university of groningen / center for information technology

iRODS Labels: A Proposal for a Complementary Labeling Mechanism in iRODS

Andrey Tsyganov

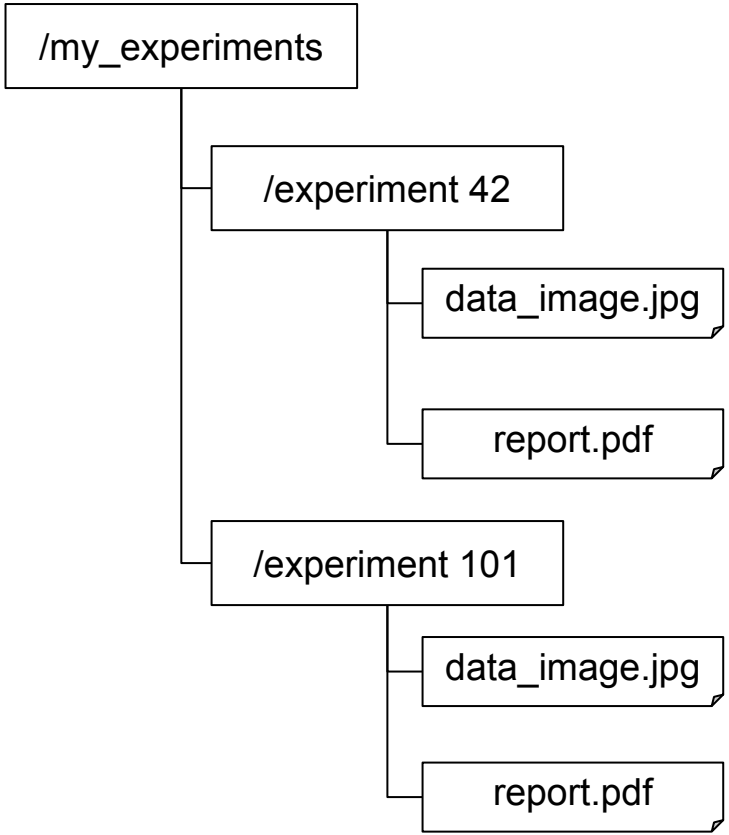
Agenda

- Introduction: iRODS AVUs vs Labels
- iRODS Labels Architecture & Implementation
- ilabel
- python client
- iRODS Labels results and future suggestio
- Example

