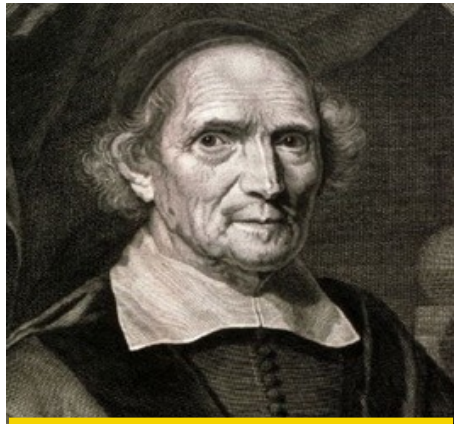
Utrecht University

06/10/2021

Retrospective: migrating Yoda from the PHP iRODS client to the Python iRODS client

Lazlo Westerhof
l.r.westerhof@uu.nl

Organisation & people



ESTABLISHED

1636

PROFESSORS

> 650 Incl. faculties Medicine

FACULTIES

7+2 teaching institutes

STAFF-MEMBERS

> 7,400

STUDENTS

> 30,000

Yoda: 'FAIR' Research Data Management

Research

Collaborate safely as a group
- membership self-managed by researchers

Vault

Maintain integrity, deposit a folder in the vault
- metadata can vary per community,
- datamanager approves deposit

Allow FAIR reuse, publish a data package
- datamanager approves publication, DOI citable data

Data deposit workflow

Researcher
requests to deposit

Data manager
checks metadata
complies with policies

System
deposits a copy in the vault

Metadata form - /research-epicpid

Close

Save ✔✔✔✔✔

Delete all metadata

Title Subjective and objective evaluations of horses for fit-to-compete or unfit-to-compete judgement.

Description At Fédération Equestre Internationale (FEI) competitions horses pass a veterinary inspection for judgement of 'fit-to-compete'. However, FEI Veterinary Delegates (VDs) often differ in opinion. The aim of the present study was to evaluate intra- and inter-observer agreements of 'fit-to-compete' judgement and compare these with objective gait analysis

Discipline Agricultural Sciences - Veterinary science (4.3) +

Version 1

Language of the data English

Collection process **Start date** 02 / 20 / 2017 **End date** 02 / 20 / 2017

Location(s) covered +

Period covered **Start date** mm / dd / yyyy **End date** mm / dd / yyyy

Tag Fit-to-compete −

Tag Objective gait analisys −

Tag Observer agreement −

Tag Motion symmetry −

Tag Optical motion capture −

Tag Motion capture videos − +

Utrecht University

Log out researcher

Yoda Portal Research Vault Statistics Group Manager

Search by filename | Search term...

Home / research-initial

research-initial Submitted

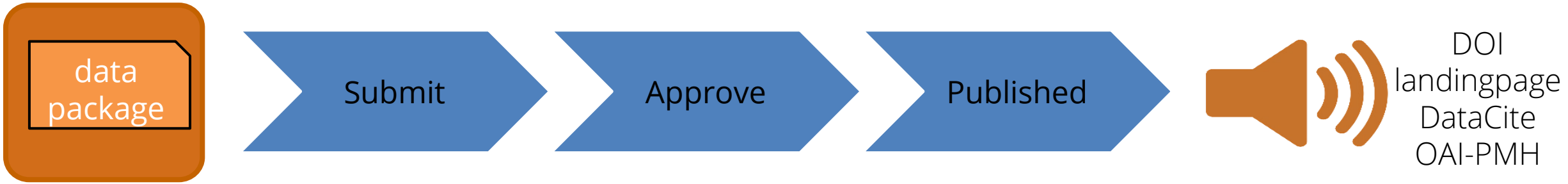
Metadata | Create Folder | Upload | Actions

Name	Size	Modified date	
clone		2021-02-23 13:13	...
testdata		2021-02-23 13:06	...
flamingos.jpg	1.2 MiB	2021-03-11 16:40	...
yoda-metadata.json	1.3 kiB	2021-02-23 13:35	...

