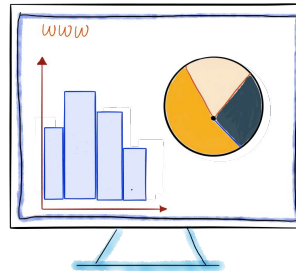


ESCAPE: a view of RUCIO + JupyterLab + ATLAS Open Data integration



Arturo Sánchez Pineda

22 January 2021, LAPP (virtual)

Overview

This is a first 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 how ATLAS Open Data is used as a Test for such technology.

Caveats:

Since I am new to the project and team, I may not have the proper jargon of clear concepts, yet. Any feedback is very welcome :)

User's context

The user's context

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.

And because this is a tool intrinsically web-based, it can be used in a cloud computing environment. So, in terms of setup, it points also to the institute sysadmins that set those tools for their academic community.

RUCIO & JupyterLab

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.

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 screenshot displays the traditional JupyterLab interface. On the left, a sidebar contains a file browser with a table of notebooks and commands. The main area is divided into several panes: a code editor for 'Lorenz.ipynb', a terminal, a console, and an output view. The code editor shows a notebook cell with text and a code input. The output view displays a 3D plot of the Lorenz attractor and a table of parameters (sigma, beta, rho) with sliders. The terminal and console panes are also visible.

Name	Last Modified
Data.ipynb	an hour ago
Fasta.ipynb	a day ago
Julia.ipynb	a day ago
Lorenz.ipynb	seconds ago
R.ipynb	a day ago
Iris.csv	a day ago
lightning.json	9 days ago
lorenz.py	3 minutes ago

In this Notebook we explore the Lorenz system of differential equations:

$$\begin{aligned}\dot{x} &= \sigma(y - x) \\ \dot{y} &= \rho x - y - xz \\ \dot{z} &= -\beta z + xy\end{aligned}$$

Let's call the function once to view the solutions. For this set of parameters, we see the trajectories swirling around two points, called attractors.

```
In [4]: from lorenz import solve_lorenz
t, x_t = solve_lorenz(N=10)
```

Output View:

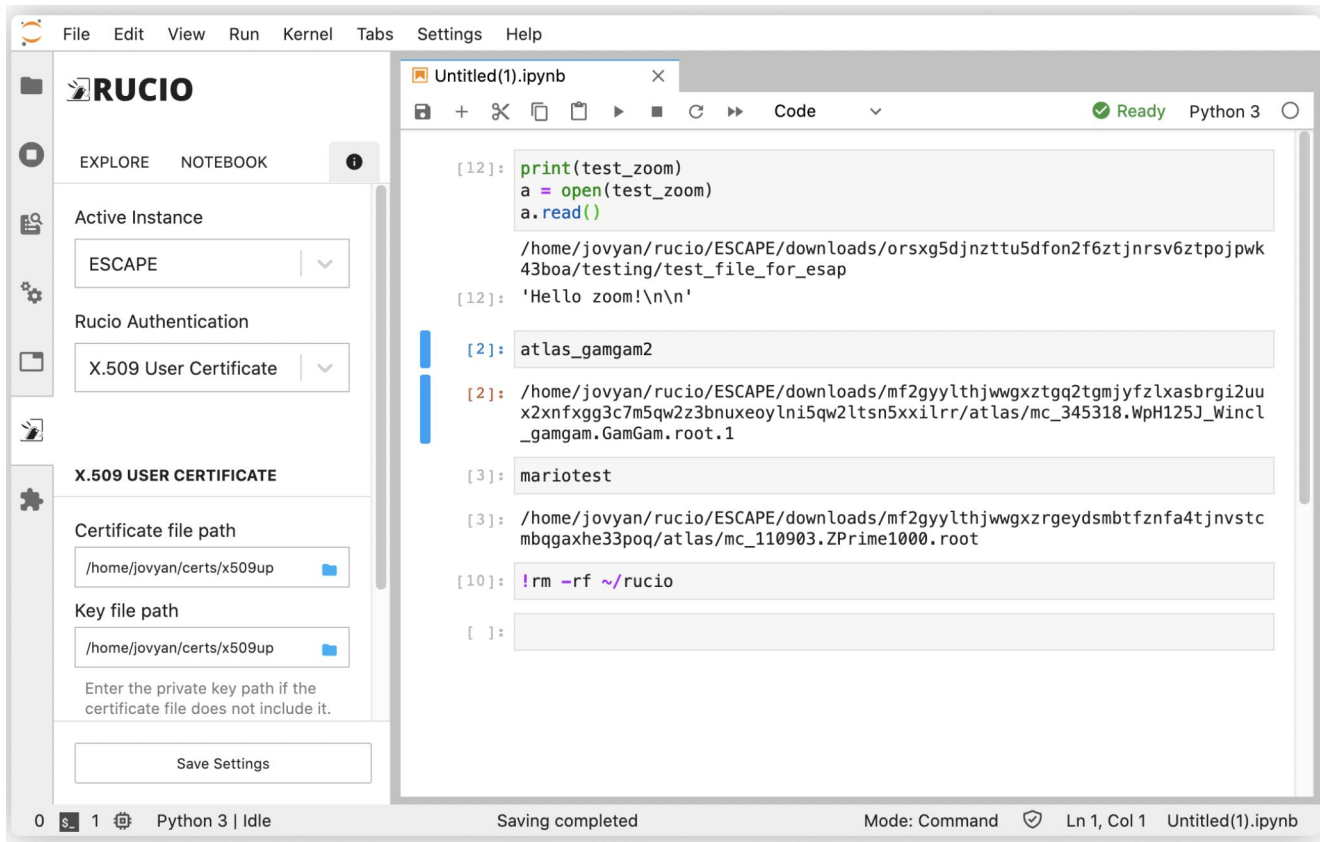
sigma: 10.00
beta: 2.67
rho: 28.00

lorenz.py:

```
9 def solve_lorenz(N=10, max_time=4.0, sigma=10.0, beta=8./3, rho=28.0):
10     """Plot a solution to the Lorenz differential equations."""
11     fig = plt.figure()
12     ax = fig.add_axes([0, 0, 1, 1], projection='3d')
13     ax.axis('off')
14
15     # prepare the axes limits
16     ax.set_xlim((-25, 25))
17     ax.set_ylim((-35, 35))
18     ax.set_zlim((5, 55))
19
20     def lorenz_deriv(x_y_z, t0, sigma=sigma, beta=beta, rho=rho):
21         """Compute the time-derivative of a Lorenz system."""
22         x, y, z = x_y_z
23         return [sigma * (y - x), x * (rho - z) - y, x * y - beta * z]
24
25     # Choose random starting points, uniformly distributed from -15 to 15
26     np.random.seed(1)
27     x0 = -15 + 30 * np.random.random((N, 3))
28
```

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.



The screenshot shows the JupyterLab RUCIO extension interface. The left sidebar contains the RUCIO logo and navigation tabs for EXPLORE and NOTEBOOK. Below these are configuration options for the Active Instance (set to ESCAPE) and Rucio Authentication (set to X.509 User Certificate). The X.509 USER CERTIFICATE section includes fields for Certificate file path and Key file path, both pointing to /home/jovyan/certs/x509up, and a Save Settings button. The main area displays a Jupyter Notebook with code cells for interacting with RUCIO datasets.

```
[12]: print(test_zoom)
      a = open(test_zoom)
      a.read()

/home/jovyan/rucio/ESCAPE/downloads/orsxg5djnzttu5dfon2f6ztjnrvsv6ztpojpwk
43boa/testing/test_file_for_esap

[12]: 'Hello zoom!\n\n'

[2]: atlas_gangam2

[2]: /home/jovyan/rucio/ESCAPE/downloads/mf2gyylthjwngxztgq2tgmjyflzasbrgi2uu
x2xfxgg3c7m5qw2z3bnuxeoylni5qw2ltsn5xxilrr/atlas/mc_345318.WpH125J_Winc1
_gangam.GamGam.root.1

[3]: mariotest

[3]: /home/jovyan/rucio/ESCAPE/downloads/mf2gyylthjwngxzrgeydsmbtfzfa4tjnvstc
mbqgaxhe33poq/atlas/mc_110903.ZPrime1000.root

[10]: !rm -rf ~/rucio

[ ]:
```