university of groningen / center for information technology

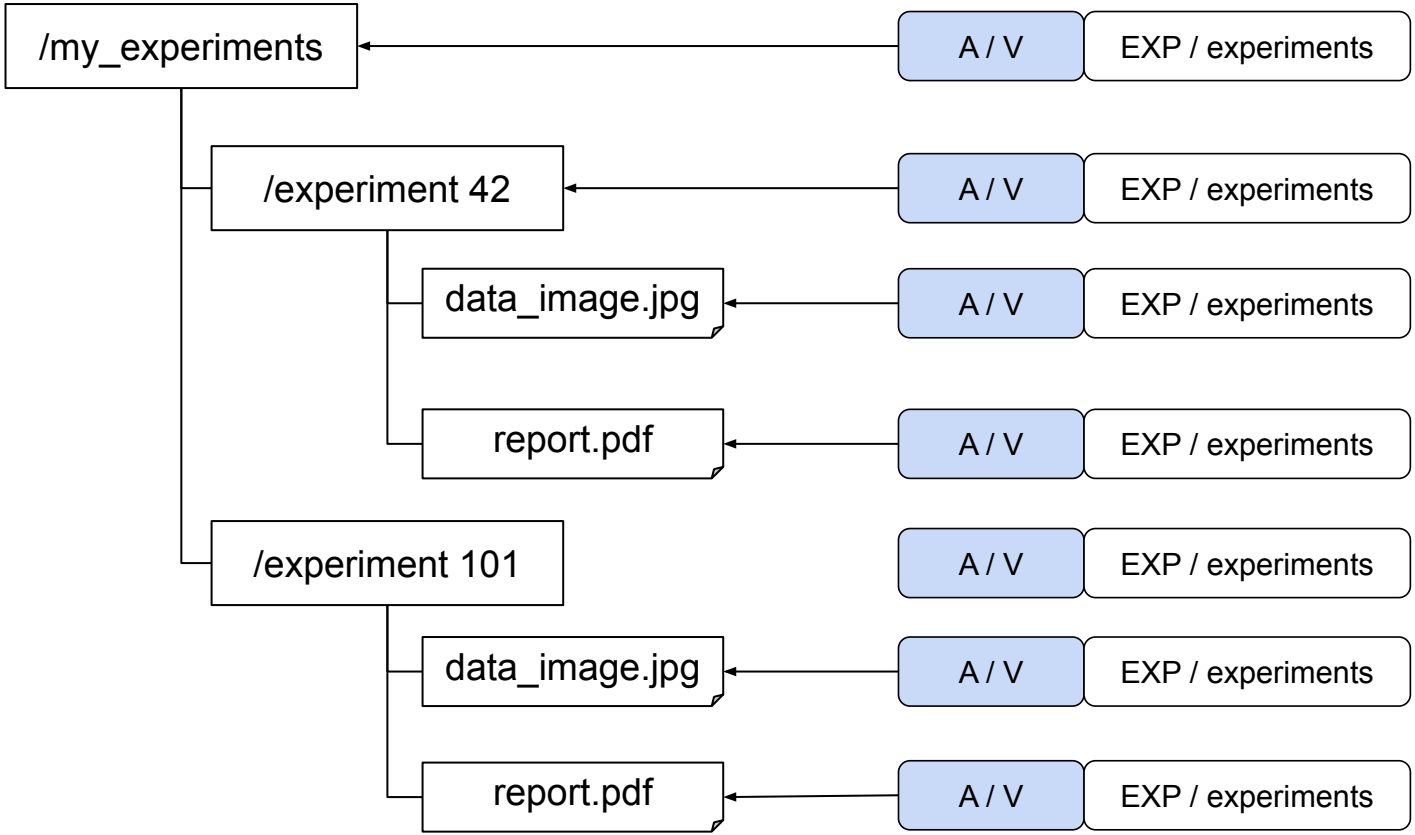
Introduction: iRODS AVUs vs Labels

Introduction: iRODS AVUs



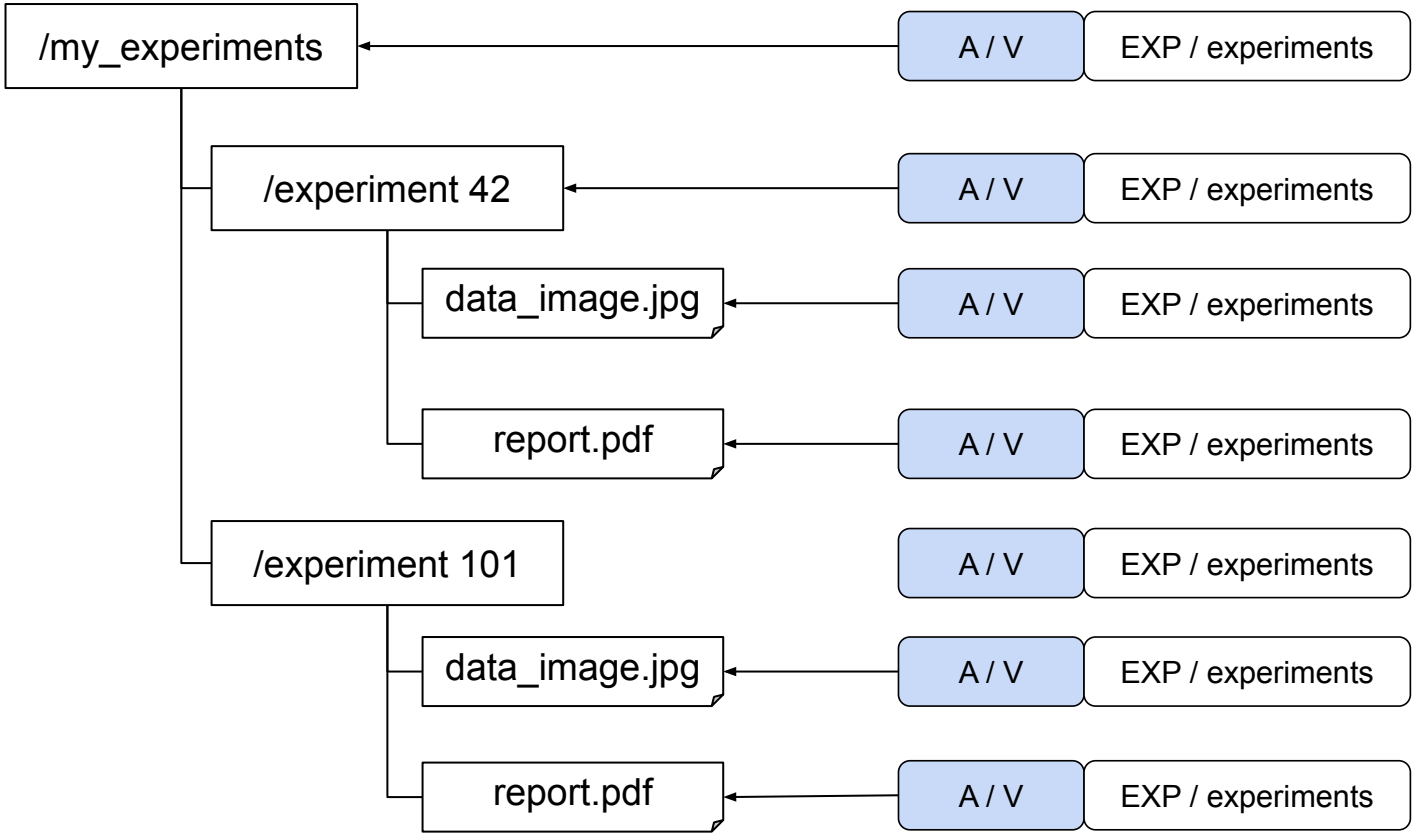
university of groningen / center for information technology

Introduction: iRODS AVUs



university of groningen / center for information technology

Introduction: iRODS AVUs



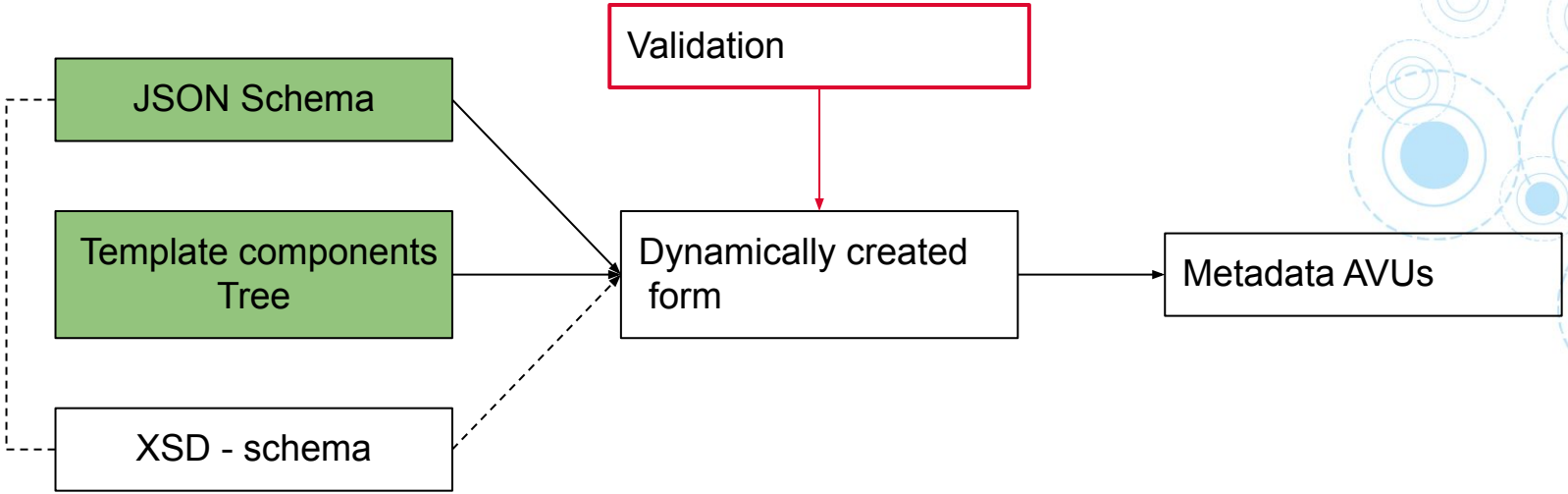
Duplication of the metadata keys / values