10

Previous 1 Next

FAIR data publication workflow



Researcher
requests to publish

Data manager
Checks if metadata complies with publication policies

System
Publishes the metadata and provides internet access to data if classified as “Open”

research-initial1[1614083569] Published

Metadata Actions

Name	Size	Modified date
original		2021-02-23 13:32
License.txt	18.2 kiB	2021-02-23 13:35
yoda-metadata[1614083570].json	793 B	2021-02-23 13:35

Serra Bragança, Filipe. & Brommer, Harold & Sloet van Oldruitenborgh-Oosterbaan, Marianne.

Subjective and objective evaluations of horses for fit-to-compete or unfit-to-compete judgement.

Publication Date: **2020-11-25T14:27:02.139485** Accessibility: **Open - freely retrievable**

At Fédération Equestre Internationale (FEI) competitions horses pass a veterinary inspection for judgement of 'fit-to-compete'. However, FEI Veterinary Delegates (VDs) often differ in opinion. The aim of the present study was to evaluate intra- and inter-observer agreements of 'fit-to-compete' judgement and compare these with objective gait analysis measurements. Twelve horses were evaluated by three experienced VDs and one veterinary specialist and video-recorded for re-evaluation later. Simultaneously, quantitative gait analysis measurements (Qhorse®) were acquired. Inter-observer agreement during live evaluations was fair (κ=0.395, 58% agreement). Intra-observer agreement between live observations and videos at one month and one year was 71% and 73% respectively. Sensitivity and specificity of motion symmetry measured with quantitative gait analysis system were 83.3% and 66.7% respectively, against the consensus of all observers as a reference. Our findings might suggest that more VDs should be used to adequately judge 'fit-to-compete'. Quantitative-gait-analysis may be useful to support decision making during fit-to-compete judgement.

Tags

Fit-to-compete | Objective gait analisys | Observer agreement | Motion symmetry | Optical motion capture | Motion capture videos

METADATA

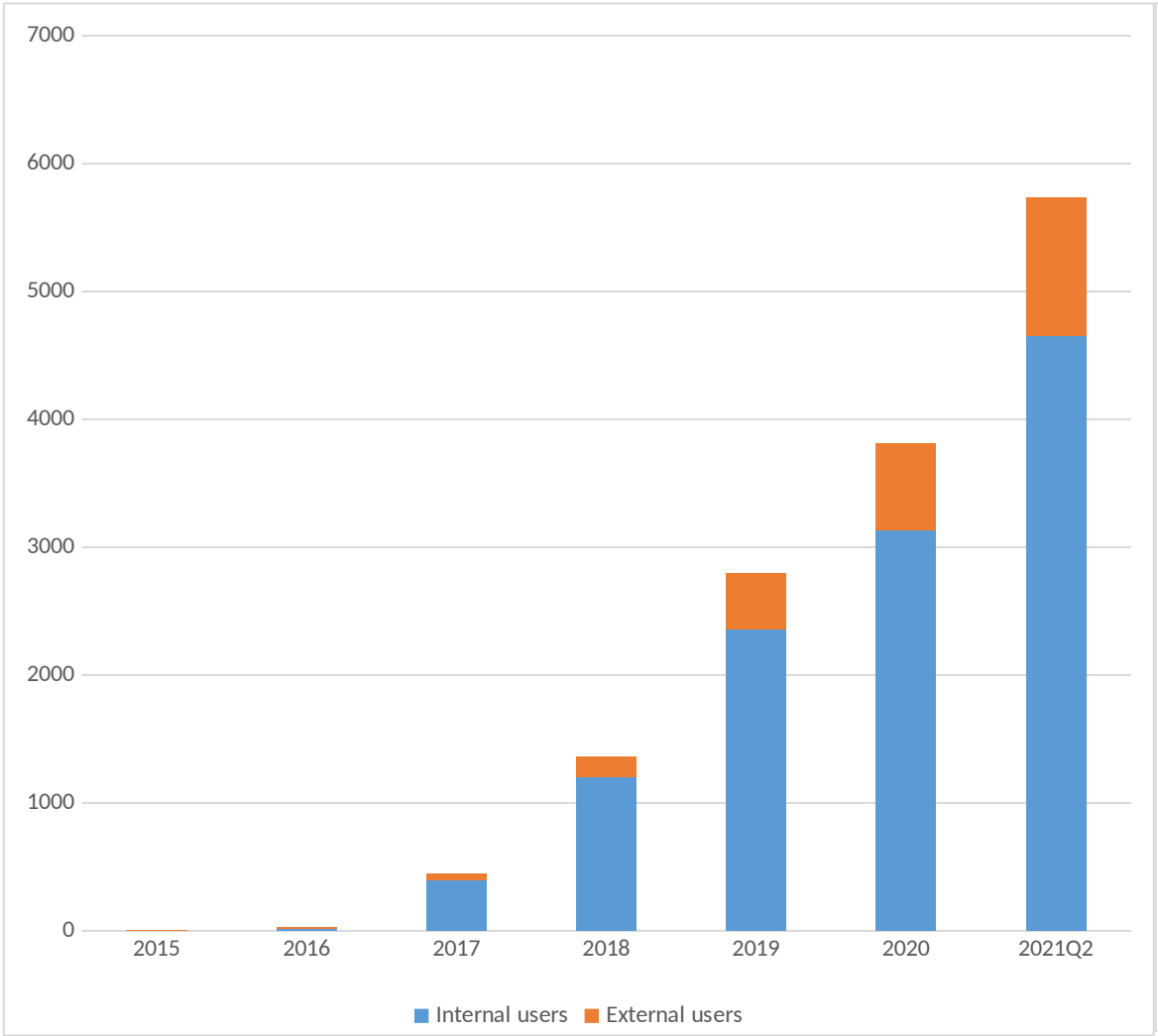
Disciplines	**Agricultural Sciences - Veterinary science (4.3)**
Version	**1**
Language	**en - English**

