



Vlaanderen  
is supercomputing

# A year of iRODS: Lessons Learned

Ingrid Barcena, KU Leuven

iRODS User Group Meeting, June 9 2021

VLAAMS  
SUPERCOMPUTER  
CENTRUM

*Innovative Computing  
for A Smarter Flanders*

[vscentrum.be](https://vscentrum.be)

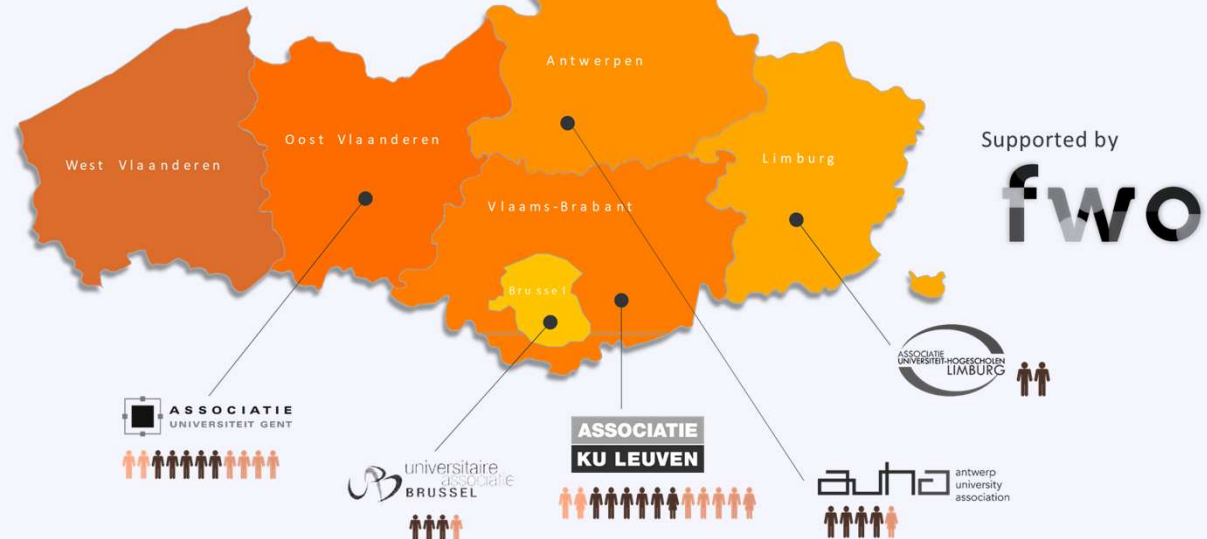
# KU Leuven



- Founded in 1425
- Belgium's largest university
- 15 faculties in 10 cities
- ~21000 employees and ~60,000 students
- ~7600 researchers and ~6,200 PhD students
- #1 Reuters Ranking of Most Innovative Universities in Europe 4 years
- #45 in the Times Higher Education World University Ranking (2020)
- #84 in the QS World University Ranking (2021)

[vscentrum.be](https://vscentrum.be)

