iRODS®

iRODS Metadata Templates Working Group: Building Blocks and Lessons Learned

Terrell Russell, Ph.D
Executive Director
iRODS Consortium

May 28-31, 2024
iRODS User Group Meeting 2024
Amsterdam, Netherlands

Metadata Templates Working Group

Founded mid-2018

Motivation

iRODS needs to help curators define and validate 'good' metadata for their pipelines and environments.

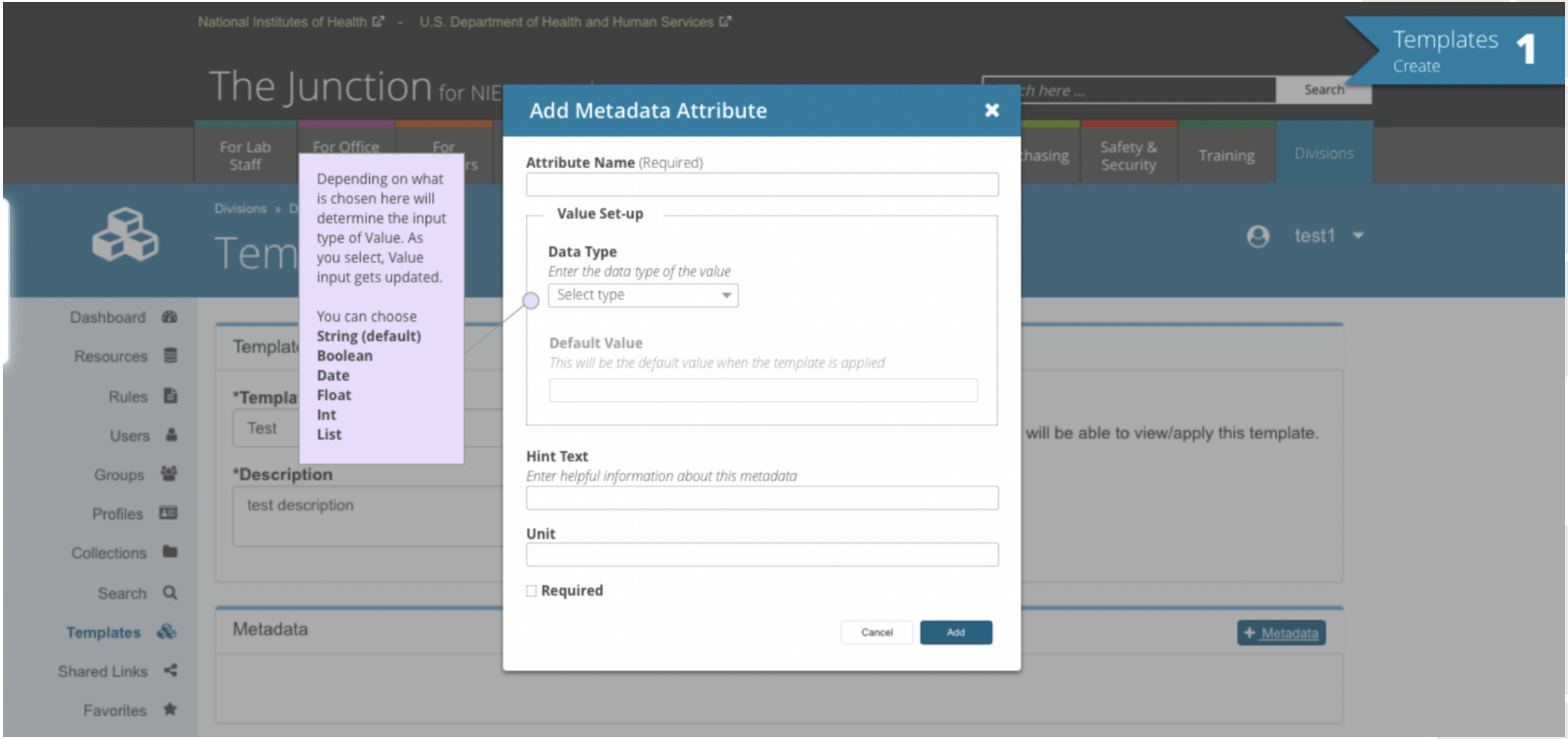
Pre-History

Applications - Boiling the Ocean

2014-2016

- Metalnx
- CloudBrowser
- Yoda
- DataHub
- Dataverse
- CyVerse

Pre-History - Metalnx



Add Metadata Attribute ✖

Attribute Name (Required)

