iRODS User Training Introduction

Vlaanderen
is supercomputing

VLAAMS
SUPERCOMPUTER
CENTRUM

Innovative Computing
for A Smarter Flanders

vscentrum.be

What is iRODS?

- iRODS (integrated Rule-Oriented Data System)
- Open Source distributed data and storage management system
- Configurable data management policies and workflows
- Scalable
- KU Leuven is part of the iRODS consortium

Tier-1 Data in the research Data Lifecycle

Pre-publication/Active Data

Post-publication/Inactive Data

Planning

Project proposal → Project start

Data management plans
Copyright, ethics, usage of data

Active Research

Collect Create → Use → Analyse

Documentation
Metadata
Storage and backup
Access control

Share with collaborators

Sharing/Reuse

Publication → End project

Publication
Link with 'persistent identifiers'
Discovery and reusage
Long term preservation

iRODS Vlaanderen is supercomputing

VSC Tier-1 Data

HPC users

Tier-1 Compute

Tier-1 Cloud

Regional / domain specific repositories

Institutional repositories

Tier-1 Data architecture

VSC Tier-1

VSC Tier-2

antwerp university association

ASSOCIATIE KU LEUVEN

universitaire BRUSSEL

ASSOCIATIE UNIVERSITEIT GENT

Users

Clients

metalnx

Cyberduck

DAVRods

iCommands

Python Client

iRODS

iRODS servers

iCAT

Mysql Database

Rule Engine

Metadata

Storage Hardware

DC Heverlee

Posix Resc1

Ceph Resc3

3km

DC Leuven

Posix Resc2
Replica Resc1

iRODS Core competencies

Unified Storage Namespace

Data virtualization of distributed storage systems

Automation

Rule Engine to enforce data polices

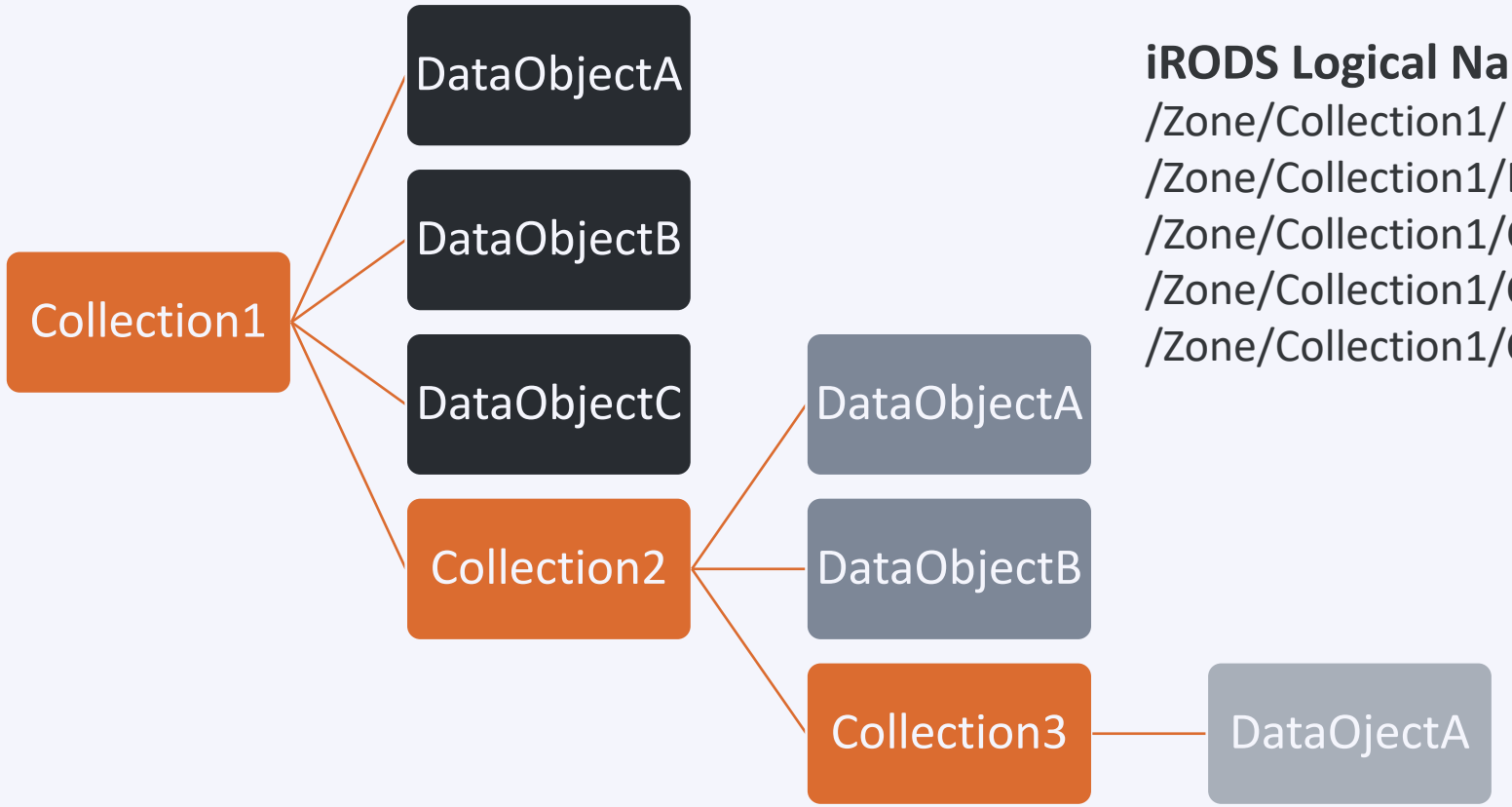
Data Discovery

Rich Metadata for collections and data objects (System metadata and user-defined metadata)

Secure collaboration

Three mechanisms: Permissions, Tickets and Federation.

Data organization in iRODS

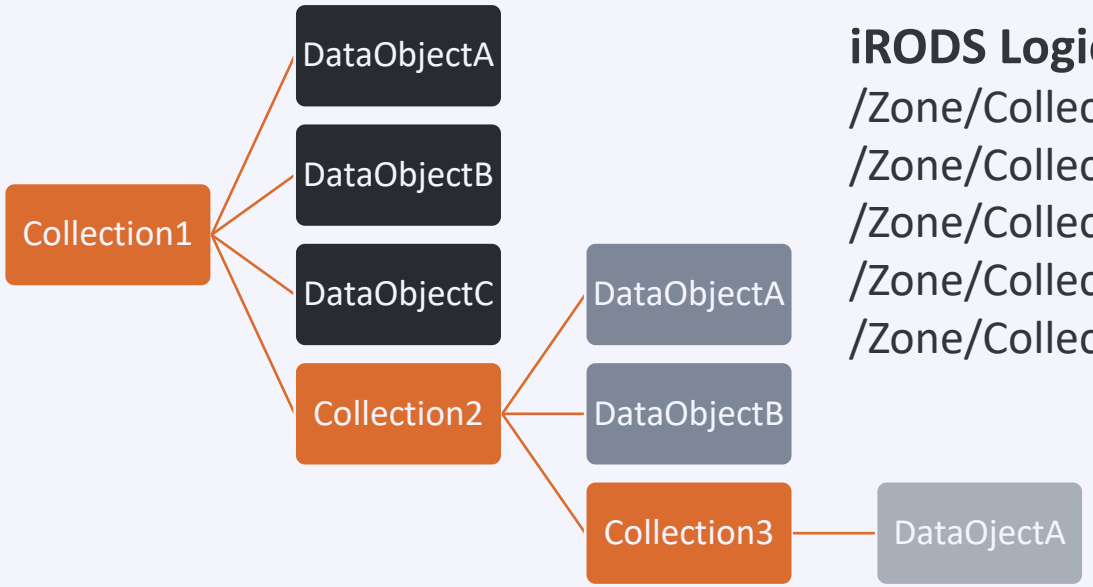


iRODS Logical Namespace
/Zone/Collection1/
/Zone/Collection1/DataObjectA/
/Zone/Collection1/Collection2/
/Zone/Collection1/Collection2/DataObjectA
/Zone/Collection1/Collection2/Collection3/DataObjectA

