# Agenda

- Introduction to iRODS (10m)
- VSC Tier-1 Data Use cases (15m)
- Introduction to the KU Leuven iRODS testbed infrastructure (10m)
- Connecting to the KU Leuven testbed (10m)
- Working with iRODS (Basic)
  - Clients:
    - GUI: Metalnx (20m)
    - WebDav (15m)

**BREAK -> People not interested on CLI/Python and advanced features can stop here**

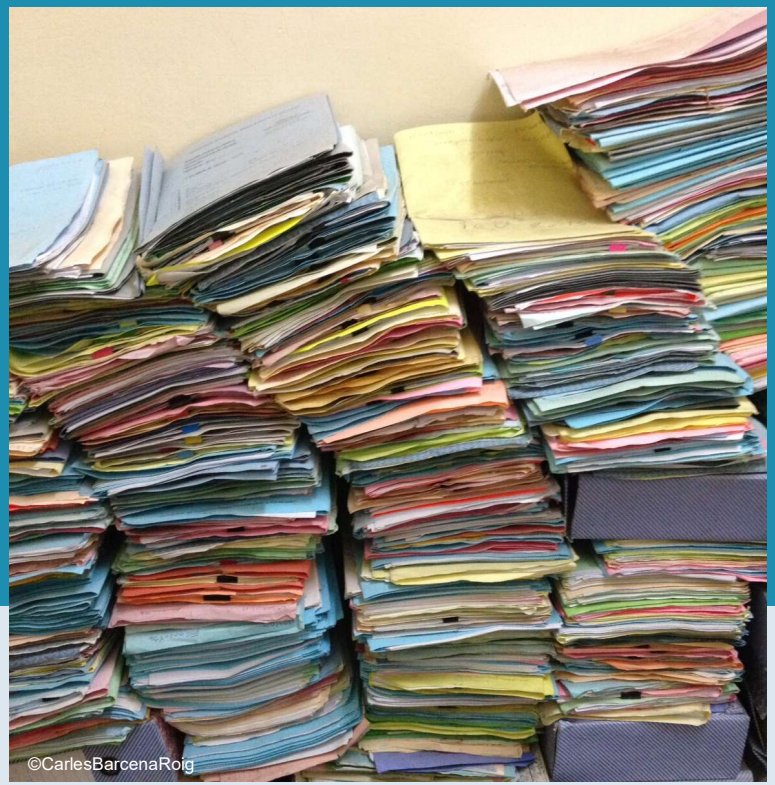
  - Command line: icommands (20m)
  - Python iRODS client (PRC) (20m)
- Working with iRODS (Advanced):
  - Task automation:
    - Rules (30m)
  - Other capabilities – not yet implemented (15m)
    - landing zone
    - auditing
    - provenance

# Training material

- All training material is available on github
  - https://github.com/hpcleuven/iRODS-local-admin-training