Yoda milestones

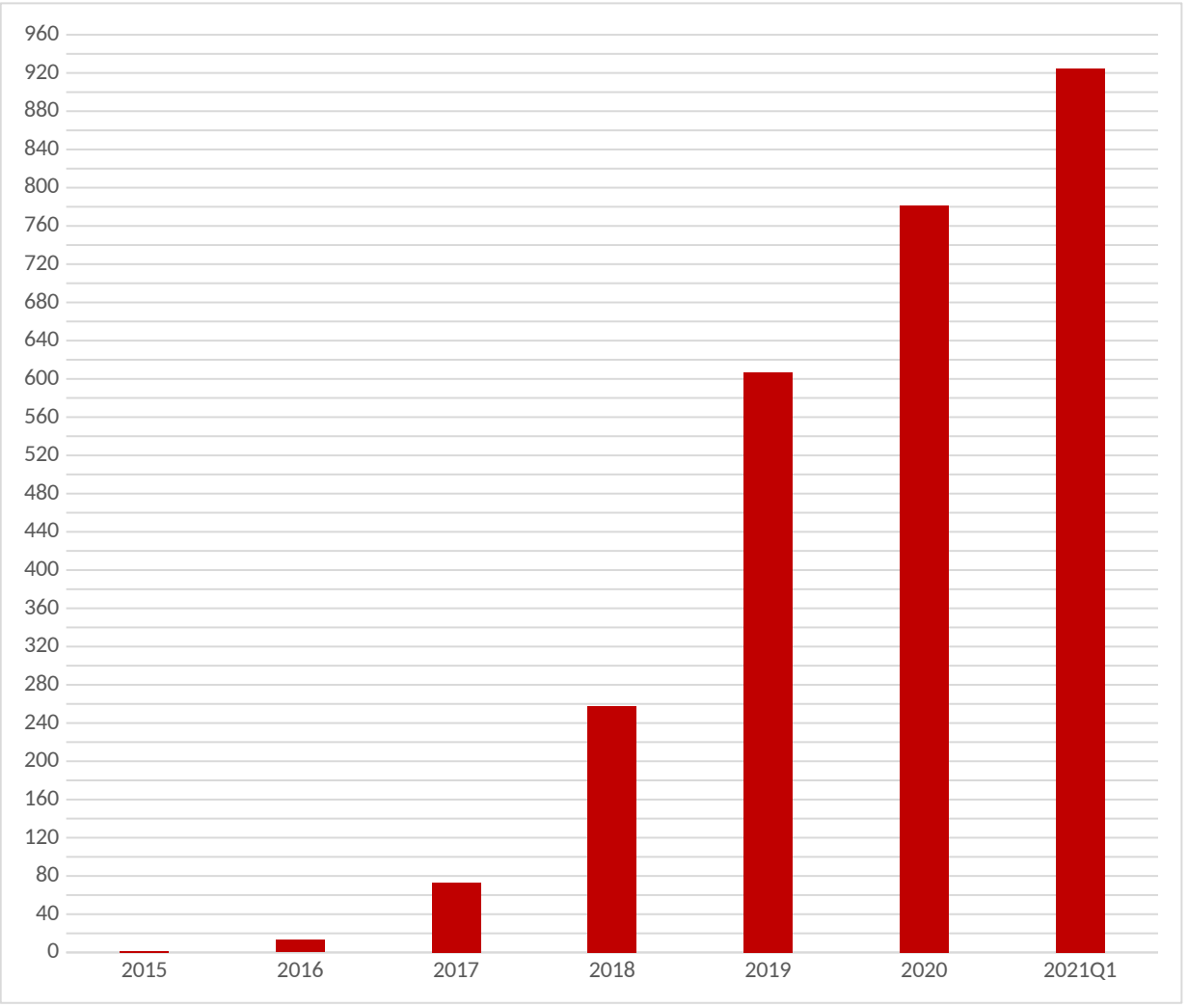
2015	• Yoda Portal and Intake module for Youth project, iRODS 3.3 based • Groupmanager module	v0.4 – v0.9
2016	• Yoda Disk (davrods module) • Yoda Portal supports dynamic Plug-in modules, iRODS 4.0	v0.9.7
2017	• Research Workspace, revisions, metadata form • Vault archive, deposit workflow, statistics, data publication workflow (DOI)	v1.0 – v1.3
2018	• Vault metadata operations, EPIC PID, External user provisioning • (de/re)publication workflows, OAI-PMH harvestable, iRODS 4.1	v1.4
2019	• Metadata-schema management, dynamic metadata forms rendering • Metadata form based on JSON schema, file up/download in Portal, iRODS 4.2.6	v1.5
2020	• Metadata format changed from XML to JSON, with JSON-AVU • Python rules engine iRODS 4.2.7	v1.6
2021	• Yoda API, OIDC authentication, iRODS 4.2.9? • Change from irods-php to python-irodsclient	v1.7

