iRODS

Integration of iRODS data workflows in an extensible HTTP REST API framework

iRODS UGM 2019

Mattia D’Antonio

m.dantonio@cineca.it

26-27 th June 2019, Utrecht, The Netherlands

CINECA

Key points

- CINECA is involved in many European projects and National initiatives
- My group in particular is committed in Data Management
- Every project has is own very specific requirements but some common needs can be identified
- We are building a common layer among all these projects
- iRODS is the base data technology adopted onto these projects

Common projects requirements

EUDAT

B2STAGE

SeaDataCloud

Telethon

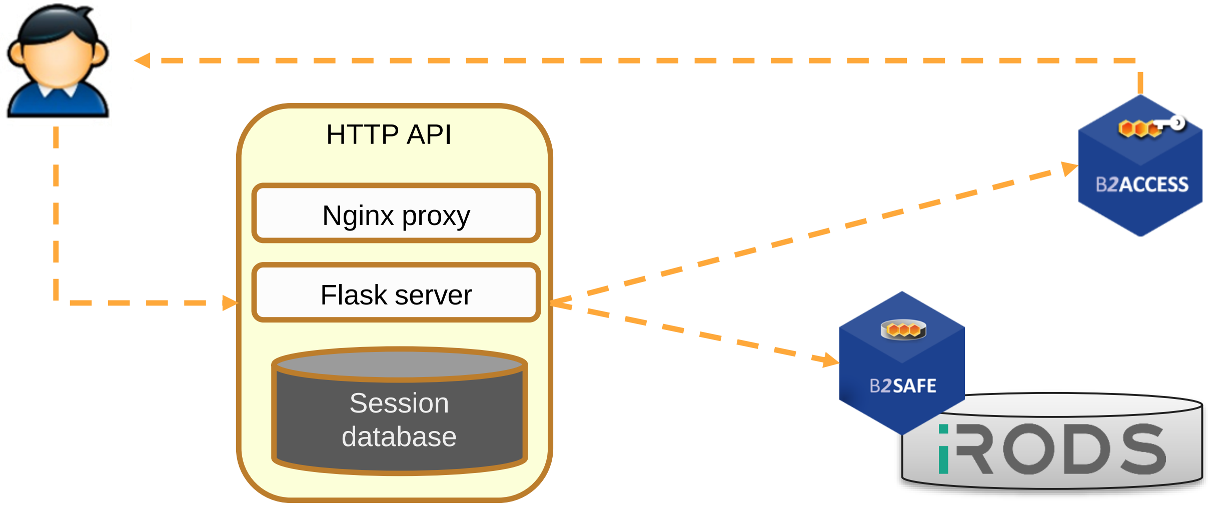
Società Italiana di Genetica Umana

EUDAT CDI

- EUDAT Collaborative Data Infrastructure (CDI) is a **network of nodes** that provide a range of services for data upload, retrieval, identification, replication. The **nodes** are essentially **data centers**
- EUDAT supports several services but I will focus on two core services:
  - B2SAFE – data and policy management service build over iRODS
  - B2STAGE – HTTP API interface for data transfer build over B2SAFE

