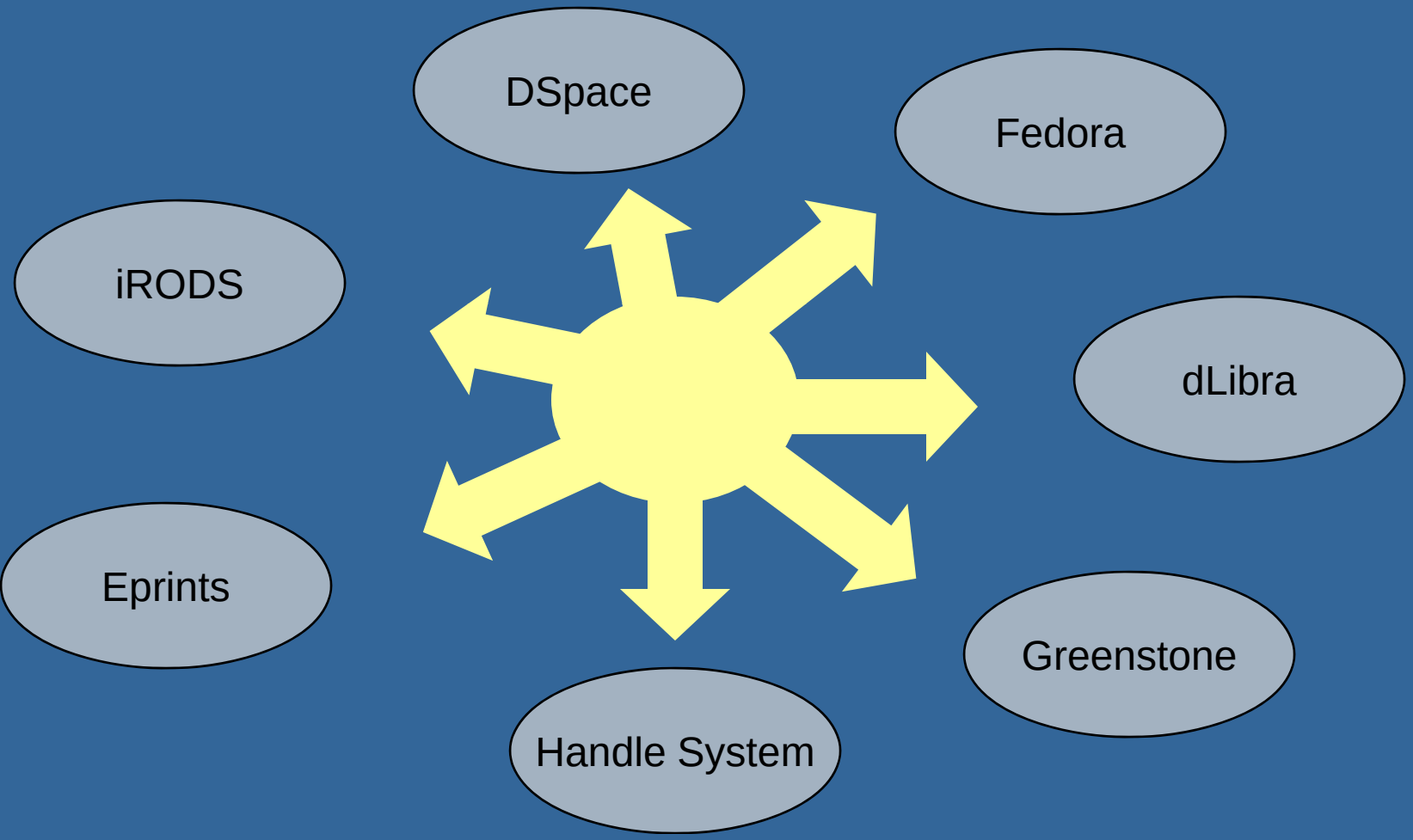
Enabling Inter-repository Access Management between iRODS and Fedora

Bing Zhu,
Uni. of California: San Diego
Richard Marciano
Reagan Moore
University of North Carolina at Chapel Hill

May 18, 2009

Atlanta

An Environment with Heterogeneous Technologies



Sharing Data Across Repositories

Enabling inter-repository data management allows us to share data by connecting the repositories of:

- Different groups, projects
- Different institutions, locations
- Different disciplines
- Diverse types of data
- Diverse hardware, software infrastructure

Issues for inter-Repository data management

- Object Model
- Virtual Registration of Digital Objects from One Repository to Another
- Inter-repository Service Management
- Policy Enforcement across Repositories

iRODS

iRODS - integrated Rule-Oriented Data System

A middleware providing functions to:

- manage distributed storages
- provide metadata support for digital preservation and search functions
- allow running distributed workflows to enforce system policies and harvest distributed computing power.

iRODS can be used for

- building datagrid
- building digital library
- building digital repositories

iRODS (continued)

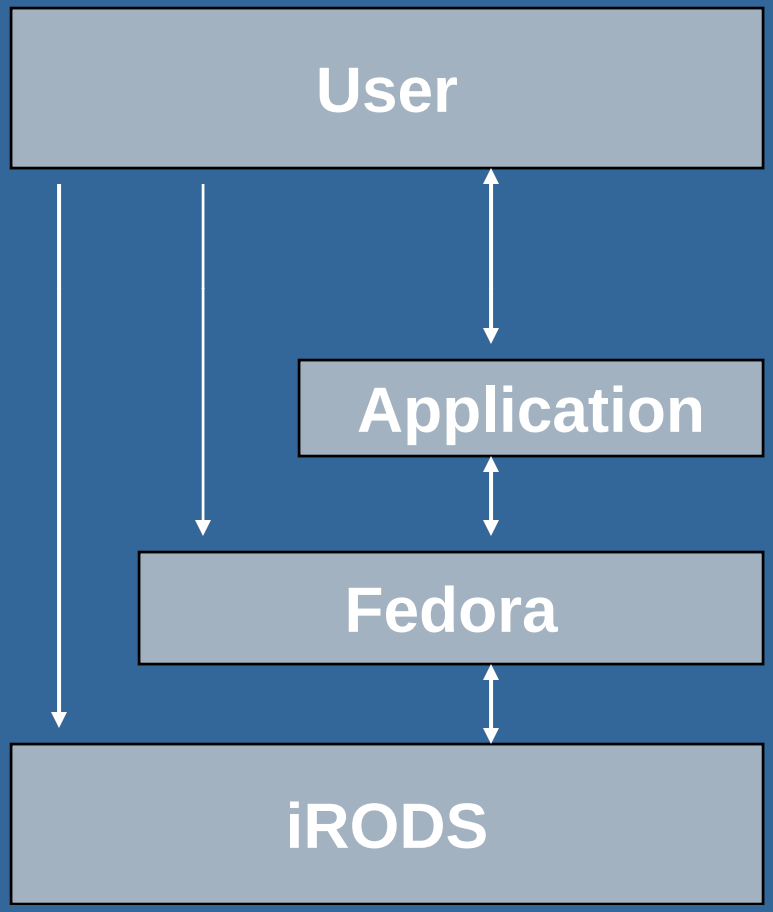
- Current version 2.02
- Scalable: Managing > 4 PB worldwide
  - Collections > PB, 100s of millions of files
- Federation of data grids
  - Flexible collaborative data sharing. Scaling >1 DB catalog.
- Transferring data
  - Parallel transfers ~70% of available bandwidth.
- Independent evaluations: NASA, etc.

