Finanziato dall'Unione europea NextGenerationEU | Ministero dell'Università e della Ricerca | Italiadomani PIANO NAZIONALE DI RIPRESA E RESILIENZA | INFN

Distributed Data Management with Rucio for the Einstein Telescope

Lia Lavezzi (INFN Torino)

on behalf of the ET Computing Team

ET EINSTEIN TELESCOPE | INFN Istituto Nazionale di Fisica Nucleare SEZIONE DI TORINO | ET ITALY Einstein Telescope

The speaker's contract is funded by the project ETIC:
Einstein Telescope Infrastructure Consortium (ETIC - IR0000004)
PNRR MISSIONE 4, COMPONENTE 2, INVESTIMENTO 3.1

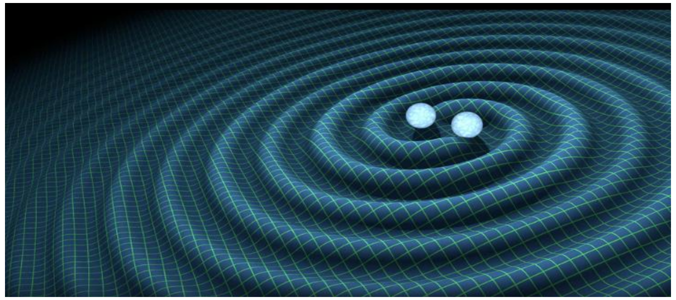
Einstein Telescope

It is a **Third Generation Ground Observatory** (3G) for Gravitational Waves, currently in the preparatory phase, to become operative in about 10 years.

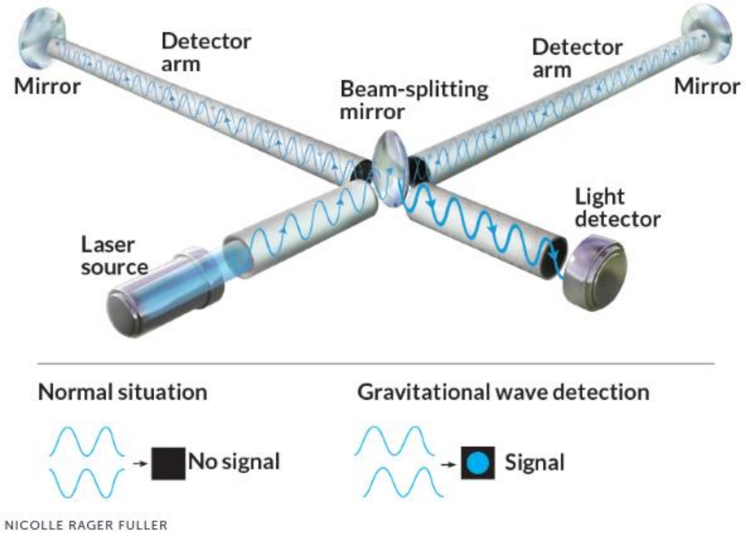
Composed by Michelson interferometers, disposed in a triangular or double-L pattern, will be built in Italy (Sardinia) or in the EU Regio Rhine-Meuse.

Gravitational Waves (GW) are ripples in the space-time fabric, generated by massive celestial bodies or highly energetic cosmic events

- 1915 theoretical prediction (Einstein)
- 2015 first experimental observation (LIGO-Virgo, BBH)
- 2017 first multi-messenger observation (LIGO-Virgo, BNS)



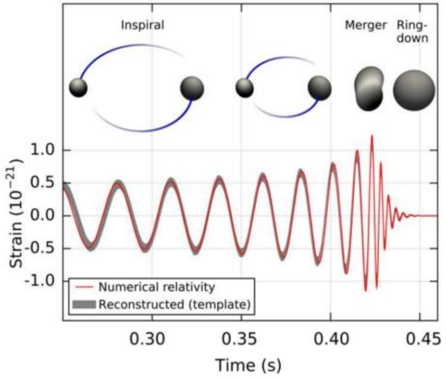
www.ligo.caltech.edu



To detect the passage of a GW the goal is to measure a change in the length of the two arms.

strain = differential arm length

This quantity is sampled in time ⇨ time series



PRL 116, 061102 (2016)

Einstein Telescope

Will improve the sensitivity of more than **one order of magnitude** with respect to current 2G interferometers, will increase the signal-to-noise ratio, suppress all kinds of noise and widen the field of observation.