## VSC PARTNERSHIP

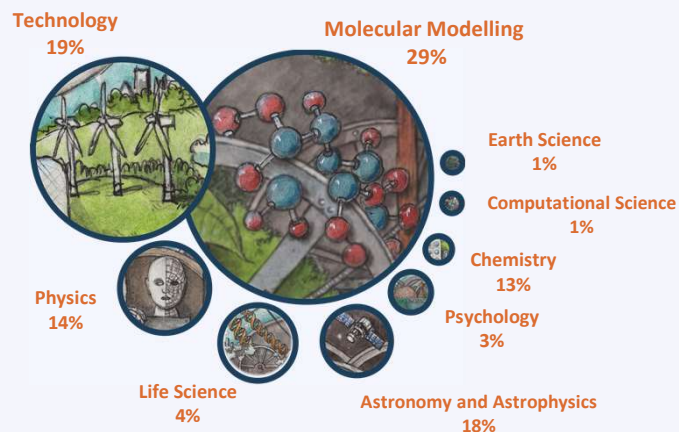


VLAAMS  
SUPERCOMPUTER  
CENTRUM

# The VSC HPC community

**Diversity!!!**

200 users  
80 Projects  
300,000 jobs



Scientific domains

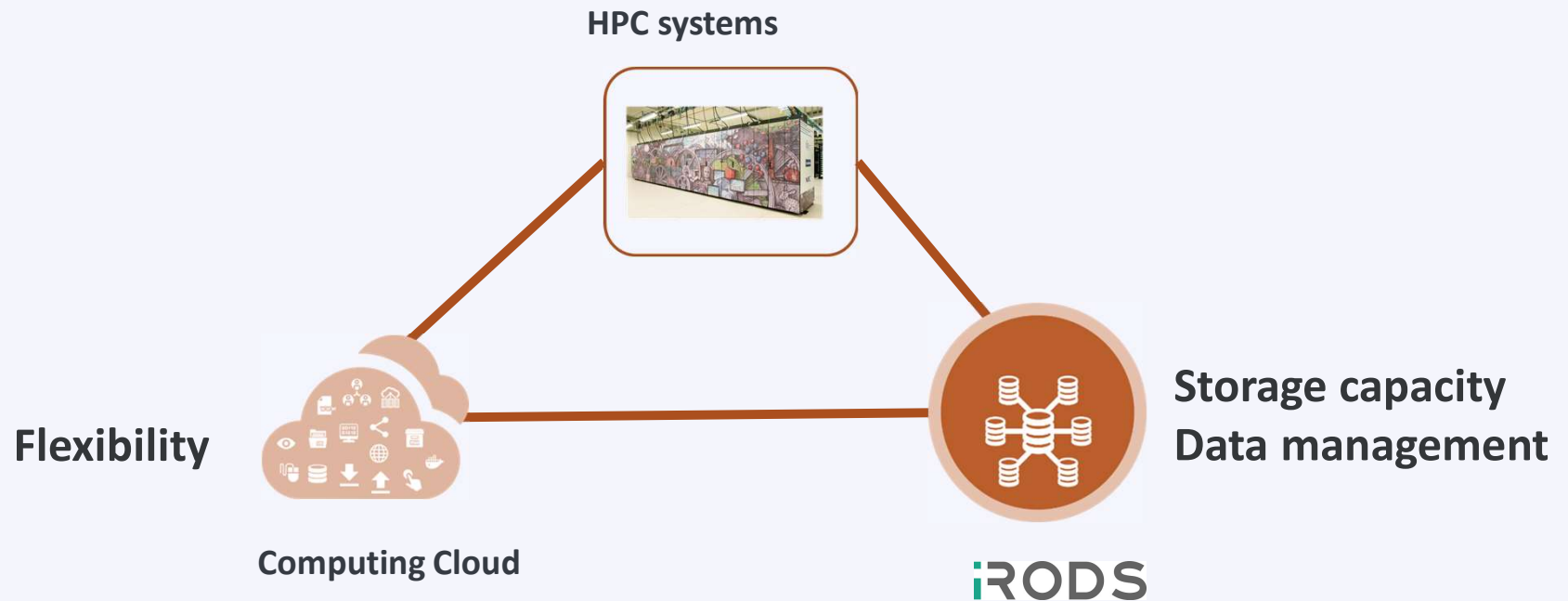


Data types



Data sizes

# How iRODS fits in here .....



# Why an (active) data repository?

**Help and encourage users to apply good data management practices**



**Give context to data**



**Search data**



**Automate data workflows**



**Share data**



**Secure data**



**Publish**

# Pilot users



**Storage4climate**  
(Climate change research)




**CLARIAH-VL**  
(History research)



**EXIMIOUS**  
(Human Exposome)

**VIB (Biotechnology) – usegalaxy.be**

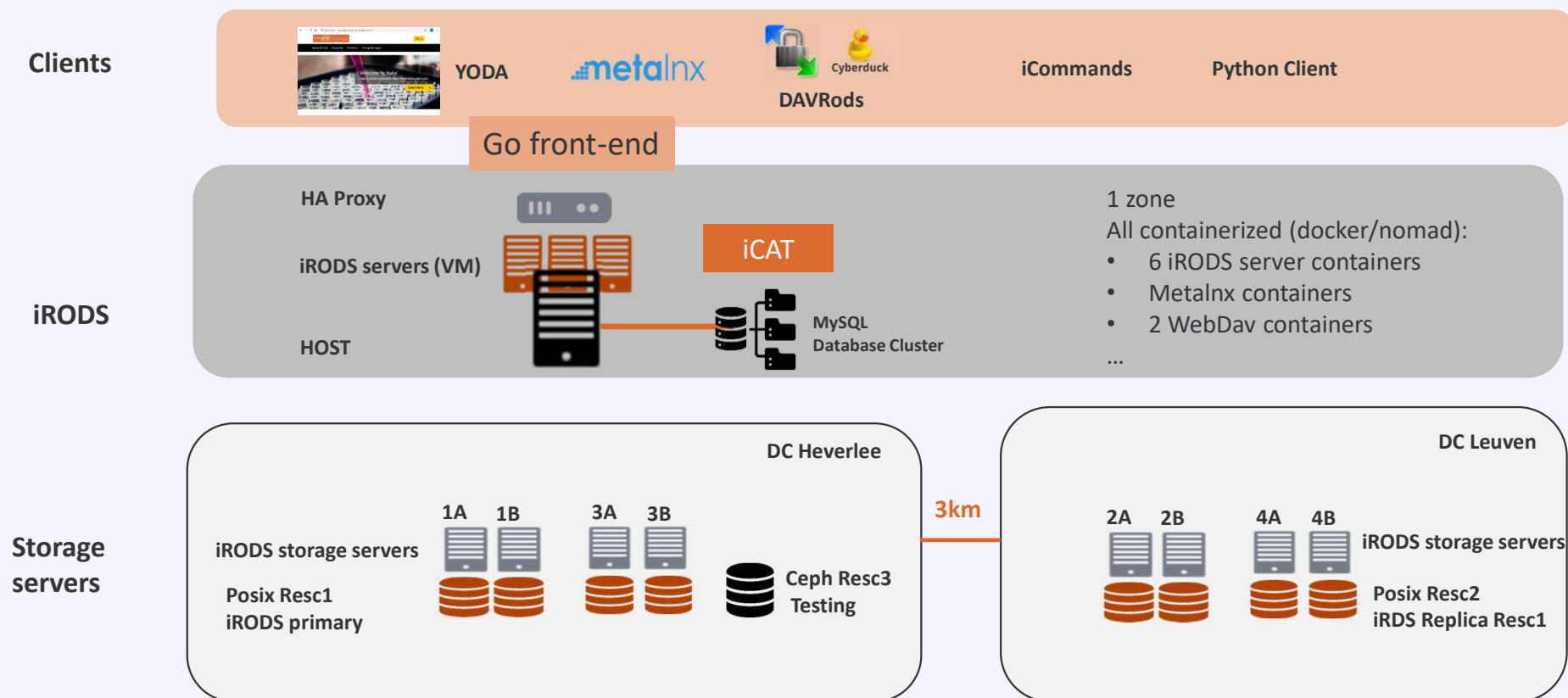
**VIB-KU Leuven (ASAP project)** 



**FLANDERS MAKE**  
(3D printing)



# System architecture (pilot environment)





# 1 year in numbers

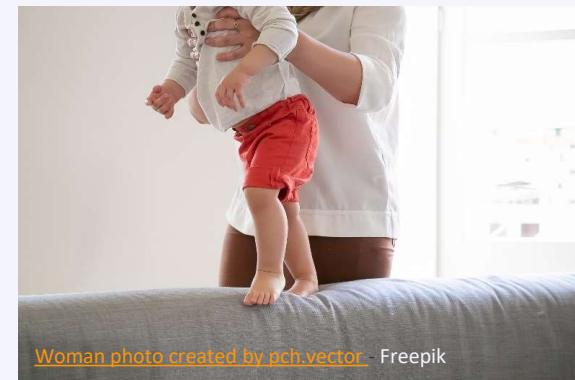
- 50 registered users (20 active)
- 475 TB uploaded
- 55,000 collections
- 6+ million files
- 200 unique metadata attributes
- 200,000 metadata values
- Average file size 180MB
- 3 times data migrated between storage devices
- 300 files lost





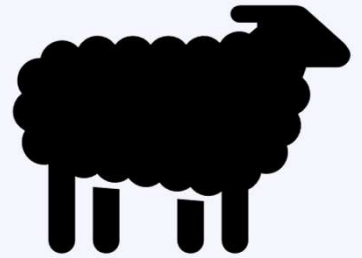
# Start is hard

- First step is easy (demo setup)
- But ...