Value Set-up

Data Type
Enter the data type of the value
Select type ▾

Default Value
This will be the default value when the template is applied

Hint Text
Enter helpful information about this metadata

Unit

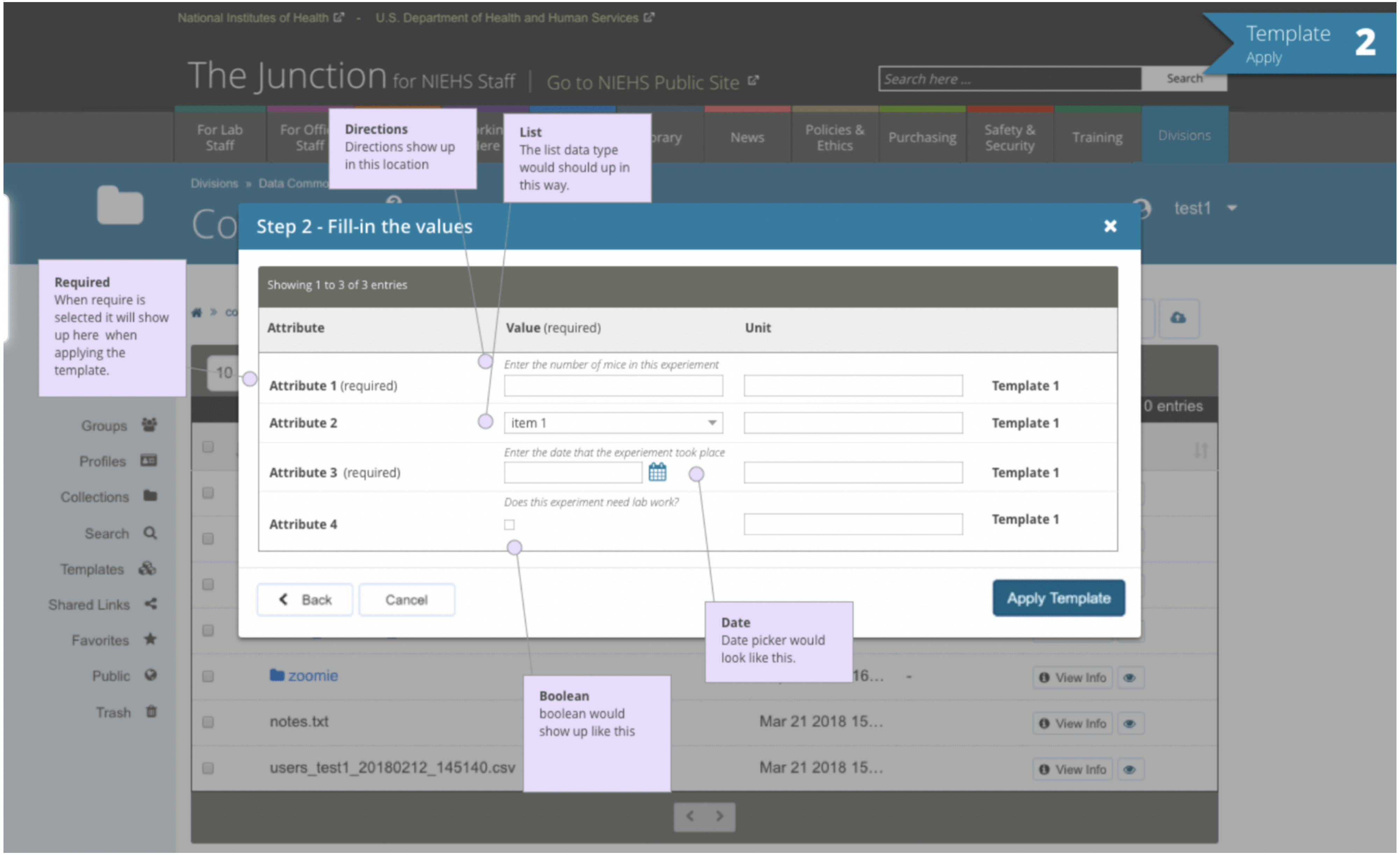
□ **Required**

Cancel | Add

Depending on what is chosen here will determine the input type of Value. As you select, Value input gets updated.

You can choose
String (default)
Boolean
Date
Float
Int
List

Pre-History - Metalnx



Step 2 - Fill-in the values

Showing 1 to 3 of 3 entries

Attribute	Value (required)	Unit	
Attribute 1 (required)	*Enter the number of mice in this experiement*		**Template 1**
Attribute 2	item 1		**Template 1**
Attribute 3 (required)	*Enter the date that the experiement took place*		**Template 1**