www.ligo.caltech.edu

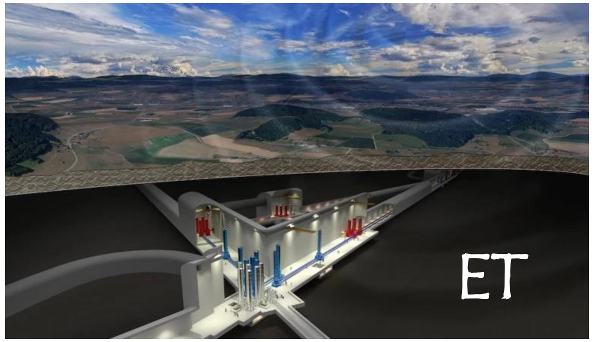
www.virgo-gw.eu

longer arms (10 km)

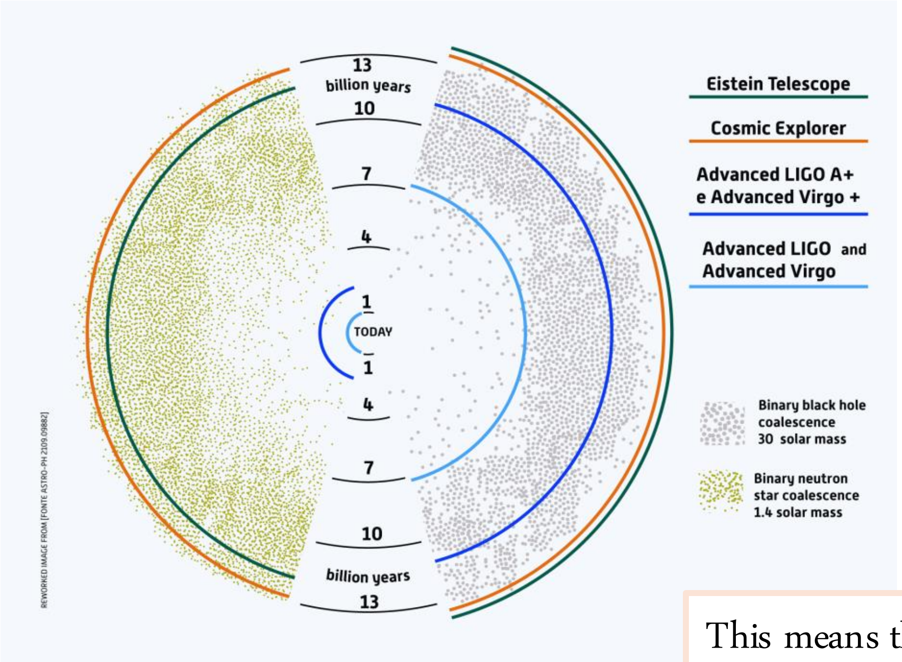
underground (**minus** 200-300 m)

each detector consists of two separate interferometers:

- cryogenic temperature [low frequency 3–30 Hz]
- room temperature [high frequency 30–10^3 Hz]



www.nature.com



Better sensitivity means to observe farther in distance ⇨ closer in time to the Big Bang (~ 0.1 billion years from BB)

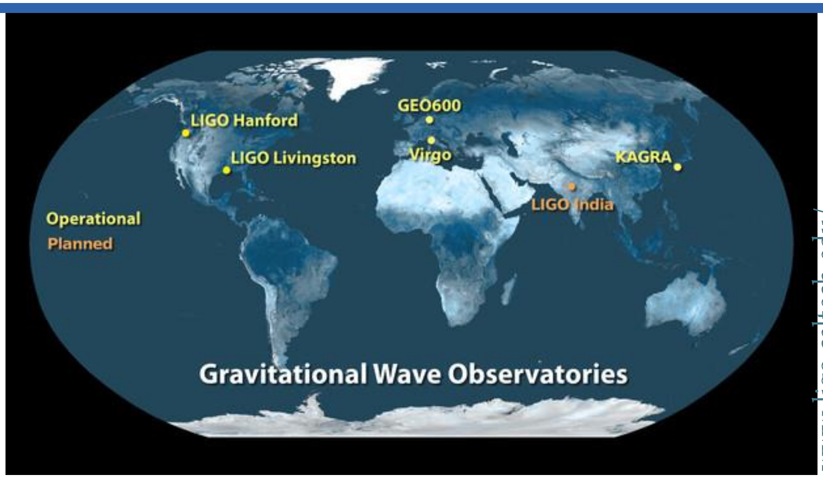
Lower frequency means to observe longer signals (hours) and heavier black holes ($M \sim 10^4 M_\odot$)

