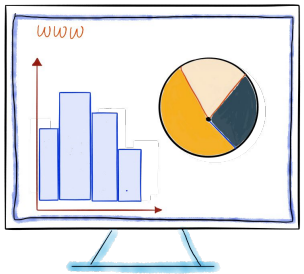
ESCAPE: a view of RUCIO + JupyterLab + ATLAS Open Data integration



Arturo Sánchez Pineda - LAPP

6th April 2021

Overview

An attempt to summarise the activities relative to *a* integration and consolidation of the Data Lake via RUCIO and a friendly web-based UI like JupyterLab, and the efforts to consolidate those in a single entity (container).

And how ATLAS Open Data is used as a Test for such technology and integrations.

In this case, the target audience refers to scientists & advanced users looking for data to perform or reproduce an analysis.

They are/should be aware of the RUCIO as a service, but enjoying the UI and features of a tool as JupyterLab.

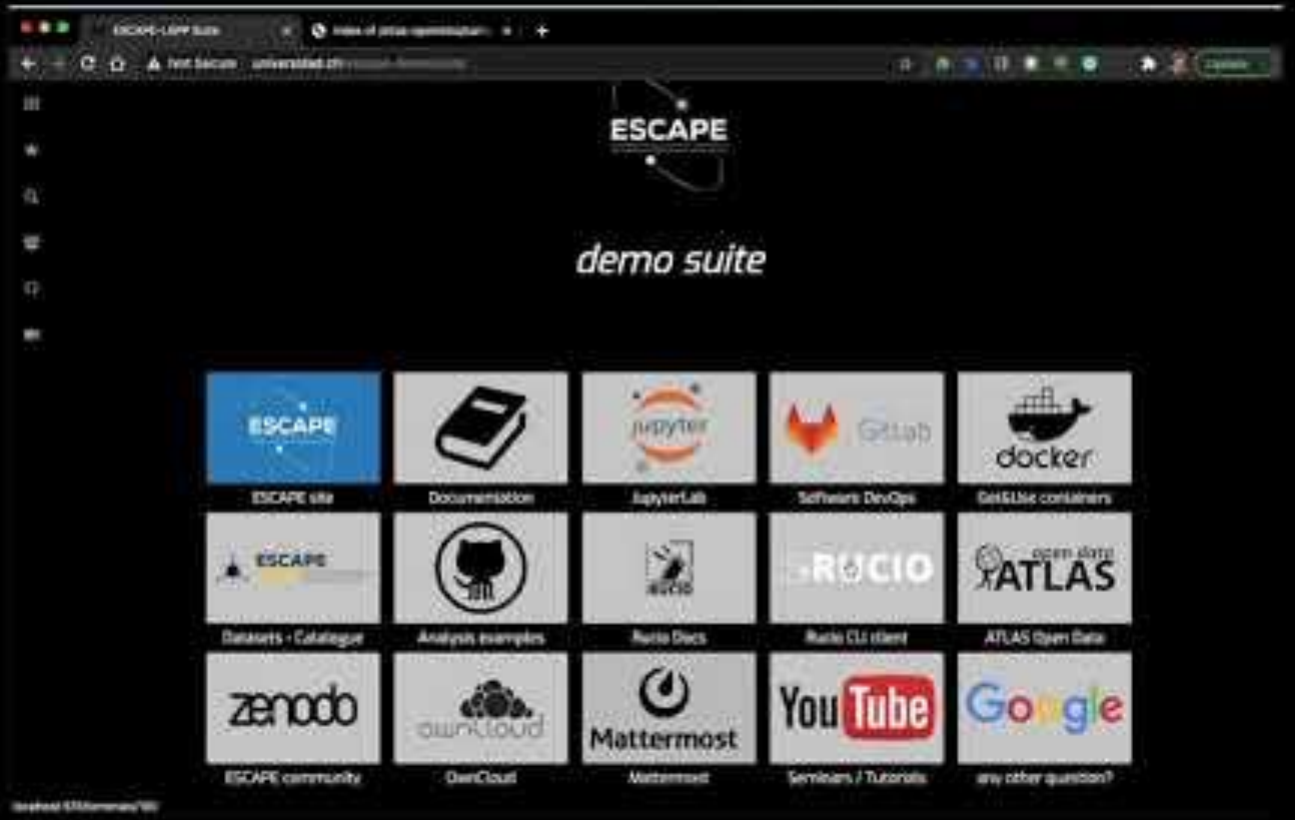
My job focuses in integrating and testing the developments of many experts.

RUCIO & JupyterLab

(container)

The RUCIO CLI client

(a 90 sec video, mainly for new users)