iRODS (Continued)

iRODS Provides:

- A uniform global naming space for managing digital datasets
- An efficient data access mechanism using parallel data transfer
- Access to data stored on distributed systems
- Rich interfaces including C/C++ API, Java API, Perl, Web Service, and Python
- Support for remote manipulation of data sets to minimize the amount of data sent over the network
- …

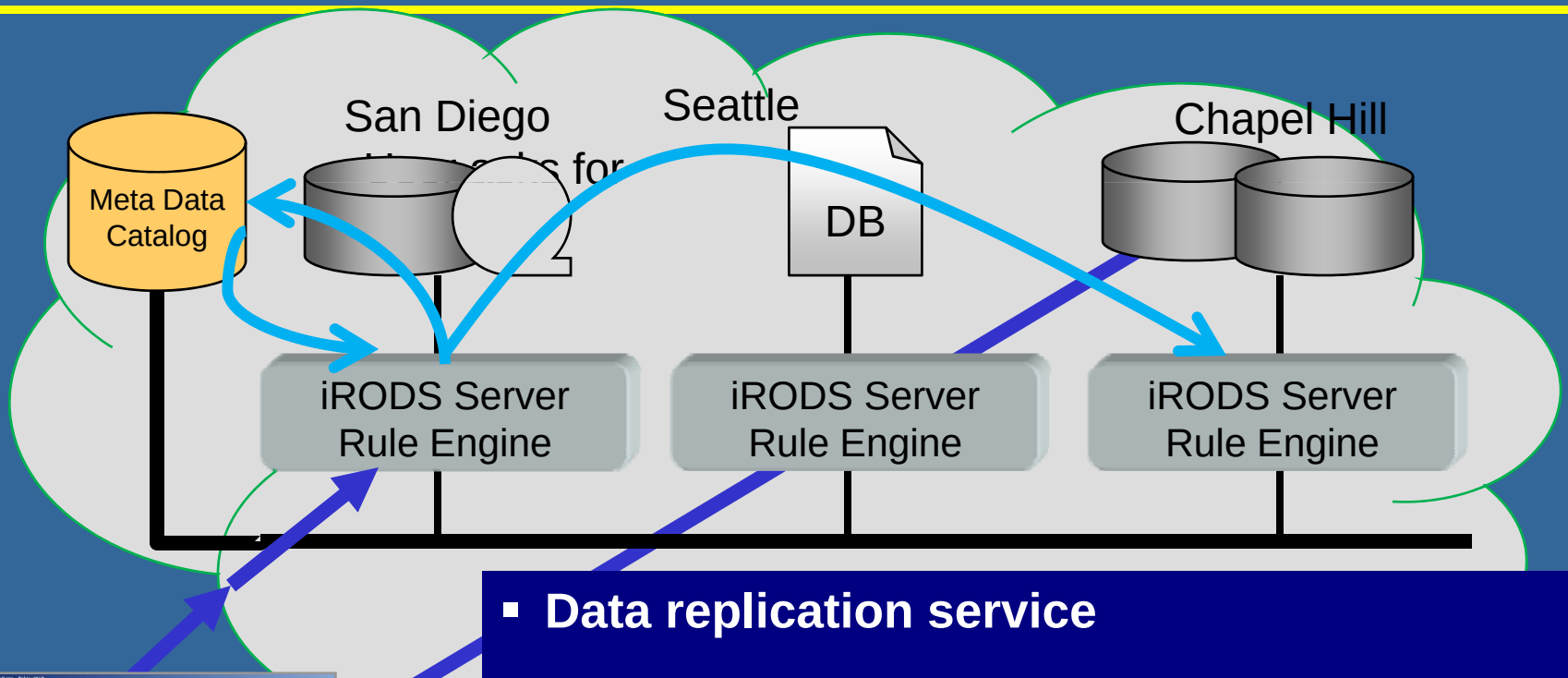
Why iRODS + Fedora?

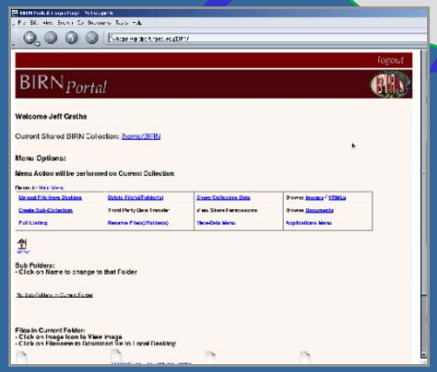


Complex Object Modeling Layer

Digital Preservation Layer

iRODS for Digital Preservation





- **Data replication service**
- **Periodic data integrity check**
- **Distributed storages for disaster recovery**
- **Metadata support for preservation description information**

