rise to a number of requirements for the sharing and management of data. Since member organizations are autonomous, there must be mechanisms for organizations to control the dissemination and access to the data they produce; seeing that INCF is about collaboration, discovery of and access to shared data must be uniform no matter where the data is located; and because data is widely distributed globally, issues of data access latency and service partitioning must be considered.

Following recommendations from the first INCF workshop on Cyberinfrastructure for neuroinformatics, the INCF Dataspace is built upon an iRODS federation (i.e. a multi-zone iRODS architecture), where iRODS zones are used at the regional (in a geographical sense) and organizational level, in order to meet the needs stated above. An iRODS zone is an independent instance of the iRODS system. It contains an ICAT-enabled server (the catalog), and possibly multiple other iRODS servers that provide access to storage resources. Since an iRODS zone is a separate administrative domain, organizations with stringent data sharing requirements may set policies on their own sharing, without affecting the policies of other regions or organizations; because the iRODS servers mediate access to another zone's data, users access remote zone data in the same fashion as local zone data; furthermore, since users interact chiefly with a local "home" zone, issues of data access latency and network partitioning are mitigated.

Figure 1 shows a high-level view of the structure of INCF's Dataspace as built on iRODS and illustrates a number of concepts of the Dataspace architecture. The first is that there is not necessarily a one-to-one mapping of zones to organizations. Zones are mainly run regionally, and organizations that are located within that region have the option of being part of that zone (for example, 'Organization3' is part of a zone that is a regional zone hosted by 'Organization2'), although a given organization might want to run their own zone if they provide access to large amounts of data, or have particular sharing constraints (such as 'Organization4' in the illustration).

Another illustrated concept is that there is a central 'incf' zone that serves as a central point for

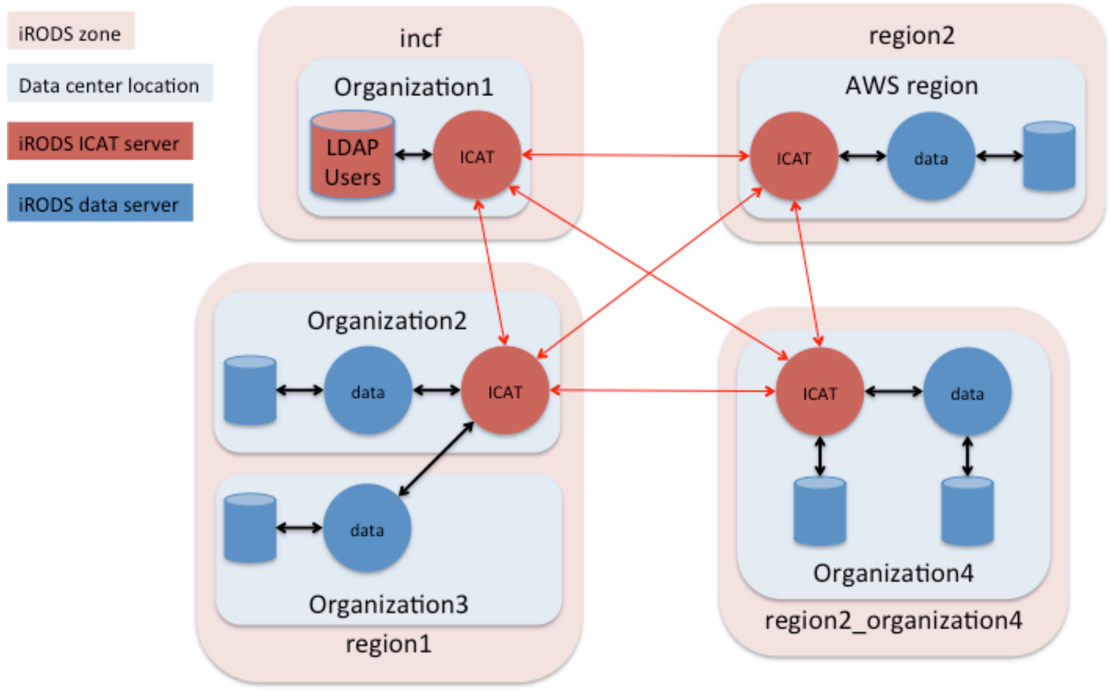

Figure 1: DataSpace Architecture

administration of the Dataspace. In addition to providing identity management services for INCF users using the existing INCF LDAP directory, this zone is used to share information needed to monitor and configure the multiple zones of the federation.