university of groningen / center for information technology

Introduction: iRODS AVUs

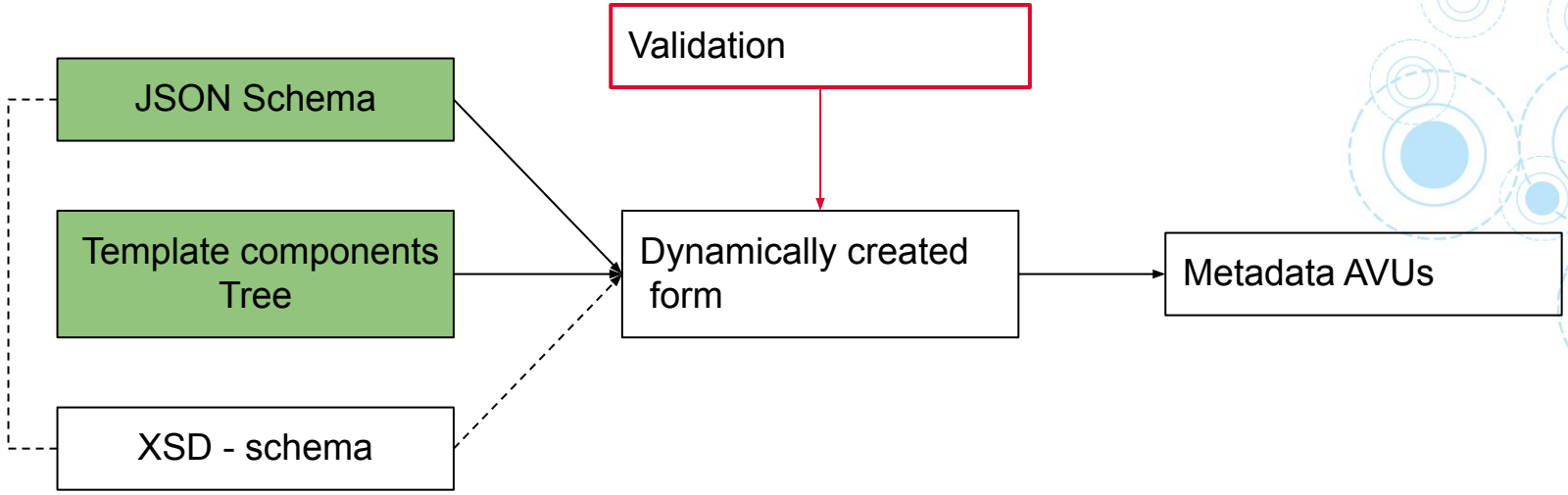
- AVUs (attribute-value-unit) are flat — no hierarchy, no relationships between metadata entries
- Object AVUs relation is 1-to-1
- No schema validation — values are untyped free text
- Searches across many collections /data objects especially with %<word>% patterns, are slow
- Problem of the metadata template for large collections:
  - data duplication when tagging all sub-collections and data objects
  - a single metadata change must be propagated to every object that carries that key

Introduction: RUG RDMS metadata templates



university of groningen / center for information technology

Introduction: RUG RDMS metadata templates

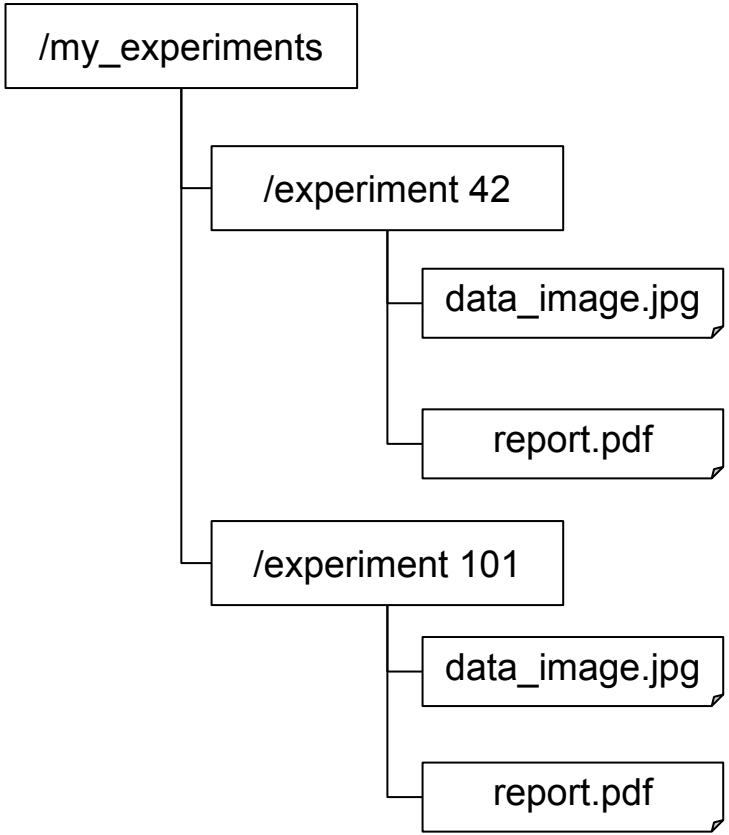


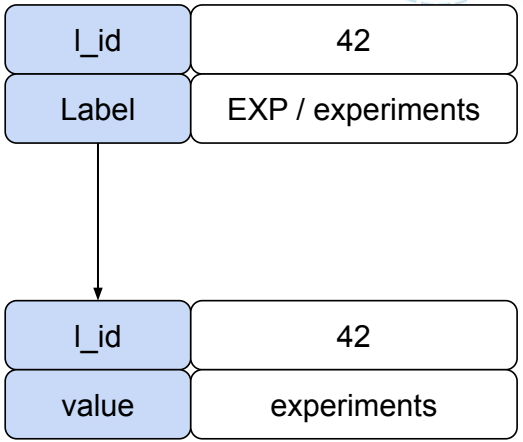
If template AVUs are on collections, then each sub-collection and data object is logically tagged with template metadata.

GUI data search processes this with a special call, but CLI tools and iRODS clients cannot do this automatically