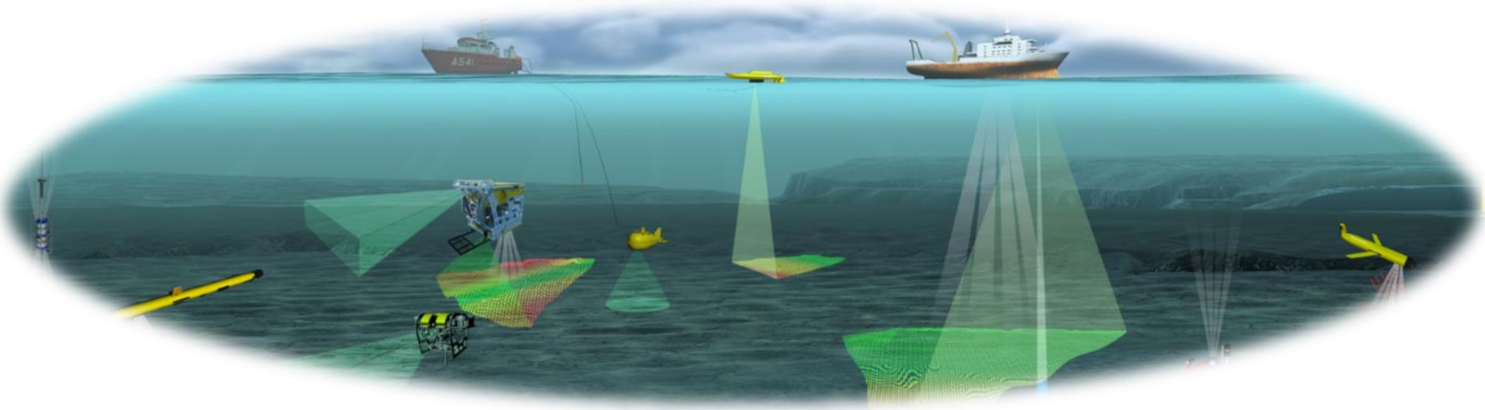
B2STAGE

- HTTP RESTful interface offering functionalities for data transfer between EUDAT resources (B2SAFE =~ iRODS) and external computational facilities

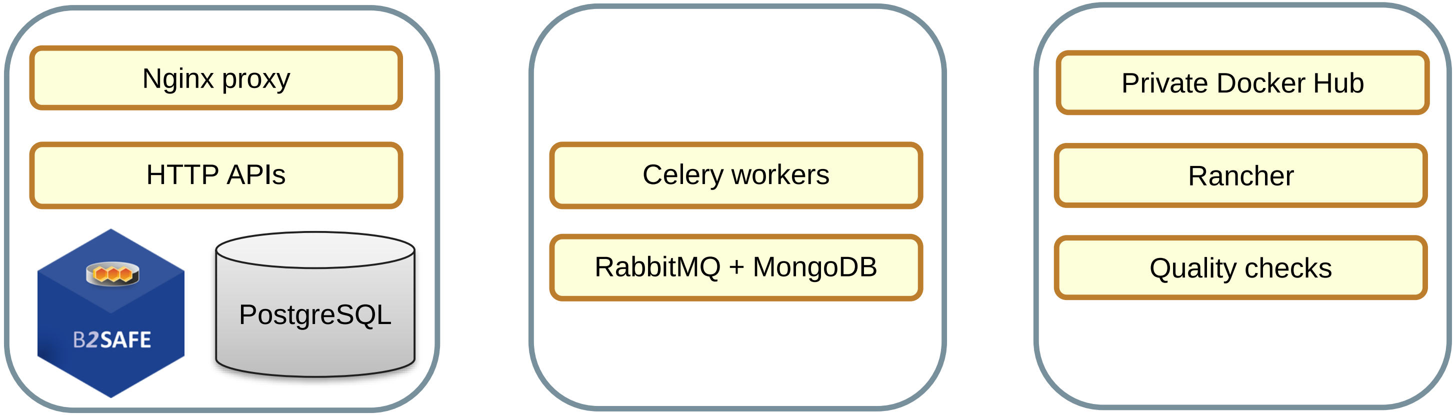


SeaDataCloud

- Pan-European infrastructure for ocean & marine data management
- Data from sensors, ships, platforms are stored in a centralized repository to be standardized, validated, indexed



SDC CDI HTTP API



Nginx proxy

HTTP APIs

Celery workers

RabbitMQ + MongoDB

Private Docker Hub

Rancher

Quality checks

Ingestion and ordering APIs are built on B2STAGE by adding custom endpoints

Heavy data management operations = asynchronous task (with Celery)

Execution of data workflows (as docker containers orchestrated through Rancher)

Genomic Repository Initiative

National initiative for the implementation of a Genomic Repository,
in collaboration with:

- **Telethon Foundation**
  - a non-profit organization for genetic diseases research
- **SIGU**
  - Italian Society for Human Genomics

