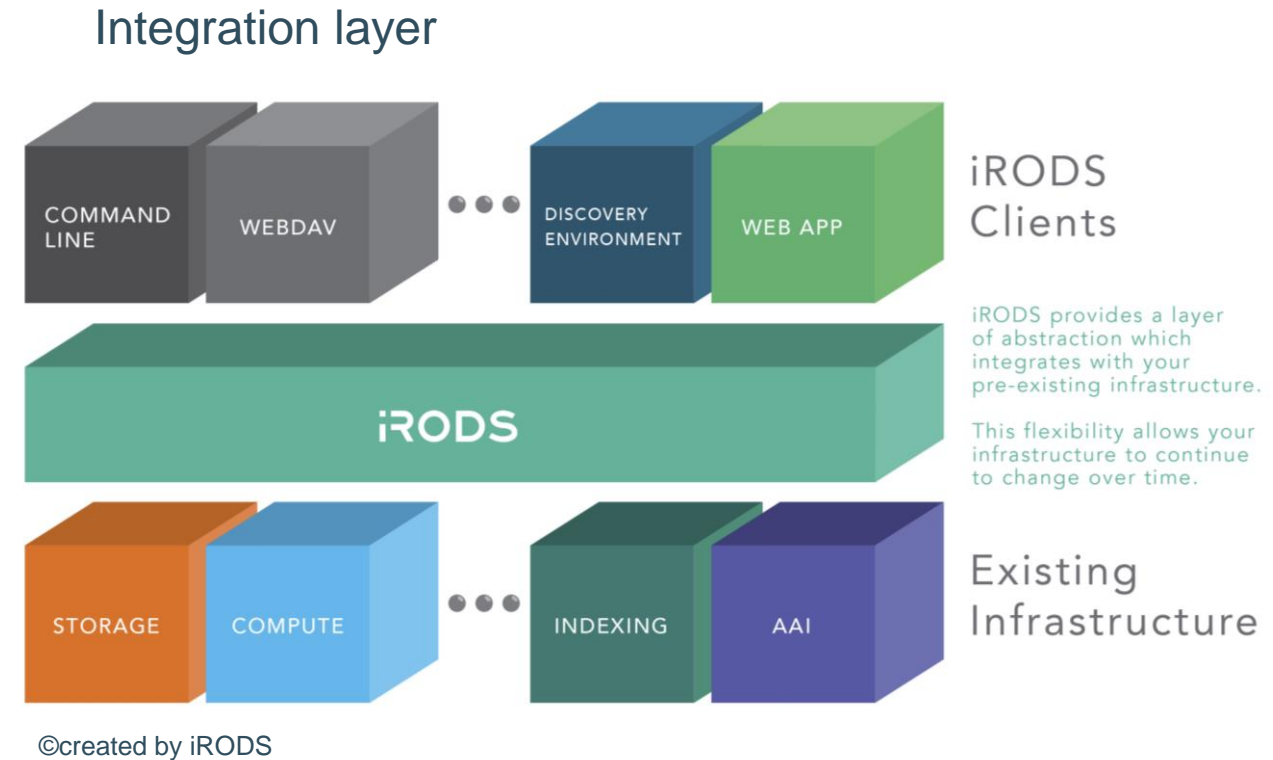


02.Introduction to iRODS



What is iRODS?

- iRODS (integrated Rule-Oriented Data System)
- Open Source distributed data and storage management system
- Configurable data management policies and workflows
- Scalable
- A flexible framework for the abstraction of infrastructure