Yoda managed research data

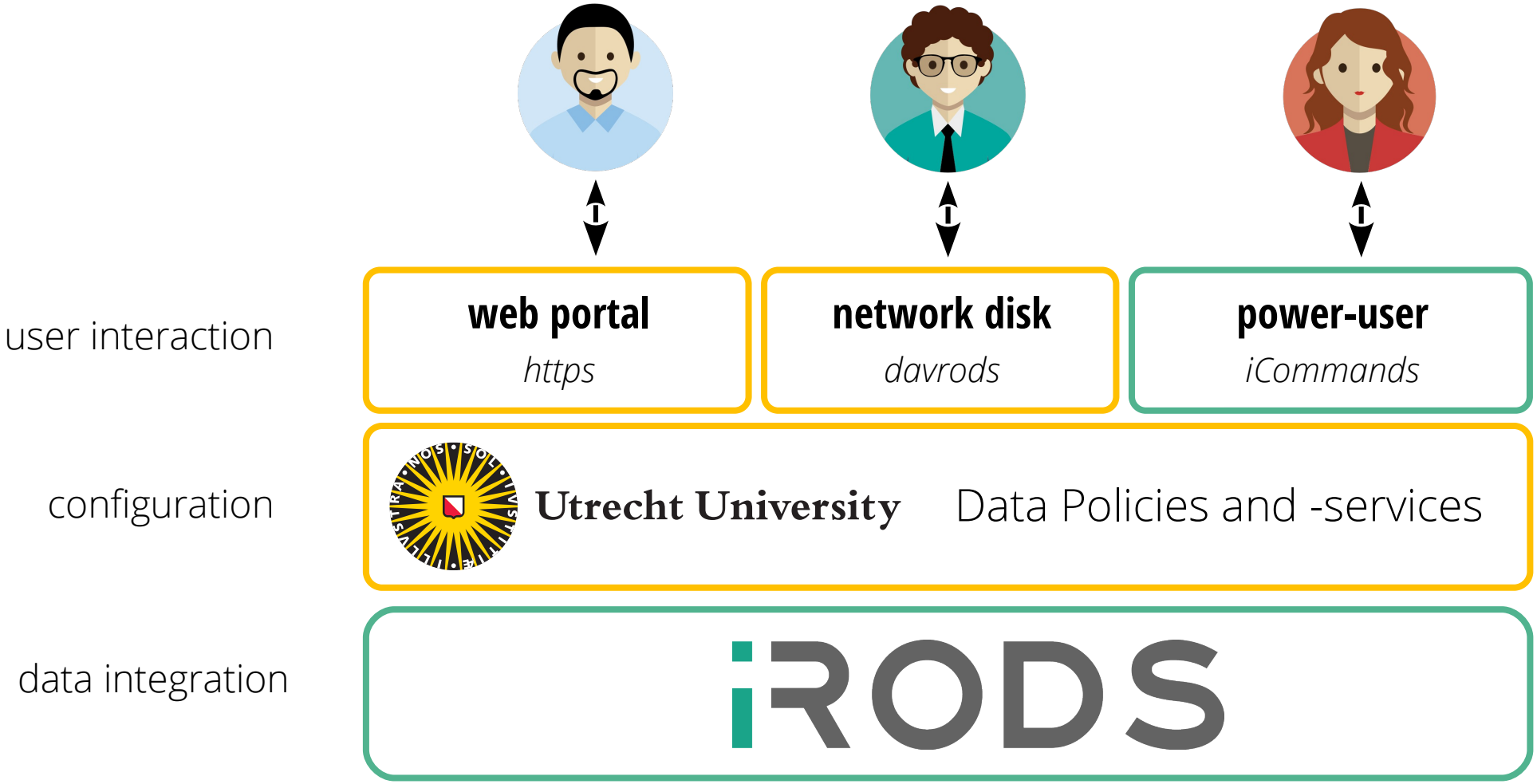
Users



Storage (TB)



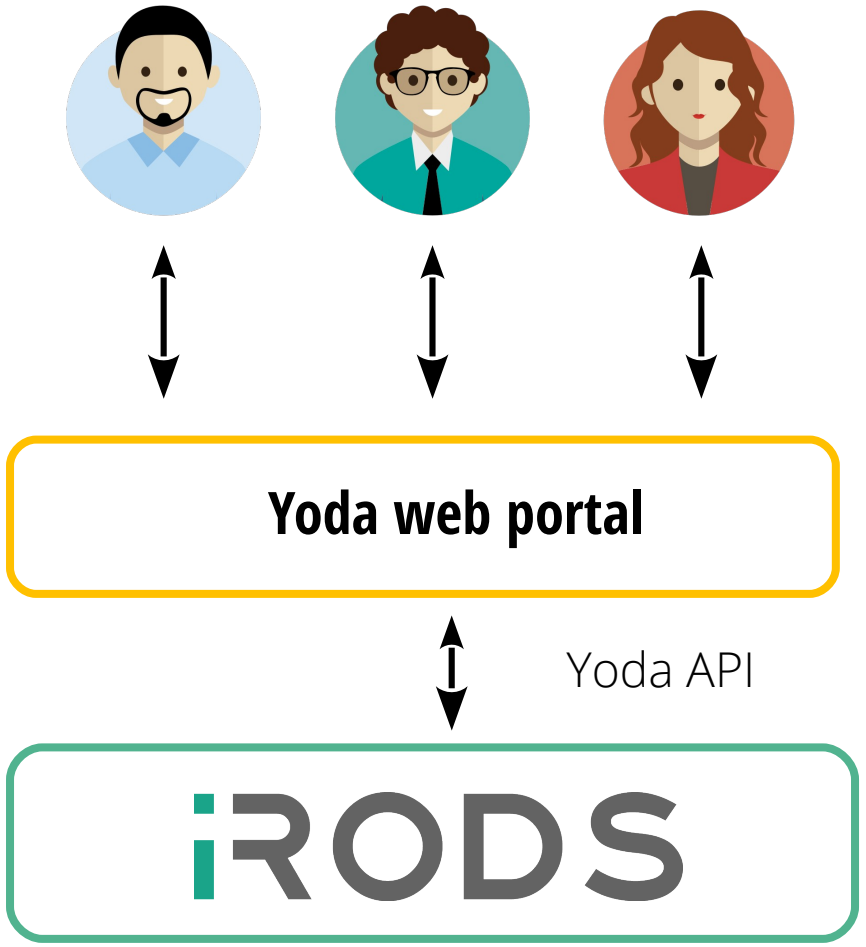
Yoda is build on iRODS



Yoda web portal & API

- Yoda web portal communicates with backend using Yoda API

- API defined in Python ruleset
  - REST API
  - exposes all Yoda functionality as API
  - all business logic in ruleset
  - very lean web portal