D·I·C·E

i·R·O·D·S

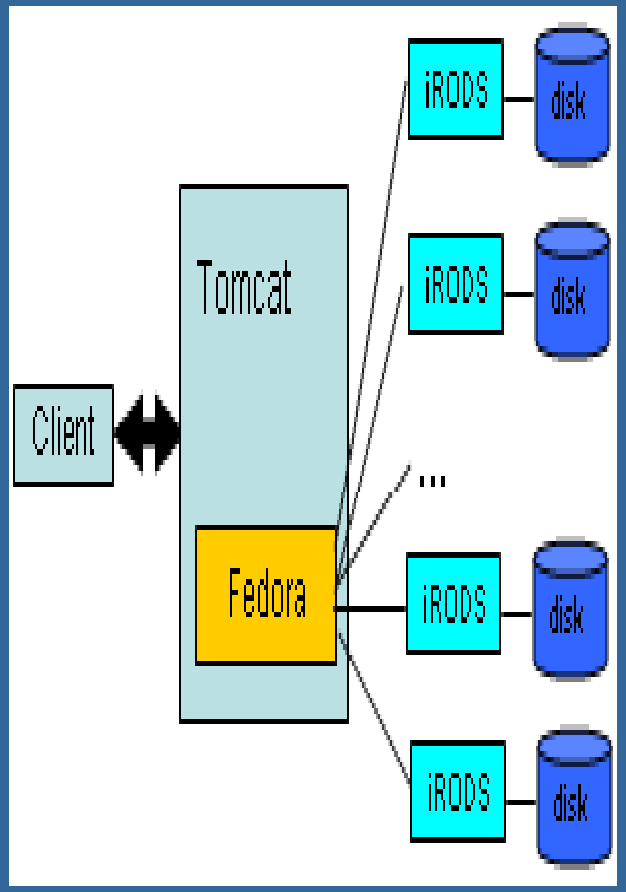
THE UNIVERSITY of NORTH CAROLINA at CHAPEL HILL

UCSD

NSF

THE NATIONAL ARCHIVES ARCHIVES.GOV

iRODS Storage Module for Fedora



- A replacement for Fedora's local storage module
- A standalone plug-in module (independent of Fedora release)
- iRODS manages both Fedora objects (XML) and data streams for Fedora (for managed content)
- Implemented based on SRB Storage Module for Fedora by DART project

iRODS Storage Module

- Manual Management of Distributed Stores - Admin Selects a Storage Resource for Storing Data Objects

- Auto Management of Distributed Storages - Use a Logical Storage Resource

Deployment of iRODS Storage Module in Fedora

- Download Jar files from iRODS web site

  https://www.irods.org/index.php/Fedora

- Copy the jar files into Fedora place under Tomcat

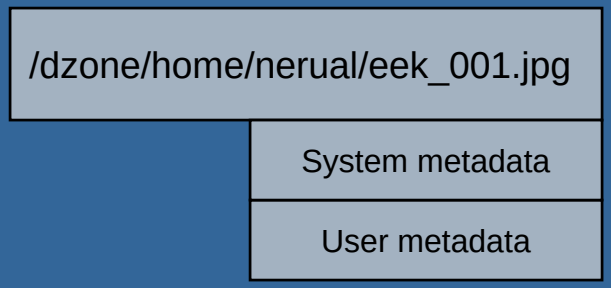
  $CATALINA_HOME/webapps/fedora/WEB-INF/lib

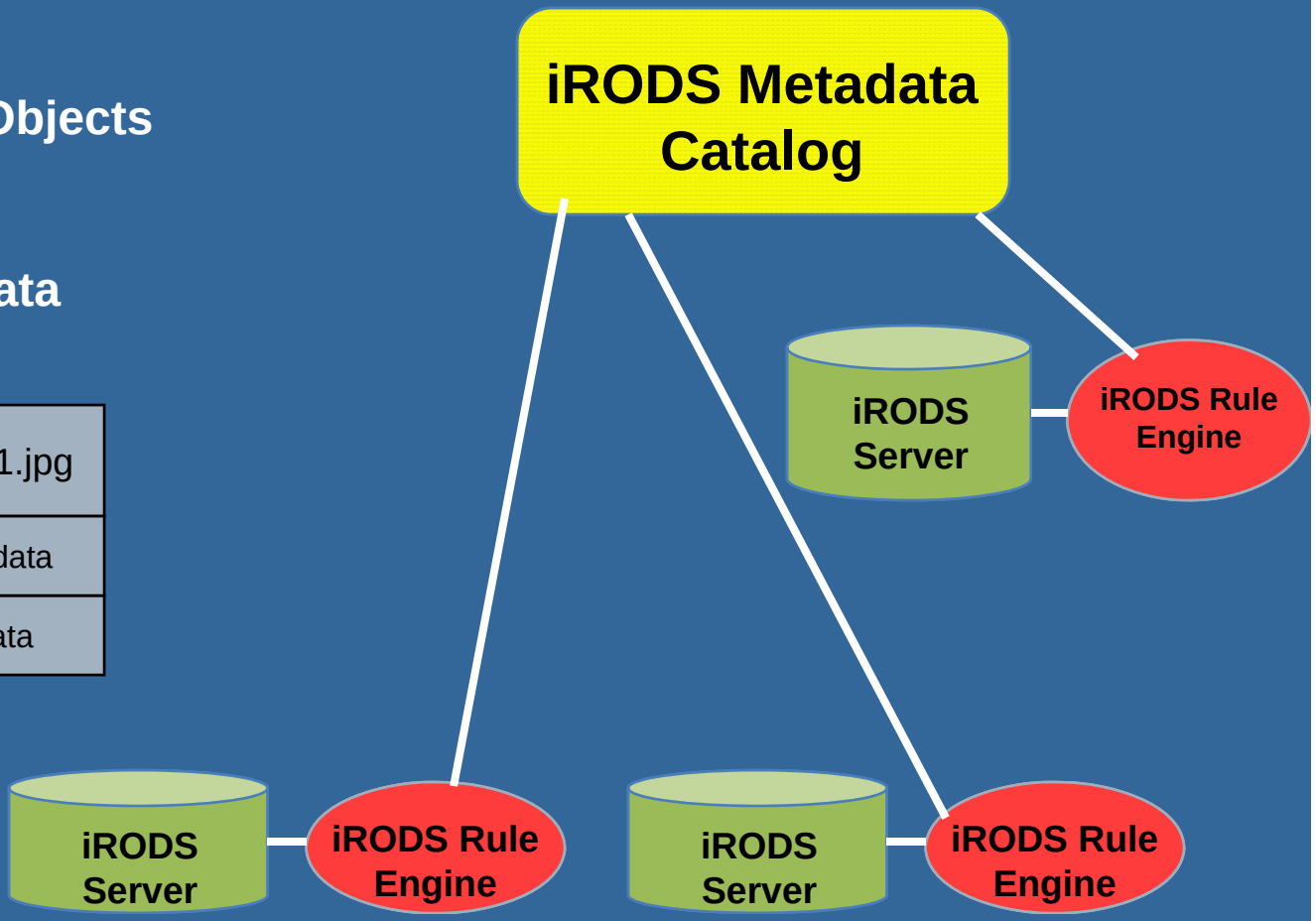
- **Edit the Fedora config file** $FEDORA_HOME/server/config/fedora.fcfg

```
<module role="fedora.server.storage.lowlevel.ILowlevelStorage"
        class="fedorax.server.module.storage.lowlevel.irods.IrodsLowlevelSt
        orageModule">
    <param
    name="file_system" :value="fedorax.server.module.storage.lowlevel.irods
    .IrodsIFileSystem"/>
    <param name="irods_host" value="irods.sdsc.edu"/>
    …
```