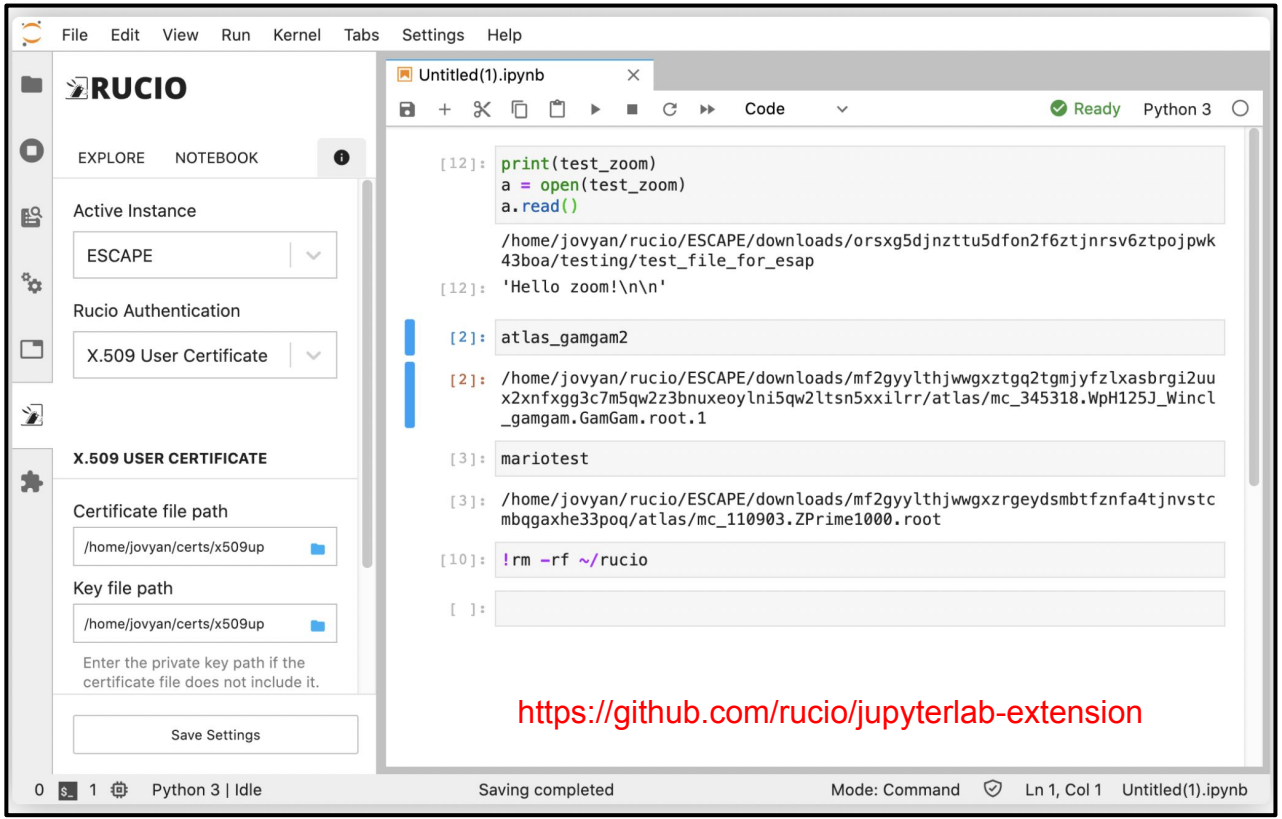
http://universidad.ch/escape-demo/suite/

The RUCIO extension for JupyterLab

The JupyterLab RUCIO extension allows to authenticate and interact with the datasets from the web UI.

Making much easier the exploration and analysis of samples in the Data Lake infrastructure.

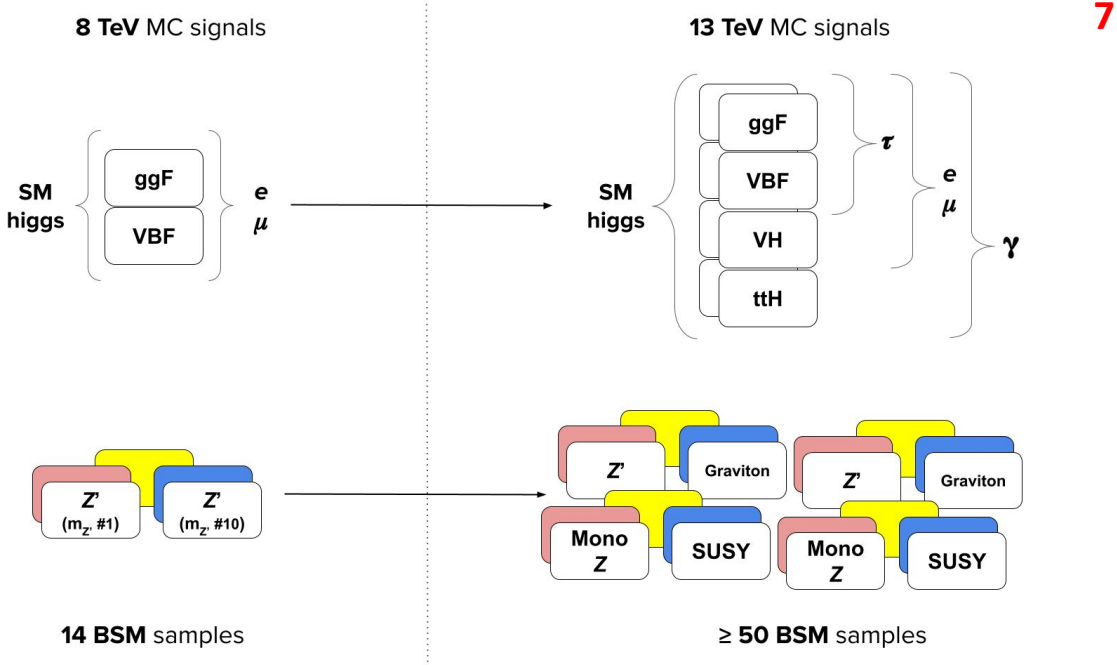
Slides in the backup for credits!

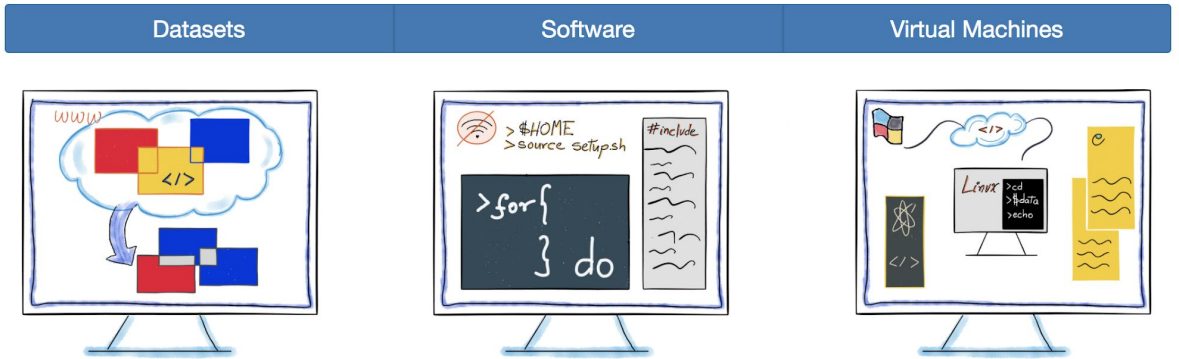


ATLAS Open Data

We release and deploy the resources on the Internet. In a nutshell, they are a series of