KU Leuven is part of the iRODS consortium – Sustained member

iRODS architecture



Clients

Provides access to iRODS



Catalogue Service Consumer – Storage server

Provides access to storage and other resources



Catalogue Service Provider – iRODS server

Provides access to the Catalogue



Metadata catalogue (iCAT)
Postgres/Mysql/Oracle

Where everything is written down

iRODS Core competencies



Unified Storage Namespace

Data virtualization of distributed storage systems



Automation

Rule Engine to enforce data policies



Data Discovery

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

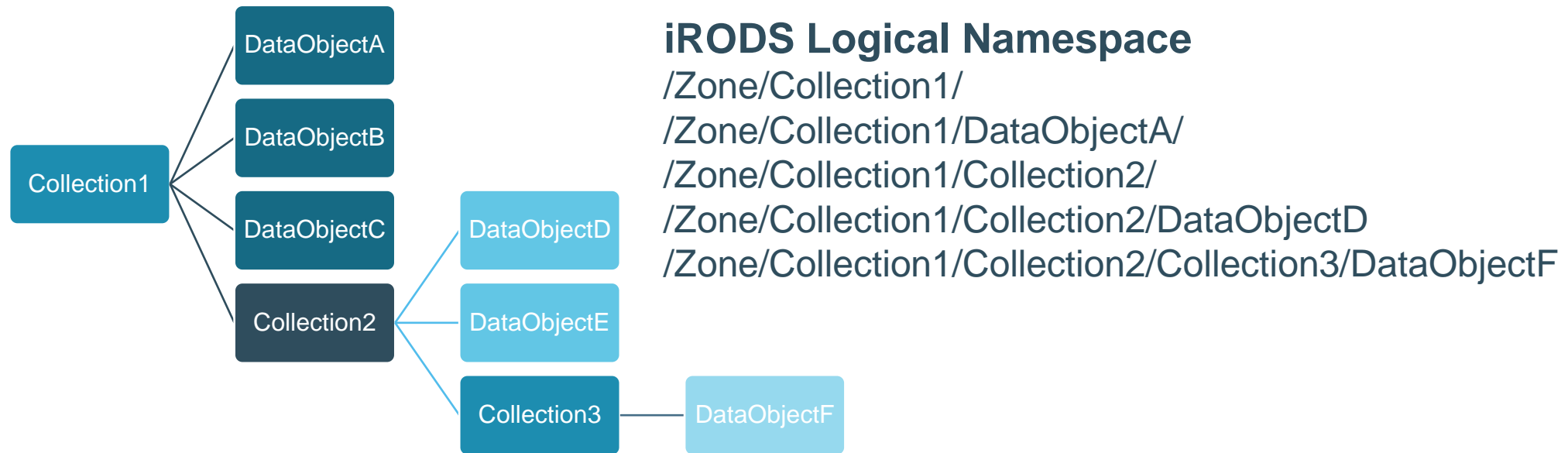


Secure collaboration

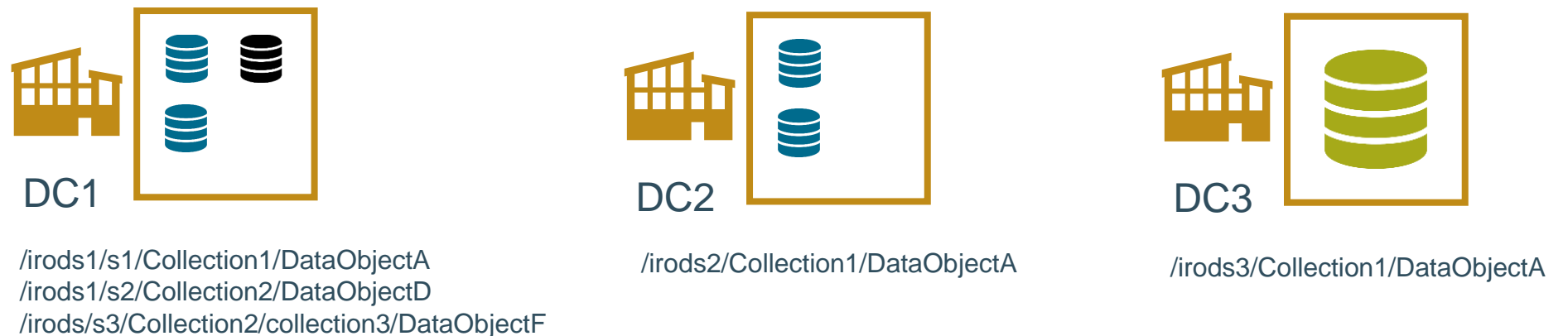
Three mechanisms: Permissions, Tickets and Federation