A third concept is that of running services within Amazon Web Services (AWS). Some regions or organizations might choose to run their zone, or provision their data, onto resources running within AWS EC2. What is important to note is that the architecture of the Dataspace does not mandate the infrastructure required to be part of the Dataspace, only the rules for zone interaction and common data management policy.

The remainder of this document will describe the core dataspace services of the INCF Dataspace architecture - those components necessary for deploying, configuring and running the data services for the Dataspace.

3. Core Dataspace Services

This section will describe the requirements and components that pertain to deploying, configuring and operating the iRODS federation itself. This includes aspects such as software packaging, configuration procedures, user authentication, and system scalability. To start, it is useful to understand the iRODS components that each organization might need to run to fulfill different logical service roles within the federation.

3.1. Component Overview

The iRODS ICAT-enabled server (IES) is a machine that provides the gateway to the ICAT database (the central repository of the zone). There must be one of these for each zone in the Dataspace. Every data server and client in the zone contacts this machine, so it must be kept available in order to access the zone's data. The zone IES is also the point of contact for servers and clients in other zones, so this machine must be reachable (in a network sense) over the Internet on the defined iRODS service ports. The IES is also responsible for authenticating the zone's users. If a user from a remote zone contacts an iRODS server, that iRODS server will forward an authentication request to the users 'home' zone.

An iRODS data server (or just "data server") is a machine that acts as the gateway to a particular storage resource (often a file system, although can be other storage types such as tape systems, etc). As long as this machine is reachable by the IES for the zone, it does not actually need to be otherwise reachable, as the IES can be used to mediate data transfer requests.

iRODS clients use the iRODS protocol to make requests of the system. iRODS clients can actually connect to any server in the zone, and have that server pass their requests on to the target IES and other data servers.