Genomic Repository

A platform on which a researcher can:

- **Deposit** sequencing data
- Manage **metadata** and annotations
- Create **correlations** between datasets
- Perform **HPC analyses** on archived data to produce more information

Common requirements among the 3 use cases

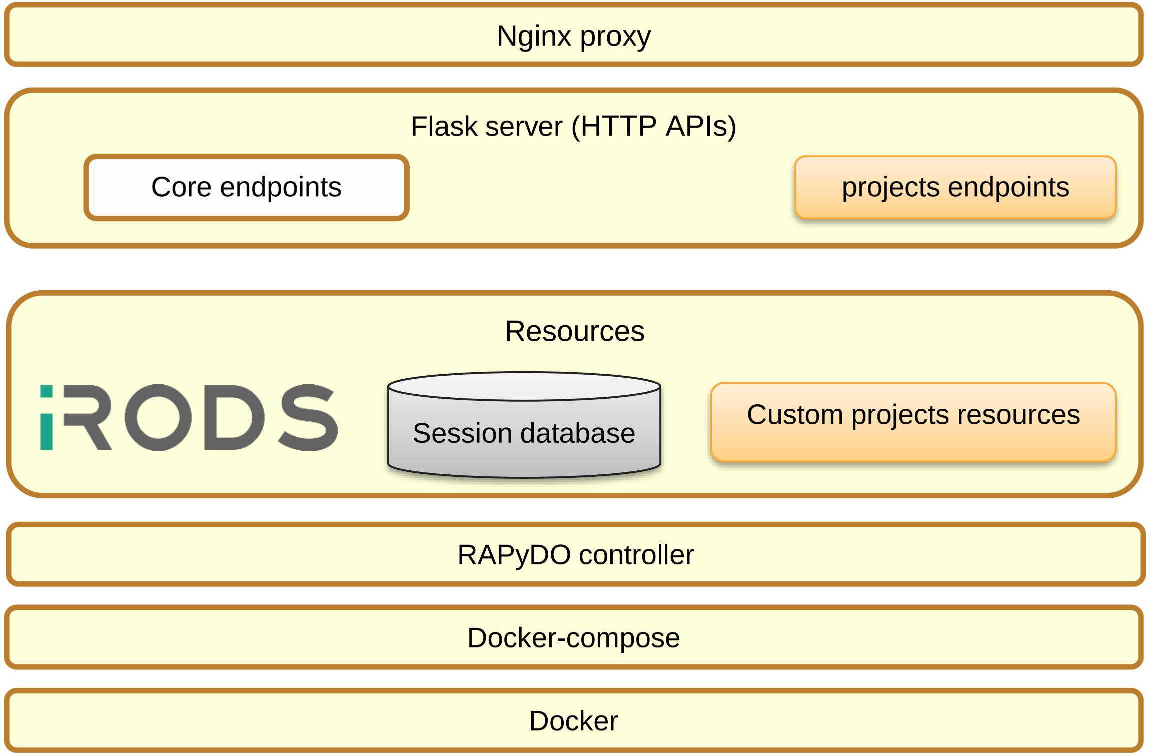
- Data storing ★★★
- Metadata management ★★
- Access via REST API ★★★
- Execution of asynchronous operations ★★
- Access from HPC cluster or other workflow manager ★★

We created a common framework (named RAPyDO) to share solutions among these projects

RAPyDO

- RAPyDO: Rest Apis with Python on Docker
- Implements a set of HTTP REST APIs (integrated with several services) to support users of different communities to implement data workflows and services
- APIs include the integration with iRODS
- Built as a wrapper of docker-compose for easy deployment on every platform
- RAPyDO is an extensible and modular framework used as a base for the projects

Architecture stack



iRODS integration

- HTTP APIs are written in **Python** by using the **Flask** framework
- A wrapper client based on the **python-irods-client** implements common operations
- The client is used from both API endpoints and celery tasks to easily interact with iRODS

```
def get(self, collection):
   if self.irods.exists(collection):
      return self.irods.list(
              collection, recursive=True, acl=True)
```