ATLAS

Open Data

We deploy the resources on the Internet.

In a nutshell, they are a series of

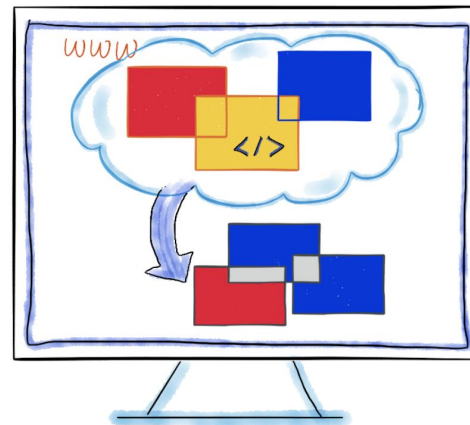
- Data samples in ROOT n-tuple format
- Software and Jupyter [Notebooks](#) in Python and C++ to analyse the samples and produce physics analysis
- JavaScript (JS) applications to produce cut-and-count analysis
- Virtual Machines with several Linux-based OS and ROOT CERN analysis framework
- [GitHub](#) & [GitLab](#) repositories
- GitBooks to document the several possible activities that can be performed

Data & Tools Repository

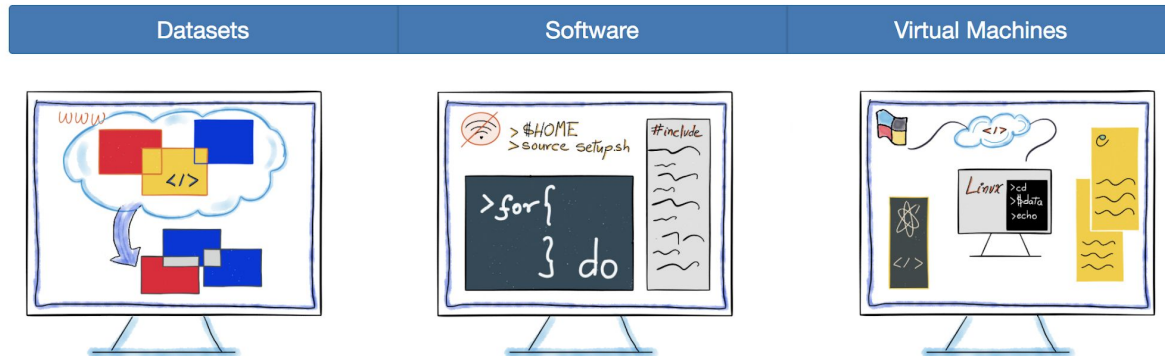
Here you have in a single place all the necessary pieces in order to start your physics analysis in a more complete way. Look into the data like an ATLAS particle physicist!

In this section, you can find where to download:

- The complete collection of available datasets
- The different analysis software
- The virtual machines to perform physics searches

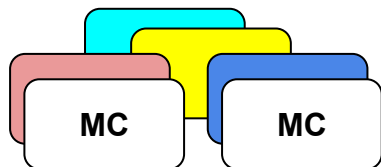
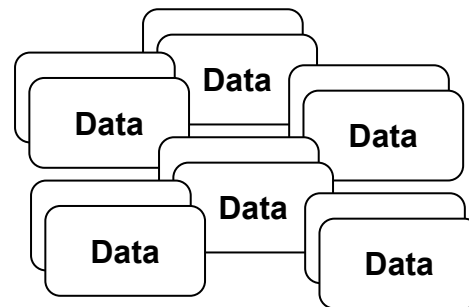