Converting a rule to Yoda API

```
# iRODS rule language.
concat(*x, *y, *foo) {
  *foo = *x ++ *y;
}
```

Converting a rule to Yoda API

- Equivalent Python rule
  - Boilerplate
  - Non-pythonic
  - Difficult to interface from Python functions

```
# iRODS rule language.
concat(*x, *y, *foo) {
  *foo = *x ++ *y;
}

# Equivalent Python rule.
def concat(rule_args, callback, rei):
    x, y = rule_args[0:2]
    rule_args[2] = x + y
```

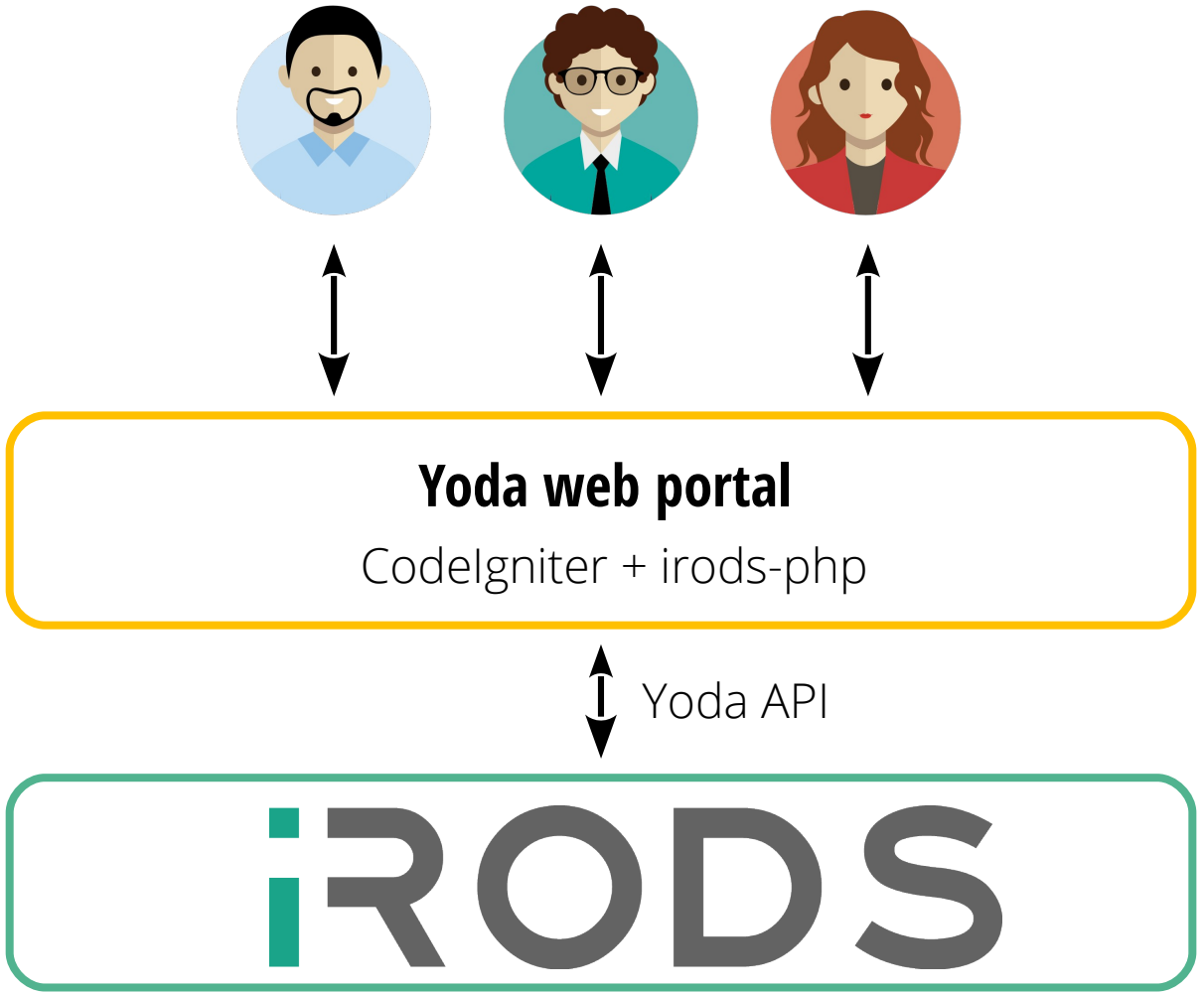
Converting a rule to Yoda API

- Equivalent Python rule
  - Boilerplate
  - Non-pythonic
  - Difficult to interface from Python functions