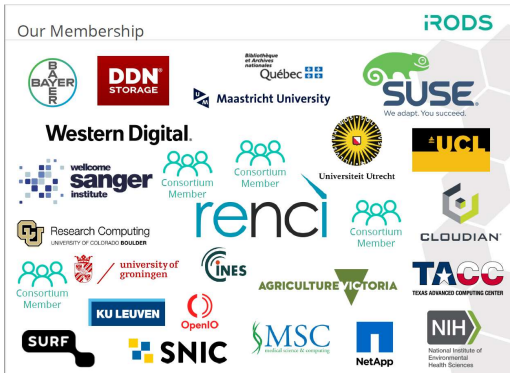
KU LEUVEN

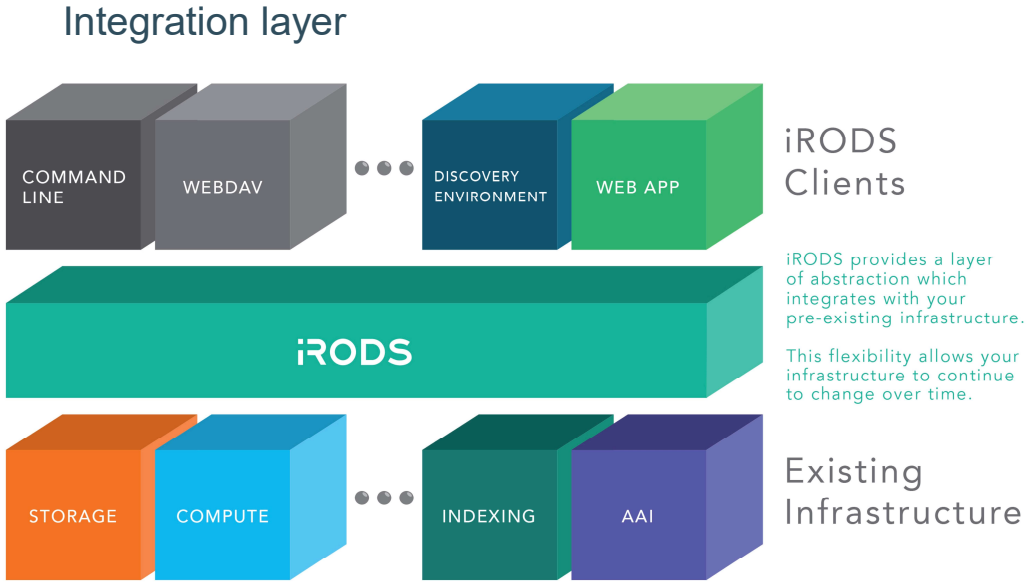
# 01.Introduction to iRODS



# 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

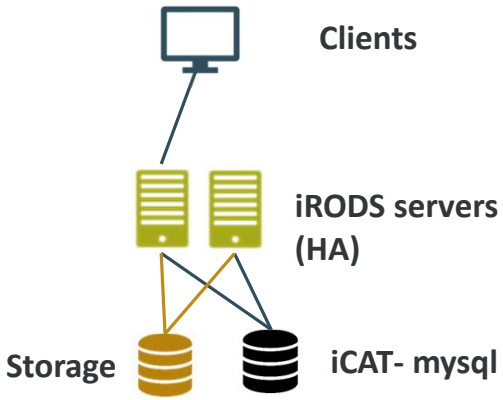
**Catalogue Service Provider – iRODS server**