Downloads

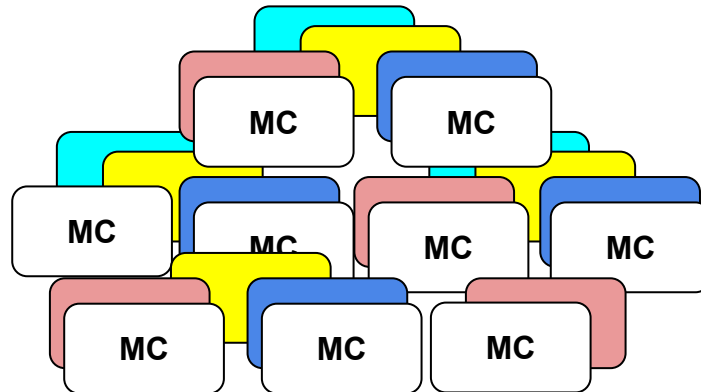


8 TeV release

13 TeV release

 1 fb^{-1}  10 fb^{-1} 

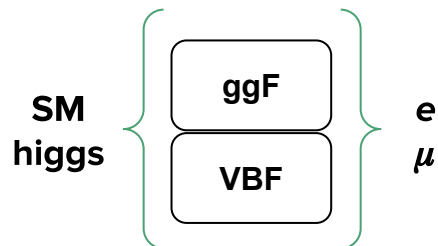
44 samples



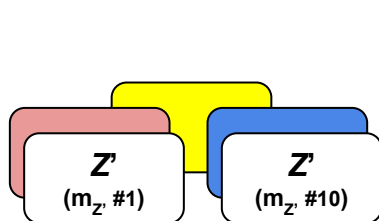
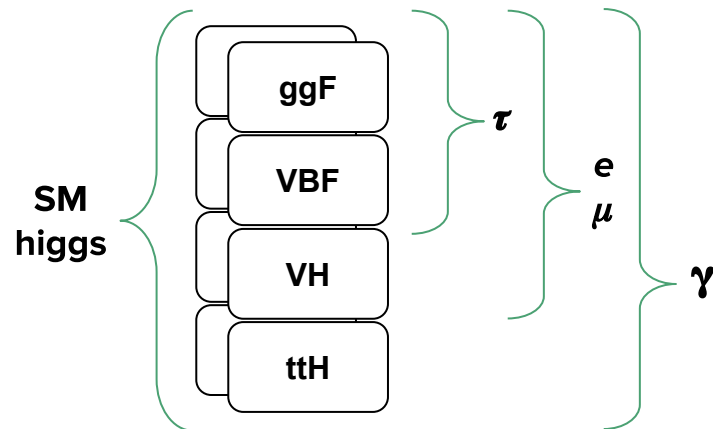
~120 samples