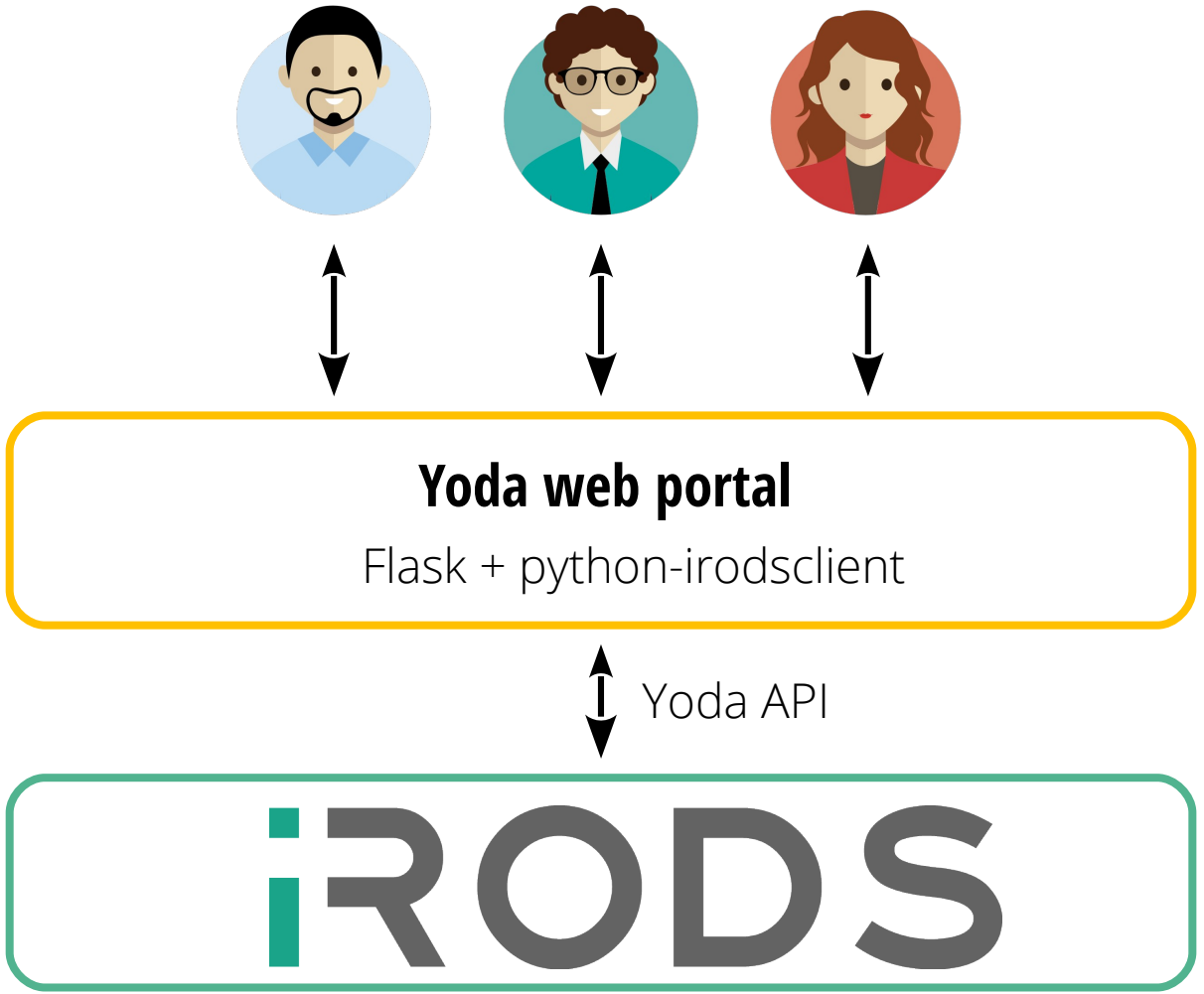
- Can we make this easier?
  - @api decorator
  - JSON input → Python arguments
  - Python return value → JSON output
  - Checks required/optional arguments
  - Supports dicts, lists…
  - Standardizes error handling

```
# iRODS rule language.
concat(*x, *y, *foo) {
  *foo = *x ++ *y;
}

# Equivalent Python rule.
def concat(rule_args, callback, rei):
    x, y = rule_args[0:2]
    rule_args[2] = x + y


# Yoda API Python rule.
@api.make()
def api_concat(ctx, foo, bar):
    return foo + bar
```