Collections ~ Directories
DataObjects ~ Files

Data virtualization in iRODS

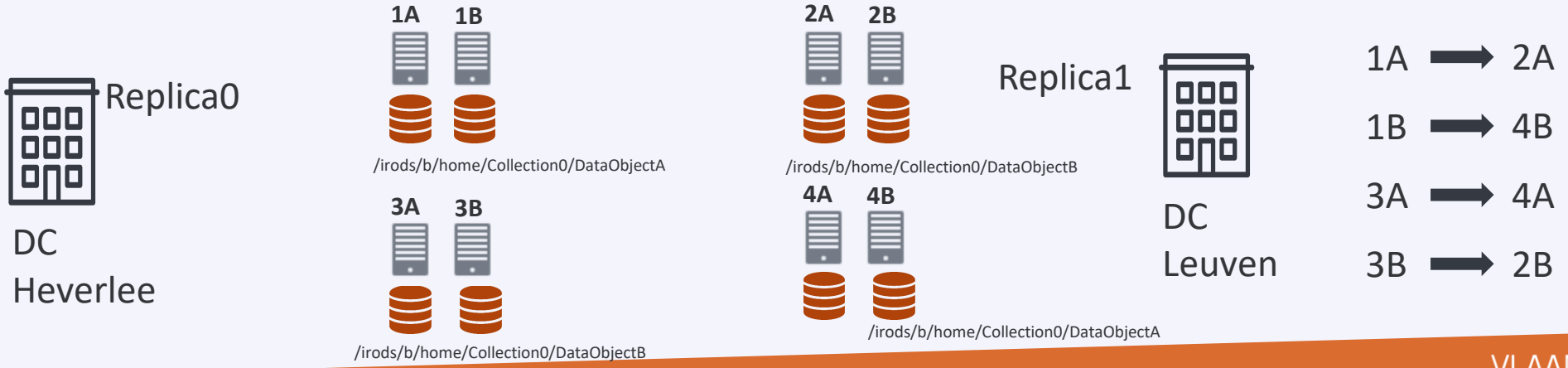
Logical representation



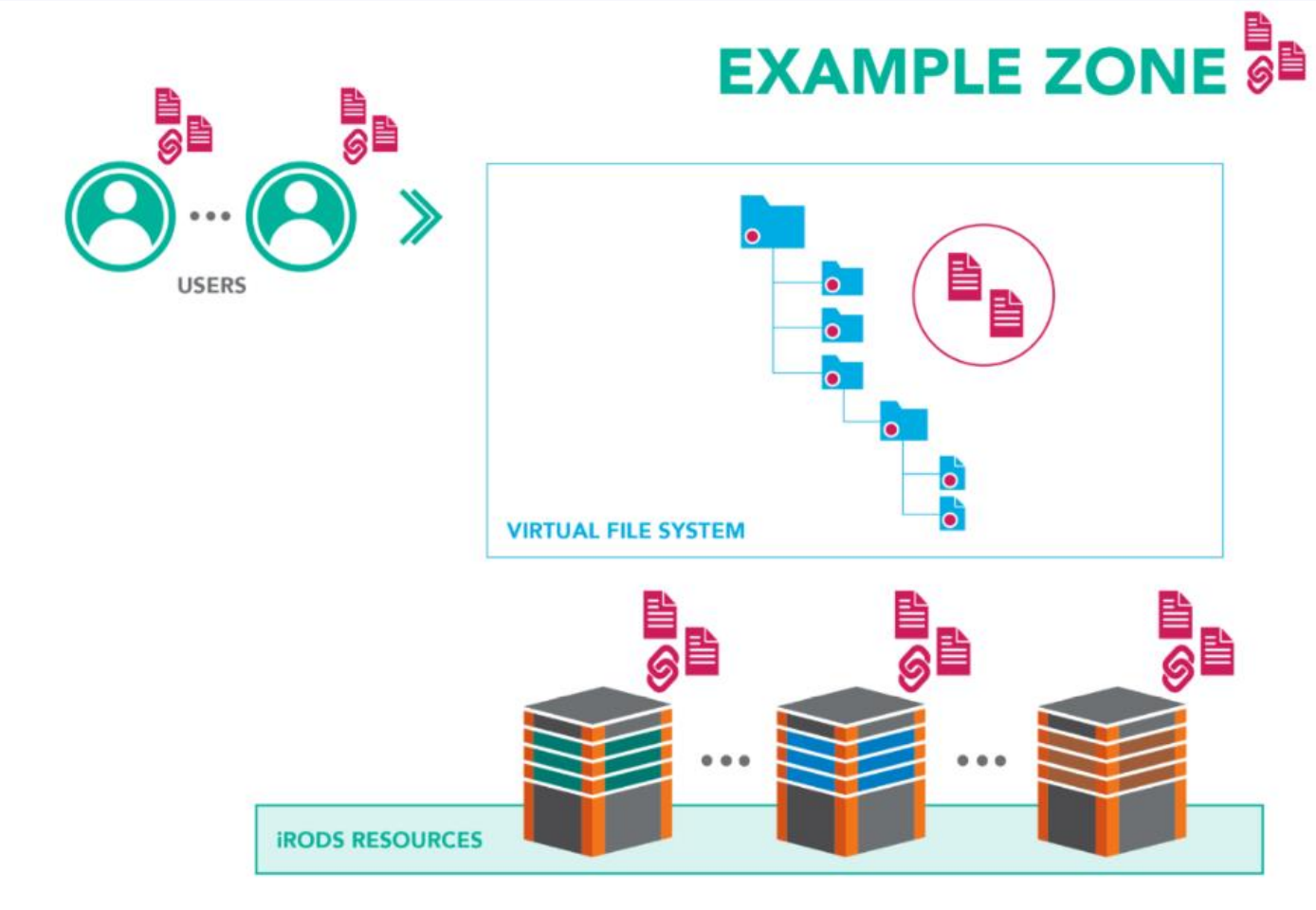
iRODS Logical Namespace

/Zone/Collection1/
/Zone/Collection1/DataObjectA/
/Zone/Collection1/Collection2/
/Zone/Collection1/Collection2/DataObjectA
/Zone/Collection1/Collection2/Collection3/DataObjectA

Physical representation



Metadata in iRODS



System Metadata:

- filename, file size, creation date …

User Metadata:

- Manual introduction
- Metadata templates
- Automation (rules/microservices)

Clients

VSC Tier-1

VSC Tier-2

User Clients

metaInx

Cyberduck

DAVRods

iCommands

PRC-Python API

Interaction with iRODS

Functionalities

- icommands:

  `iput – iget- irsync –imeta...`

- uploading/downloading data
- adding metadata to data objects/collections
- querying based on metadata
- deleting data objects/collections
- synchronization of data
- ACLs to data objects/collections

```
vsc33731@login1 ~
$ imkdir research

vsc33731@login1 ~
$ icd research

vsc33731@login1 ~
$ iput dataset1

vsc33731@login1 ~
$ iput dataset2

vsc33731@login1 ~
$ ils
/kuleuven_tier1_pilot/home/vsc33731/research:
  dataset1
  dataset2
```

Functionalities

- VSC-PRC:

  `Python3, python-irodsclient`

- working with data objects/collections
- adding metadata to data objects/collections
- querying based on metadata
- deleting data objects/collections
- listing the disk usage
- ACLs to data objects/collections

```
In [1]: from vsc_irods.session import VSCiRODSSession

In [2]: session = VSCiRODSSession(txt='-')

In [3]: irods_path = session.path.get_irods_home() + "/research"

In [4]: session.path.imkdir('research')

In [5]: session.path.ichdir('research')

In [6]: session.bulk.put("./dataset*", irods_path)

In [7]: for item in session.search.find(irods_path, types='f'):
   ...:     print(item)
   ...:
/kuleuven_tier1_pilot/home/vsc33731/research/dataset1
/kuleuven_tier1_pilot/home/vsc33731/research/dataset2
```