iRODS Digital Object Model

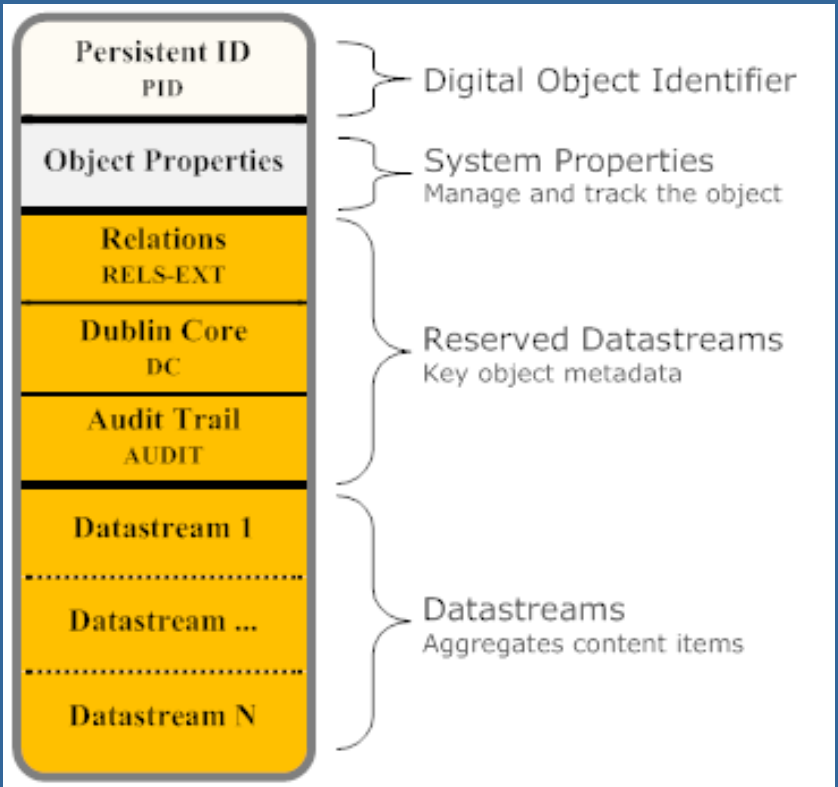
- File Based
- Distributed Digital Objects
- system metadata
- user defined metadata





Fedora Digital Object Model

A Compound Object Model

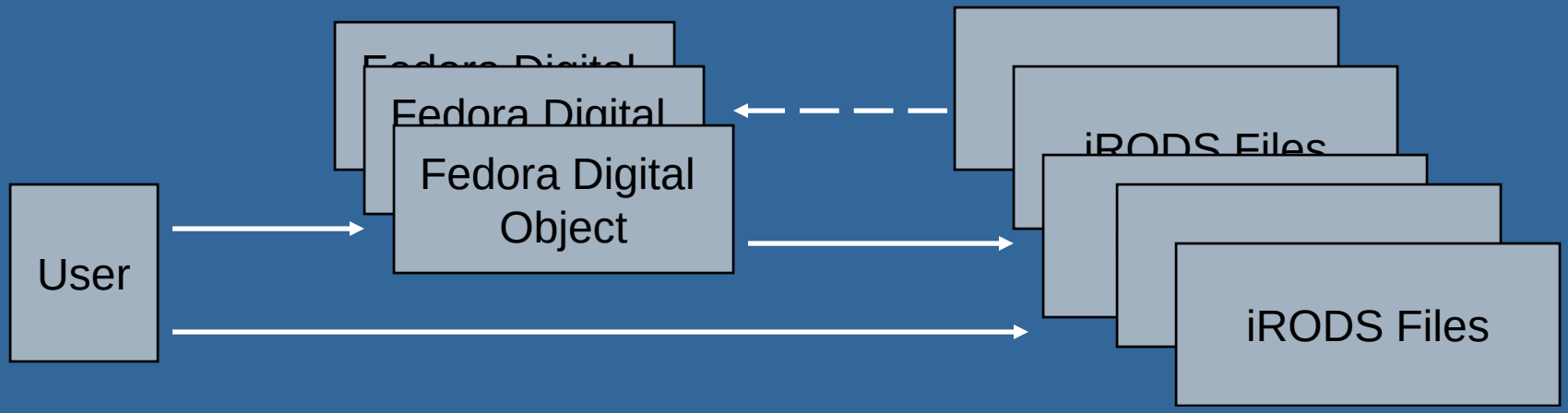


Four Types of control group for Fedora datastream:

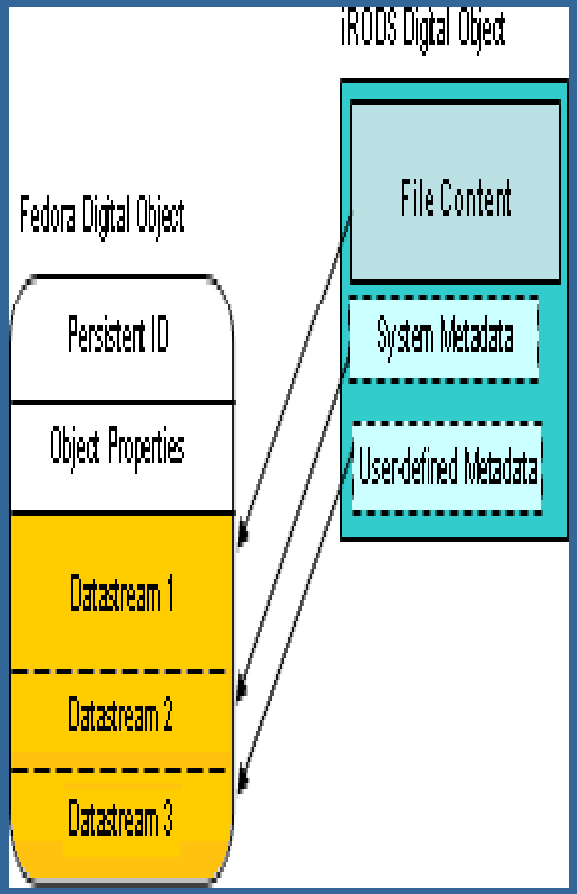
- **Internal XML Metadata**
- **Managed Content**
- **External Reference Content**
- **Redirect**

Referencing iRODS objects as Fedora External Datastreams

- An extension to the Fedora's external content manager
- iRODS files can be virtually linked inside Fedora
- Syntax:
  - http://irods_hostname:port/irodsfullpath?fs=irods
  - future: irods://…