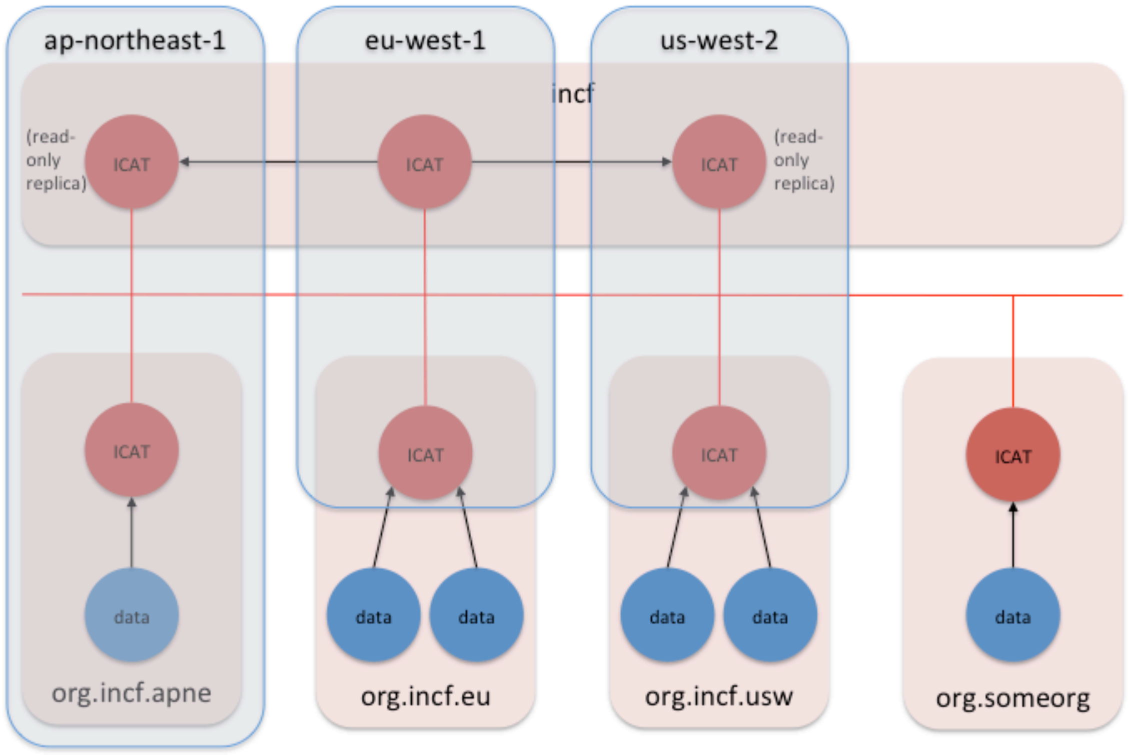

Figure 2: Physical Architecture of IDS in AWS

For the INCF Dataspace, there are three levels of zones: a central INCF zone, multiple regional zones, and multiple organizational zones. Each zone has at least one IES instance, and shares a set of administratively defined rules. The central INCF zone is mainly used to provide user authentication services against the INCF LDAP directory, but also initially hosts organization's data until the need for a zone for a particular region arises (due to large number of organizations or a large amount of data). The INCF zone also provides a small amount of storage space for administrative use, such as sharing federation configuration and providing a repository for access/audit logs. The INCF zone is replicated across geographical regions (using 4 AWS regions) with a single master IES and multiple slave IES instances. Figure 2 shows how the iRODS architecture maps into AWS regions.

Regional zones are composed of an IES node that is either hosted by a volunteer organization, or possibly could be hosted in AWS if no suitable host is available. Each regional zone are responsible for managing the data servers hosted by organizations within the zone's region. The regional ICAT servers can be replicated in a master-slave mode if required for performance reasons. Each organization that wishes to provide federated access to some data will run a data server, and the data server and data "vault path" will be defined within the regional zone's ICAT. This vault path can be anywhere in the file system, including on distributed storage such as NFS mounts, Lustre, or other distributed file systems.

Organizational zones are created for those organizations that provide access to large amounts of data (and require their own ICAT DB instance for scalability reasons), have specific data sharing requirements (e.g. regulatory), or just feel they need to run their iRODS system autonomously (perhaps due to special requirements for resource types or customized iRODS rules). These sites run an IES and any number of data servers necessary to support their environment. Any organization that wants to maintain the highest level of autonomy over their data should run their own zone, although the administrative burden is higher than providing a data server alone.

3.2. Software Deplyment and Configuration

This section describes all the parts of the system needed to get an iRODS server up and running, whether it will be the IES for a zone, or an organization's data server. The packages and processes defined make it as easy as possible for a researcher to publish data and join the Dataspace, as described in the use cases.

3.2.1. Server Software Packages

INCF provides customized iRODS builds at https://github.com/INCF/ids-tools (i.e. stock iRODS releases, potentially with INCF-specific patches applied) packaged for the following target operating systems.

- Debian 6.0 ("squeeze")
- Ubuntu 10.04 LTS
- Ubuntu 12.04 LTS

Binary software is provided in deb package format for installation. There might be different flavours of packages available, depending on whether a data server is being set up (no DB support required) or an IES using Postgres, MySQL or Oracle is being set up. For those who have requirements for other systems, the iRODS source distribution with INCF patches will also be made available for "do it yourself" installations.

3.2.2. Server Configuration

iRODS server configuration falls into two categories: the server runtime environment, and the iRODS zone configuration.

The server runtime environment is configured as part of the deb package installation, as it mainly treats with configuration such as the location of configuration files and log files, and mostly follows Debian standard practices for locations and startup mechanism.

The iRODS zone configuration can be "semi-automated", but still must be driven by an organization's system administrator and the zone's administrator. Configuration management and automation is managed using Fabric (http://www.fabfile.org).

Note: this document will not cover how to make an iRODS server "visible" to the other nodes in the federation. Operational parameters such as required port numbers and protocols will be provided in documentation, but it is up to an organization to make sure the iRODS server can properly communicate over the network.