Attribute 4	*Does this experiment need lab work?* □		**Template 1**

Required
When require is selected it will show up here when applying the template.

Directions
Directions show up in this location

List
The list data type would should up in this way.

Date
Date picker would look like this.

Boolean
boolean would show up like this

Pre-History - CloudBrowser

- JSON Schema to define template
- template themselves defined schema for metadata
- stored in .irods collection
- parser, validator, resolver, exporter
- handled combining/merging templates into java object
- Mike Conway, Cesar Garde, Terrell Russell

Pre-History - CloudBrowser and Jargon

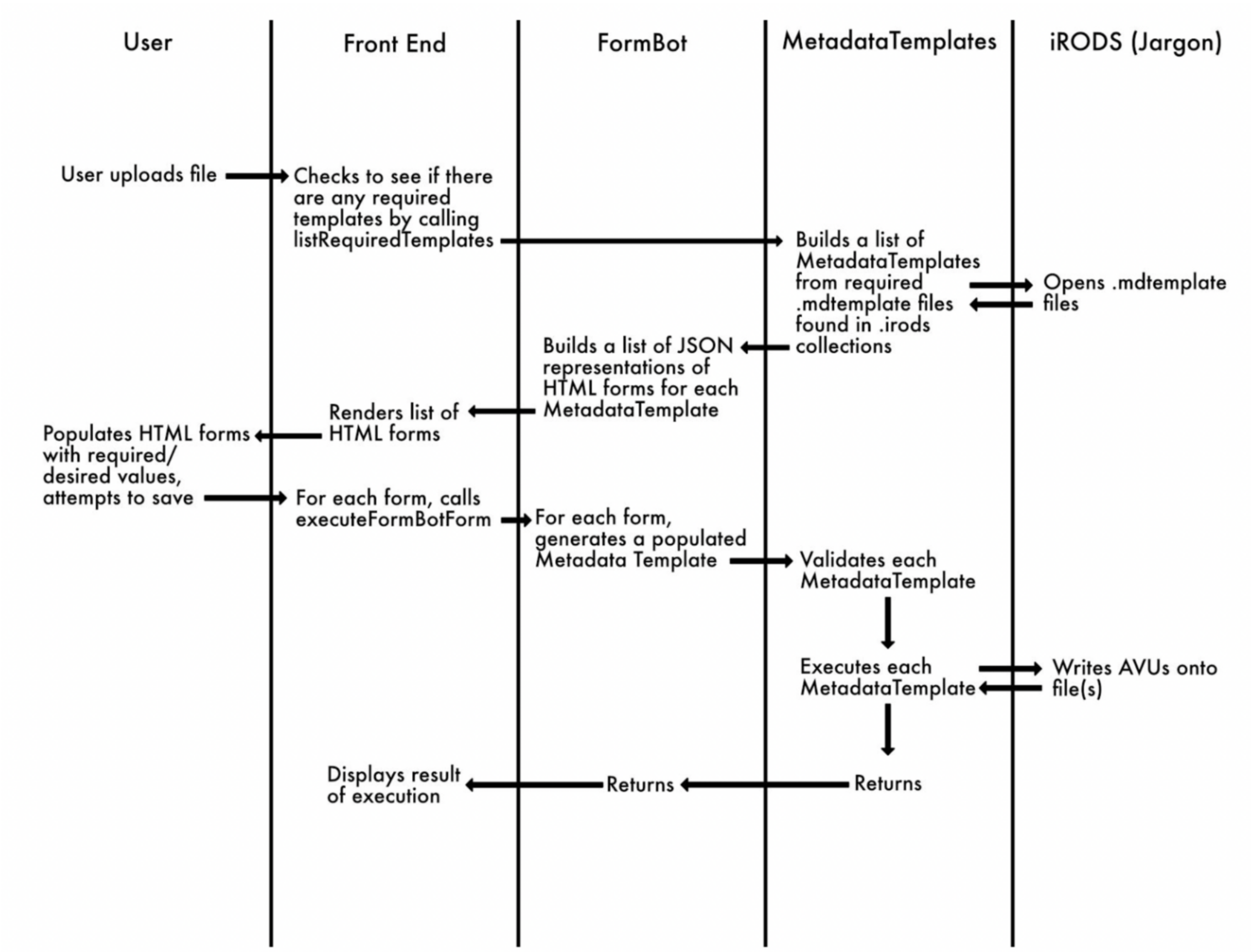
Defining some of these endpoints in the web client and the Java client library led to discussion about a Swagger API (later known as OpenAPI).