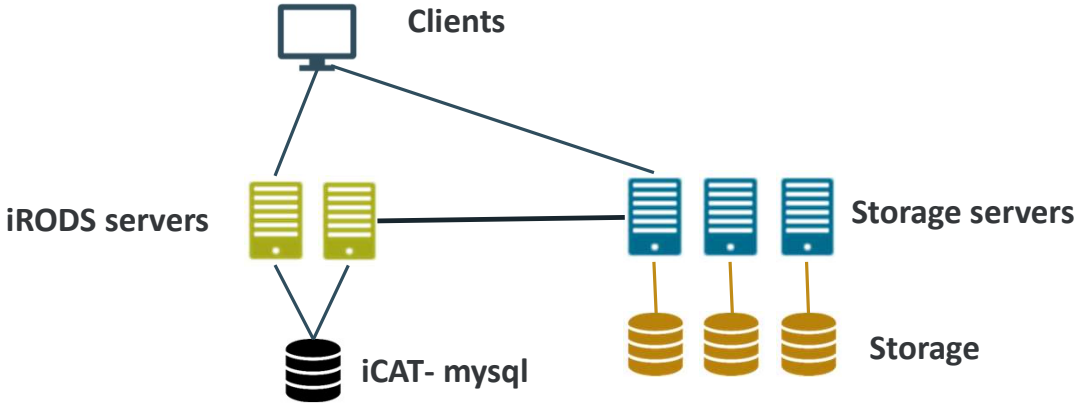
Provides access to the Catalogue

**Metadata catalogue (iCAT)**
Postgres/Mysql/Oracle

Where everything is written down

# iRODS architecture: examples



# iRODS Core competencies

## Unified Storage Namespace

Data virtualization of distributed storage systems

## Automation

Rule Engine to enforce data policies

## Data Discovery

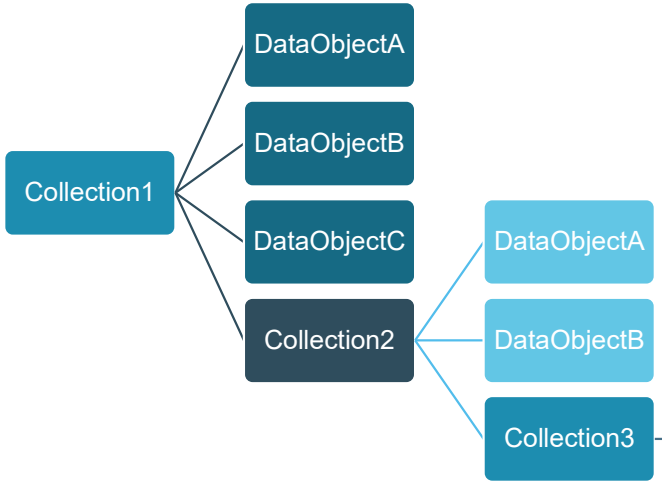
Rich Metadata for collections and data objects
(System metadata and user-defined metadata)

## Secure collaboration

Three mechanisms: Permissions, Tickets and Federation

# Data virtualization in iRODS

**Logical representation**




**iRODS Logical Namespace**

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

**Physical representation**

DC1

/irods1/s1/Collection1/DataObjectA
/irods1/s2/Collection2/DataObjectA
/irods/s3/Collection2/collection3/DataObjectA