3.2.3. INCF Federation Configuration Repository

In order to facilitate the uniform configuration of the zones within the Dataspace, INCF provides a configuration repository that can be leveraged both by setup automation scripts (e.g. to download current configuration information, fabfiles, etc) and zone and organization administrators (e.g. to retrieve documentation, policy rule examples, etc). The configuration repository is mostly composed of files (e.g. fabfiles, software packages). This has the benefit of being able to manage these files within a version control system. Files can be retrieved directly via the version control system commands (e.g. git clone), and they can also be served up via HTTP in some form of package format.

The use of a version control system for the repository files also allows for updating the configuration in a managed way. There can be identified

"INCF federation admins" in the INCF user database that have rights to update the repository information. This can be used during the zone and data server deployment steps so that the central configuration is kept up to date, but in a controlled and audited fashion.

3.2.4. Data Server Setup

To configure a new data server, the person doing the software installation can run the script *ids-setup-data-server*. This script will prompt for information such as the zone that the data server will be part of, the short name of the organization the server belongs to, the location (in the file system) of the new storage resource's vault path, the Unix user that should be used to run the service and own the files in the vault, and other bootstrap information. It will then retrieve a Fabric *fabfile* corresponding to the data server configuration steps and run a driver script to apply the configuration. This will include setting up the top-level organization namespace as defined within the INCF Dataspace Policy document, and updating any of the system rules such as those to manage storage access and default resource selection.

Note that some of the configuration will require the organization's administrator to authenticate to their zone's IES using 'storageadmin' credentials. The zone administrator should establish this credential, and send it to the person doing the installation, before installing the data server.

3.2.5. IES Setup

Setting up the zone server is a bit more involved, as the iRODS catalog needs to be initialized, and the various peer zone relationships need to be established. After installing the appropriate (for the database type) server package, the zone administrator can run the script *ids-setup-zone*. This script will prompt for some bootstrap information needed to get the zone running (e.g. zone name, the IES's FQDN, service user, iRODS admin username and password, etc, etc). It will then retrieve fabfiles needed to set up the database, bootstrap the iRODS zone using that database, apply any configuration defined by the policy document, and register the existence of the new zone within the INCF federation configuration, by making an update to the configuration repository.

3.2.6. Ongoing Zone Configuration

A number of tasks need to be run periodically, or on-demand, to keep each zone in sync with the overall federation configuration. Information that needs to periodically be synchronized and applied locally includes:

- The list of INCF users and groups. The script *ids-sync-users* manages this process.
- Local zone updates to rules. The script *ids-sync-zone-rules* is used to run this process. Note that any rules defined locally and not part of the zone's central rule base will be overwritten in this process.
- List of zones in the federation. The script *ids-sync-peer-zones* sets up the remote zones in the local zone. Note that since this synchronization process will not necessarily be performed at the same time for all zones, there might be a period of time that some zones aren't reachable.

In all cases, these scripts could be run from cron (e.g. daily), or potentially by receiving a notification from the INCF Federation Configuration Repository, if the underlying version control system can support generating notifications, or could even be initiated via ssh from the central zone if a local zone is willing to delegate some administrative rights to INCF administration staff.

3.3. Publishing Existing Datasets

A common use case when adding an organization to the federation will be that an organization will want to share existing datasets through the Dataspace. Moreover, they will most likely not want to move or replicate this data, especially if it is large. Again, addressing the use case that any researcher should be able to publish data, this process should be made as easy as possible, and should lead the researcher through data registration, and proper setting of access controls.

Once an organization has installed their data server, they can "register" existing data into iRODS without having to move it. The only two requirements are that: a) the data server can "see" the datasets (i.e. if the data is on a shared filesystem … say Lustre … then the iRODS data server can mount the volume), and b) the iRODS service account has read access to the files and directories that compose the dataset. If these criteria are met, then it's straightforward to run the iRODS 'ireg' command to make this data available at a chosen part of the iRODS namespace. Once registered in iRODS, it can be made subject to the same ACLs as normally imported data.

Detailed documentation will be provided that describes the process to register data into an organization's namespace, and to set proper initial ACLs.