Implemented methods

- Methods mapped on icommands
    - e.g. list(), mkdir(), put(), get(), move(), remove(), set_permissions(), ticket(), etc
    - mapped on ils, imkdir, iput, iget, imv, irm, ichmod, iticket, etc
- Simple utilities methods without a corresponding icommand
    - e.g. exists(), is_collection(), is_dataobject() and others
- Method to perform more complex operations, e.g.
    - Methods to read and write file content as strings, chunks or Flask data streams

Authentication

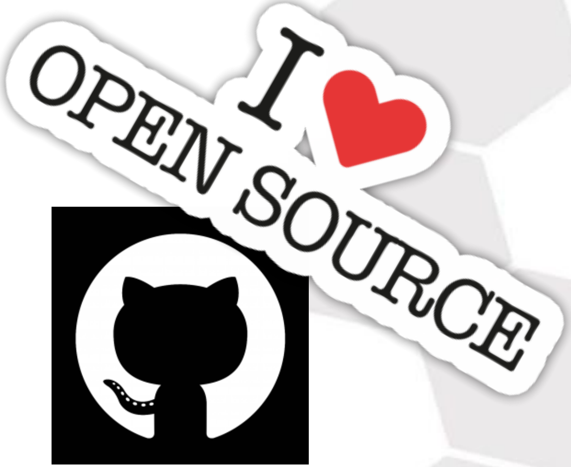
- HTTP APIs support all iRODS authentication protocols:
  - Native credentials
  - Pluggable authentication modules (PAM)
  - Grid Security Infrastructure (GSI)

Native credentials are natively supported by python-irods-client

PAM and GSI modules

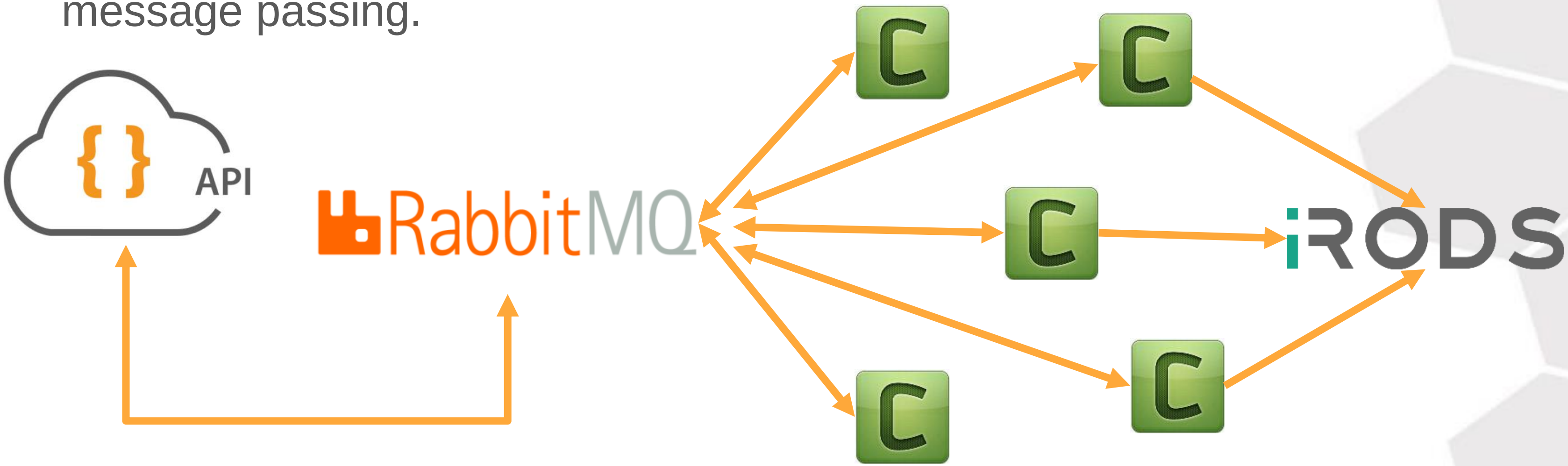
We contributed to the PRC by developing authentication modules for:

- Grid Security Infrastructure (GSI)
  - Merged on main branch on Jan 2017
  - Status: completed
- Pluggable authentication modules (PAM)
  - Merged on main branch on Dec 2018
  - Status: partially completed, some issues to be fixed
    - e.g. #156 PAM authentication and irods_environment.json



Asynchronous operations

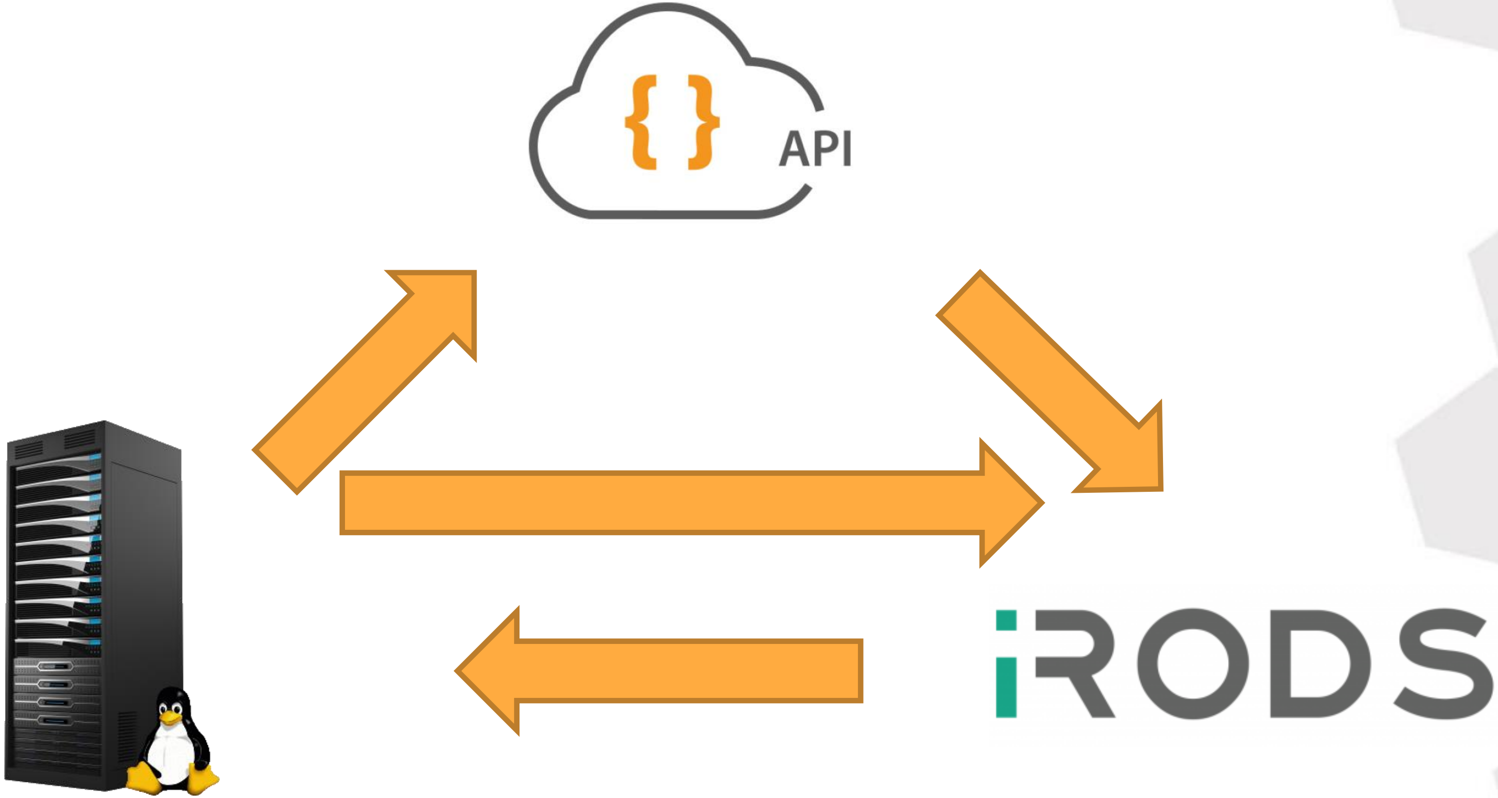
- Some operations are (quite) fast and can be execute synchronously
- To be able to execute data intensive and complex workflows we also introduced an asynchronous layer
- Implemented on Celery, a task management queue based on distributed message passing.