- Data samples in ROOT n-tuple format
- Framework software and **Jupyter [Notebooks](#) in Python and C++ (kernels)** to analyse the samples and produce physics analysis
- JavaScript (JS) applications to produce cut-and-count analysis
- Linux-based Virtual Machines with ROOT-CERN analysis framework and other multiple DevOps tools
- [GitHub](#) & [GitLab](#) repositories
- Web-based documentation sites to present the resources, activities and projects that can be performed





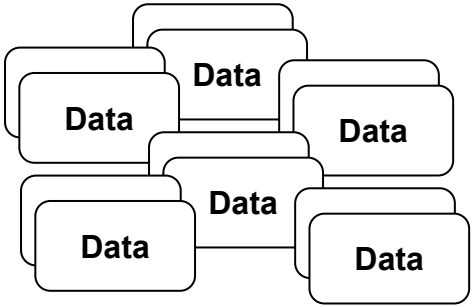
8 TeV release

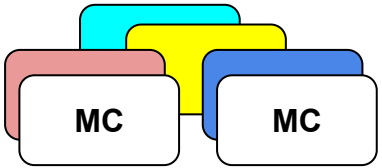
13 TeV release

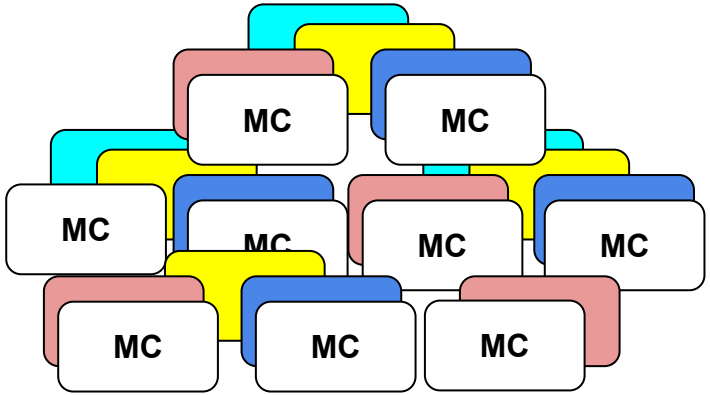
1 fb^{-1}

Data

10 fb^{-1}





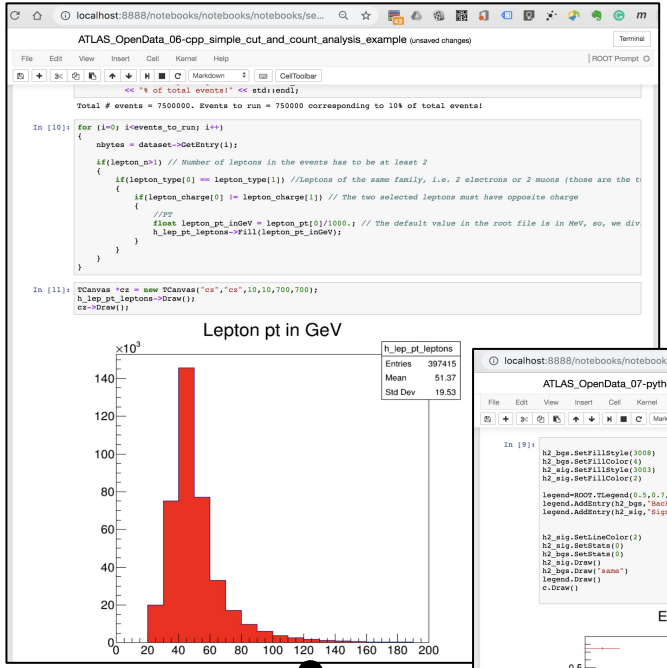


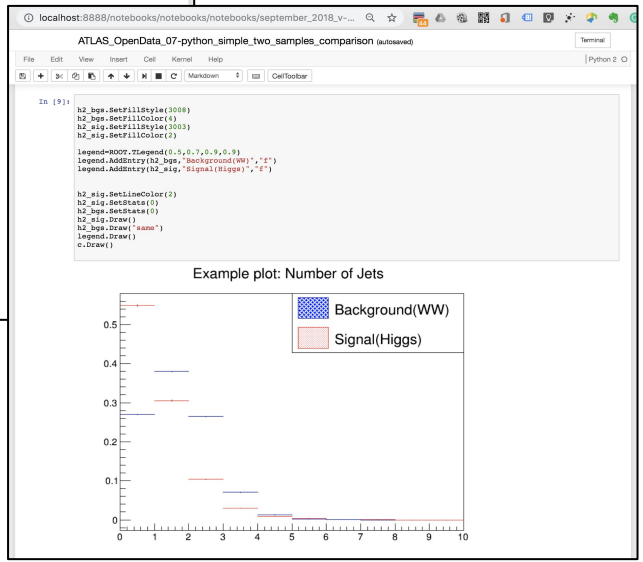
44 samples

940 samples

1+7

Collections based in final states





Use ROOTJS

Jupyter notebooks can run ROOT commands

- We produce a series of examples for training on the usage of the notebooks, reading of the samples and plotting simple analysis.
- **The notebooks use both the Python and the C++ ROOT kernel to produce results that can be adjusted by teachers and trainers.**
- **Also ROOT-independent exercises complement the collection of examples, using upROOT.**
- The notebooks can read the samples directly from the Internet (using https protocol) or run locally if present in the machine.

