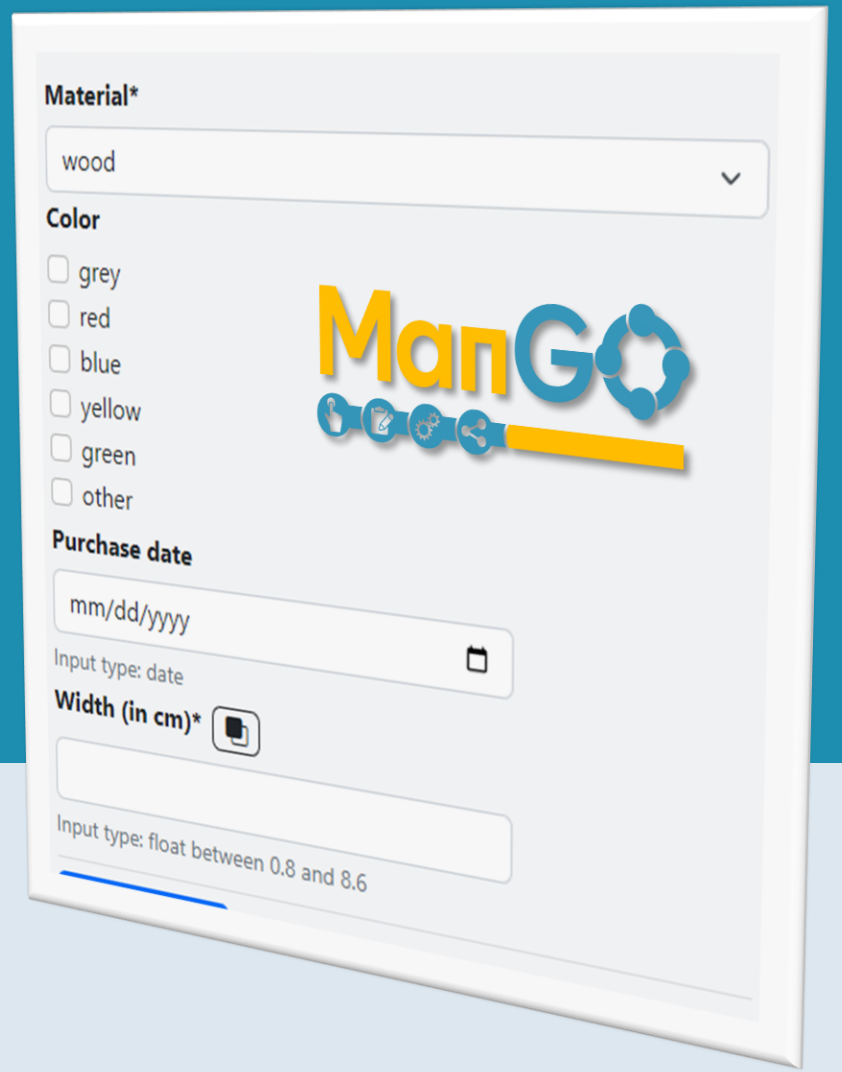
KU LEUVEN

Metadata schemas updates

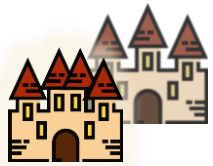
JSON schemas and storage in iRODS



FOZ-RDM – ICTS (KU Leuven)