university of groningen / center for information technology

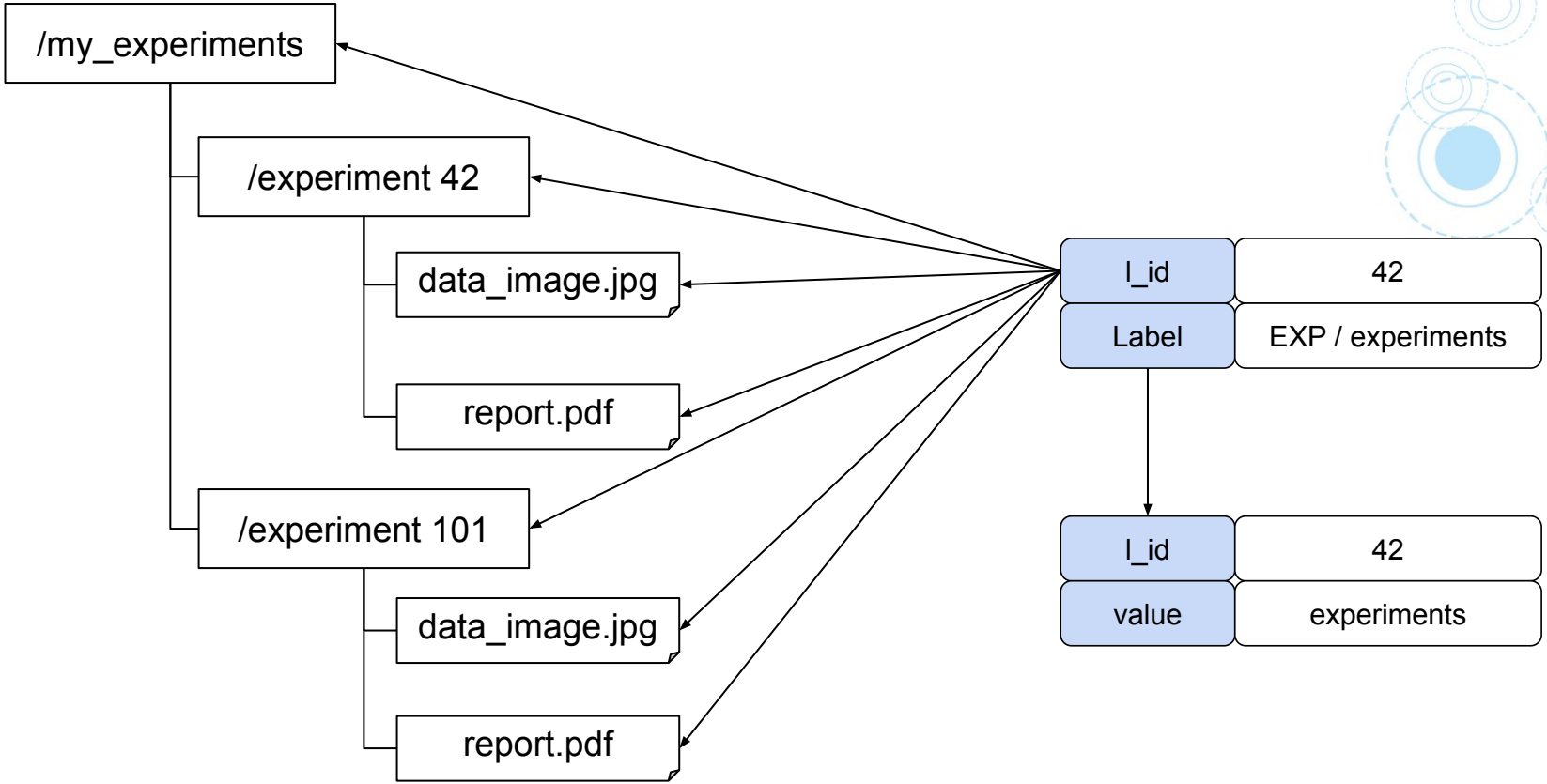
Introduction: iRODS Labels





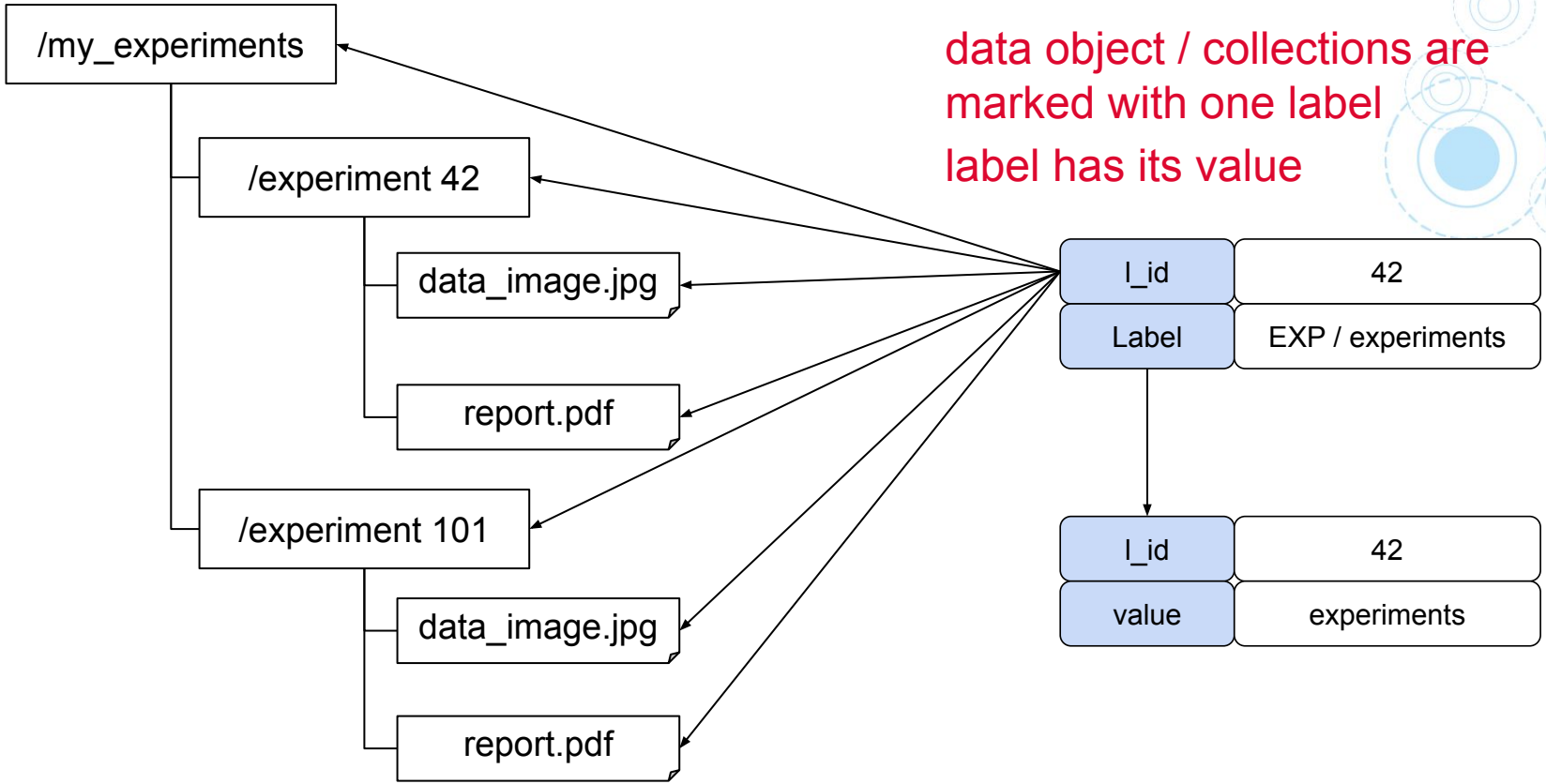
university of groningen / center for information technology

Introduction: iRODS Labels



university of groningen / center for information technology

Introduction: iRODS Labels



university of groningen / center for information technology

iRODS Labels Architecture & Implementation

iRODS Labels Architecture & Implementation

- Labels are a new metadata type that lives alongside AVUs
- A label has a name, optional description, and an optional value for validation
- Labels can build a tree — every label can have a parent label
- One label can be attached to many data objects and/or collections simultaneously
- Labels have their own ACL, separate from object ACLs

iRODS Labels Architecture & Implementation

- Labels are a new metadata type that lives alongside AVUs
- A label has a name, optional description, and an optional value for validation
- Labels can build a tree — every label can have a parent label
- One label can be attached to many data objects and/or collections simultaneously
- Labels have their own ACL, separate from object ACLs