High Performance Computing

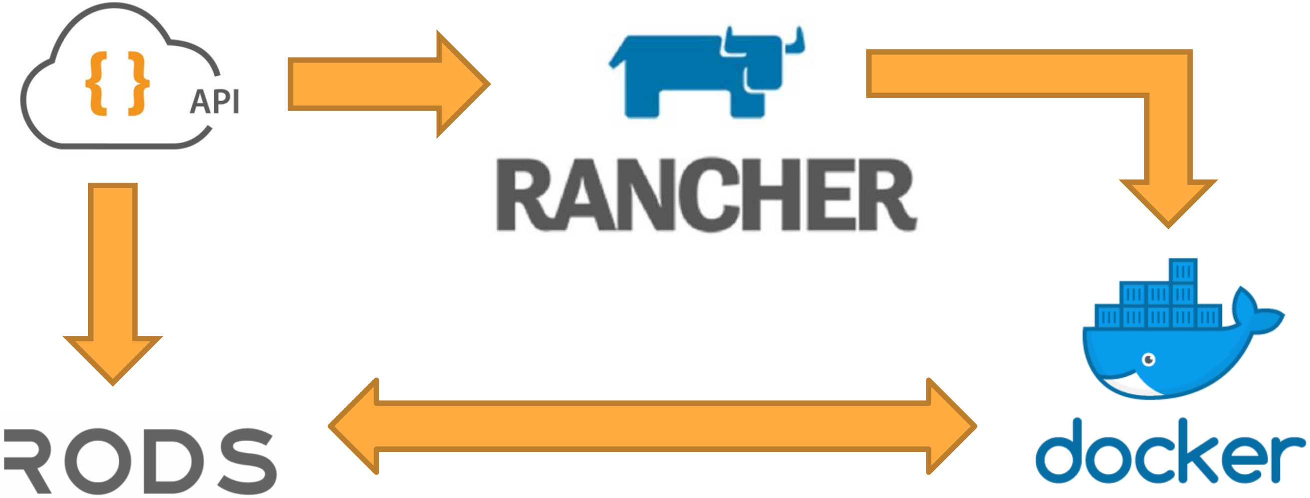
- Many projects need to store data for archiving purpose to be treated as read-only resources (e.g. for data search / retrieval)
- Other projects use archived data as inputs for analyes
- The use of iRODS ensure data to be easily shared beetwen all the components
- The use of ACL ensure data security by preserving access rights

Complete workflow



Dockerized environments

- HPC clusters are not always the solution
- More flexibility can be achieved through **docker**
- Docker containers can be orchestrated by using services like **Rancher**
- We implemented a **Rancher client** integrated into RAPyDO



iRODS main benefits

- Stability and scalability, also for big data projects
- Accessibility from different locations (REST APIs, HPC cluster)
- Security and access policies (preserved regardless the access method)
- Many authentication methods (some of our projects are certificates-based, other are defined on LDAP servers -> GSI, PAM)
- Data replication
- Rules

Conclusions

- iRODS is the perfect technology as base for many data-oriented projects
- Projects need higher-level services to be built over it
- Common requirements can be translate in common solutions
    - Don’t reinvent the wheel…
- Risk of fossilization on obsolete solutions
    - Every new project can start from previous solutions
    - … and perfect it

Don’t reinvent, perfect it

iRODS

Thank you for your attention

Mattia D’Antonio – m.dantonio@cineca.it

https://github.com/rapydo