Functionalities

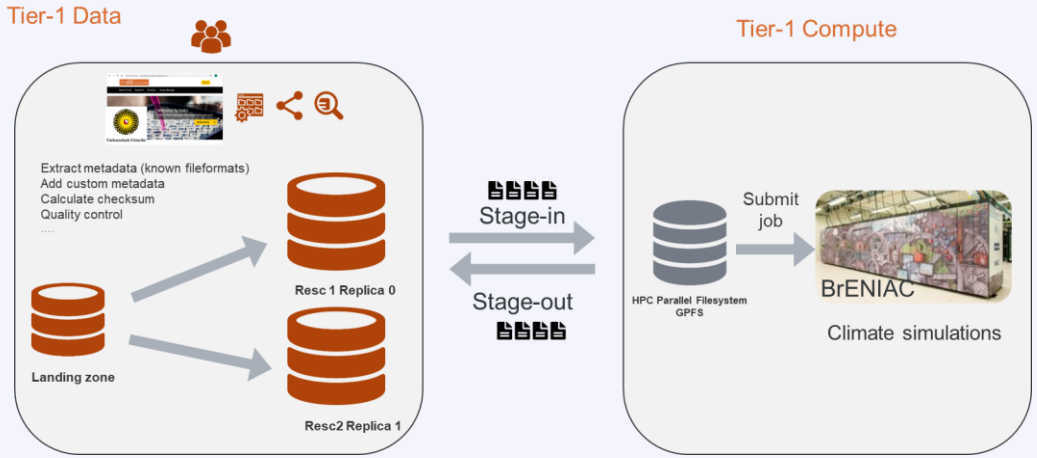
- HPC_to_Data:

- icommands:

  `iput – iget- irsync -ibun`

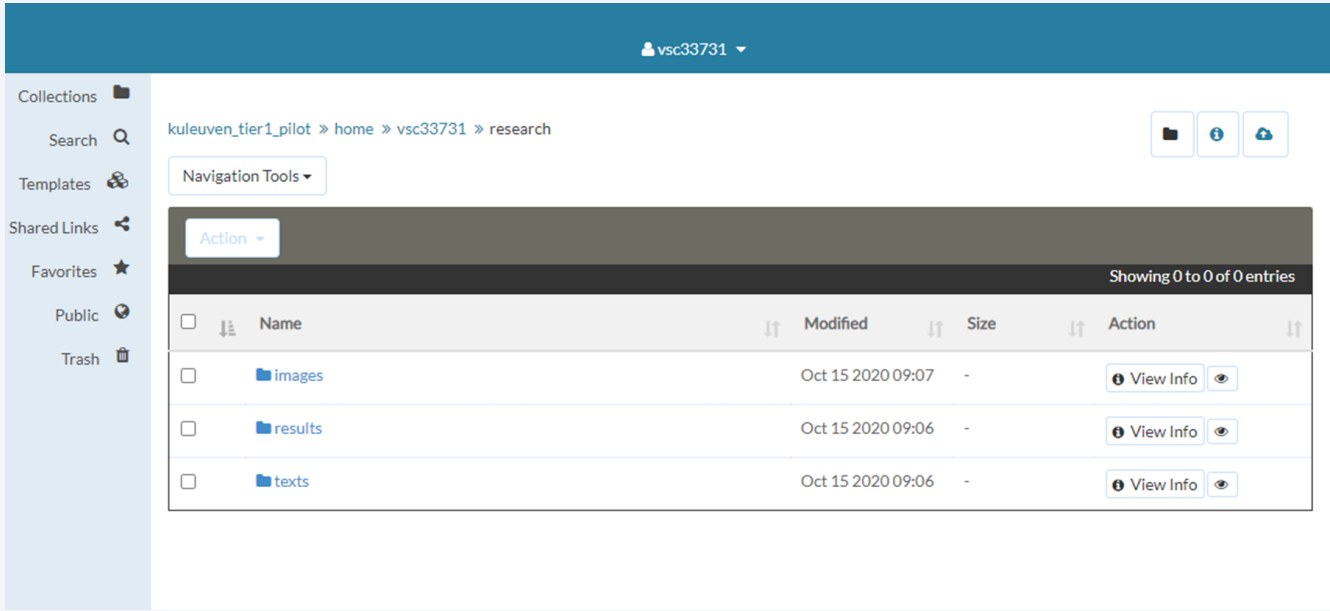
- VSC Python client:

  `vsc-prc-iget- vsc-prc-iput`



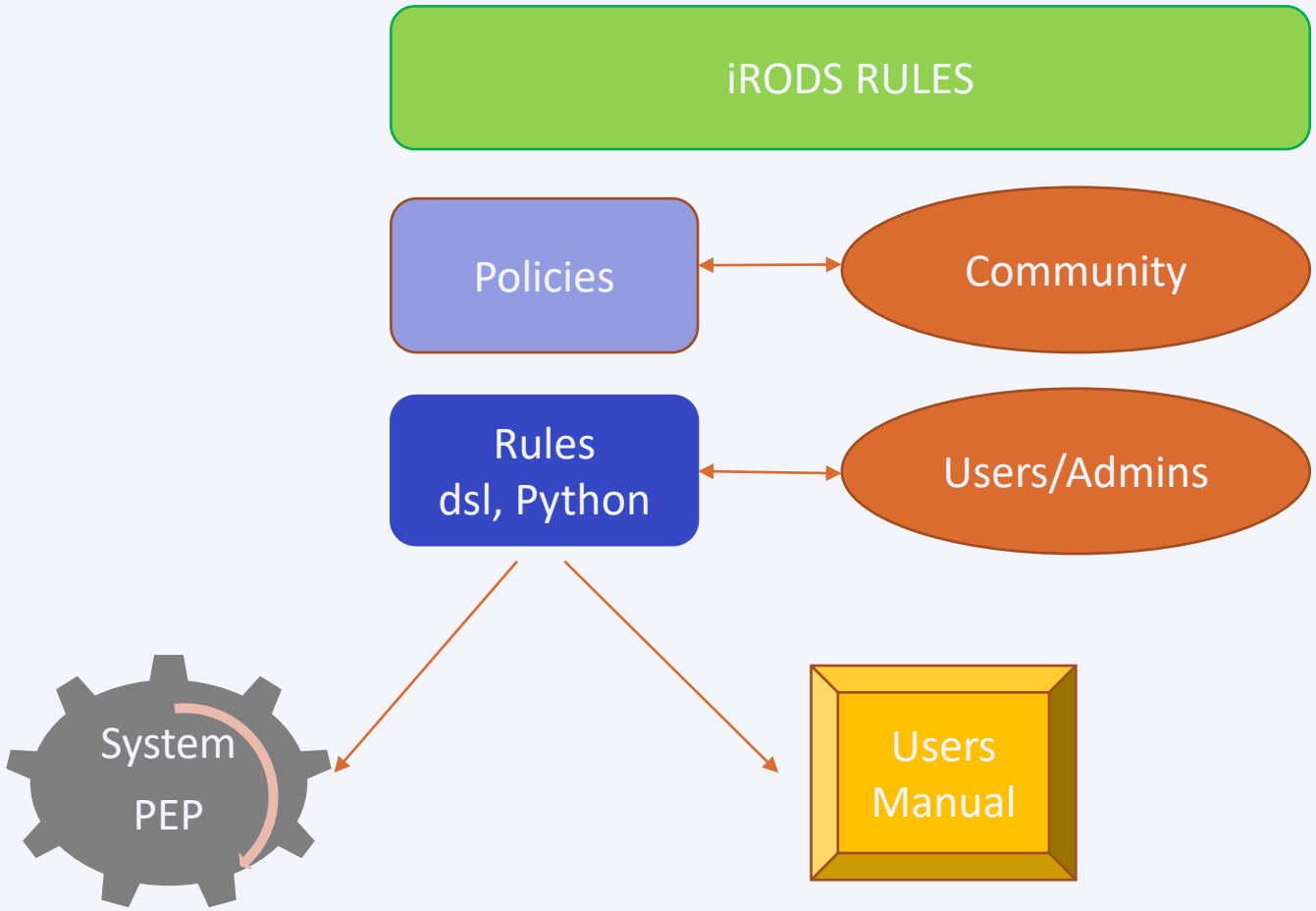
Functionalities

- Metalnx:
  - graphical user interface easiness
  - working with data objects/collections
  - adding metadata to data objects/collections
  - downloading data objects
  - permission
  - iRODS design



Functionalities

- irule:
  - user level rules
  - written into a local file
  - execute it when you need



Functionalities

- Other clients (not mentioned in video):

- webDav
- Cyberduck
- WinSCP

Documentation and support

- Documentation

https://docs.vscentrum.be/en/data_m/

- Support

  data@vscentrum.be

Stay Connected to VSC

LinkedIn®