-  **Claude AI**, Model: Sonnet 4.6 - project assistant

university of groningen / center for information technology

iRODS Labels Architecture & Implementation: Database Schema

- New tables in the iRODS catalog (ICAT_DB):
  - r_label — label definitions; name, parent, **visibility**
  - r_label_value — one value payload per label; optional JSON Schema validation
  - **r_label_access** — per-label ACL; user + access level (read/modify/own)
  - r_data_label — attaches labels to data objects
  - r_coll_label — attaches labels to collections; includes inherit flag
- No foreign keys by design — consistency maintained by explicit hooks in **db_plugin.cpp (!!!)**, following the iRODS catalog convention.

university of groningen / center for information technology

iRODS Labels Architecture & Implementation: Database Schema

iRODS default database schema

university of groningen / center for information technology

iRODS Labels Architecture & Implementation: Software

- **Platform**
  - OS Ubuntu 22.04 LTS (jammy)
  - VM **Vagrant** + VirtualBox (ubuntu/jammy64)
  - Database PostgreSQL 14 (system package) · ODBC via odbc-postgresql
- **iRODS**
  - Version **4.3.5** — compiled from source (github.com/irods/irods, tag 4.3.5)
  - Build dir /opt/irods-src → /opt/irods-build
  - Plugin build /opt/irods-labels-build
- **Compiler & Build Tool**
  - Compiler Clang 13.0.1 (from irods-externals, not system clang)
  - C++ standard C++20
  - CMake 3.21.4 (from irods-externals)
  - All compilation uses the irods-externals toolchain to guarantee ABI compatibility with libirods_client.so and libirods_server.so.

university of groningen / center for information technology

iRODS Labels Architecture & Implementation: Software

Dependencies (all from `irods-externals`)

Library	Version	Role
Boost	1.81.0	Filesystem, program_options, thread
fmt	8.1.1	String formatting
nlohmann/json	3.10.4	JSON transport payload
nanodbc	2.13.0	C++ ODBC wrapper for catalog SQL
spdlog	1.9.2	Logging
ZeroMQ	4.1.8	iRODS message bus
cppzmq	4.8.1	ZeroMQ C++ bindings
Avro	1.11.0	iRODS serialisation
Catch2	2.13.8	Unit testing framework
Clang RT	13.0.1	Runtime library (`libc++`)

iRODS Labels Architecture & Implementation: Software

Python Client

- Runtime : Python 3.10 (Ubuntu 22.04 system)
- iRODS binding : python-irodsclient (latest via pip)
- Schema validation : python3-jsonschema
- DB introspection : python3-pyodbc

Docker image/container build that re-uses Vagrant configuration scripts to build the iRODS Labels test container

university of groningen / center for information technology

iRODS Labels Architecture & Implementation

- Implemented as an iRODS server plugin
- Plugin registered as API number 20100
- Transport layer: JSON over bytesBuf_t (same pattern as built-in iRODS touch API)
- All SQL executes server-side via nanodbc — clients make RPC calls only
- The iRODS source is patched to extend GenQuery with label columns and lifecycle hooks

iRODS Labels Architecture & Implementation

university of groningen / center for information technology

iRODS Labels Architecture & Implementation

iRODS core files that need to be modified for the iRODS Labels plugin

university of groningen / center for information technology

iRODS Labels Architecture & Implementation

university of groningen / center for information technology

iRODS Labels Architecture & Implementation

iRODS Labels custom files including ilabel cli tool

university of groningen / center for information technology

iRODS Labels Architecture & Implementation

API Registration

plugins/api/include/irods/plugins/api/api_plugin_number_data.h

Registers API slot #20100 for the label RPC

GenQuery v1 (iquest)

lib/core/include/irods/rodsGenQueryNames.h

Adds LABEL_* column name string constants

plugins/database/src/general_query_setup.cpp

Registers sTable / sColumn / sFklink entries for label tables

plugins/database/src/general_query.cpp

genqAppendAccessCheck() ~L1649: injects visibility ACL WHERE clause for private labels on every label-touching query

server/icat/src/icatGeneralQuerySetup.cpp

Wires label columns into the iCAT query engine setup

university of groningen / center for information technology

iRODS Labels Architecture & Implementation

GenQuery v2 (iquery)

server/genquery2/include/irods/private/genquery2_table_column_mappings.hpp

Maps label column IDs 11000–11032 to table.column

server/genquery2/src/genquery2_sql.cpp

Adds r_label join logic for v2 SQL generation*

Catalog Lifecycle Hooks

plugins/database/src/db_plugin.cpp

Handles major data consistency intervention across the object lifecycle (create / delete / update).

iRODS Labels Architecture & Implementation

plugins/database/src/general_query_setup.cpp

```
/* Labels plugin — table registrations */
sTable( "R_LABEL",         "R_LABEL",         0 );
sTable( "R_LABEL_VALUE",   "R_LABEL_VALUE",   0 );
sTable( "R_DATA_LABEL",    "R_DATA_LABEL",    0 );  sTable( "R_COLL_LABEL",    "R_COLL_LABEL",    0 );
```

```
/* Labels plugin — column mappings (COL_LABEL_* defined in irods/label_ops.h) */
sColumn( COL_LABEL_ID,                "R_LABEL",         "label_id" );
sColumn( COL_LABEL_NAME,              "R_LABEL",         "label_name" );
sColumn( COL_LABEL_DESCRIPTION,       "R_LABEL",         "label_description" );
sColumn( COL_LABEL_CREATOR_ID,        "R_LABEL",         "creator_id" );
sColumn( COL_LABEL_CREATE_TS,         "R_LABEL",         "create_ts" );
sColumn( COL_LABEL_PARENT_ID,         "R_LABEL",         "parent_label_id" );

sColumn( COL_LABEL_VALUE,             "R_LABEL_VALUE",   "value" );
sColumn( COL_LABEL_VALUE_MODIFY_TS,   "R_LABEL_VALUE",   "modify_ts" );

sColumn( COL_DATA_LABEL_DATA_ID,      "R_DATA_LABEL",    "data_id" );
sColumn( COL_DATA_LABEL_LABEL_ID,     "R_DATA_LABEL",    "label_id" );
sColumn( COL_DATA_LABEL_ASSIGN_TS,    "R_DATA_LABEL",    "assign_ts" );

sColumn( COL_COLL_LABEL_COLL_ID,      "R_COLL_LABEL",    "coll_id" );
sColumn( COL_COLL_LABEL_LABEL_ID,     "R_COLL_LABEL",    "label_id" );
sColumn( COL_COLL_LABEL_ASSIGN_TS,    "R_COLL_LABEL",    "assign_ts" );
```