3.4. User Authentication and Authorization

In order to support the ability of researchers to limit the scope of data sharing within the Dataspace, the Dataspace supports proper identification of users, and provide the means to limit user access to data.

iRODS maintains its own database of users and groups that it uses to identify object ownership and for use in ACLs. In a federation remote zone users can be defined within a local zone by adding a reference to the remote user into the local user database. For example, if 'csmith' is a user in 'zone1', then the administrator of 'zone2' needs to add the user 'csmith#zone1' to the 'zone2' user database. Once added, this user name (with zone qualifier) can be used in ACLs. And if 'csmith#zone1' makes a request of a server in 'zone2', the server will redirect the authentication request to the 'zone1' IES.

For the INCF Dataspace, INCF synchronizes the list of users from the INCF LDAP directory into the INCF central zone (named 'incf'). Then, every other zone in the federation can use the approach above to reference the users in the incf zone when setting ACLs on data shared in the INCF context. This provides a consistent username space across the entire federation. Note that this does not preclude the definition of and use of other user names in a given zone.

Note that in iRODS, group names cannot be referenced from other zones (i.e. whereas one can add 'user#incf' to the local zone, one cannot add 'group#incf'. INCF provides a list of groups that need to be defined within every zone to provide uniformity across the zones of the Dataspace, and these groups are synchronized using the processes described previously.

A side effect of this design decision is that a Dataspace client must connect through the incf zone in order to browse all other zones in the dataspace. The implication is that all queries for all zones will use the incf zone servers as a gateway. In order to avoid the bottleneck of a single iRODS server for all access, the incf zone ICAT server is configured with a number of slave ICAT servers, with a number of supporting iRODS servers (non-ICAT) that can also perform the gateway function, or some combination of both. The extra servers are spread in different geographies to enable "local" network connections as much as is possible in order to reduce the latency of connections.

3.4.1. Authentication

The default mode of operation in iRODS is to authenticate against the user information in ICAT. There is some level of integration with external systems, namely Kerberos and GSI, but the user still must exist in the iRODS DB. Moreover, if one moves away from the iRODS CLI, the only "common" authentication mechanism between all the iRODS clients is username (only the iRODS CLI supports Kerberos and GSI).

PAM authentication in iRODS has been implemented for the IDS and became available in the iRODS 3.2 release in early October 2012. This has the effect of allowing users to authenticate with username and password, but the verification of identity is being done through the PAM sub-system, rather than against credentials in the iRODS user DB. The INCF Dataspace leverages this new PAM support for the initial roll-out of software packages. The main benefit of this PAM support is that users no longer have to remember two passwords: one for INCF and one for INCF's iRODS federation, as authentication happens (via PAM) directly against the INCF LDAP server.

In future phase after the PAM support, the intention is to augment the iRODS authentication to allow people to log into the Dataspace using OpenID credentials, so that using, for example, a Google identity to log in to INCF's iRODS federation would be possible. This support will depend in part on work that DICE is doing to support Shibboleth, which exhibits similar characteristics to using OpenID.

Once all done, the intention is to support the following types of authentication use cases:

- Local zone users can authenticate with local zone username and password (i.e. the administrative user case).
- 'incf' zone users can authenticate to any IES or data server within the federation using their INCF username and password.
- 'incf' zone users can authenticate to any IES or data server within the federation using a third-party identity such as a Google ID via OpenID, where the OpenID has been associated with their INCF account.

3.4.2. Authorization

iRODS allows users to put ACLs onto collections and data objects within the federation. When a user creates an object in iRODS, that user "owns" the object, and can assign read access, read and write access, or full ownership to other iRODS users and groups. An owner can also set the "inherit" mode on a collection, so that any sub-collections or objects created within that collection inherit its ACLs. Much like for the authentication use case, iRODS leverages its internal database of users and groups for use in iRODS ACLs.