Speaker: Mariana Montes

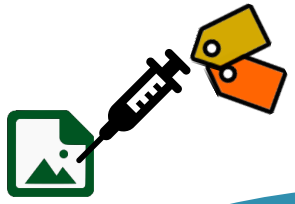
A history of ManGO metadata schemas



2023

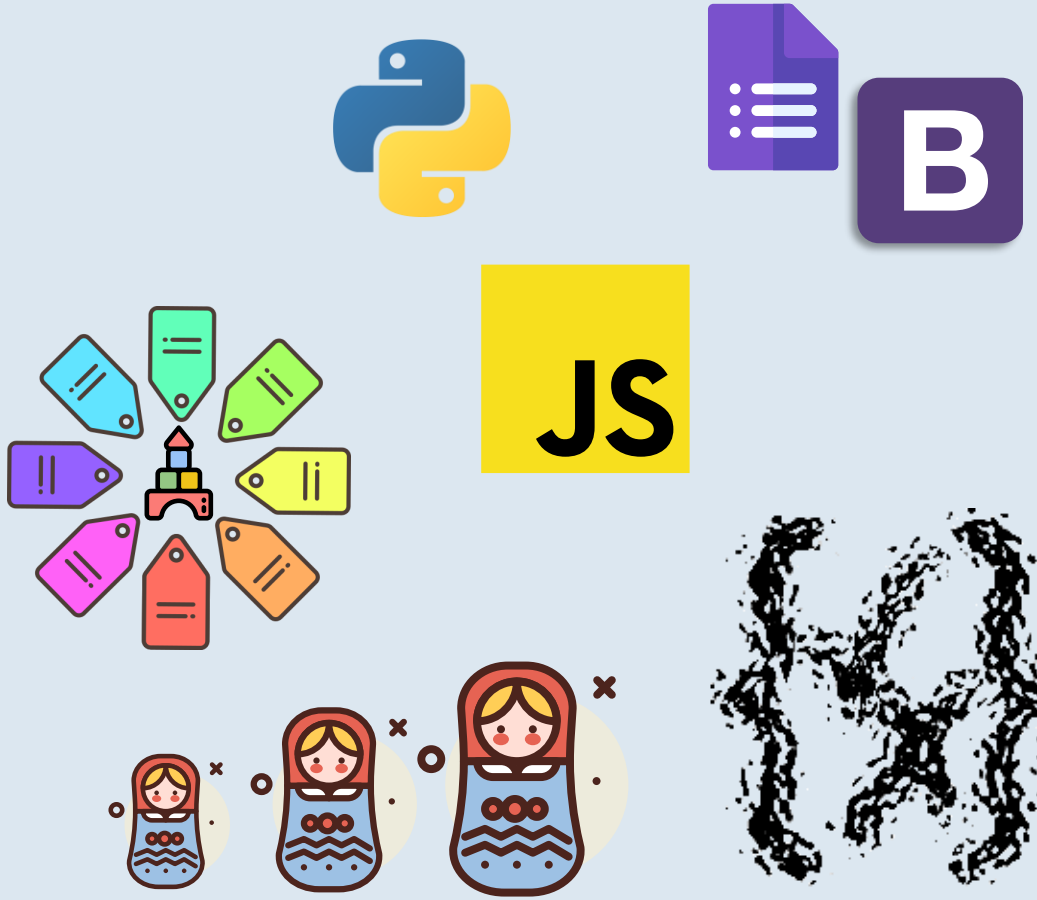
```
{
  "schema_name" : "book" ,
  "version" : "1.0.0" ,
  "status" : "published" ,
  "title" : "Book information" ,
  "properties" : {
    "title" : {
      "type" : "text" ,
      "title" : "Book title" ,
      "required" : true,
      "help" : "Title in the book (also translations)" ,
      "repeatable" : true
    }
  },
  "edited_by" : "username" ,
  "realm" : "project_collection" ,
  "parent" : ""
}
```

2024



2025

How we started



2023

```
{
  "schema_name" : "book" ,
  "version" : "1.0.0" ,
  "status" : "published" ,
  "title" : "Book information" ,
  "properties" : {
   "title" : {
    "type" : "text" ,
    "title" : "Book title" ,
    "required" : true,
    "help" : "Title in the book (also translations)" ,
    "repeatable" : true
   }
  },
  "edited_by" : "username" ,
  "realm" : "project_collection" ,
  "parent" : ""
}
```

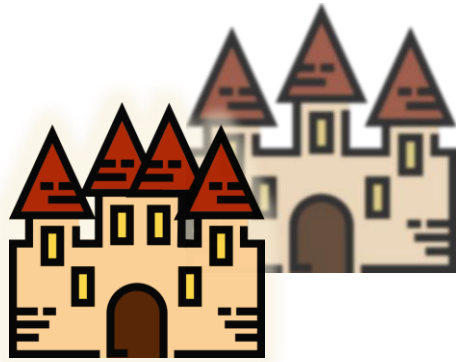
2024

2025

First steps: features

Apply rich and standardized metadata

Hierarchical structure for metadata



Schema life cycle

Why ManGO Metadata Schemas?