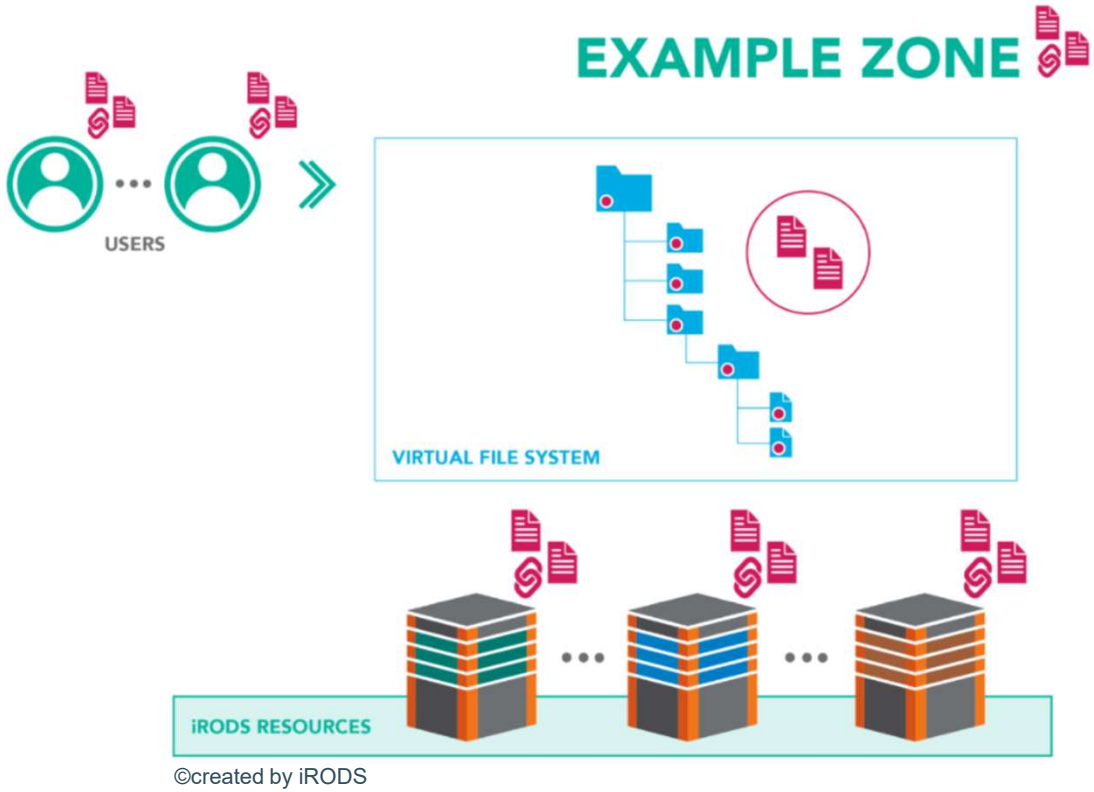
DC2

/irods2/Collection1/DataObjectA

DC3

/irods3/Collection1/DataObjectA

# Data discovery: Metadata everywhere



©created by iRODS

System Metadata:
- filename, file size, creation date …

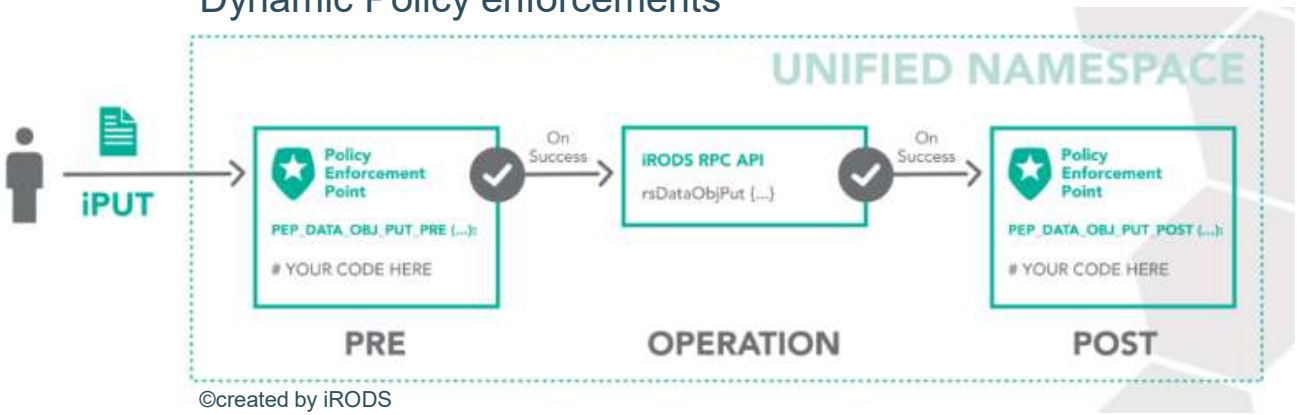
User Metadata:
- Manual introduction
- Metadata templates
- Automation (rules/microservices)

# Automation

Integrated policy engine can be triggered by any operation:

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

Dynamic Policy enforcements




©created by iRODS

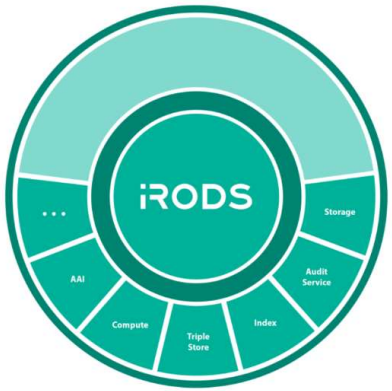
The iRODS rule may:

- restrict access
- log for audit and reporting
- provide additional context
- send a notification
- Execute a process on the file