Within the INCF Dataspace, there is a need to be able to share data and collections from the level of individual users, through labs, organizations, all the way to INCF-wide. The iRODS ACLs support this, in conjunction with the use of a central database of users and groups from the ‘incf’ zone. The following use cases can be supported:

- Can share data with individuals (as many as needed) by adding a read or read/write ACL for ‘username#incf’ on a collection or data object.
- To share with groups, such as labs or organizations, once the group is created, read or read/write ACLs for ‘groupname#incf’ can be set on a collection or data object. The INCF Dataspace policy states that the ‘incf’ zone will define a group for every organization and potentially other subsets of INCF users (such as projects or labs).
- Group membership management can be assigned to selected members of groups by making them part of the special “groupadmin” group. This can be used to let certain users within an organization manage that organization’s groups.
- The special group ‘incf_public’ will be defined in every zone, and will consist of every user defined within the ‘incf’ zone, and thus would include every INCF member, so ACLs could be assigned to objects to allow sharing amongst all INCF members.

More direct integration with external user management systems, such as LDAP directories, will be explored with the DICE group. This type of integration would allow one to reference LDAP groups and users directly in an ACL, rather than requiring their definition within the iRODS user DB. This support is on the roadmap for iRODS, but currently has no timeline associated with it. In the interim, scripts are implemented that will synchronize LDAP user and group information with the iRODS user DB for the ‘incf’ zone, so that administrators only need to make adjustments in one location (i.e. LDAP).

3.5. System Scalability

iRODS itself is well architected to support a secure and reliable platform for managing data, but to achieve higher levels of performance and reliability, some choices need to be made that often involve trade offs in cost and complexity. For instance, if an organization wants more reliable storage, it needs to double its storage capacity and mandate some level of replication of data within the organization using iRODS rules.

3.5.1. ICAT Scalability

The use of multiple zones helps increase the scalability of the entire Dataspace, since each zone is responsible for only a subset of files, but if any particular zone supports a large number of users, or a large number of data servers, it could become a bottleneck. Number of files and directories is of particular concern when scaling the ICAT. The system can support many millions of objects in the ICAT, but query performance can tend to suffer if this number gets too large.

In addition to approaches such as making copies of popular datasets in other zones to move traffic from a “hot” zone, there are a few ways that the IES and ICAT DB can be made more scalable.

1. Increase the capabilities of the IES node mainly through more RAM and better IO capabilities. The IES is very infrequently CPU-bound, but multiple cores are helpful in serving a large number of concurrent requests.
2. Set up a second IES as a “slave”. This node can be used to service queries, but not allow modification, so if the IES’s problem is in supporting lots of queries, this could help.
3. Set up a second IES “master” node. While not mentioned in the iRODS documentation, this mode of operation (in production at IN2P3, for example) consists of two (or more) iRODS servers that are both configured as IESes. DNS round robin is used to select between the servers, and each are equivalent in terms of the operations they can perform. The only usage mode that might cause problems is if a site expects lots of concurrent write operations to the same data object in iRODS, which isn’t particularly common in much scientific type data analysis. Note that this mode also provides some level of availability, since the iRODS service is available as long as one of the IES nodes is operational (although the iRODS client might not be particularly good at reconnection).
4. Use a more scalable database backend, using active replication such as pgpool for Postgres, or RAC for Oracle. RAC in particular might have some opportunities to optimize queries across nodes.

Note that configuring an INCF iRODS zone into the configurations described above in 2, 3 and 4 require manual configuration, and will not be supported via automated mechanisms.

3.5.2. Data Server Scalability

Given that the primary purpose of the data server is to perform data transfer operations, the system running the data server must be optimized for that function. To that end, data server scalability usually boils down to sizing the server properly for concurrent I/O operations. For systems serving local disk arrays, this could mean a requirement to stripe data across multiple disks, using hardware or software RAID. For systems using NFS for backend storage (or Lustre, etc), it would mean adding adequate network capacity (10GigE, or bonded NICs). For servers that have data "hotspots", lots of RAM will help by giving the system lots of buffer cache space.

If a data server starts becoming a bottleneck, a good approach to ameliorate this would be to create a second data server serving another resource, and using a resource group to spread iRODS data accesses across the two nodes. iRODS can be configured to use the RMS to choose data servers based on load.

3.6. Monitoring and Notification

In the context of iRODS, monitoring has a couple of connotations. One is the monitoring of the operational state of the system, and the other is the monitoring of data access and usage, for the purposes of data lifecycle management and for sending notifications when particular data events occur.