university of groningen / center for information technology

iRODS Labels Architecture & Implementation

irods-src/lib/core/include/irods/rodsGenQueryNames.h

```
{ COL_LABEL_ID,                "LABEL_ID"                },
{ COL_LABEL_NAME,              "LABEL_NAME"              },
{ COL_LABEL_DESCRIPTION,       "LABEL_DESCRIPTION"       },
{ COL_LABEL_CREATOR_ID,        "LABEL_CREATOR_ID"        },
{ COL_LABEL_CREATE_TS,         "LABEL_CREATE_TS"         },
{ COL_LABEL_PARENT_ID,         "LABEL_PARENT_ID"         },
{ COL_LABEL_VALUE,             "LABEL_VALUE"             },
{ COL_LABEL_VALUE_MODIFY_TS,   "LABEL_VALUE_MODIFY_TS"   },
{ COL_DATA_LABEL_DATA_ID,      "DATA_LABEL_DATA_ID"      },
{ COL_DATA_LABEL_LABEL_ID,     "DATA_LABEL_LABEL_ID"     },
{ COL_DATA_LABEL_ASSIGN_TS,    "DATA_LABEL_ASSIGN_TS"    },
{ COL_COLL_LABEL_COLL_ID,      "COLL_LABEL_COLL_ID"      },
{ COL_COLL_LABEL_LABEL_ID,     "COLL_LABEL_LABEL_ID"     },
{ COL_COLL_LABEL_ASSIGN_TS,    "COLL_LABEL_ASSIGN_TS"    },
```

iRODS Labels Architecture & Implementation

irods-src/server/genquery2/src/genquery2_sql.cpp

```
// Labels plugin tables (irods labels plugin)
"R_LABEL",                          // 20
"R_LABEL_VALUE",                    // 21
"R_DATA_LABEL",                     // 22
"R_COLL_LABEL",                     // 23
```

```
// Labels plugin FK edges (irods labels plugin)
{to_index("R_LABEL"),       to_index("R_LABEL_VALUE")},
{to_index("R_LABEL"),       to_index("R_DATA_LABEL")},
{to_index("R_LABEL"),       to_index("R_COLL_LABEL")},
{to_index("R_DATA_LABEL"), to_index("R_DATA_MAIN")},
{to_index("R_COLL_LABEL"), to_index("R_COLL_MAIN")},
```

```
// Labels plugin join conditions (irods labels plugin)
{"{}.label_id = {}.label_id", 20},    // R_LABEL <-> R_LABEL_VALUE
{"{}.label_id = {}.label_id", 21},    // R_LABEL <-> R_DATA_LABEL
{"{}.label_id = {}.label_id", 22},    // R_LABEL <-> R_COLL_LABEL
{"{}.data_id = {}.data_id", 23},      // R_DATA_LABEL <-> R_DATA_MAIN
{"{}.coll_id = {}.coll_id", 24},      // R_COLL_LABEL <-> R_COLL_MAIN
```

university of groningen / center for information technology

iRODS Labels Architecture & Implementation

There is a key architectural question on how to implement consistency when a label is set with inherit=TRUE.

Actions:

- new data objects/collections are added under the folder with the label + inheritance=True. Thus, a label should be added into r_coll_label/r_data_label
- data objects/collections are removed from the collection.

There are two options:

1. use the same logic as currently in iRODS, then db_plugin needs to be patched
2. ON DELETE CASCADE at the database level, with foreign keys

ilabel

ilabel

- **ilabel** is an extra icommand tool similar to the standard iRODS icommands — it is designed to be installed alongside iput, iget, imeta, and other built-in tools. It follows iRODS conventions: reads ~/.irods/irods_environment.json, uses the current iRODS session, and exits with standard return codes.
- Design: zero SQL — every operation is an RPC call to API #20100 on the server. The only client-side logic is resolving iRODS paths to object IDs via GenQuery before dispatching.

ilabel

USER

```
$ ilabel …
```

COMMANDS

LABEL CRUD
add_label
del_label
get
set_value
set_visibility

HIERARCHY
search
children
path
tree

ATTACH/DETACH
add -d / rm -d
add -C / rm -C
add -C -r

LIST
list_data
list_colls
list_labels
search_by_value
list_data_by_value
list_colls_by_value

ACCESS
grant
revoke

ilabel

ilabel.cpp - CLI

label_client

JSON - bytesBuf_t
label_client.cpp

API #20100

procApiRequest

IRODS SERVER

irods_label_api_plugin.so

rs_label_op()

labels_server.cpp

lsv_*() -> nanodbc -> SQL

ICAT DB

PostgreSQL -> r_label*

university of groningen / center for information technology

ilabel: commands

```
# Create
ilabel add_label project/2026 "Q1 experiment data"
ilabel add_label --parent project/2026 project/2026/raw "raw files"
ilabel add_label myLabel --visibility private
```

```
# Inspect
ilabel get       project/2026          # name, description, value, schema, timestamps
ilabel search    project/              # all labels matching prefix
ilabel children  project/2026          # direct children in the hierarchy
ilabel path      project/2026/raw      # full path from root
ilabel tree      project/              # entire subtree
```

```
# Mutate
ilabel set_visibility project/2026 private
ilabel set_value project/2026 '{"status":"active"}' --validation '{"type":"object"}'
ilabel del_label project/2026          # cascade-deletes all attachments and ACL rows
```

university of groningen / center for information technology

ilabel: commands

```
# What objects carry a given label?
ilabel list_data  project/2026          # data objects
ilabel list_colls project/2026          # collections
ilabel list_data_by_value  %active%     # data objects by value fragment
ilabel list_colls_by_value %active%     # collections by value fragment

# What labels are on a given object?
ilabel list_labels -d /zone/home/user/data.csv
ilabel list_labels -C /zone/home/user/experiments
ilabel search_by_value     %active%     # labels whose value matches fragment
```

```
# Access control
ilabel grant  project/2026 alice tempZone read
ilabel grant  project/2026 alice tempZone write
ilabel grant  project/2026 alice tempZone own
ilabel revoke project/2026 alice tempZone
```