This means that the # observations will increase: 100 obs/year (current) ⇨ 100 thousands obs/year (ET expected)

Gravitational Wave Data Requirements

Two kinds of data:

- **physics data** are time series with a defined duration of the *strain* variable, do not scale with sensitivity: **expected ET per year some 100 TB**
- **raw data** are collected from all the sensors, do scale with interferometer complexity: **expected ET per year some 10 PB**



www.ligo.caltech.edu/

2G interferometers LIGO, Virgo and KAGRA participate to the International GW Network (IGWN) and use a common infrastructure:

- data distribution via OSDF (*) (within OSG, US) using Rucio (partially)
- publication of the data on CVMFS for POSIX view
- job scheduling via HTCondor (80% local + 20% grid)

More details in James Clark's talk

ET is evaluating similar and updated tools:

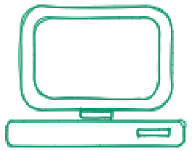
- data distribution and management Data Lake using Rucio
- RucioFS for POSIX view
- ESCAPE (**) VRE JupyterLab for interactive development
- analysis tools as Snakemake (backends: HTCondor grid, Slurm HPC clusters, clouds) and REANA for open science

Once again, next generation Interferometer will create a strong **network**, then ET will need to share data with the Cosmic Explorer (3G interferometer in the US) and LISA (ESA mission space interferometer) which will be taking data in the same decades as ET ⇨ we will need a multi-RI access to the Rucio Data Lake

(*) Open Science Data Federation osg-htc.org/services/osdf.html

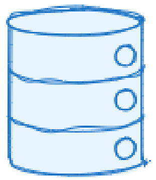
(**) European Science Cluster of Astronomy & Particle physics ESFRI research infrastructures projectescape.eu/

Current Rucio Setup in Einstein Telescope



Torino Rucio Server

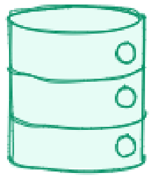
- authentication: userpass, X.509/proxy
- used for testing and development
- hosts:
    - Rucio Server
    - Rucio DB (`Postgres`)



Louvain RSE

One of the two `XRootD` *origin* data servers federated to OSDF was dedicated to host ET data:

- ~2 TB simulated data
- published also on CVMFS
- used for Mock Data Challenges



Torino RSE

- `XRootD` standalone server
- 260 GB space
- used for testing and development

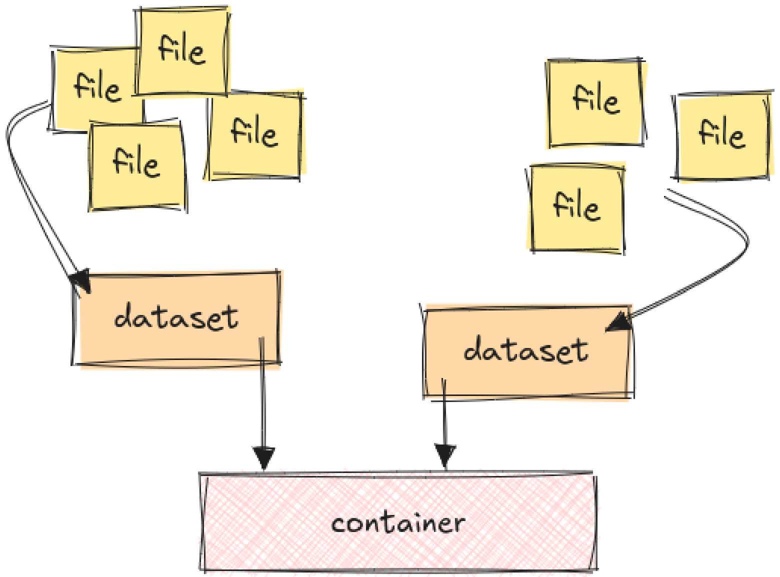
Pictures drawn on excalidraw.com/

RucioFS: Rucio Filesystem

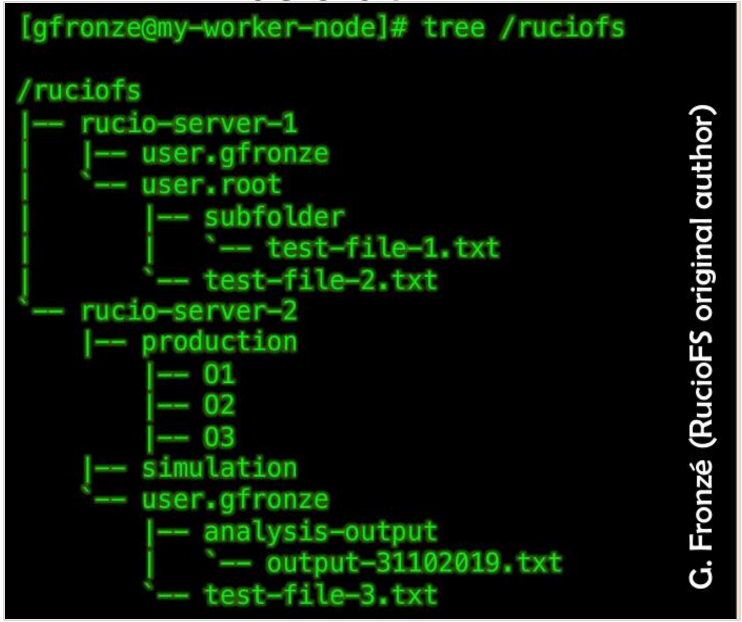
GW users and analysts *love* POSIX-like filesystem, in fact currently all experimental data are published on CVMFS.

RucioFS is a POSIX-like FUSE mount filesystem that transforms the dataset/container/file scheme in a hierarchical structure of directories.

STANDARD RUCIO DATA



RUCIOFS DATA



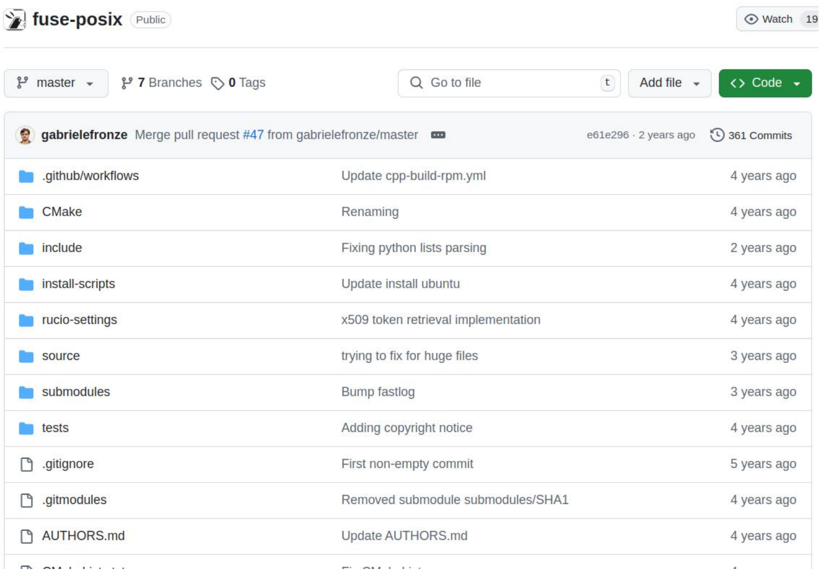
- written in C/C++11
- wraps around python Rucio client
- read-only filesystem
- supports multiple servers
- files downloaded @ first access
- tested on CentOS7, Ubuntu 18.04 LTS and AlmaLinux 9

Pictures drawn on excalidraw.com/

RucioFS: Installation

- Can be downloaded from GitHub repo and used to build a Docker container starting from the `rucio-client` image
- The code to do it is available in `rucio/containers` repo
- The `rucio-fs` image is also available on DuckerHub

github.com/rucio/fuse-posix

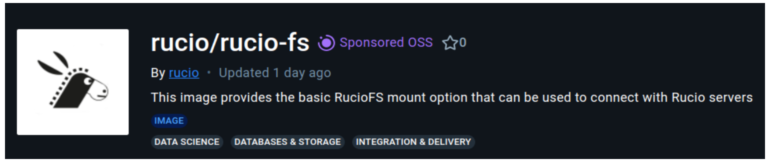


(G. Fronzé, original author)