Data virtualization in iRODS

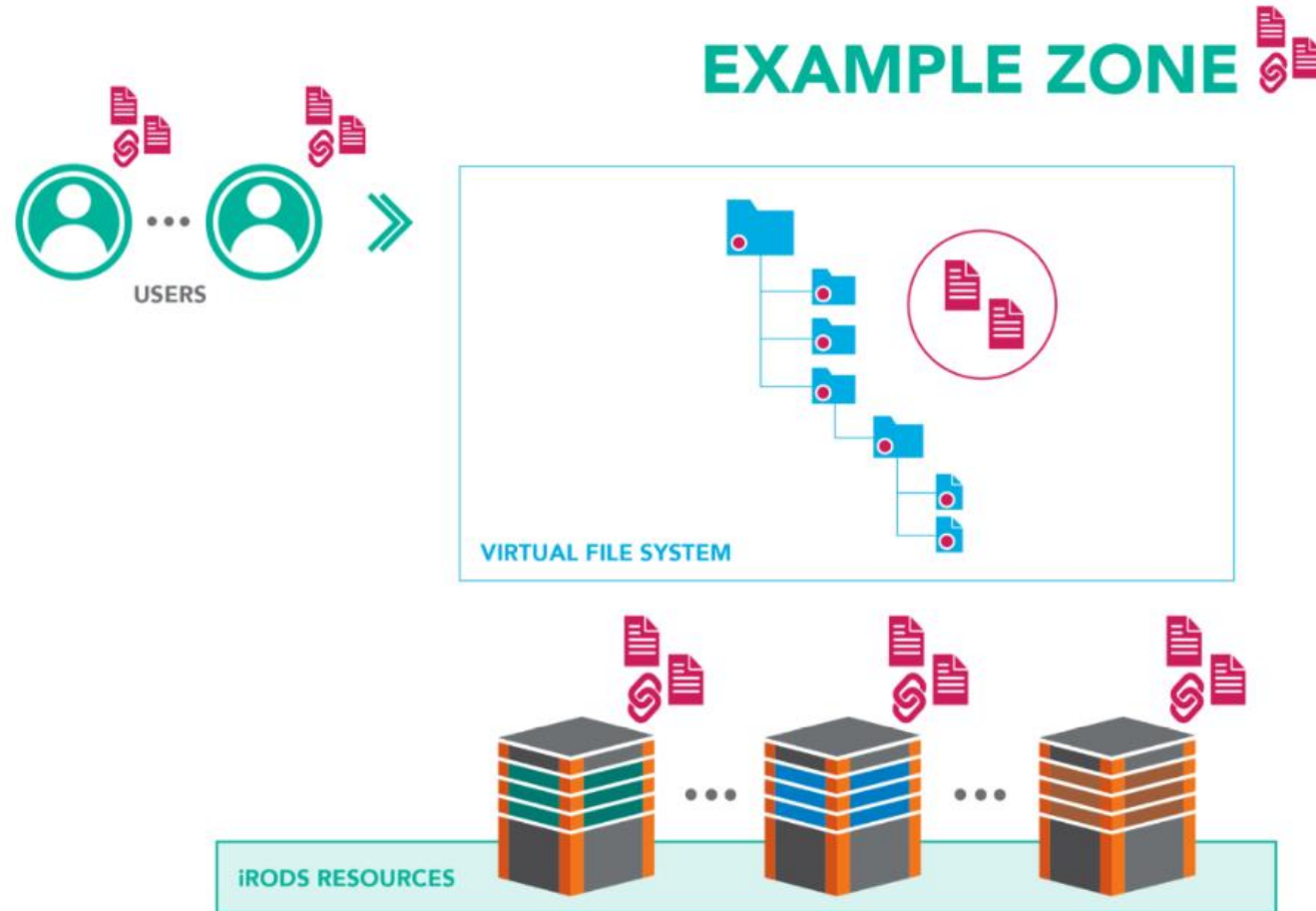
Logical representation



Physical representation



Data discovery: Metadata everywhere



©created by iRODS

System Metadata:

- filename, file size, creation date ...