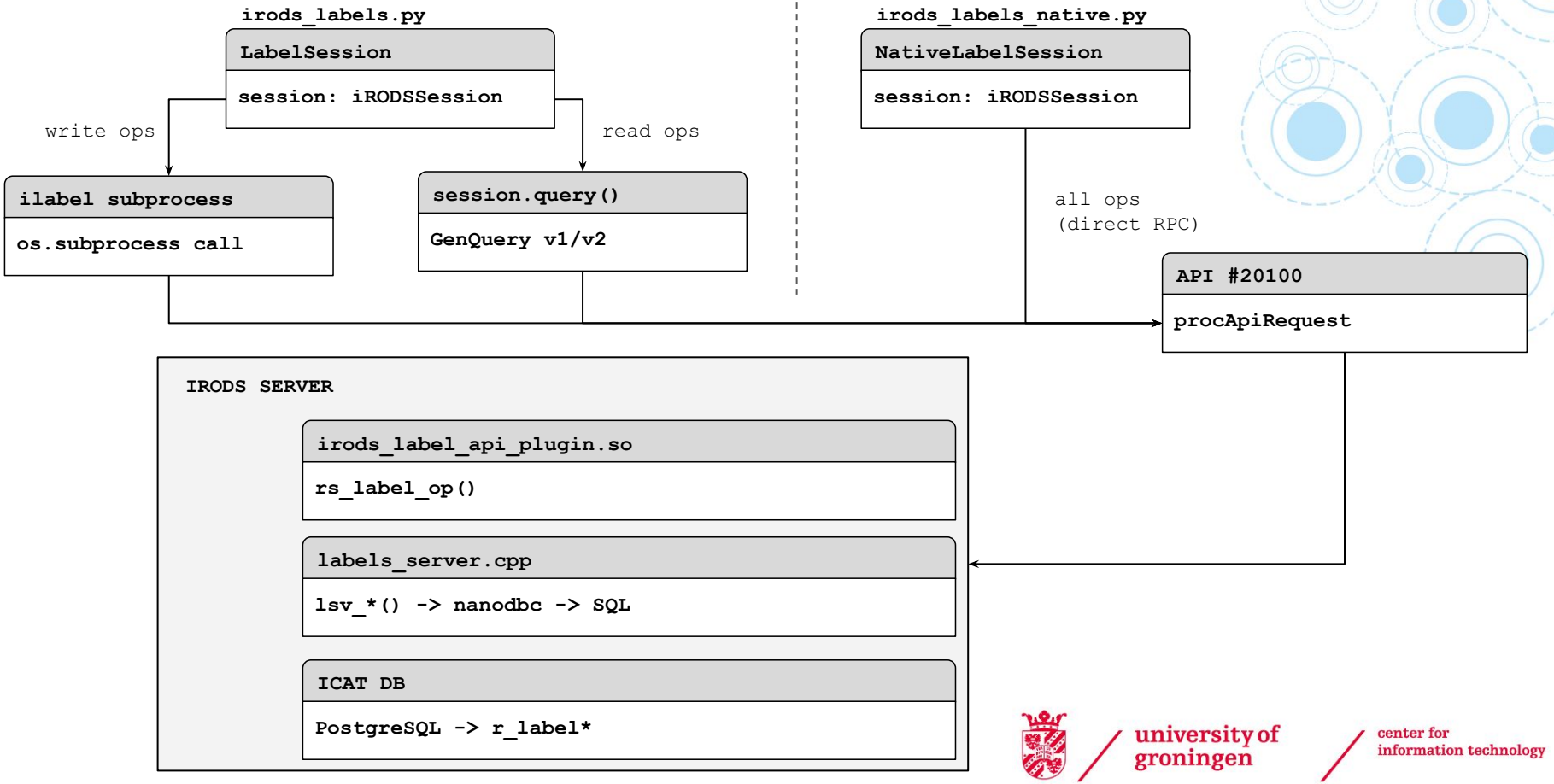
university of groningen / center for information technology

ilabel: commands

Flag	Applies to	Meaning
`--visibility private|public`	`add_label`	Controls name discovery via GenQuery
`--parent <name>`	`add_label`	Places label in the hierarchy
`--validation <json>`	`set_value`	Stores a JSON Schema alongside the value
`-d`	`add`, `rm`, `list_labels`	Target is a data object
`-C`	`add`, `rm`, `list_labels`	Target is a collection
`-r`	`add -C`, `rm -C`	Recursive — sets `inherit=TRUE` on `r_coll_label`

python client

python client



university of groningen / center for information technology

python client

Two implementations, identical public API

LabelSession — irods_labels.py:

- **Write ops → ilabel subprocess → API #20100 → labels_server.cpp SQL**
- **Read ops → session.query() → GenQuery v1/v2 → ICAT_DB**

NativeLabelSession — irods_labels_native.py

- **All ops → API #20100 directly → labels_server.cpp SQL**
- **No subprocess, no GenQuery, no ilabel binary required on the client**
- **Path → ID resolution uses one lightweight session.query() call before attach/detach ops**

```
from irods_labels import LabelSession
with LabelSession(session) as ls:
/
from irods_labels_native import NativeLabelSession
with NativeLabelSession(session) as ls:

    ls.add_label(
        "project/2026"
       ,"Experiment data for Q1")
    ls.attach to data(
        "project/2026",
        "/zone/home/user/data.csv")
    ls.grant(
        "project/2026",
        "alice",
        "tempZone",
        "read")
    results = ls.list_data_by_label("project/2026")
```

university of groningen / center for information technology

iRODS Labels results and future suggestion

Result

- A new plugin that compiles successfully within iRODS core
- Labels provide additional functionality for tagging and data management within iRODS
- iRODS Labels is not a replacement but rather an addition to the existing AVUs mechanism that has its own great value and purpose

Future

- Integration into the iRODS core to avoid custom compilation?
- Adding MySQL and Oracle databases support
- db_plugin verification for locks or FK integration instead
- Cross–tree references - one label with multiple parent labels
- Label value versioning: very useful to preserve the history of template changes

university of groningen / center for information technology

Example

Example

```
vagrant@irods-labels-dev:~$ bash /vagrant/tests/demo_publication_label.sh

── Step 1: Create dataset: paper + presentation + src/ with plugin ──
  $ imkdir /tempZone/home/rods/irods-labels-ugm2026
  $ imkdir /tempZone/home/rods/irods-labels-ugm2026/src
  $ iput paper.pdf /tempZone/home/rods/irods-labels-ugm2026/
  $ iput presentation.pptx /tempZone/home/rods/irods-labels-ugm2026/
  $ iput plugin.tar.gz /tempZone/home/rods/irods-labels-ugm2026/src/
  ✓ Created: /tempZone/home/rods/irods-labels-ugm2026/{paper.pdf, presentation.pptx, src/plugin.tar.gz}

── Step 2: Create publication label ──
  $ ilabel add_label publication/irods-labels-ugm2026 'Publication metadata — DataCite minimal schema'
  ✓ Label 'publication/irods-labels-ugm2026' created
```

university of groningen / center for information technology

Example

```
── Step 3: Set JSON value (DataCite metadata) + JSON Schema validation ──
 $ ilabel set_value publication/irods-labels-ugm2026 '<value>' --validation '<schema>'
 Value:
   {
     "title":           "iRODS Labels: A Proposal for a Complementary Labeling Mechanism in iRODS",
     "creator":         "A. Tsyganov",
     "resourceType":    "Software",
     "publicationYear": "2026",
     "rights":          "CC-BY-4.0",
     "subject":         "iRODS, metadata, labels, data management, iRODS UGM 2026"
   }
 Validation schema:
   {
     "type": "object",
     "required": ["title", "creator", "resourceType", "publicationYear", "rights"],
     "properties": {
       "title":           { "type": "string" },
       "creator":         { "type": "string" },
       "resourceType":    { "type": "string", "enum": ["Dataset", "Software", "Collection"] },
       "publicationYear": { "type": "string", "pattern": "^[0-9]{4}$" },
       "rights":          { "type": "string", "enum": ["CC-BY-4.0", "CC0-1.0", "CC-BY-SA-4.0"] },
       "subject":         { "type": "string" }
     },
     "additionalProperties": false
   }
 ✓ Metadata and validation schema stored on label 'publication/irods-labels-ugm2026'
```

university of groningen / center for information technology

Example

```
── Step 4: Attach label to dataset collection recursively (inherit=TRUE) ──
  $ ilabel add -C -r publication/irods-labels-ugm2026 /tempZone/home/rods/irods-labels-ugm2026
  ✓ Label attached to /tempZone/home/rods/irods-labels-ugm2026 and all children with inherit=TRUE

── Step 5: List all objects tagged with this label ──
  $ ilabel list_data publication/irods-labels-ugm2026
  Data objects:
    /tempZone/home/rods/irods-labels-ugm2026/paper.pdf
    /tempZone/home/rods/irods-labels-ugm2026/presentation.pptx
    /tempZone/home/rods/irods-labels-ugm2026/src/plugin.tar.gz
  $ ilabel list_colls publication/irods-labels-ugm2026
  Collections:
    /tempZone/home/rods/irods-labels-ugm2026
    /tempZone/home/rods/irods-labels-ugm2026/src
```