This also restarted a conversation about a REST API for all of iRODS itself, but now to include some metadata template endpoints.

Pre-History - May 2017

TRiRODS

Rick Skarbez



User

Front End

FormBot

MetadataTemplates

iRODS (Jargon)

User uploads file

Checks to see if there are any required templates by calling listRequiredTemplates

Builds a list of MetadataTemplates from required .mdtemplate files found in .irods collections

Opens .mdtemplate files

Builds a list of JSON representations of HTML forms for each MetadataTemplate

Renders list of HTML forms

Populates HTML forms with required/ desired values, attempts to save

For each form, calls executeFormBotForm

For each form, generates a populated Metadata Template

Validates each MetadataTemplate

Executes each MetadataTemplate

Writes AVUs onto file(s)

Returns

Returns

Displays result of execution

Metadata Templates Working Group - Formed

June 2018

and by March 2019...

Metalnx metadata templates stored in the Metalnx database as jsonschema

Metadata Templates Working Group - June 2019

Maastricht and Utrecht demonstrated iRODS rules to provide a round trip from JSON to AVU to JSON

- https://github.com/MaastrichtUniversity/irods_avu_json
- https://github.com/MaastrichtUniversity/irods_avu_json-ruleset
- Paul van Schayck, Ton Smeele, Daniel Theunissen and Lazlo Westerhof
- included type information, nesting, used unit for nesting
- handled metadata on data objects and collections

Non-Consortium implementations - some convergence appearing...