User Metadata:

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

Automation

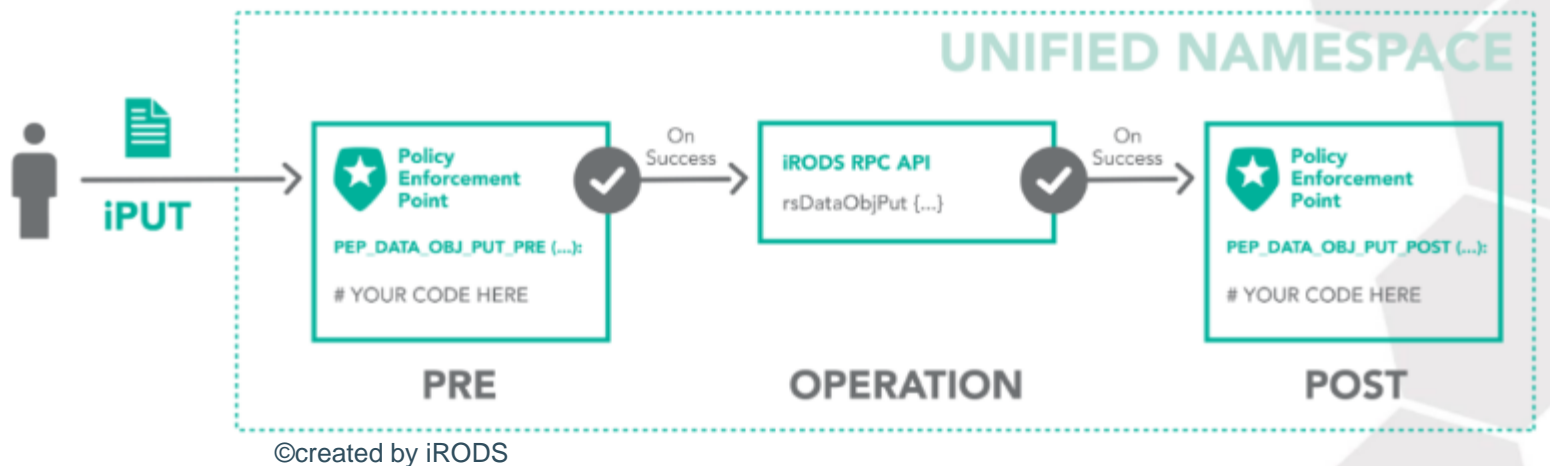
Integrated policy engine can be triggered by any operation:

- Authentication
- Storage Access
- Database Interaction
- Extensible RPC API

Delay rules to do repetitive works - periodically:

- execution time
- execution frequency

Dynamic Policy enforcements



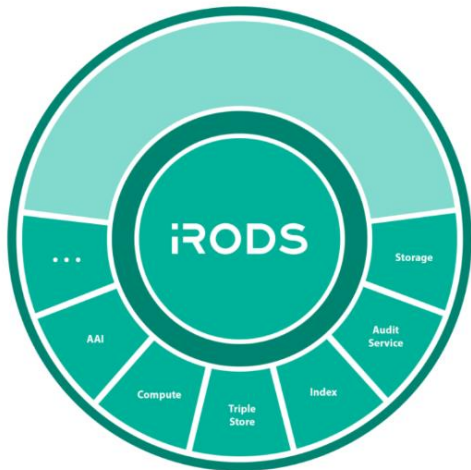
The iRODS rule may:

- workflow automation – event based
- restrict access
- log for audit and reporting
- provide additional context
- send a notification
- execute a process on the file