ATLAS Open Data datasets in the Datalake

- ROOT yet need to be importable from a notebook
  - It is deployed for testing in DockerHub
- Add more datasets to the Datalake
  - All the 13 TeV and 8 TeV ATLAS Open Data samples
  - 16 datasets → 940 samples (ROOT files)
  - < 200 GB
  - Scope used: **ATLAS_OD_EDU** (for **ATLAS O**pen **D**ata for **EDU**cation)
  - Source of the datasets:

    http://opendata.atlas.cern/samples-13tev/ **&** http://opendata.atlas.cern/samples-8tev/
    - Another set of 10 ROOT files to come (dedicated Jet MC samples) → 1 dataset, ~21 GB.

The pieces together

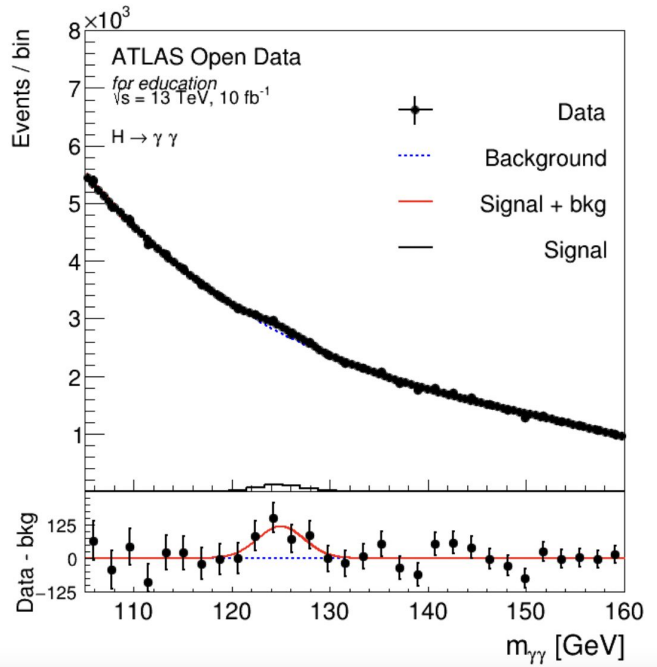
ATLAS Jupyter Notebooks and JupyterLab RUCIO extension

The ATLAS Open Data as a test field

```
In [8]: for i in range(len(hyy_0)):
            print(i, hyy_0[i])

0 /home/jovyan/rucio/ESCAPE/downloads/mf2gyylthjsxqzlsmnuxgzk7jb4xs
1 /home/jovyan/rucio/ESCAPE/downloads/mf2gyylthjsxqzlsmnuxgzk7jb4xs
2 /home/jovyan/rucio/ESCAPE/downloads/mf2gyylthjsxqzlsmnuxgzk7jb4xs
3 /home/jovyan/rucio/ESCAPE/downloads/mf2gyylthjsxqzlsmnuxgzk7jb4xs
4 /home/jovyan/rucio/ESCAPE/downloads/mf2gyylthjsxqzlsmnuxgzk7jb4xs
5 /home/jovyan/rucio/ESCAPE/downloads/mf2gyylthjsxqzlsmnuxgzk7jb4xs
6 /home/jovyan/rucio/ESCAPE/downloads/mf2gyylthjsxqzlsmnuxgzk7jb4xs
7 /home/jovyan/rucio/ESCAPE/downloads/mf2gyylthjsxqzlsmnuxgzk7jb4xs
8 /home/jovyan/rucio/ESCAPE/downloads/mf2gyylthjsxqzlsmnuxgzk7jb4xs
```

```
In [10]: show_image('histograms/hist_mYY_bin1.png')
```



Once the Open Data datasets are registered in RUCIO they can be downloaded and read, using the JupyterLab extension, including search features

Example in nbviewer.jupyter.org

ATLAS Open Data → C++ examples framework

To run C++ analyses

More computational-complex particle physics analysis examples using the existing publicly available data

More in [Opendata.atlas.cern - documentation 13 TeV - physics]

Also use PROOF, adding a parallel component to the examples.

SM Higgs boson production in the H → ZZ decay channel in the four-lepton final state

Physics analysis examples

Overview of physics analysis examples

Brief introduction to the physics of the Higgs boson

SM W-boson production in the single-lepton final state

Single-top-quark production in the single-lepton final state

Top-quark pair production in the single-lepton final state

SM Z-boson production in the two-lepton final state

SM Higgs boson production in the H → WW decay channel in the two-lepton final state

Search for supersymmetric particles in the two-lepton final state

SM WZ diboson production in the three-lepton final state

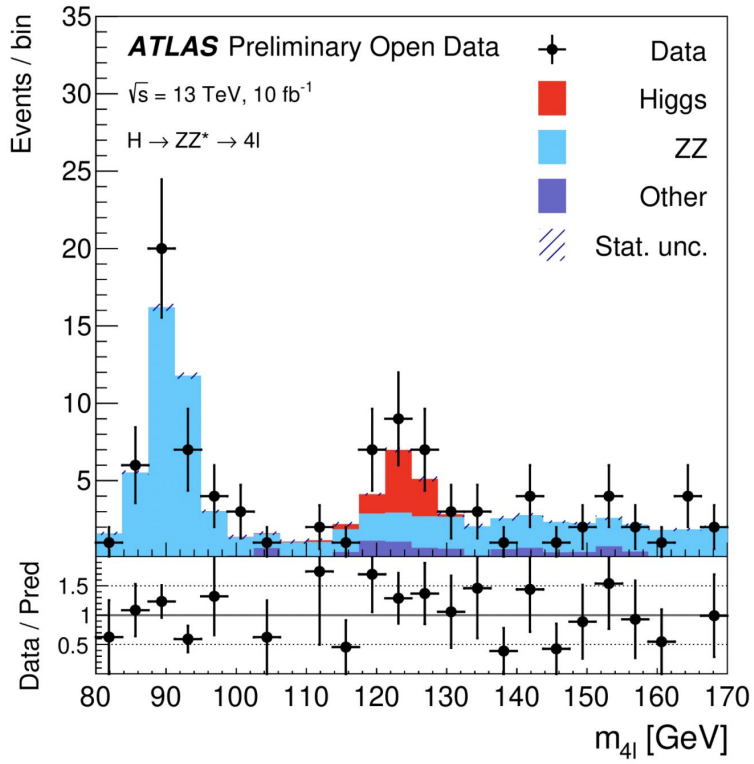
SM ZZ diboson production in the four-lepton final state