- Users not used to working with metadata
- Where to start, with flat AVUs?
- Forms as a user-friendly way of annotating structured metadata
- Flexible, domain-agnostic, user-friendly form manager

First steps: setup

Vanilla JS frontend
Python backend
Bootstrap 5.3

Based on JSON Schema **but different**

Schemas stored in persistent storage

The Code

- Limiting dependencies (no JS framework)
- It grew as we learned what to provide and how
- Open source but rather messy

The JSON

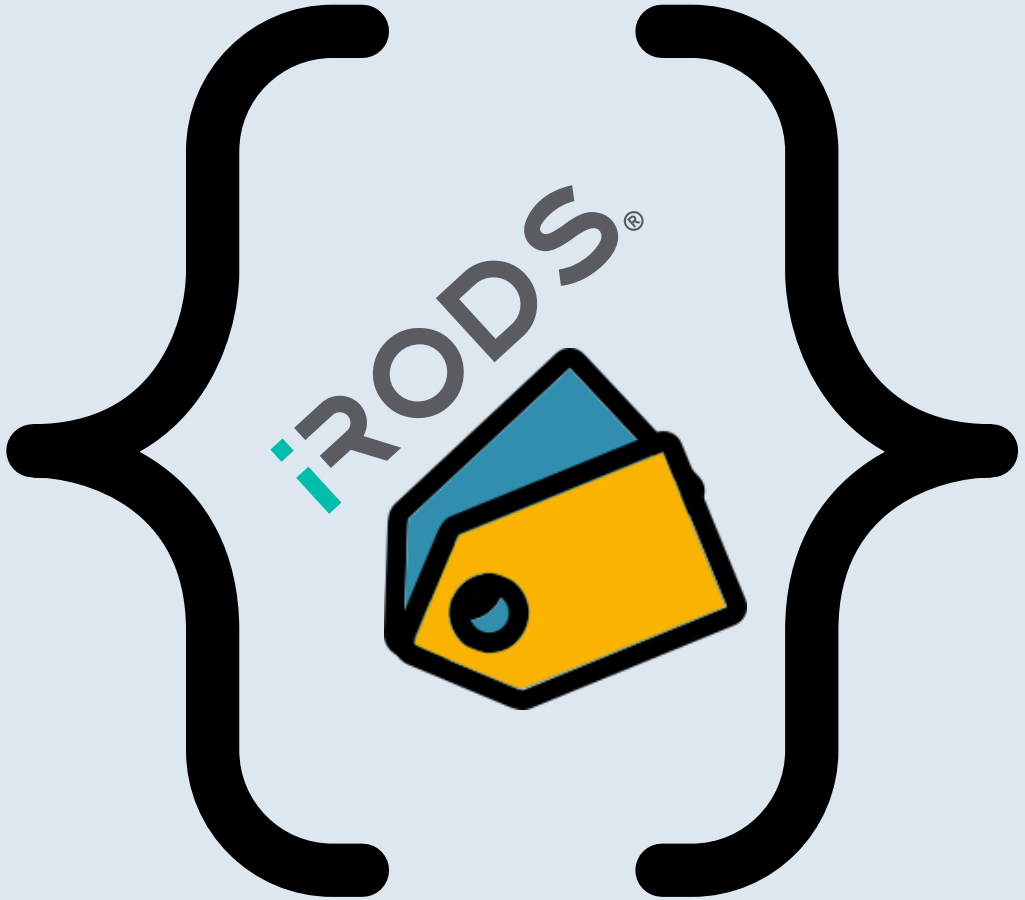
Based on JSON Schema **but different**

- Based on JSON Schema, known standard
- Goal: describing a form, NOT validation of JSON
- Examples of differences:
  - types are input types
  - “required” as a property of the fields