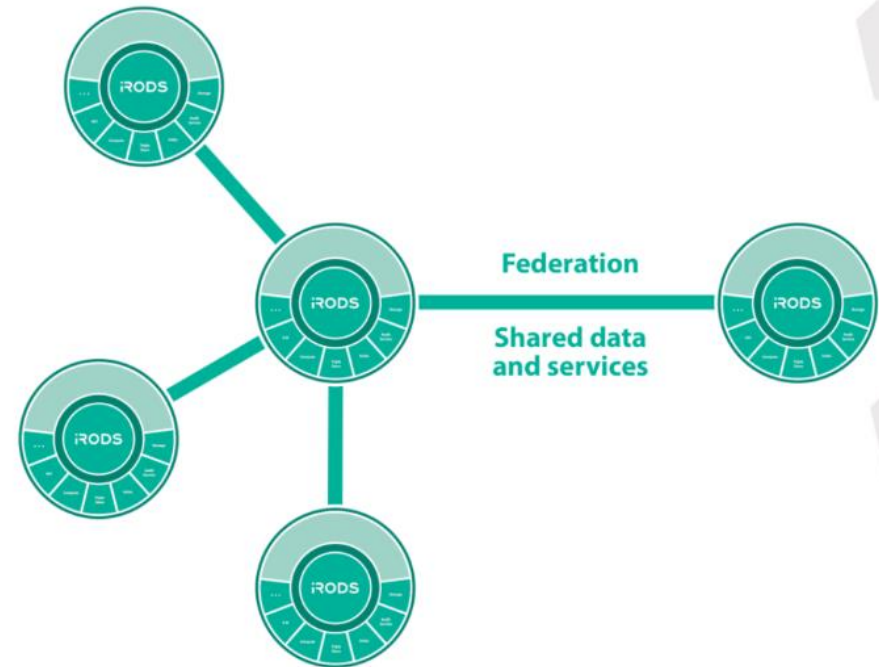
Secure collaboration

Inside a zone

- ACL (users, groups)
- Tickets:
 - Temporary access
 - No iRODS account needed



Between zones: federation

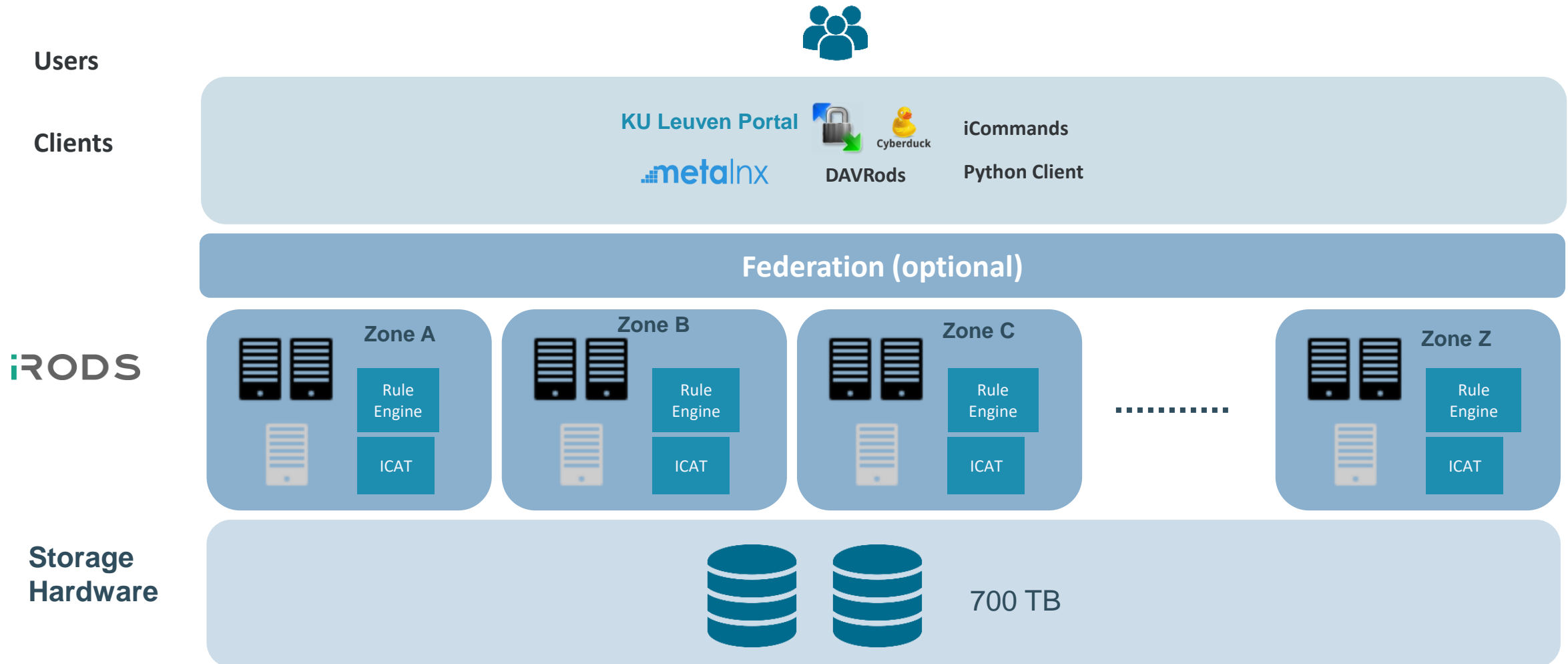


Secure collaboration: ACL's

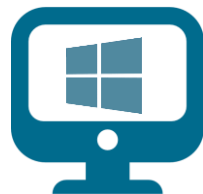
Access level	Possible dataobject operations
read	Download, preview (metalnx), view metadata, view permissions
write	Upload, overwrite, edit (metalnx), edit metadata
own	Edit permissions, remove file