SM Higgs boson production in the H → ZZ decay channel in the four-lepton final state

SM Z-boson production in the two-tau-lepton final state

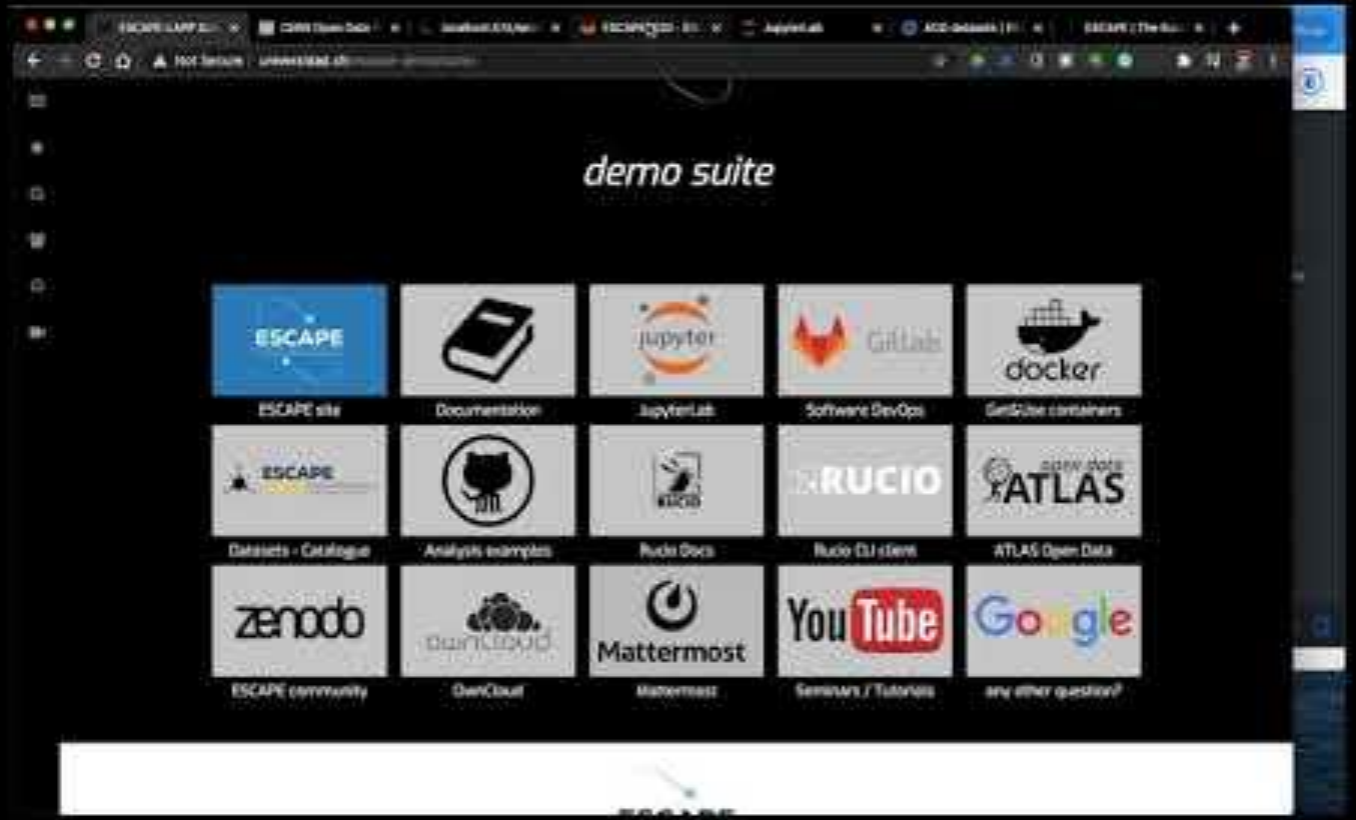
Search for BSM Z' → tt in the single-lepton boosted final state

SM Higgs boson production in the H → yy decay channel in the



Ongoing developments with **JupyterLab & RUCIO extension**

(a 150 sec video)



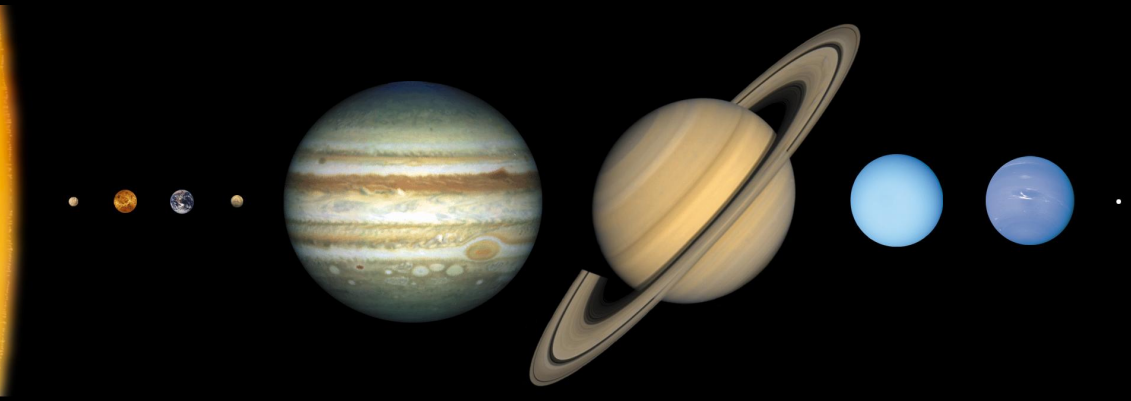
More tools to finish to integrate in the container, like more kernels, PROOF, CVMFS

http://universidad.ch/escape-demo/suite/

Idea:

A collection of containers

A supported container *system*



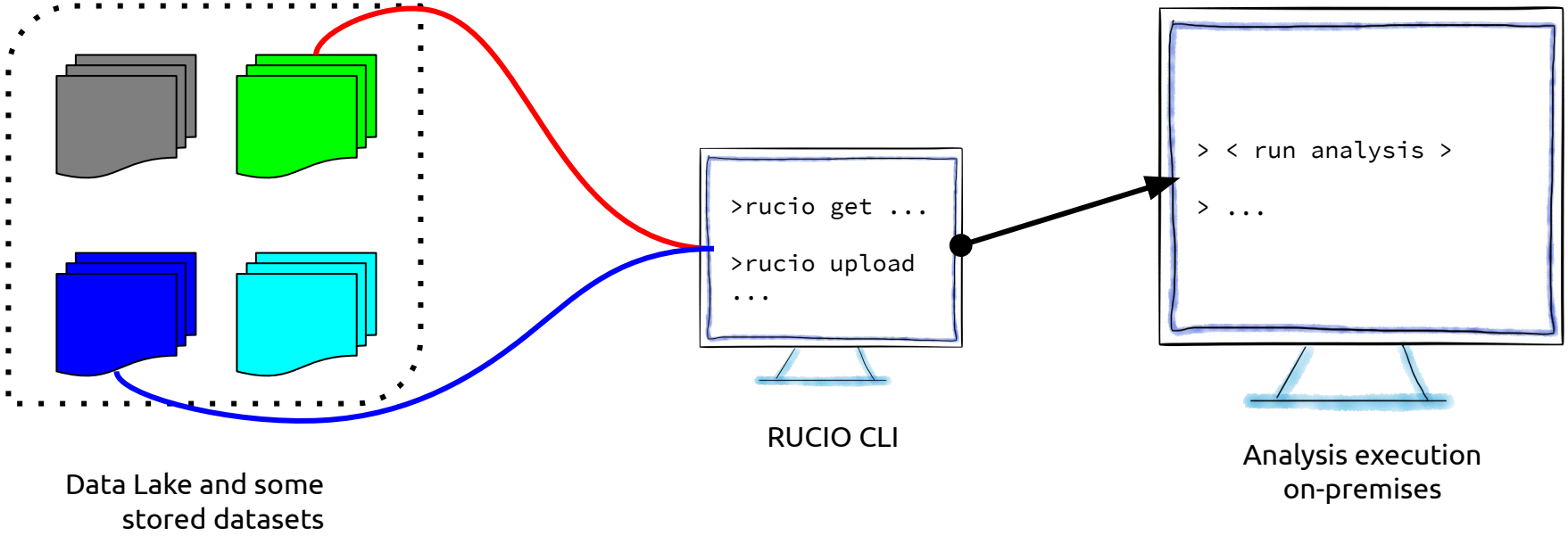
A collection of supported containers will allow maximising the reach of the target audiences while keeping a realistic objective in term of human capital for the creation, maintenance and user's support

- Mercury → it is the collection's base container. It has a minimal setup, including JupyterLab and rucio.
- Venus →a "hotter" version of Mercury, with standard DevOps software tools.
- Earth → The most popular container. Including a series of common HEP tools that *most* users have requested.
- Mars → A dedicated HEP container, slimming version of the Earth.
- Jupyter → the largest container. It has all the tools. For who want to have it "all".
- Saturn → Some experiment's custom version. Same with Uranus, Neptune**?**

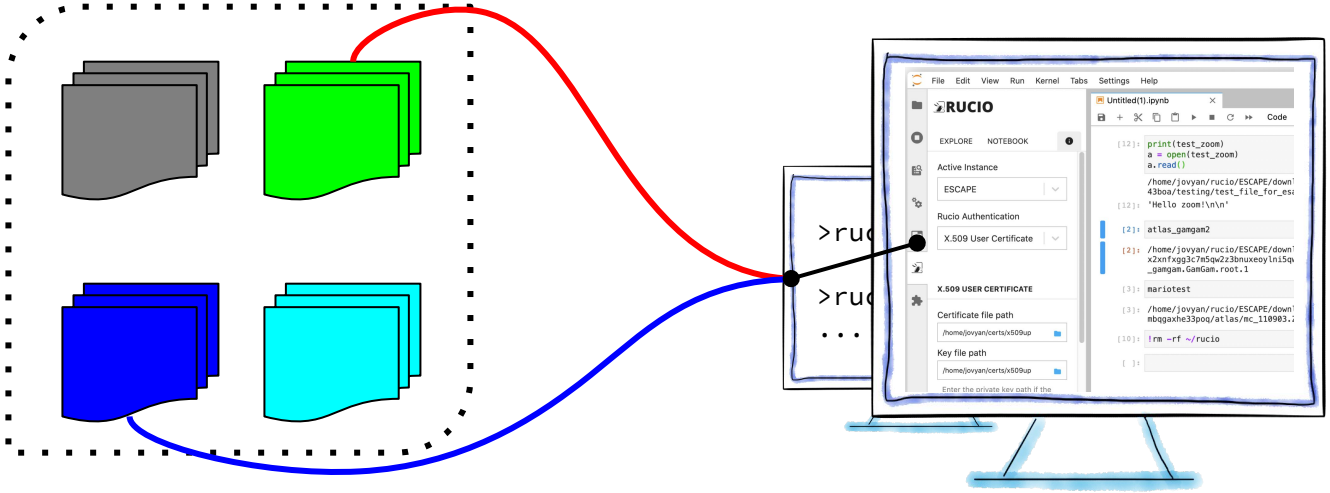
Why the Solar System?

it forces a fix and manageable number :)