Example

```
── Step 6: Inspect label - title, rights, validation schema ──
  $ ilabel get publication/irods-labels-ugm2026
name: publication/irods-labels-ugm2026
description: Publication metadata - DataCite minimal schema
creator_id: 10002
parent_id: 0
value: {
  "title":           "iRODS Labels: A Proposal for a Complementary Labeling Mechanism in iRODS",
  "creator":         "A. Tsyganov",
  "resourceType":    "Software",
  "publicationYear": "2026",
  "rights":          "CC-BY-4.0",
  "subject":         "iRODS, metadata, labels, data management, iRODS UGM 2026"
}
value_validation: {
  "type": "object",
  "required": ["title", "creator", "resourceType", "publicationYear", "rights"],
  "properties": {
    "title":           { "type": "string" },
    "creator":         { "type": "string" },
    "resourceType":    { "type": "string", "enum": ["Dataset", "Software", "Collection"] },
    "publicationYear": { "type": "string", "pattern": "^[0-9]{4}$" },
    "rights":          { "type": "string", "enum": ["CC-BY-4.0", "CC0-1.0", "CC-BY-SA-4.0"] },
    "subject":         { "type": "string" }
  },
  "additionalProperties": false
}
create_ts: 0
modify_ts: 1782768132
```

Example

```
── Step 7: Search by value fragment: find all objects with rights=CC-BY-4.0 ──
  --- ilabel (CLI) ---
  $ ilabel list_data_by_value %CC-BY-4.0%
  Data objects:
    /tempZone/home/rods/irods-labels-ugm2026/paper.pdf
    /tempZone/home/rods/irods-labels-ugm2026/presentation.pptx
    /tempZone/home/rods/irods-labels-ugm2026/src/plugin.tar.gz
  $ ilabel list_colls_by_value %CC-BY-4.0%
  Collections:
    /tempZone/home/rods/irods-labels-ugm2026
    /tempZone/home/rods/irods-labels-ugm2026/src
  --- Python client (NativeLabelSession) ---
  $ python3 /vagrant/tests/search_by_label_value.py %CC-BY-4.0%
    Data objects (value matches '%CC-BY-4.0%'):
      /tempZone/home/rods/irods-labels-ugm2026/paper.pdf
      /tempZone/home/rods/irods-labels-ugm2026/presentation.pptx
      /tempZone/home/rods/irods-labels-ugm2026/src/plugin.tar.gz

    Collections (value matches '%CC-BY-4.0%'):
      /tempZone/home/rods/irods-labels-ugm2026
      /tempZone/home/rods/irods-labels-ugm2026/src

    Total: 3 data object(s), 2 collection(s)
```

Example

```
── Step 8: Update license: CC-BY-4.0 → CC0-1.0  (ONE command – all objects) ──
  $ ilabel set_value publication/irods-labels-ugm2026 '<updated: rights=CC0-1.0>'
  ✓ License updated – no per-file changes needed

── Step 9: Search again: all objects now found under CC0-1.0 ──
  --- ilabel (CLI) ---
  $ ilabel list_data_by_value %CC0-1.0%
  Data objects:
    /tempZone/home/rods/irods-labels-ugm2026/paper.pdf
    /tempZone/home/rods/irods-labels-ugm2026/presentation.pptx
    /tempZone/home/rods/irods-labels-ugm2026/src/plugin.tar.gz
  $ ilabel list_colls_by_value %CC0-1.0%
  Collections:
    /tempZone/home/rods/irods-labels-ugm2026
    /tempZone/home/rods/irods-labels-ugm2026/src
  --- Python client (NativeLabelSession) ---
  $ python3 /vagrant/tests/search_by_label_value.py %CC0-1.0%
    Data objects (value matches '%CC0-1.0%'):
      /tempZone/home/rods/irods-labels-ugm2026/paper.pdf
      /tempZone/home/rods/irods-labels-ugm2026/presentation.pptx
      /tempZone/home/rods/irods-labels-ugm2026/src/plugin.tar.gz

    Collections (value matches '%CC0-1.0%'):
      /tempZone/home/rods/irods-labels-ugm2026
      /tempZone/home/rods/irods-labels-ugm2026/src

    Total: 3 data object(s), 2 collection(s)
  ✓ Same objects found under CC0-1.0 – updated with a single 'ilabel set_value' command
```

Example

```
── Step 10: Add new file + new sub-collection — both inherit the label automatically ──
  $ iput tests.tar.gz /tempZone/home/rods/irods-labels-ugm2026/src/
  $ imkdir /tempZone/home/rods/irods-labels-ugm2026/docs
  $ iput changelog.txt /tempZone/home/rods/irods-labels-ugm2026/docs/
  $ ilabel list_labels -d /tempZone/home/rods/irods-labels-ugm2026/src/tests.tar.gz
  Labels on tests.tar.gz:
    publication/irods-labels-ugm2026  — Publication metadata — DataCite minimal schema
  $ ilabel list_labels -C /tempZone/home/rods/irods-labels-ugm2026/docs
  Labels on docs/:
    publication/irods-labels-ugm2026  — Publication metadata — DataCite minimal schema
  ✓ New file and collection inherited 'publication/irods-labels-ugm2026' automatically — no manual tagging needed

── Step 11: List by value again — new file and collection appear in results ──
  $ ilabel list_data_by_value %CC0-1.0%
  Data objects (now includes tests.tar.gz and changelog.txt):
    /tempZone/home/rods/irods-labels-ugm2026/src/tests.tar.gz
    /tempZone/home/rods/irods-labels-ugm2026/src/plugin.tar.gz
    /tempZone/home/rods/irods-labels-ugm2026/paper.pdf
    /tempZone/home/rods/irods-labels-ugm2026/presentation.pptx
    /tempZone/home/rods/irods-labels-ugm2026/docs/changelog.txt
  $ ilabel list_colls_by_value %CC0-1.0%
  Collections (now includes docs/):
    /tempZone/home/rods/irods-labels-ugm2026/src
    /tempZone/home/rods/irods-labels-ugm2026
    /tempZone/home/rods/irods-labels-ugm2026/docs
  ✓ All objects — including newly added ones — found via value search
```

Q & A?