Remember: the user should also have read access to the parent collection

KU Leuven iRODS architecture



Clients



 metalnx

KU Leuven Portal



DAVRods

PRC-Python API



Cadaver client

DAVRods

iCommands

CentOS7
Ubuntu16
Ubuntu18

PRC-Python API

 metalnx

KU Leuven Portal

References:

KU Leuven portal: <https://{yourZone}.irods.icts.kuleuven.be>

Metalnx: <https://{yourZone}.icts.kuleuven.be/metalnx/>

Cyberduck: <https://cyberduck.io/>

WinSCP: <https://winscp.net/eng/download.php>

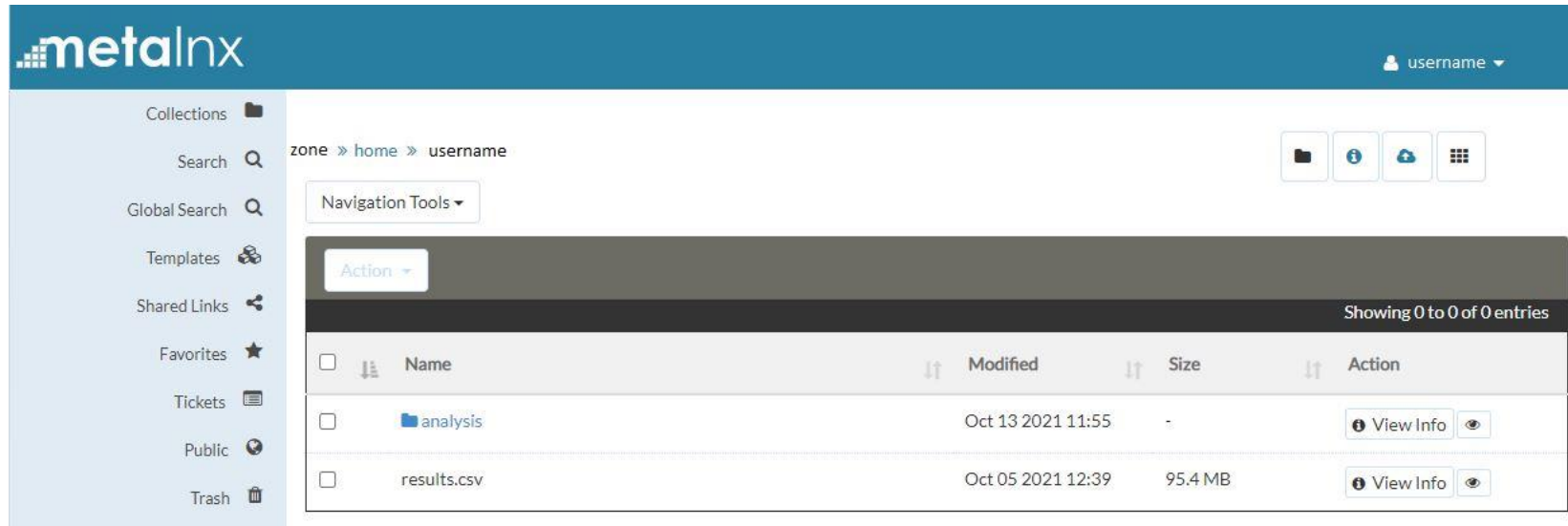
Cadaver client: <http://www.webdav.org/cadaver/>