Recap



CLI interaction with samples

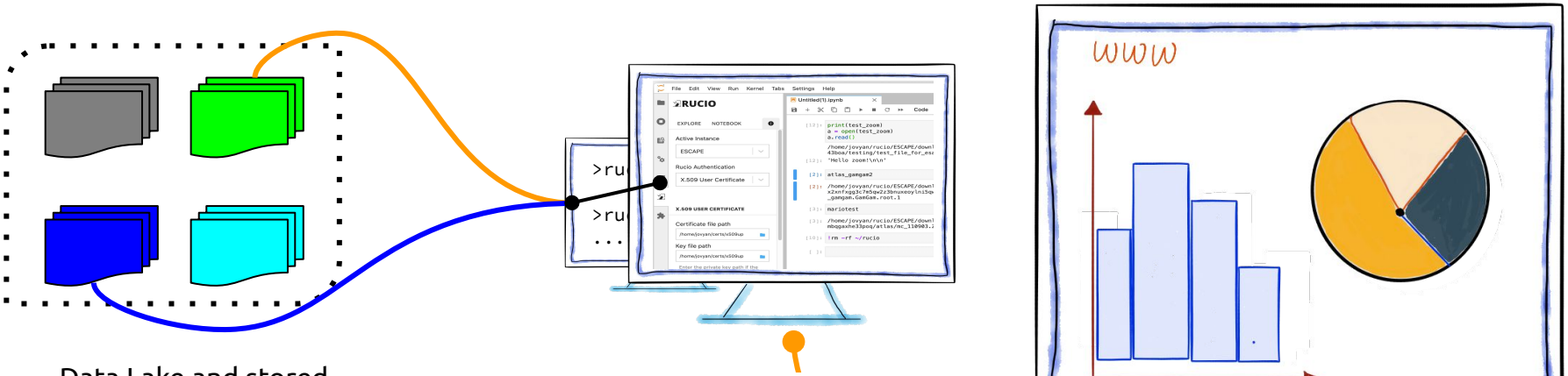


Data Lake and stored Open Data datasets

Container with RUCIO + JupyterLab extension

Authentication, discovery and download of the samples using the Web UI

RUCIO+JupyterLab (container) interaction for users



Data Lake and stored
ATLAS Open Data datasets

RUCIO + JupyterLab with ATLAS
open data notebooks for testing

Analysis code, results and
visualisation

A view of the service

Summary

The job now is the testing, consolidation and use of the mentioned resources in a consistent way that resembles a single service + analysis of real experimental data.

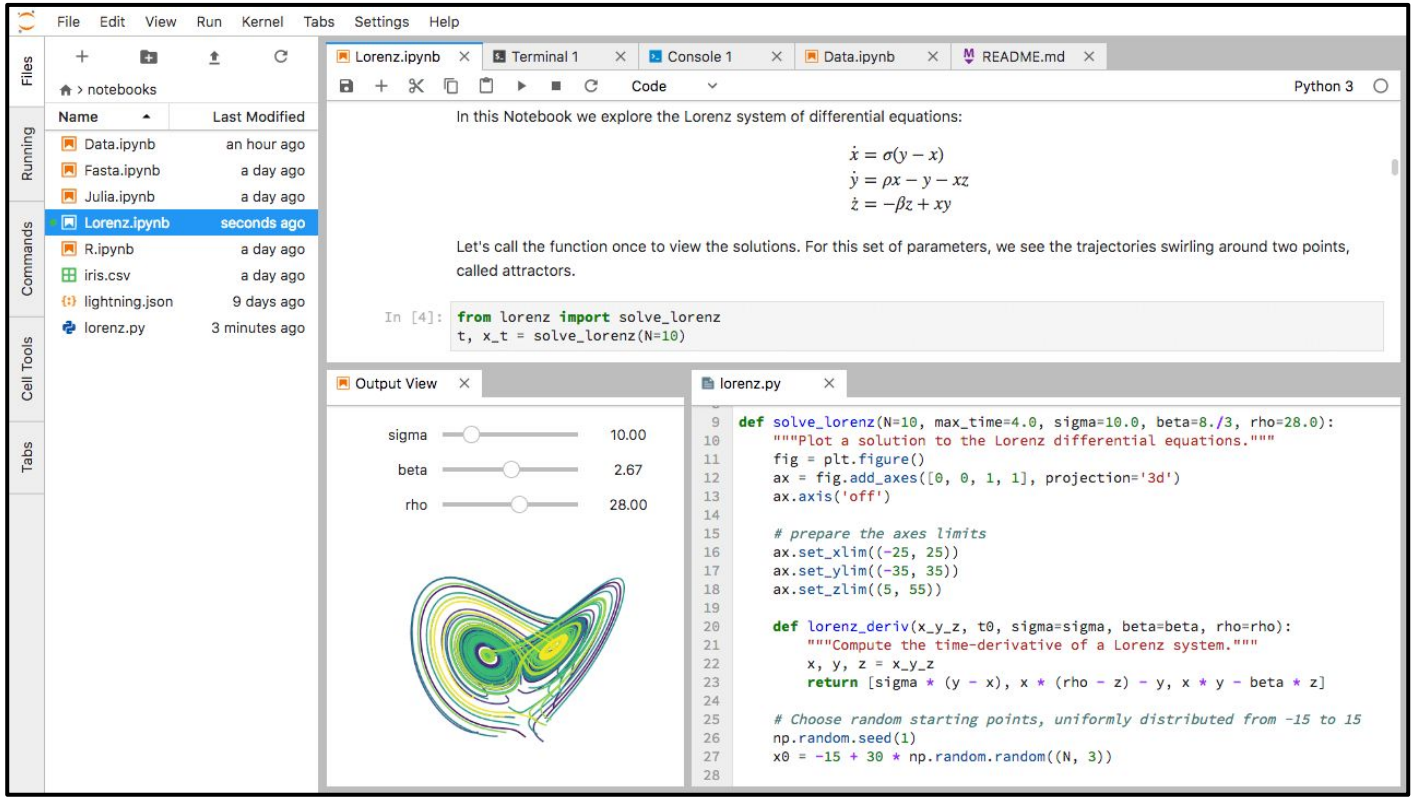
More complex data access to be explored, and also workflows.

Backup

The traditional JupyterLab UI

A well-known tool for all of us (data analysis and visualisation) is the Jupyter notebook.

JupyterLab is a suite of tools and features that allow interacting with multiple elements in a single view. And do the computation, of course.