  - move from demo to real life is more complicated
  - Learning curve (finding the optimal iRODS way of working)
  - Documentation is not always clear (or is not enough)
  - Need to figure out when and how to create rules inside iRODS to support domain specific use cases



# MySQL or not MySQL?

- It is possible (we do!)
- No big performance differences noticed so far (more tests needed)
- But ...
  - MySQL iCAT initial installation needs patches (UTF-8 encoding, ...)
  - Metalnx search does not work
    - Missing MySQL implementation in upstream dependency/repo (<https://github.com/irods-contrib/metalnx-web/issues/174>)
    - See: <https://github.com/DICE-UNC/jargon-irods-ext/blob/4a20477a4a4dcb3eb59cfcf73b7ebdcdbd584d2/emc-metalnx-services/src/main/java/com/emc/metalnx/services/irods/utils/SpecificQueryProviderFactoryImpl.java#L44-L53>
  - Issues with atomic metadata
    - Inconsistent use of low level drivers, MySQL conflict (<https://github.com/irods/irods/issues/4917>)
  - Database schema uses case insensitive VARCHAR columns by default, where the PostgreSQL schema is case sensitive (breaks imeta upper qu ... for example)
  - MySQL transactions behaves different that PostgreSQL transactions



# The storage dilemma

- Which storage type is best? Which is the best configuration/storage hierarchy?