- Utrecht / Yoda
- Maastricht / DataHub
- NIEHS / Data Commons
- Arizona / CyVerse

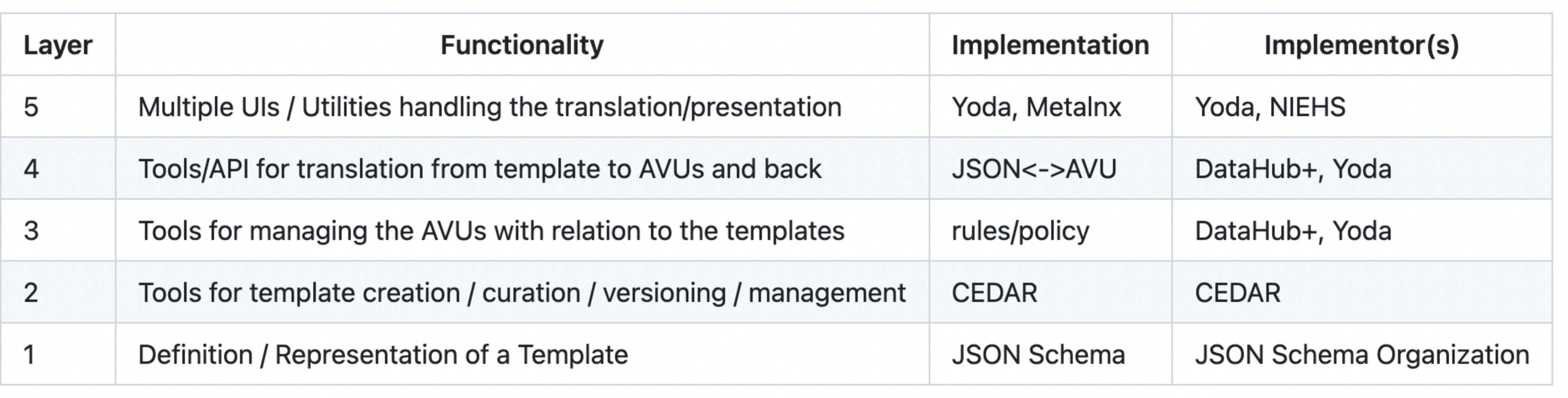
CEDAR coming online as interface / home for editor

Metadata Templates Working Group - June 2019

Identified Five Elements

- Definition / Representation of the Schema (CEDAR itself?, NIEHS)
- Tools for template/schema creation / curation / versions / management (CEDAR itself?, NIEHS)
- Tools for managing the data with relation to the templates (DataHub+, Yoda)
- Translation from schema to AVUs and back (DataHub+, Yoda)
- Multiple UIs / utilities handling the translation/presentation (Yoda, NIEHS)
- (saved) Search queries and results, virtual collections

Five Layers, reordered



Layer	Functionality	Implementation	Implementor(s)
5	Multiple UIs / Utilities handling the translation/presentation	Yoda, Metalnx	Yoda, NIEHS
4	Tools/API for translation from template to AVUs and back	JSON<->AVU	DataHub+, Yoda
3	Tools for managing the AVUs with relation to the templates	rules/policy	DataHub+, Yoda
2	Tools for template creation / curation / versioning / management	CEDAR	CEDAR
1	Definition / Representation of a Template	JSON Schema	JSON Schema Organization

Metadata Templates Working Group - Late 2019

September 2019

- swimlanes, more separation of layers
- first use of external schema applied to iRODS AVUs, Yoda
- identification that atomic application of AVUs is more important than batch/multiple

October 2019

- Operations in a Swagger API
  - Resolve MTs based on an object/collection
  - List attached MTs on an object/collection
  - Attach/Apply MT to an object/collection as required/optional
  - Remove MT from an object/collection
  - List overall available MT in the pool
  - Resolve JSON schema(s) that defines the metadata to be applied via template X to collection Y
  - POSSIBLE - Rasterize? Set of nested/attached schemas down into a single schema

Metadata Templates Working Group - Early 2020

February 2020

- discussion of creation of 3-4 CEDAR JSON schemas for testing
- discussion of using CEDAR as editor, then export to local defined schema
- discussion of using API PEPs rather than database PEPs

April 2020 - Atomic AVUs merged into iRODS

July 2020

- CEDAR as editor, but not publisher/host, needs to be elsewhere
- investigation of schemas/json to xml/html/forms (jsonforms)