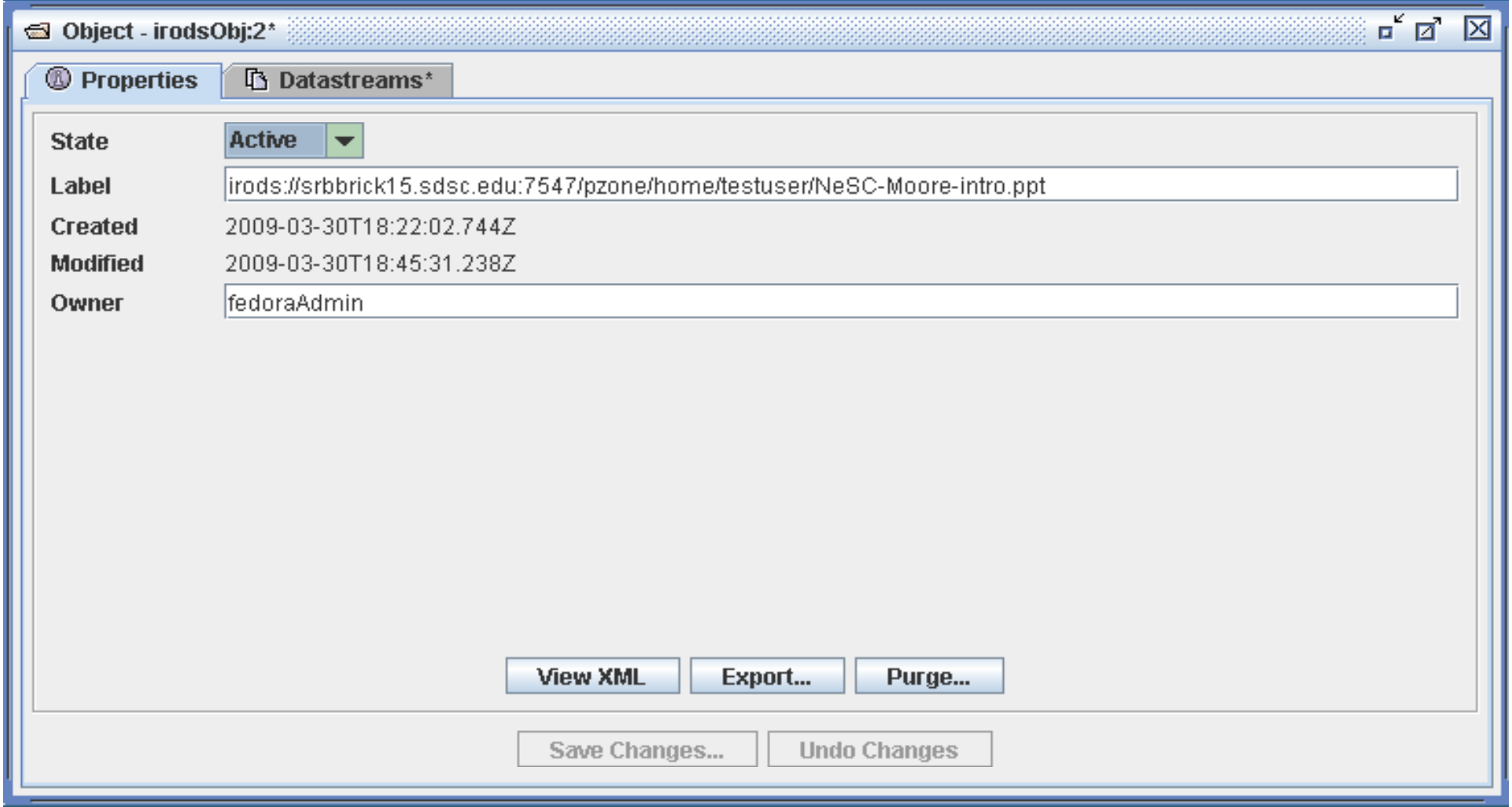


Registering iRODS Objects into Fedora

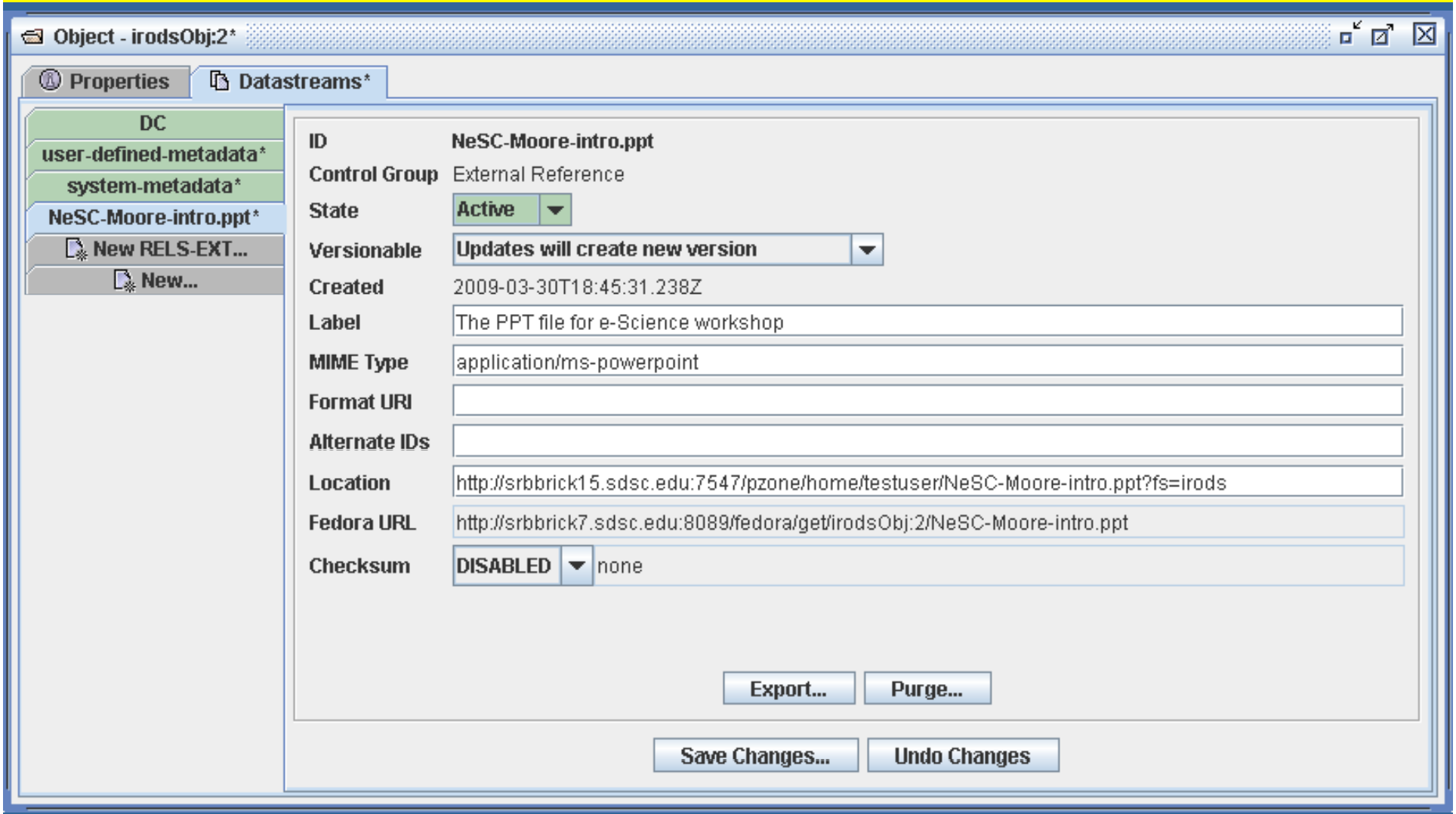


- Create a Fedora object. The full path of the iRODS object becomes the label of the Fedora object.
- Create an external reference datastream for the iRODS file.
- iRODS system metadata is registered as an externally referenced datastream in Fedora.
- iRODS user-defined metadata is registered as an externally referenced datastream in Fedora.