3.6.1. Operational Monitoring

Monitoring the operation of the iRODS services is mainly out of scope for this document, as individual sites will leverage the tools and frameworks that have been chosen to generally monitor all the services that site provides.

It is worth mentioning the iRODS Resource Monitoring System (RMS … https://www.irods.org/index.php/Resource_Monitoring_System). This system has been specially developed to monitor the attributes of the storage resources being provided by the data servers in an organization. Other than just providing information about storage volume usage and data server availability (up or down), the metrics that are collected by this system can be leveraged by the resource selection algorithm when dealing with resource groups (see the section System Scalability on data server scalability). When used for this purpose, data servers can be selected for 'put' operations based on "load", where "load" is configurable based on a number of metrics of system health such as disk space, cpu/mem utilization, etc.

3.6.2. Monitoring Data Activity

A more interesting type of monitoring, in the context of data management, is monitoring accesses to the objects within the Dataspace. The INCF Dataspace Policy document describes the events that each zone will report, and some of the use cases.

iRODS provides audit logs, which when turned on log events into a special table in the zone's ICAT. This data can then be extracted from the database and turned into a report of system activity. One downside to the audit logs is that if objects or users are removed from the zone's ICAT (or the 'incf' ICAT), then some of the fields are not resolvable.

Thus, a better approach is to generate an event trail via the rules engine. For each event of interest, an appropriate system rule entry-point will be identified, and then an "event generation" micro-service, implemented as part of the initial INCF iRODS software package, will be used to generate a log of event records for each zone. These event logs will be post-processed according to the policy described in the INCF Dataspace Policy document, using scripts and pipelines that will be provided with the software package.

Moreover, this same mechanism of generating event logs could also be used to generate notification events. That is, the event-logging micro-service could forward an event to a service that maintains a list of subscribers who would like to know when particular events occur. If an event occurs, and the subscriber has rights (known from the ACL) to see the event, the service will forward the event. There are various options for implementing this, but any solution would most likely include some sort of messaging system for the micro-service to use to send the event to the subscription service.

A simple option is to use iRODS own XMessage system (https://www.irods.org/index.php/XMessage_System). It has the advantage that it's already part of the iRODS installation and integrated with the rules engine. On the downside, it is specific to iRODS, so doesn't have many programming language bindings, so the subscription service would need to fit the XMessage interfaces. It also doesn't have a lot of features, especially for durability of messages. Limitations notwithstanding, the initial implementation of the audit trail is using the XMessage system. The iRODS rules engine already has hooks to generate messages when rules are triggered, so capturing logs can be implemented with a simple XMessage client that filters on the messages of interest before storing the log trace in log files, which can be used for further processing.

If the XMessage system approach does not meet future needs, other options could include using one of the many AMQP implementations (http://www.amqp.org/), which has the advantage of being very well used and tested, and is "standardized", so there are many choices for both server and client implementations. It also has a lot more features and allows one to implement many more messaging patterns that might be useful (e.g. queuing or publish-subscribe). Another option could be to use Amazon's Simple Queuing Service (https://aws.amazon.com/sqs/), which doesn't have all the features of AMQP, but would allow easy deployment of subscription services to AWS as the system scales up to the size of the INCF Dataspace.

On the client side, it would make sense to present information through existing, well-known client tools. For example, each INCF user could have a personalized (and password protected) RSS feed of events of interest, or the subscription service could deliver event messages through XMPP, allowing for integration with (say) a Google Talk client. A more heavyweight, but featureful integration might be to use AMQP on the client side, with some kind of "system tray" tool that subscribes to a personalized event topic, allowing the subscription service to send messages directly to the client.

3.7. Namespace Durability

The INCF Dataspace Policy document describes the mandated structure of the iRODS logical namespace as new regions and organizations are added to the Dataspace, but it does not deal with the case of how the namespace might evolve over time, especially given that PIs, labs and researchers are liable to move between organizations and regions. In these cases, there might be cause to move data from one part of the namespace to another. For example, the data associated with a particular moving lab needs to re-host data within the infrastructure of a new organization.