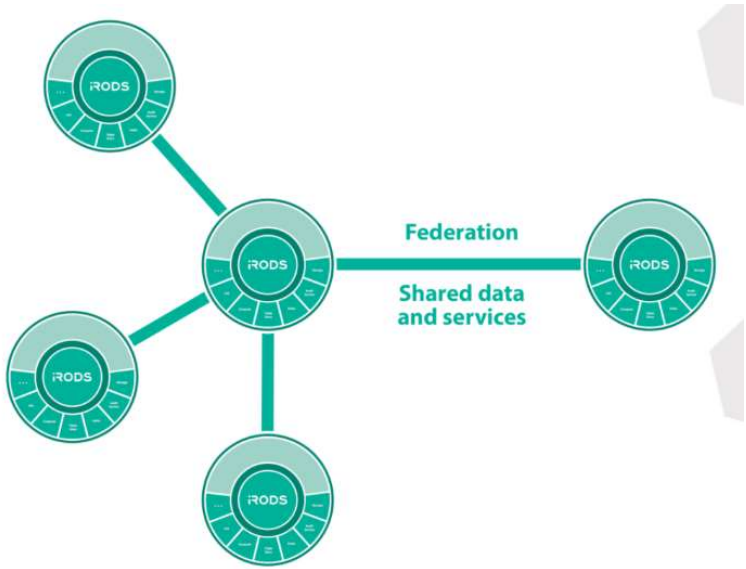
# Secure collaboration

Inside a zone

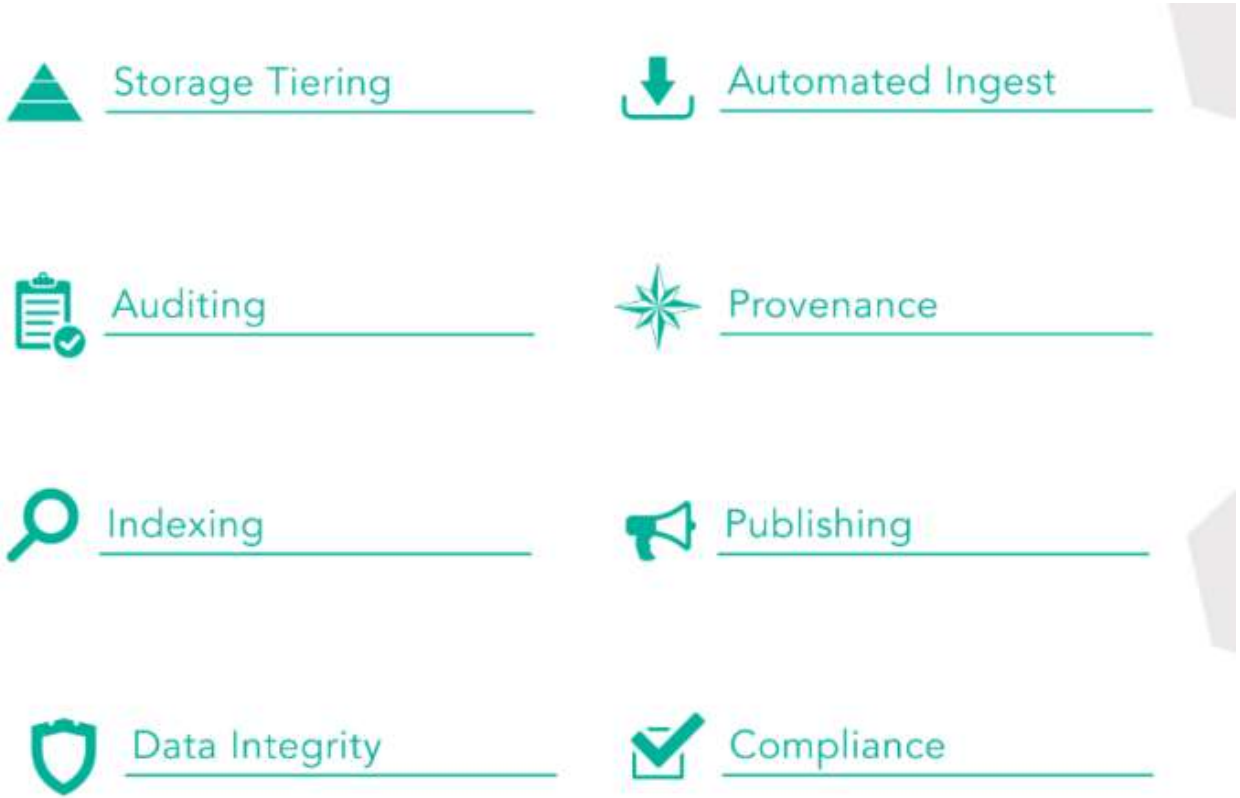
- ACL (users, groups)
- Tickets:
  - Temporary access
  - No iRODS account needed




Between zones: federation



# iRODS capabilities




©created by iRODS