github.com/rucio/containers/blob/master/fs/Dockerfile

```
ARG TAG
FROM rucio/rucio-clients:release-$TAG

USER root
RUN dnf install -y git cmake3 libcurl-devel fuse-devel tree g++


  ...

ARG CACHEBUST=12
WORKDIR /opt
RUN git clone --recursive https://github.com/rucio/fuse-posix.git
WORKDIR /opt/fuse-posix
RUN /bin/bash ./build.sh


  ...
```

hub.docker.com/r/rucio/rucio-fs



- I have Rucio 32.8.0 installed so I built my `rucio-fs` image and container
- Had to fix the Dockerfile for missing installation of g++

RucioFS: How it Works

```
[root@et-rucio-server fuse-posix]# cd /ruciofs/torino
-> fuse-op.h: rucio_getattr: path /
-> fuse-op.h: rucio_getattr: path /torino
-> fuse-op.h: rucio_getattr: this is server mountpoint for torino server
--> REST-API.cpp: rucio_list_scopes
---> curl-REST: GET: url: https://et-rucio-server.to.infn.it:443/scopes/
```

Implemented callback functions:

- `rucio_getattr;`
- `rucio_readdir;`
- `rucio_read;`

READ ONLY

- `fuse-op/rucio_getattr`: checks the full PATH and decides if it is a server, scope, did, container, file
  - It is a server ⇨ `REST-API/rucio_list_scopes`: checks if the list for this server is in the cache
    - It is not ⇨ `curl-REST/GET`: gets the information from the Rucio DB via `libcurl`

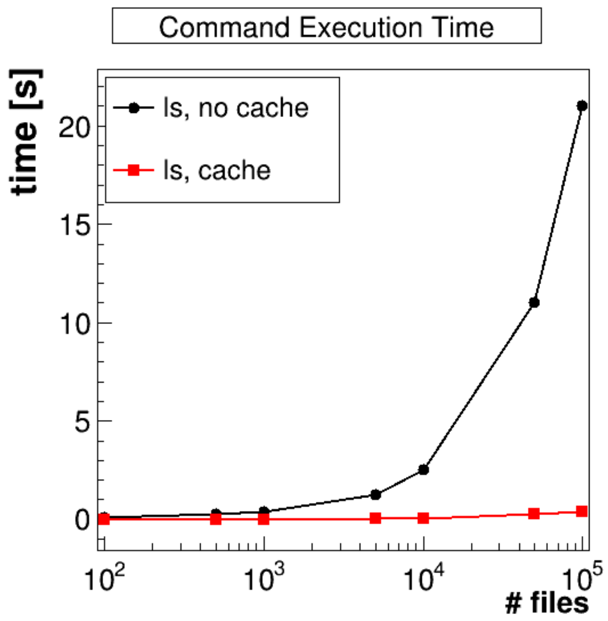
- The scope (server, did, container, file) names are all cached by means of a
  `std::unordered_map< std::string, std::vector < std::string > >`
  i.e. associations of parent_directory and list of subdirectories (*e.g.* server_name, vector < file_scopes >)

- The caching system speeds up the execution time, but a mechanism to automatically refresh the cache is still missing

⇨ the cache is updated only after the `/ruciofs` directory is unmounted and re-mounted (*i.e.* the Docker container is stopped and re-started)

THIS NEEDS TO BE INVESTIGATED & IMPROVED

RucioFS: Scalability Test



Test: run **`ls`** in the directory `/ruciofs/torino/lavezzi`, filled with an **increasing number** of files **with/without cache**

- uploaded *n* single files on Torino Rucio RSE
- each one is a text file with a random number written in it
- size ~ 4 kB/file

TIME IS LINEAR WITH THE NUMBER OF FILES

- with caching the time stays < 0.5 s even for 10^5 files

⇨ having a caching system is the optimal choice

- with caching, there is the need to unmount/re-mount the filesystem, while without caching there is not

⇨ need to implement a way to refresh the cache periodically

ET use case numbers:

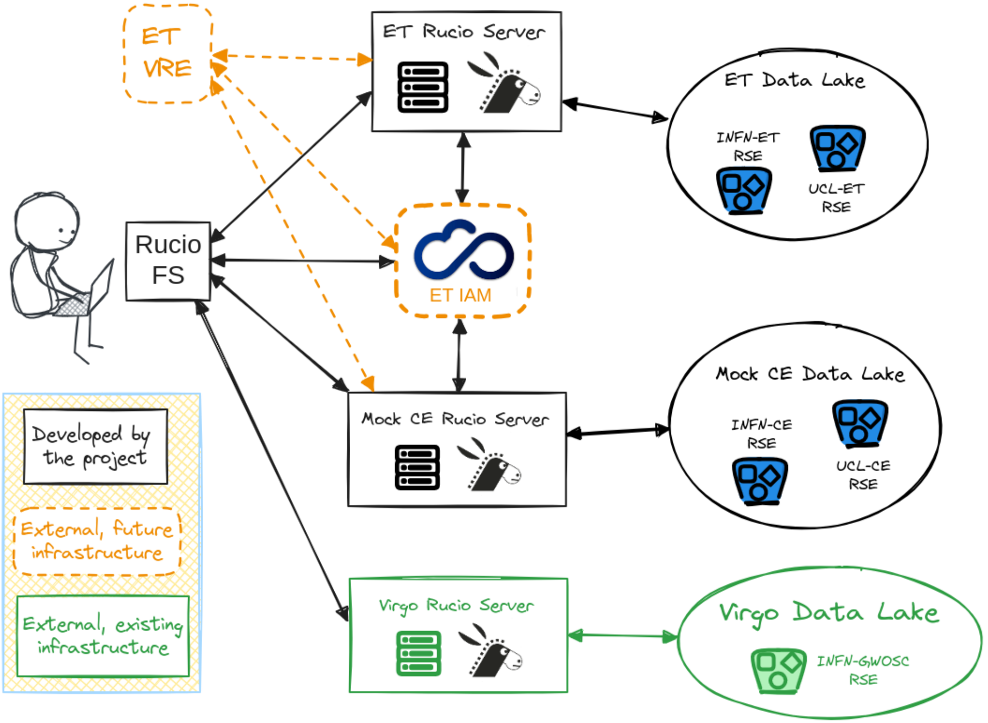
- MDC1 dataset: 1300 files, 155 GB per interferometer (~ 1month data taking)
- 3 interferometers + 1 null channel, noise only + noise + signal ⇨ total 10400 files

Virgo use case numbers

- take O4 as a reference, from Spring 2024 to now
- ~ 57000 files, ~ 3.2 TB (strain + environmental)

MADDEN Multi-RI Access and Discovery of Data for Experiment Networking

Federica Legger (INFN Torino)
Andres Tanasijczuk (UCL)



Participating organizations:

- INFN Torino (PI: Federica Legger)
- UC Louvain (coord.: Andres Tanasijczuk)

- Targeted start date: January 2025
- Duration of the project: 24 months
- Overall funding: 210 K€