We have been playing around: xfs, NFS, S3 (Ceph, NetApp Storage Grid) , BeeGFS, GPFS

- So far ...
  - NFS the most stable and performant
  - S3 was unstable and less performant (TBD: test new S3 plugin and Ceph NFS gateway)
  - BeeGFS was less performant
    - ibun did not work – hard links not supported
  - GPFS stable and similar in performance as NFS
  - Replica management is iRODS not yet robust enough (logical lock might help)
  - We have seen memory leaks issues in some storage systems (not yet known why)

# iRODS user experience

- CLI/Python works for most HPC users (advanced users)
  - We created user guides and training (incl. hands-on and examples)
  - We developed a PRC extension:
    - to allow to use wildcards
    - Add job information as metadata
- But also needed ...
  - Other direct access to data (NFSRODS, WebDav (Davrods), irodsfs)
  - A user friendly GUI

# The GUI wars

## YODA

- User friendly
- Implements the whole data life cycle (ingest to publication) -> RDM tool
- Implement full control ACLs and data access
- Implements a rich set of policies (for specific workflows)
- Metadata at collection level and configurable templates (based on DataCite)



## Metalnx ...

- User friendly (but sometimes too much options)
- Direct interface to iRODS (no policies)
- Metadata at collection and files level (simple templates possible)
- Search function (so far not for us)



# The GUI wars

Although Yoda and Metalnx works well for some use cases

- They do not coexist nicely
- YODA shadow groups broke some Metalnx functions

So ...

- We will stop using YODA
- We will still keep using Metalnx (and try to fix the problems)
- In parallel:
  - We are doing a PoC for an own developed portal
  - We have developed a light front-end (based on the Go Client):
    - Custom authentication based on oauth (while waiting for interactive PAM)
    - Users and group management features
    - <https://github.com/cyverse/go-irodsclient>



[Gunner111](#) (CC BY-NC-ND 2.0)



[Alvin Trusty](#) (CC BY-NC-ND 2.0)

# Yet another iRODS portal?

Rationale ....

- Easy to use, domain specific web based interface(s)
- Support a wide range of use cases
- Mix (external) business logic and iRODS rules
- Modular approach, (re-usable) building blocks
- Centrally supported (ICT services, us 😎), possibly extended by end users
- Leverage search engines for discovery, analysis and query scaling



# PoC: technology choices

- Python with iRODS Python Client
  - Extended with a home developed higher level API
  - Leverage the many available scientific libraries when appropriate
- Flask (lightweight) web application framework
  - (nested) Blueprints (modularity)
  - Flask-WTF (forms)
  - Waitress/Gunicorn (HTTP server)
- Client side: Vue.js javascript framework
  - Standardised @ KULEuven web teams
- Search engine: OpenSearch (Elasticsearch)
  - Apache Solr as a future option
- Meta-data templates
  - TBD (Yoda/JSON, TWG initiative, RUG, our own ..)

# Conclusion

- iRODS is a flexible and powerful tool for data management
- The learning curve is high but the results make it very valuable
- Some additional documentation and how-to could help to find the way to beginners
  - Community curated wiki on best practices, links and more ...
- Some components need to be more tested against other DB than PostgreSQL
  - we will try to contribute to it
- And there is a great, dynamic and fantastic community!
  - Thank you for the help and support!

# Next steps



iRODS



iRODS



- Finalize design and acquire the production storage infrastructure (min. 6-8PB)
- iRODS features to explore, test, (deploy)
  - Federation
  - NFSRODS
  - Landing zone/automatic ingestion
  - User portal including metadata standards/templates
  - Search engine
  - Globus
  - GDPR (access control, security, auditing, ...)

# The team (growing)

Paul  
Borgermans



Mustafa  
Dikmen



Jef  
Scheepers



Maxime  
Van den Bossche



Ingrid  
Barcena



User Support  
User tools  
Policies  
Documentation



Peter  
Verraedt

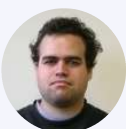


Jan  
Bongers

iRODS  
middleware



Tom  
Vanmierlo



Kristoff  
Van Buggenhout



Bart  
Vanneste

Storage



Thank you!