The storage

- Fast and easy to access (file system)
- No need to handle permissions
- Users don’t need to access the files themselves

AVUs and JSON

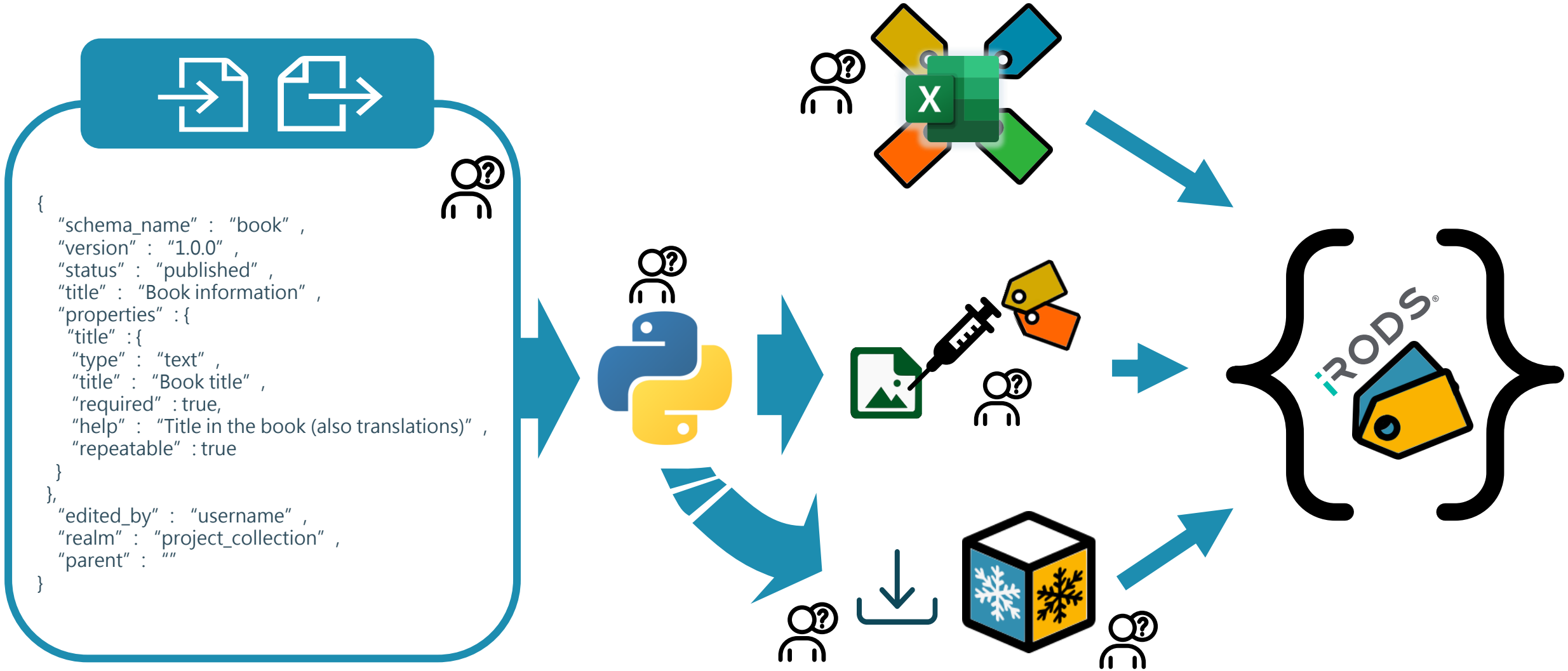


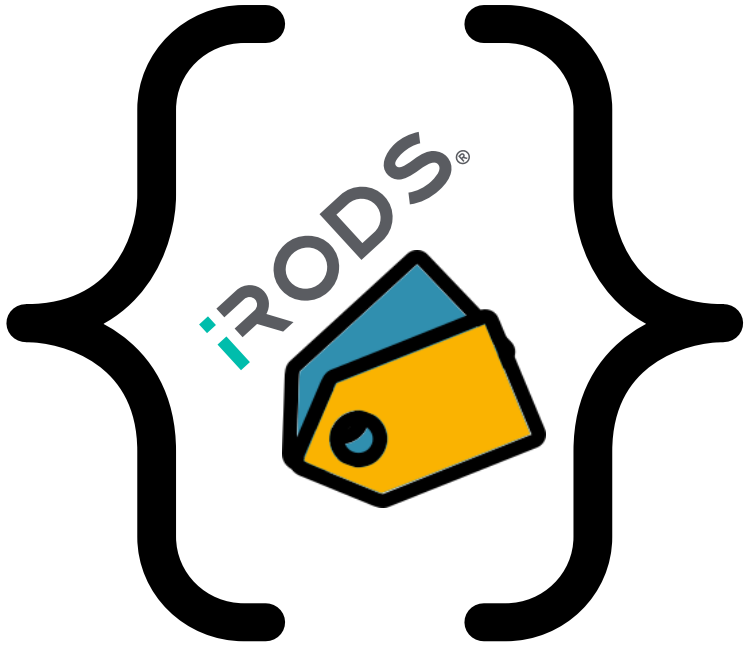
2023

2024

2025

```
{
  "schema_name" : "book" ,
  "version" : "1.0.0" ,
  "status" : "published" ,
  "title" : "Book information" ,
  "properties" : {
   "title" : {
    "type" : "text" ,
    "title" : "Book title" ,
    "required" : true,
    "help" : "Title in the book (also translations)" ,
    "repeatable" : true
   }
  },
  "edited_by" : "username" ,
  "realm" : "project_collection" ,
  "parent" : ""
}
```


{
"schema_name" : "book" ,
"version" : "1.0.0" ,
"status" : "published" ,
"title" : "Book information" ,
"properties" : {
"title" : {
"type" : "text" ,
"title" : "Book title" ,
"required" : true,
"help" : "Title in the book (also translations)" ,
"repeatable" : true
}
},
"edited_by" : "username" ,
"realm" : "project_collection" ,
"parent" : ""
}
X
iRODS



Schema metadata from Python dicts

From list of Python dicts to AVUs

All AVUs to JSON format

Now we **want** validation and standard representation!

iRODS®

What now?

23

2024

2025

```
{
  "schema_name" : "book" ,
  "version" : "1.0.0" ,
  "status" : "published" ,
  "title" : "Book information" ,
  "properties" : {
    "title" : {
      "type" : "text" ,
      "title" : "Book title" ,
      "required" : true,
      "help" : "Title in the book (also translations)" ,
      "repeatable" : true
    }
  },
  "edited_by" : "username" ,
  "realm" : "project_collection" ,
  "parent" : ""
}
```

The Code

- Modern web and browser standards: Web Components + lit.dev
  - No major JS framework
  - HTML and data together
  - Lightweight, reactive
- Rewrite
  - Multiple developers together
  - Full specs in sight
  - Learning from experience