Metadata Templates Working Group - Late 2020

August 2020

- Yoda has an atomic endpoint
- discussion about aggregating templates recursively
- ELEMENTS OF ARCHITECTURE
    - CREATION/DEFINITION of templates (punt to CEDAR / others)
    - HOSTING of templates (perhaps CEDAR, perhaps irods.org or github)
    - BINDING/MANAGEMENT of templates to collections/data (part of MTWG MVP)
    - USE of templates in GUI (part of MTWG MVP)
- relevant API components
    - CLIENT/BROWSER: some javascript code to execute client side, wraps an Ajax POST call to the web server
    - WEB SERVER: passes the rule call onto iRODS
    - iRODS PYTHON RULE ENGINE: processes the api call

November 2020 - CEDAR moving to JSON-LD

Metadata Templates Working Group - Late 2020

August 2020

- Yoda has an atomic endpoint
- discussion about aggregating templates recursively
- ELEMENTS OF ARCHITECTURE
  - CREATION/DEFINITION of templates (punt to CEDAR / others)
  - HOSTING of templates (perhaps CEDAR, perhaps irods.org or github)
  - BINDING/MANAGEMENT of templates to collections/data (part of MTWG MVP)
  - USE of templates in GUI (part of MTWG MVP)
- relevant API components
  - CLIENT/BROWSER: some javascript code to execute client side, wraps an Ajax POST call to the web server
  - WEB SERVER: passes the rule call onto iRODS
  - iRODS PYTHON RULE ENGINE: processes the api call

November 2020 - CEDAR moving to JSON-LD

Metadata Templates Working Group - 2021

January 2021

- gofair using jinja templates, mostly rendering/layout
- assessment - maybe there is no 'one ring'
  - different applications will choose to handle rendering themselves
  - stick to the API
  - GUI asks for templates, renders it, sends filled information

February 2021

- decision to be schema/application agnostic

July 2021

- subject areas should drive this work
- iRODS should not define or manage templates for anyone
- iRODS should validate

Metadata Templates Working Group - 2022

February 2022

- eResearchNZ validates this work
- curators want to know/define what is required
- and then enforcement / flagging for humans to come help

March 2022

- KU Leuven building a portal (became ManGO)
- templates / editor / required/optional, collection and data objects

June 2022 - We should have a working group whitepaper

August 2022

- Community is 'ahead' of consortium
- iRODS server should provide building blocks
- Python now -> C++ later once agreed/good

October 2022

- MIAME (Minimum Information About a Microarray Experiment)
- machine actionable data management plans
- validating - iRODS should be a consumer of these efforts

Metadata Templates Working Group - Early 2023

March 2023

- RDA20 - Sweden - they are struggling with getting consensus
  - every discipline has own language/details
  - consistency is really hard / impossible
- KU Leuven
  - wrote editor schema in javascript
  - working on versioning, new template affects old data
  - templates are namespaced, so no collisions
  - based on project-level management of associated templates

June 2023

- KU Leuven - namespacing!

Metadata Templates Working Group - July 2023

- MT get their own database table in iRODS?
- KU Leuven - using template to render the form, not validate the metadata itself
    - two types of template? form and metadata
- IT4I - forcing users to only use a single schema
    - exporting to elastic for search / multiple zones
- Microservices
    - Attach (type, schema, object_id)
        - Initially, type will just be 'url'
        - Could later be 'irods_schema' and store the id from the new table
        - Or 'form' for ManGO wrapper/form information
    - Detach (type, schema, object_id)
    - Validate (object_id, recursive)
        - Run gather (below) to build the effective json schema
        - Get and build json payload with all current AVUs
        - Run payload and schema through validatorReturn result (OK or failure/explanation)
    - Export/Collapse/Rasterize/Gather/Dump (object_id, recursive)
        - Find all associated schemas and construct effective schema
        - Recursive would check/gather all parents up to root
- JSON Schema only, no JSON-LD

Metadata Templates Working Group - Late 2023

August 2023

- Utrecht has done this separation
  - UI schema - react
  - metadata schema - JSON schema
  - research space - not required
  - vault space - requirements, full schema
- schema information to be protected by metadata guard?