Yoda web portal

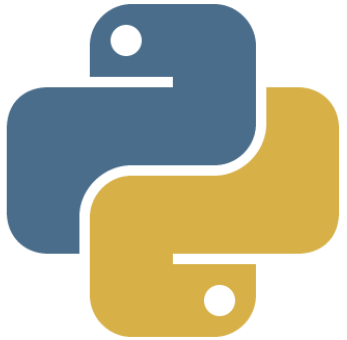


Yoda web portal



Why are we migrating from the PHP client to Python client?

- Client actively developed
- Maintainability
- Performance improvement
- One programming language less
- Libraries and frameworks
- Available tooling



python

Flask iRODS

- Two modules for communication with iRODS

- Connection manager module
  - manage python-irodsclient sessions
  - session per authenticated user

- API module
  - handles API calls from web portal
  - JSON encoding / decoding

```
    x = rule.Rule(
        g.irods,
        body='a {{ api_{}(*x); }}'.format(fn),
        params={'*x': '"{}"'.format(sanitized_params)},
        output='ruleExecOut')
    x = x.execute()
```

Interfacing with Yoda API

- Yoda API rule
  - JSON input → JSON output

```
# Yoda API Python rule.
@api.make()
def api_concat(ctx, foo, bar):
    return foo + bar
```

Interfacing with Yoda API

- Yoda API rule
  - JSON input → JSON output

- Calling API from Flask
  - dictionary input → dictionary output

```
# Yoda API Python rule.
@api.make()
def api_concat(ctx, foo, bar):
    return foo + bar


# Callable from Flask frontend .
response = api.call('concat',
                    {'foo': 'test',
                     'bar': '123'})
```

Interfacing with Yoda API

- Yoda API rule
  - JSON input → JSON output

- Calling API from Flask
  - dictionary input → dictionary output

- Calling API from JavaScript
  - dictionary input → dictionary output

```
# Yoda API Python rule.
@api.make()
def api_concat(ctx, foo, bar):
    return foo + bar


# Callable from Flask frontend .
response = api.call('concat',
                    {'foo': 'test',
                     'bar': '123'})


# Callable from JavaScript frontend .
let str = await
       Yoda.call('concat',
                 {'foo': 'test',
                  'bar': '123'});
```

Data archive geosciences

- Simplify deposit workflow to three steps
  - Upload data
  - Add metadata
  - Submit data package

- Reuses existing Yoda API functionality

Modern web file upload suport

- Existing client library (https://github.com/flowjs/flow.js)
  - HTML5 File API
  - Support for folders
  - Chunked uploads
  - Resumable uploads

- python-irodsclient made it easy to implement upload backend

Python client challenges

- Session cleanup after rule execution
  - re-open a connection to iRODS after each rule execution
  - significant performance overhead
  - https://github.com/irods/python-irodsclient/issues/190

- Character limits?
  - strlen 1030 of msg > dim size 1024

Yoda is free and open source

Everything is available on GitHub: https://github.com/UtrechtUniversity

Deployment: https://github.com/UtrechtUniversity/yoda/tree/pyportal
Yoda portal: https://github.com/UtrechtUniversity/yoda-portal/tree/pyportal
iRODS ruleset: https://github.com/UtrechtUniversity/irods-ruleset-uu/tree/pyportal

More information? → UGM Slack / l.r.westerhof@uu.nl

```
$ iexit
```

Utrecht University

DISCLAIMER

The information in this presentation has been compiled with the utmost care,
but no rights can be derived from its contents.