The JSON

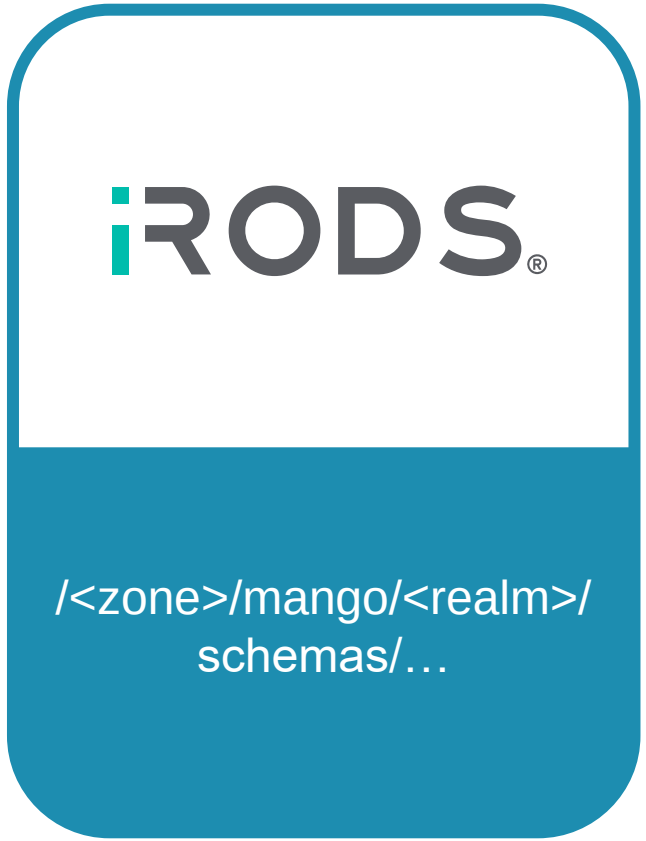
- Validation of schema metadata beyond the portal
- Schema manager as general form editor and JSON-schema writer

How much of the UI can be described in it?

Write and read JSON Schema

Convert existing schemas

The storage

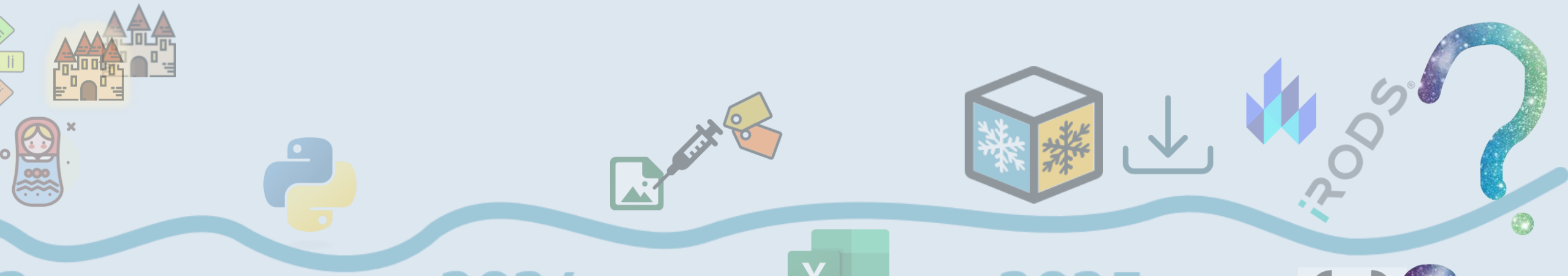


- User access to read/download
- Editing permissions managed by Portal
- Access by rules for MD protection

☑ Write and read in iRODS

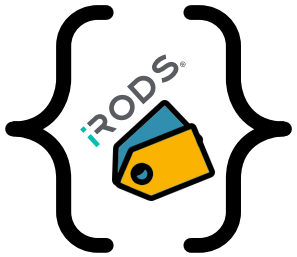
☑ Script to migrate existing schemas

☐ Change filenames and use metadata!



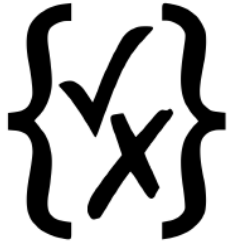
Moving forward…

Challenges and uncertainties



☐ Custom conversion needed to validate with JSON Schema

github/kuleuven repos: mango-mdconverter, mango-mdschema



? Implement protection rules, maybe using the code from the Metadata Templates WG?

To keep up!

Work together on the code,
sharing knowledge and ideas

Listen to our users,
our source of inspiration

Thank you!

mariana.montes@kuleuven.be

github.com/kuleuven

- mango-metadata-schemas
- mango-mdconverter
- mango-mdschemas

KU LEUVEN