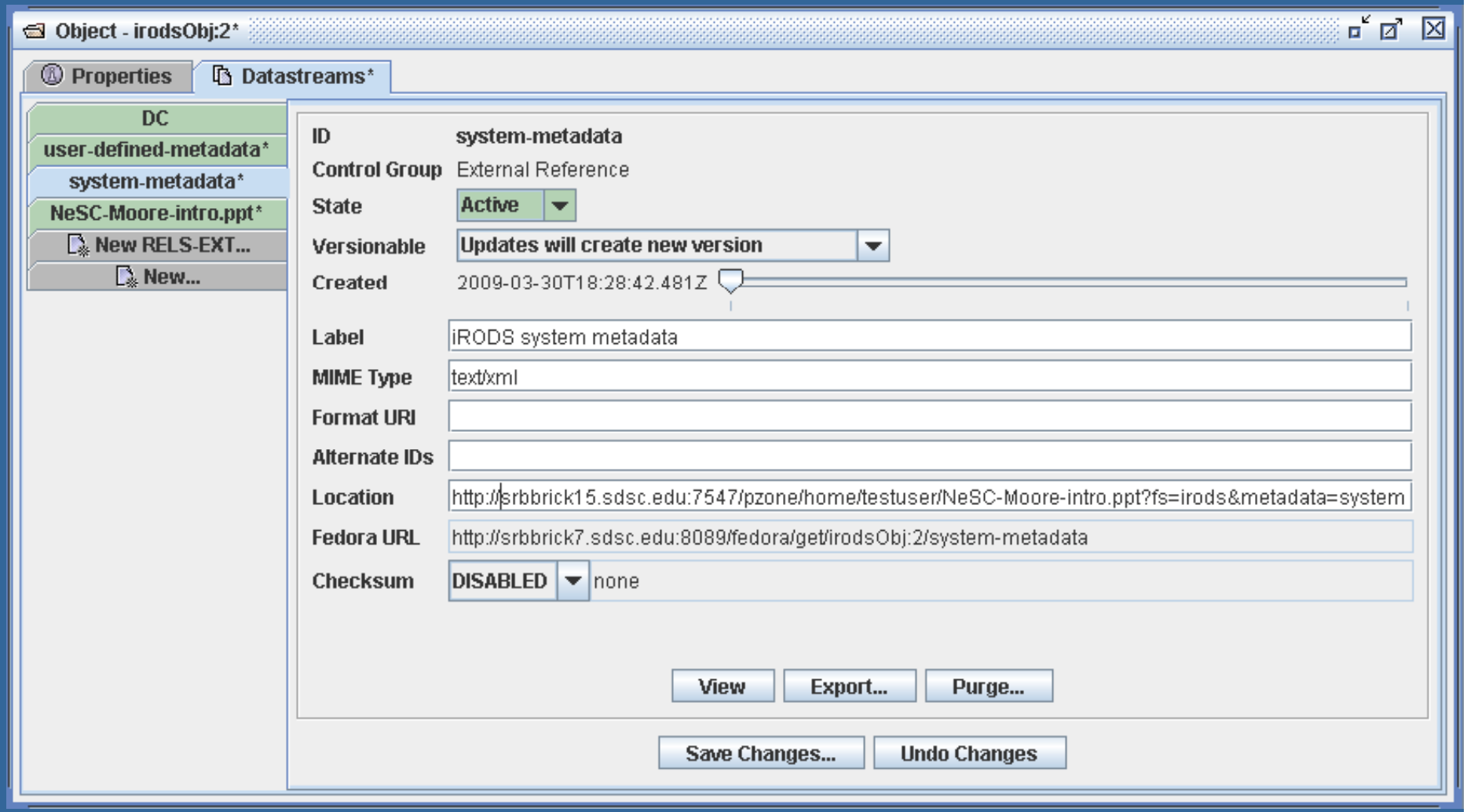
Example: Create a Fedora Object



Example: Create a datastream for the iRODS file

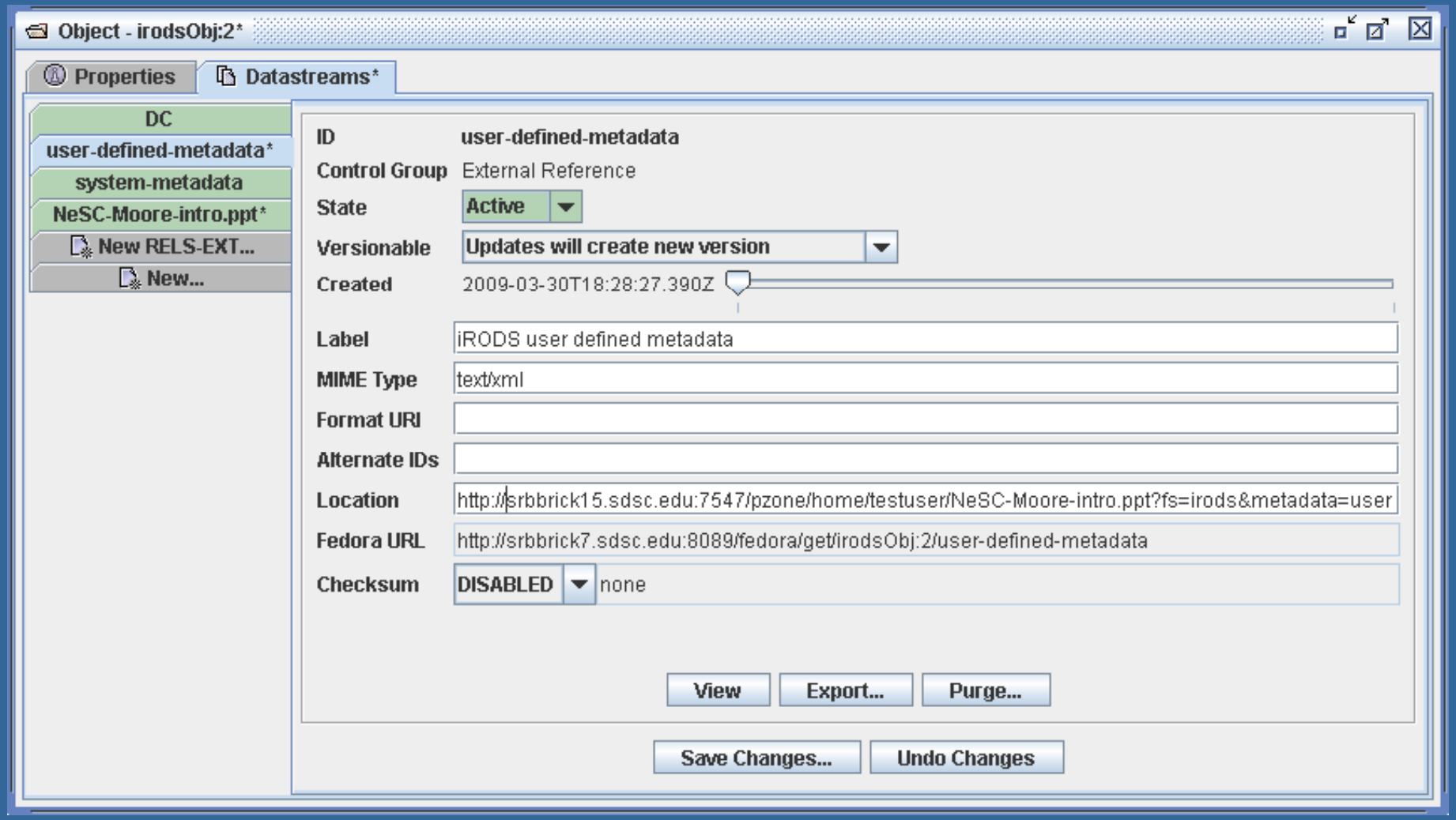


Example: a dynamic reference to iRODS system metadata

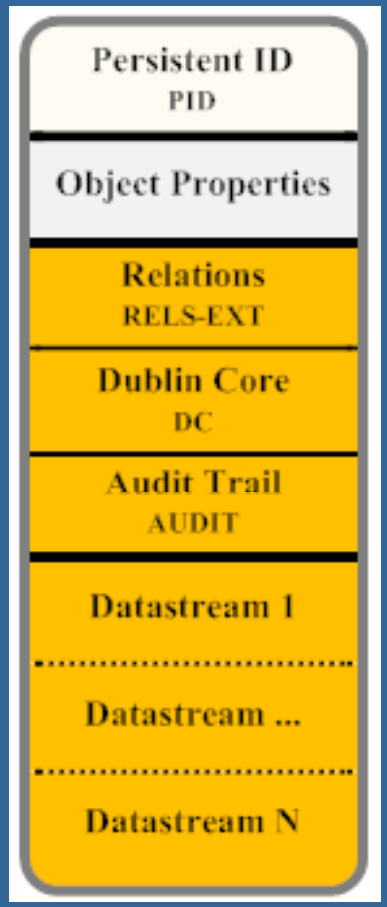


D·I·C·E
i·R·O·D·S
THE UNIVERSITY of NORTH CAROLINA at CHAPEL HILL
UCSD
NSF
THE NATIONAL ARCHIVES

Example: a dynamic reference to iRODS user metadata



Mapping Fedora Objects in iRODS



Tar File Format

User

Access compound tar file

iRODS Server

iRODS Rule Engine

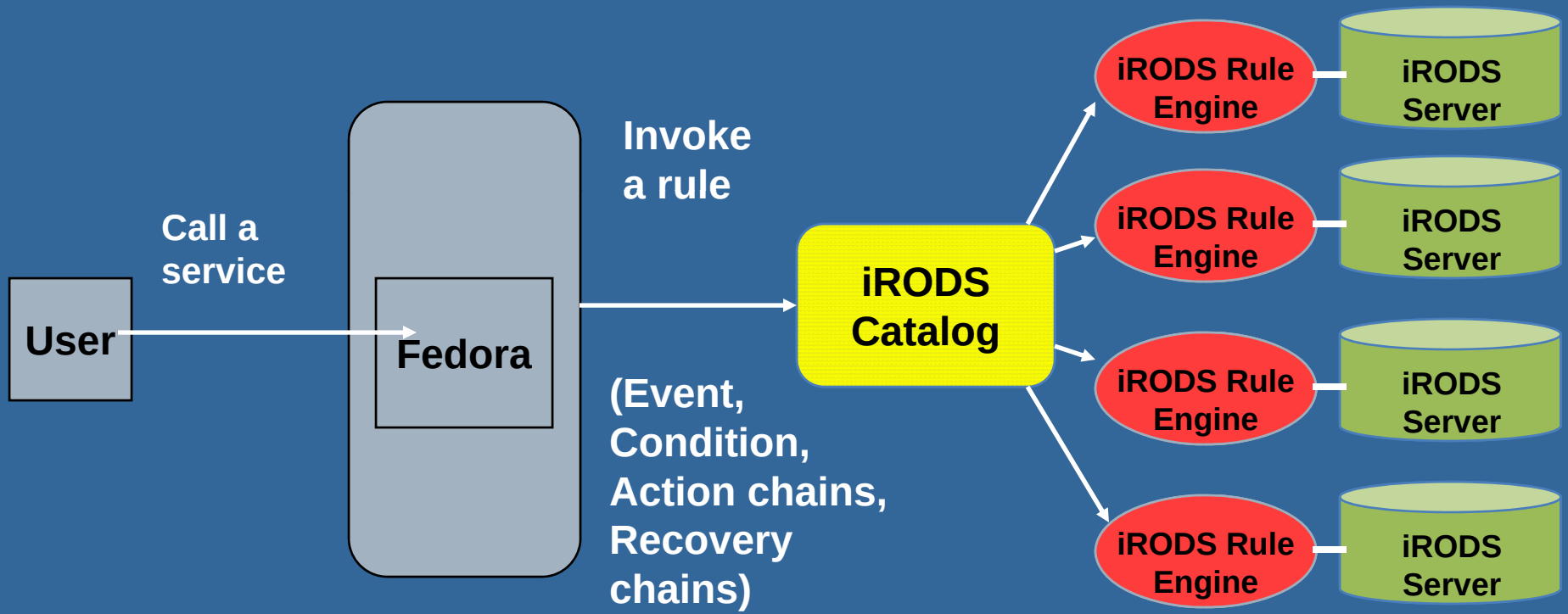
Meta Data Catalog

Manage the Tar file as a container

iRODS Rules

- Implement Policies
- Verify enforcement (audit trails)
- Automate management of exploding data
  - Let you handle petabytes in hundreds of millions of files
- Each Rule defines
  - Event, Condition, Action chains (micro-services, other Rules), Recovery chains
- Rule types
  - Atomic (immediate), Deferred, Periodic
- Rules are executed by iRODS Rule Engine
  - Applied where data is (server-side)

Invoking iRODS service in Fedora



iRODS Policy

- To express community goals for data access and sharing, management, long-term preservation, uses, etc.

- Implemented through iRODS rules
  - Each rule is a chain of micro-services
  - Invoked by the iRODS Rule Engine
  - Currently C functions; PHP, Java coming soon
  - Can wrap Web-services

Sample iRODS Policies

- Run a particular workflow when a “set of files” is ingested into a collection (e.g. make thumbnails of images, post to website).
- Automatically replicate a file added to a collection into 3 geographically distributed sites.