Icommands: <https://irods.org/download/>

Python iRODS Client (PRC) : <https://github.com/irods/python-irodsclient>

Metalnx

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

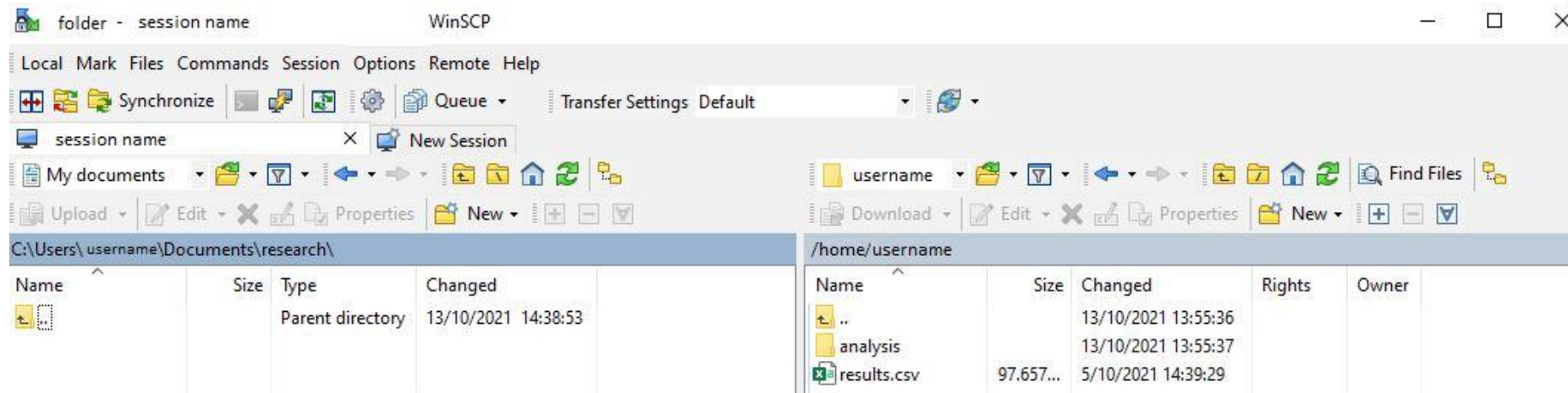


The screenshot displays the Metalnx web interface. The top header is blue with the 'metalnx' logo on the left and a user profile 'username' on the right. A left sidebar contains navigation links: Collections, Search, Global Search, Templates, Shared Links, Favorites, Tickets, Public, and Trash. The main content area shows a breadcrumb path 'zone » home » username' and a 'Navigation Tools' dropdown. Below this is a table with columns: Name, Modified, Size, and Action. The table contains two entries: 'analysis' (a folder) and 'results.csv' (a file). Both entries have 'View Info' and 'View' icons. The status bar at the bottom of the table indicates 'Showing 0 to 0 of 0 entries'.

<input type="checkbox"/>	Name	Modified	Size	Action
<input type="checkbox"/>	analysis	Oct 13 2021 11:55	-	View Info View
<input type="checkbox"/>	results.csv	Oct 05 2021 12:39	95.4 MB	View Info View

WebDav client

- graphical user interface easiness
- working with data objects/collections
- downloading data objects
- read and edit easily
- iRODS design



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

```
$ iput results2.csv
$ iget results.csv
$ ils
/zone/home/username:
  results.csv
  results2.csv
C- /zone/home/username/analysis
$ icd analysis
$
```

Python iRODS Client (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

```
import os
import ssl
from irods.session import iRODSSession

try:
    env_file = os.environ['IRODS_ENVIRONMENT_FILE']
except KeyError:
    env_file = os.path.expanduser('~/.irods/irods_environment.json')

ssl_context = ssl.create_default_context(purpose=ssl.Purpose.SERVER_AUTH, cafile=None, capath=None, cadata=None)
ssl_settings = {'ssl_context': ssl_context}
with iRODSSession(irods_env_file=env_file, **ssl_settings) as session:
    collection = session.collections.get("/path/to/collection")
    for data_object in collection.data_objects:
        print(data_object.name)
```


Hands-on

Connecting to your zone



Connecting to iRODS

- Go to: <https://{yourZone}.irods.icts.kuleuven.be>
- Authenticate using your u-account:
 - You will be sent to the KU Leuven login page
- You are in!

Login portal

[HOME](#) [BROWSER](#) [METALNX](#) [GROUPS](#) [ADMIN](#)

IRODS ZONE

You can connect to the irods zone by using the following information:

Hostname

Port

Zone

Username

Password

Hands-on

- Demo of the functionalities of the Login Portal
- Explore the demonstrated functionalities
 - Navigate to your home
 - Create a new folder
 - Upload a file
 - Download the file
 - Add a metadata attribute value to the file
 - Look at the metadata
 - Delete the file