The JupyterLab RUCIO plugin

In 2020 at CERN, Muhammad Aditya Hilmy created a JupyterLab extension that allows the proper authentication (login/pass or certificate) and access to the datasets in the Data Lake using RUCIO.

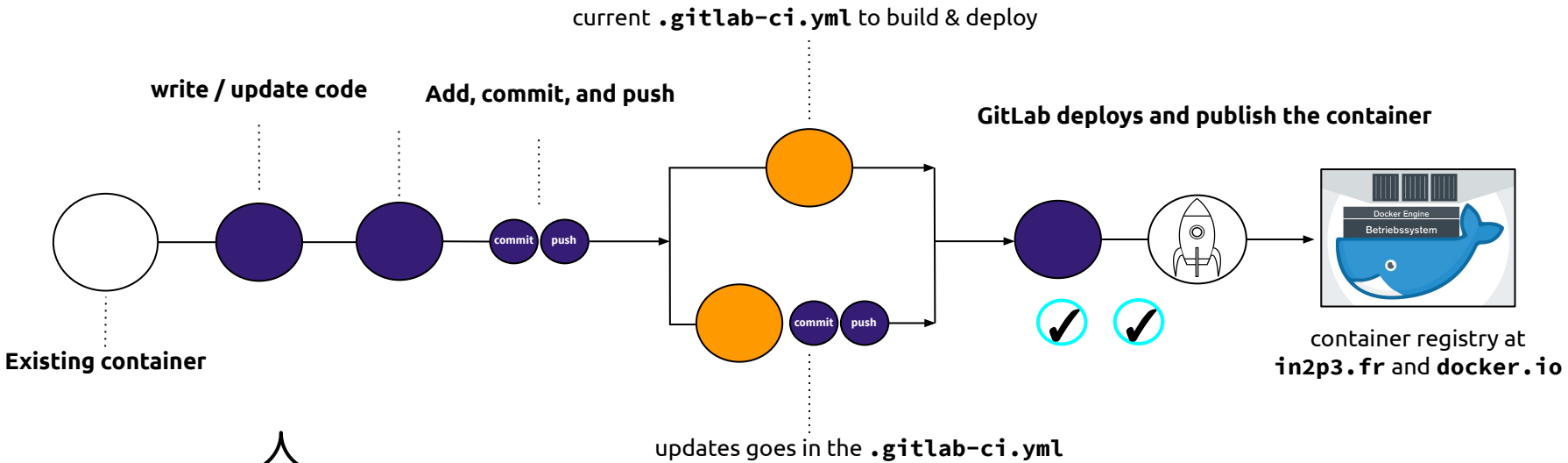
More on how it looks like in one of Muhammad's presentations

The main idea is to deliver an easy and transparent way to access, download and use datasets replicated in the Data Lake.
It hides all the complexity on that access and allows a seamless usage of the data in a Jupyter notebook analysis.

Work at LAPP

- What I understood until now - any missing info or mistake is mine

A very first view to the current container



Container CI / CD

- The series of resources is package in a single container
- The CI setup automatically handles the publication of the container

Frédéric & Berkay's job

Several developments and deployment already in place

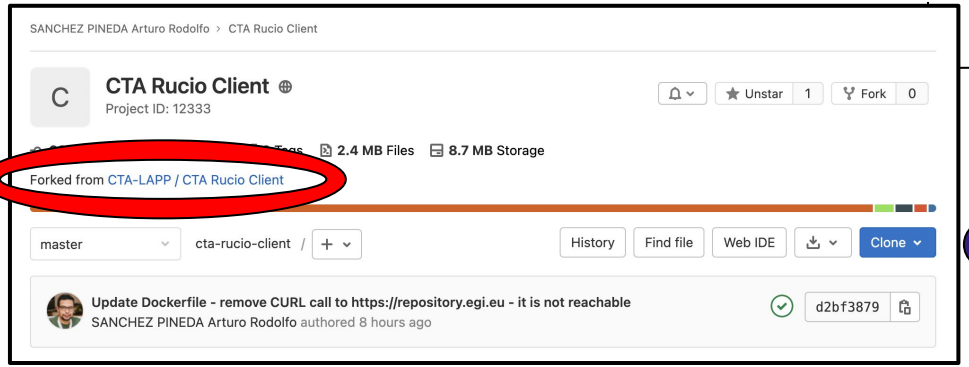
- The compendium of resources includes the current rucio client + JupyterLab + RUCIO extension, proxy & authentication, ...

A very first view to the current container

current **`.gitlab-ci.yml`** to build & deploy

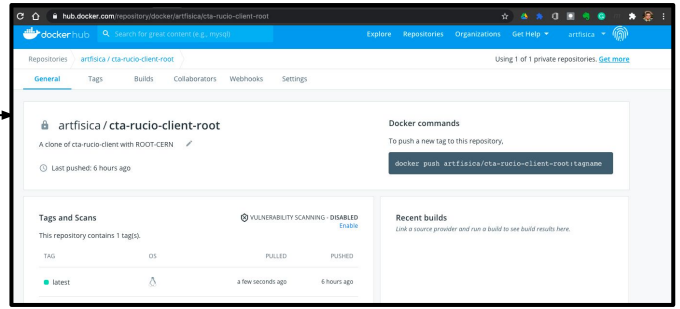
write / update code

Add, commit, and push



push

GitLab deploys and publish the container



updates goes in the **`.gitlab-ci.yml`**

Container CI / CD

- The series of resources is package in a single container
- The CI setup automatically handles the publication of the container

Arturo profiting from Frédéric & Berkay's job

Several tools and updates added

- Mainly ROOT + some dependencies and extra tools…
- Jupyter conf file to handle the usage of the rucio extension (Muhammad feedback, see later)
- From JupyterLab-3 the widgets are installed using ipywidgets instead of labextension

The C++ code also runs in the container

Caveats: this example reads the samples from the internet, but changing one code-line allows to usage of the same samples from the notebook above.