- Automatically extract metadata for a file of a certain type and store in metadata catalog.
- Periodically check integrity of files in a Collection and repair/replace if needed/possible.
- Automatically pick a certain storage location based on user or collection or size or type.
- Let a user access a collection only if using certificate-based login.
- Send a notification when a certain file is ingested.

Policy-level Interoperability

- Can a preservation environment be assembled from two existing repositories with differing management policies?
- Can the policies of the federation be enforced across both repositories, ensuring consistent management of the archives?
- Can policies be migrated between repositories, either by association of the policies with the storage repositories, or through control of repository procedures?
- What fundamental mechanisms are needed within a repository to implement new policies?

Embedding Approaches

- **ra-DM driving OM. Design policy federation models that are implemented at the storage level through the ra-DM.**
- **OM driving ra-DM. Design policy federation models in which the workflows within the OM model enforce the policies, but deposit the objects into the ra-DM.**
- **ra-DM and OM co-driving. Design policy federation models in which policies are enforced by both types of preservation environments.**

OM : Objetc Model

ra-DM: rule-aware distributed model

DICE Center

iRODS:

- **http://www.irods.org**
- **http://www.dice.unc.edu**
- **http://www.diceresearch.org**

Fedora-iRODS Integration:

- **https://www.irods.org/index.php/Fedora**