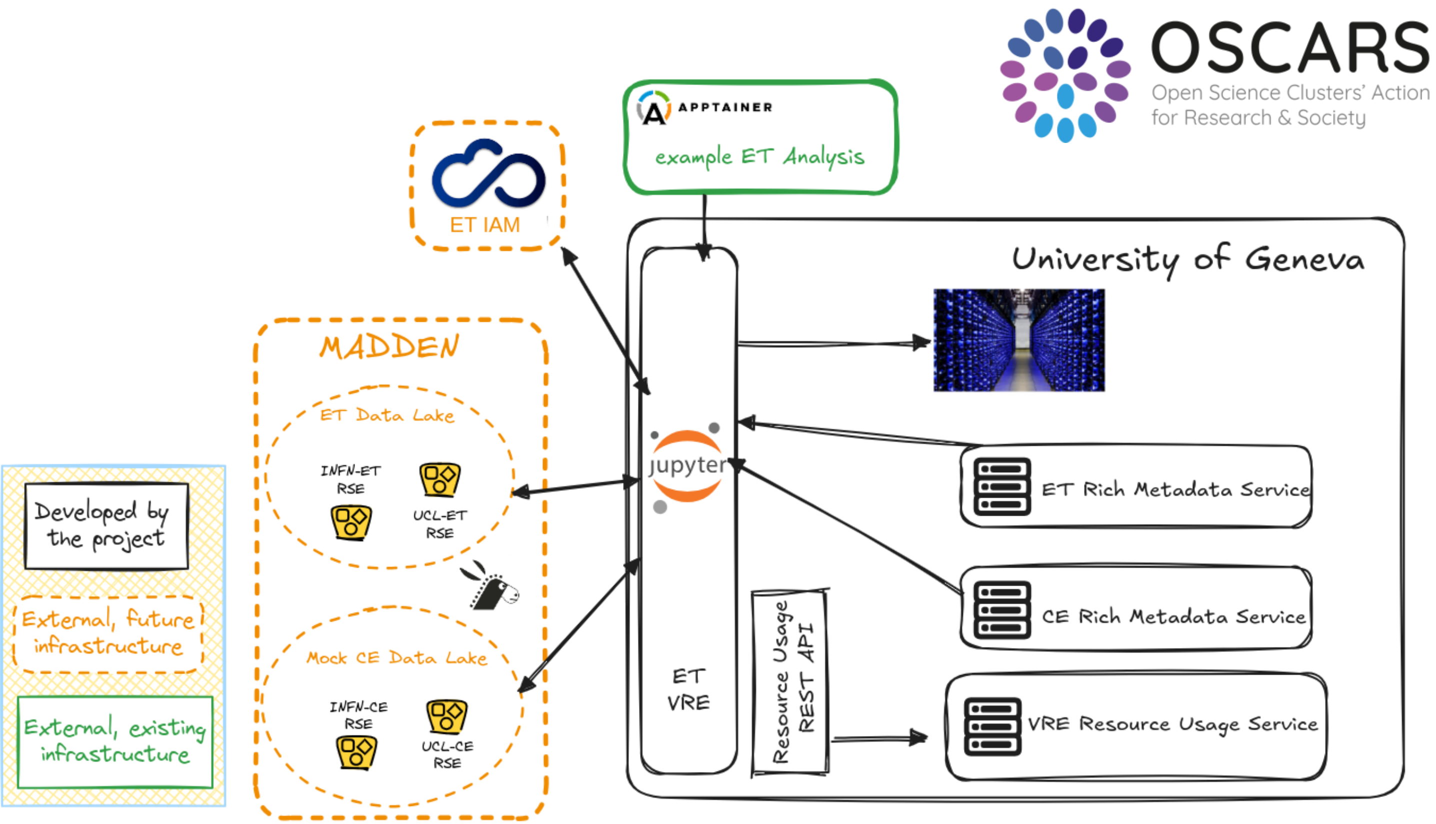
The main objectives of this project are:

- Build a multi-RI Data Lake managed with Rucio.
- Develop and test RucioFS, a tool to provide a POSIX-like view of the Rucio catalogue in a multi-RI environment.
- Extend RucioFS to support advanced querying capabilities using Rucio metadata.

Pictures drawn on excalidraw.com/

ETAP Einstein Telescope Analysis Portal

Paul Laycock (UniGe)



Participating organizations:

- University of Geneva (PI: Paul Laycock)

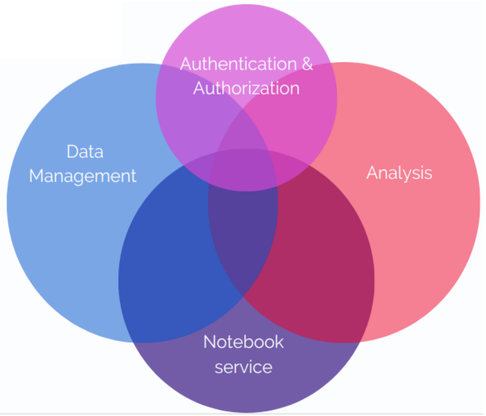
- Targeted start date: January 2025
- Duration of the project: 18 months
- Overall funding: ~250 K€

- Deploy the CERN ESCAPE VRE at University of Geneva
- Connect to multi-RI Data Lakes managed by Rucio (MADDEN)
- Deploy multi-RI Metadata services from the HEP Software Foundation (HSF)
- Design a flexible computing resource monitoring service

Pictures drawn on excalidraw.com/

VRE – Rucio Metadata in the JupyterLab Extension

Georgy Skorobogatov, ICCUB



VRE(*) (CERN) is a collaborative online platform to enable the sharing of analysis tools and data access.

One of the components is a JupyterLab to provide access to the various tools.

The Data Management is performed via Rucio ⇨ Rucio Jupyter Extension

GW use case

GW **physics data** are time series with a defined duration of the *strain* variable.