November 2023

- initial Python rules
  - Attach and Detach - initial work done, need error checking
  - Gather - next
  - Validate - last, depends on Gather

Metadata Templates Working Group - 2024

January 2024

- collection can have more than one schema

March 2024

- gather - using AllOf to combine schemas or loop through all
- users and groups and data objects and resources?   not for now

May 2024

- implementation
  - attach
  - detach
  - gather - returns array of attached schemas, possible recursive
  - validate - data object
  - validate - collection (all data objects below)

Conclusions

- Site-specific knowledge and interfaces are too diverse
- Template management is too big a task for the server/policy

- iRODS should focus on the capabilities and functionality
  - Rather than defining policy/schemas for applications and users
- iRODS cannot / should not be defining the templates for anyone
  - Should provide PEPs / microservices / functions to validate
    - But not manage the templates themselves

- Provide 70-80% of the original intent of metadata templates
- Community to use/test/incorporate prototype Python functions
  - Once good... we port to C++ and ship with the server as microservices

Running Code

```
# attach a template
$ irule -r irods_rule_engine_plugin-irods_rule_language-instance \
    "metadata_templates_collection_attach('*logical_path', '*schema_location', 'url')" \
    '*logical_path=/tempZone/home/rods/thedir%*schema_location=\
    https://raw.githubusercontent.com/fge/sample-json-schemas/master/jsonrpc2.0/jsonrpc-request-2.0.json' \
    ruleExecOut
```

```
# show AVU
$ imeta ls -C thedir
AVUs defined for collection /tempZone/home/rods/thedir:
attribute: irods::metadata_templates
value: https://raw.githubusercontent.com/fge/sample-json-schemas/master/jsonrpc2.0/jsonrpc-request-2.0.json
units: url
```

```
# detach a template
$ irule -r irods_rule_engine_plugin-irods_rule_language-instance \
    "metadata_templates_collection_detach('*logical_path', '*schema_location', 'removeme')" \
    '*logical_path=/tempZone/home/rods/thedir%*schema_location=doesnotexist' \
    ruleExecOut
```

Running Code

```
# gather, print to stdout
$ irule -r irods_rule_engine_plugin-irods_rule_language-instance \
    "metadata_templates_collection_gather('*logical_path', '*recursive', *schemas); \
    writeLine('stdout', *schemas)" \
    '*logical_path=/tempZone/home/rods/thedir%*recursive=0%*schemas=""' \
    ruleExecOut


# validate data object
$ irule -r irods_rule_engine_plugin-irods_rule_language-instance \
    "metadata_templates_collection_gather('*logical_path', '*recursive', *schemas); \
    metadata_templates_data_object_validate('*data_object_path', *schemas, *rc); \
    writeLine('stdout', *rc)" \
    '*logical_path=/tempZone/home/rods/thedir%*recursive=0%*schemas=""%\
    *data_object_path=/tempZone/home/rods/thedir/a.txt%*rc=""' \
    ruleExecOut


# validate a collection
$ irule -r irods_rule_engine_plugin-irods_rule_language-instance \
    "metadata_templates_collection_gather('*logical_path', '*recursive', *schemas); \
    metadata_templates_collection_validate('*logical_path', *schemas, *recursive, *errors); \
    writeLine('stdout', *errors)" \
    '*logical_path=/tempZone/home/rods/thedir%*recursive=0%*schemas=""%*errors=""' \
    ruleExecOut
```

Running Code

```
$ bash bats-core/bin/bats test_metadata_templates.bats
test_metadata_templates.bats
✓ collection - attach, gather, detach template
✓ attach bad schema
✓ validate data object
✓ validate collection

4 tests, 0 failures
```

Thank you!

Questions?