While it is possible to technically move data within an iRODS federation (or copy from source to destination, and then remove from source), this poses problems for researchers who might need to reference the data in a stable namespace. In order to alleviate this issue, future consideration must be made for keeping a registry of namespace changes that can be hosted by INCF. This would be composed of a database that maps old namespace elements to new namespace elements. In the case where data needed to be moved from one part of the logical namespace to another, the move operation would update this central repository in order to maintain the old path to new path mapping.

When dereferencing data objects that have moved, a "file not found" type error could result in a lookup within this repository. Native iRODS clients might not be able to automatically perform this dereferencing, but an enhanced INCF Dataspace client could be implemented to consult this repository.

Alternatively, data objects and collections could be referenced using a persistent identifier system such as the one provided by the European PID Consortium (EPIC; http://www.pidconsortium.eu/).

4. Conclusions

The goal of a global computational infrastructure for neuroscience is becoming increasingly important as data sharing and integration grows in importance and data sets and analysis grow in scale and complexity. The scale of the problem makes it intractable to solve at once, and smaller scale solutions have appeared to address subsets of the problem in subdomains of neuroscience. This makes any effort to reach an integrative solution more important.

The main objective of IDS - the INCF Dataspace - is to be the basis of an international infrastructure that spans various countries, in several subdomains, and supports several technical systems currently in use. A data sharing and querying infrastructure is the primary target, allowing users to easily add their data to the system, attach permissions to the data, and allow for other users to access the data. Being able to search for and discover data is of great importance. An important trait of IDS is its ease of use; an idealized goal is to be able to interface with the infrastructure in a Dropbox-like manner, fully integrated with a user's local machine.

Future steps will address the identification and support of common terminologies and ontologies for annotation and search as well as being able to support analysis frameworks and workflow systems.

Appendix A. IDS Command Summary

The table below lists various scripts that are mentioned in this document and in the policy document, along with a brief description of the intended purpose. Most of the scripts essentially wrap the Fabric 'fab' command calling a specific set of fabfiles for a given task. Most are administrative in function, but a couple might be called by end-users directly.

Script Name	**Description**
ids-setup-zone	This script is responsible for performing all the tasks needed to set up a new zone. It includes setting up the local zone configuration, and then calling other scripts to perform initial synchronization of the Dataspace configuration and namespace setup.
ids-setup-data-server	This script sets up a new data server within a given organization, and also sets up the new data server within the zone's configuration. It includes calling other scripts to set up the initial namespace and synchronize zone-level rules.

ids-setup-namespace	This script makes sure that the namespace either at a zone or an organizational level is set up properly, or it reports non-compliance with INCF Dataspace policy.
ids-sync-ldap-users	This script is used to synchronize the INCF LDAP directory of users and groups with the iRODS user and group database in the ‘incf’ zone.
ids-sync-users	This script causes the list of users in the ‘incf’ zone to be synchronized within a given regional or organizational zone, and also sets up groups to reflect the groups defined within the ‘incf’ zone.
ids-sync-peer-zones	This script sets up the zone names and endpoints within a given zone. That is, it defines all the zones that are remote to a particular region or organization’s zone.
ids-sync-zone-rules	This script sets up iRODS system rules in a given zone to reflect the rules required by INCF Dataspace policy. For example, a rule to enforce the INCF public read ACL on the ‘public’ collection.
ids-process-audit-logs	This script runs the pipeline that is used to process a particular zone’s audit logs, preserving required read ACLs, and rolling up and optionally publishing results to a central location within the ‘incf’ zone.
ids-event-logger	This script consumes rule engine audit messages using the iRODS XMessage client, and filters messages of interest into log files.
ids-init	This script, to be called by end-users, sets up the initial iRODS environment variables needed for the iRODS i-commands (i.e. command line utilities) to work within the IDS.
ids-copy-dataset	This script, to be called by end-users, copies a given dataset (i.e. a number of collections and data objects) from one zone to another, preserving ACLs and meta-data attributes.
ids-search-meta	This script, to be called by end-users, performs simple searches over meta-data terms (using the iRODS ‘iquest’ command), in a way that searches over all zones within the INCF Dataspace.