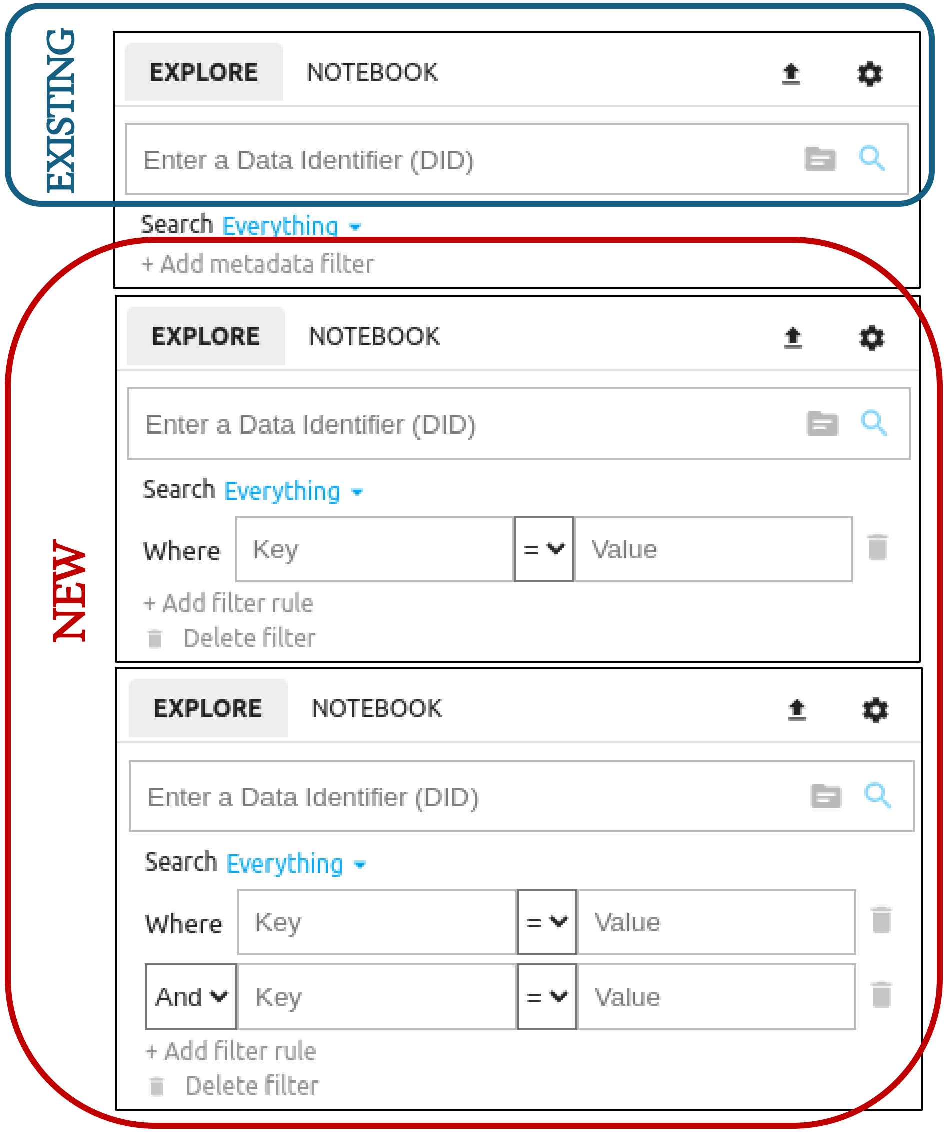
Rucio metadata (timestamp, channel, delta time…) are useful to find data.

⇨ Added a button for opening a **new** metadata filter interface.

Metadata filter interface

- Metadata **key + value + comparison operator** selector
- More filter rules can be added/deleted
- Logical operator selectors

Contributions are welcome at: github.com/rucio/jupyterlab-extension/issues/36



(*) Virtual Research Environment vre-hub.github.io/

Conclusions

- ET is a 3G ground-based observatory for Gravitational Waves in preparatory phase that will become operative in about 10 years.
- It is currently defining the computing model and evaluating Rucio for data management.
- Wished features:
  - Support for multi-RI Data Lake
  - POSIX-like view of the file catalogue
  - Extended metadata support

We are actively working on the development of these features through current and future funded EU projects

Thank you for the attention