$\geq \mathbf{X7}$
Collections
based in
final states

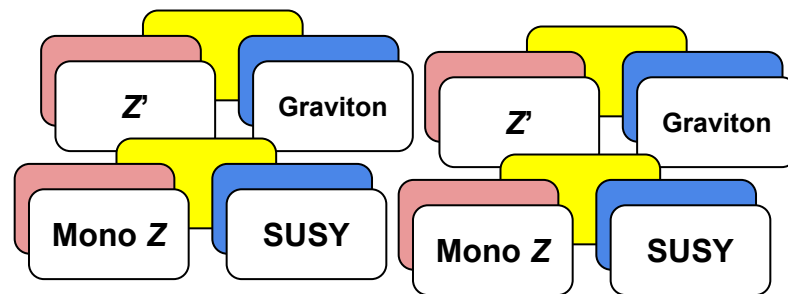
8 TeV MC signals



13 TeV MC signals



14 BSM samples

 ≥ 50 BSM samples

Jupyter Notebooks

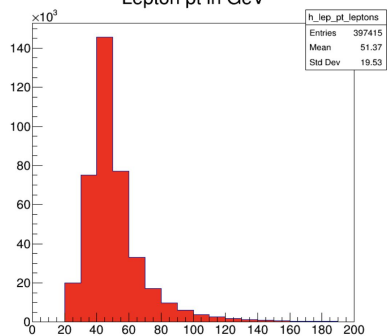
```

ATLAS_OpenData_06-cpp_simple_cut_and_count_analysis_example (unsaved changes)
File Edit View Insert Cell Kernel Help | ROOT Prompt
In [10]:
for (i=0; i<events_to_run; i++)
{
    nbytes = dataset->GetEntry(i);
    if(lepton_n>1) // Number of leptons in the events has to be at least 2
    {
        if(lepton_type[0] == lepton_type[1]) //Leptons of the same family, i.e. 2 electrons or 2 muons (those are the t
        {
            if(lepton_charge[0] != lepton_charge[1]) // The two selected leptons must have opposite charge
            {
                //PT
                float lepton_pt_inGeV = lepton_pt[0]/100.; // The default value in the root file is in MeV, so, we div
                h_lep_pt_leptons->Fill(lepton_pt_inGeV);
            }
        }
    }
}

In [11]: TCanvas *cs = new TCanvas("cs","cs",10,10,700,700);
h_lep_pt_leptons->Draw();
cs->Draw();

```

Lepton pt in GeV



Use
ROOTJS

```

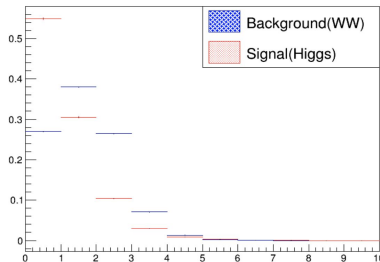
ATLAS_OpenData_07-python_simple_two_samples_comparison (autosaved)
File Edit View Insert Cell Kernel Help | Python 2
In [9]:
b2_bgs.SetFillStyle(3008)
b2_bgs.SetFillColor(4)
b2_sig.SetFillStyle(3003)
b2_sig.SetFillColor(2)

legend=ROOT.TLegend(0.5,0.7,0.9,0.9)
legend.AddEntry(b2_bgs,"Background(WW)","f")
legend.AddEntry(b2_sig,"Signal(Higgs)","f")

b2_sig.SetLineColor(2)
b2_sig.SetState(0)
b2_bgs.SetState(0)
b2_sig.Draw()
b2_bgs.Draw("same")
legend.Draw()
c.Draw()

```

Example plot: Number of Jets



Jupyter notebooks can run ROOT commands

- We produce a series of examples for basic 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.
- The notebooks can read the samples directly from the Internet (using http protocol) or run local (if there is limited Internet access)

The pieces together:

ATLAS Jupyter Notebooks and
JupyterLab RUCIO extension

The ATLAS Open Data as a test field

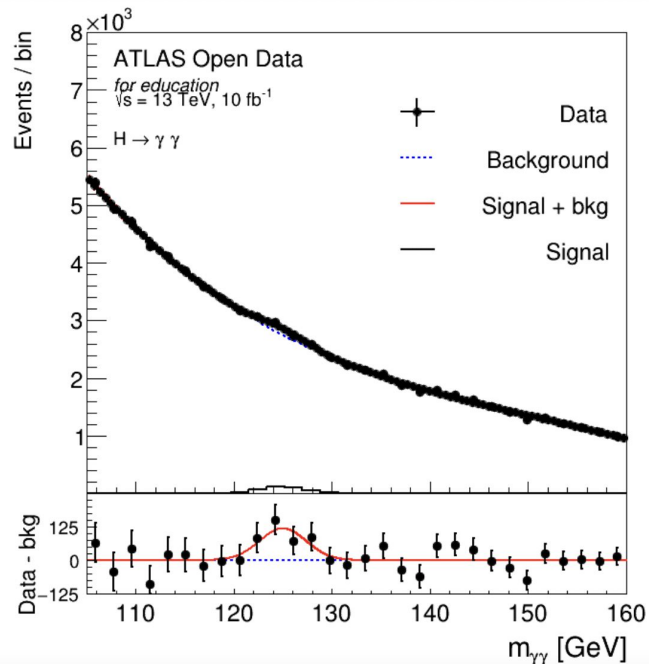
```
In [8]: for i in range(len(hyy_0)):
        print(i, hyy_0[i])
```

```
0 /home/jovyan/rucio/ESCAPE/downloads/mf2gyylthjsxqz1smnuxgzk7jb
1 /home/jovyan/rucio/ESCAPE/downloads/mf2gyylthjsxqz1smnuxgzk7jb
2 /home/jovyan/rucio/ESCAPE/downloads/mf2gyylthjsxqz1smnuxgzk7jb
3 /home/jovyan/rucio/ESCAPE/downloads/mf2gyylthjsxqz1smnuxgzk7jb
4 /home/jovyan/rucio/ESCAPE/downloads/mf2gyylthjsxqz1smnuxgzk7jb
5 /home/jovyan/rucio/ESCAPE/downloads/mf2gyylthjsxqz1smnuxgzk7jb
6 /home/jovyan/rucio/ESCAPE/downloads/mf2gyylthjsxqz1smnuxgzk7jb
7 /home/jovyan/rucio/ESCAPE/downloads/mf2gyylthjsxqz1smnuxgzk7jb
8 /home/jovyan/rucio/ESCAPE/downloads/mf2gyylthjsxqz1smnuxgzk7jb
```

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

Once the Open Data datasets are registered in RUCIO (*like is the case for the current 13 TeV samples, like those in this Higgs into two photons example, thanks to Stephane*) they can be downloaded and read, using the JupyterLab extension, including search features

Example in nbviewer.jupyter.org



The ATLAS Open Data as a test field

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 $H \rightarrow WW$ 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 $H \rightarrow ZZ$ decay channel in the four-lepton final state

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

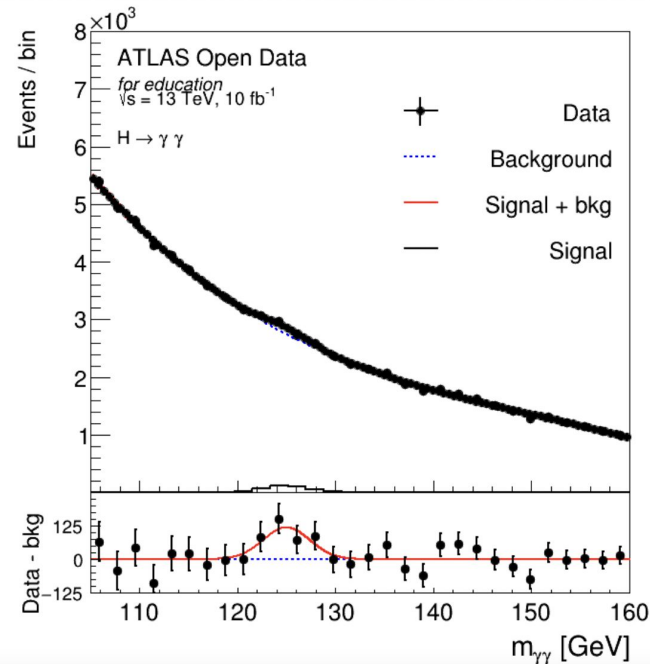
Search for BSM $Z' \rightarrow t\bar{t}$ in the single-lepton boosted final state

SM Higgs boson production in the $H \rightarrow \gamma\gamma$ decay channel in the two-photon final state

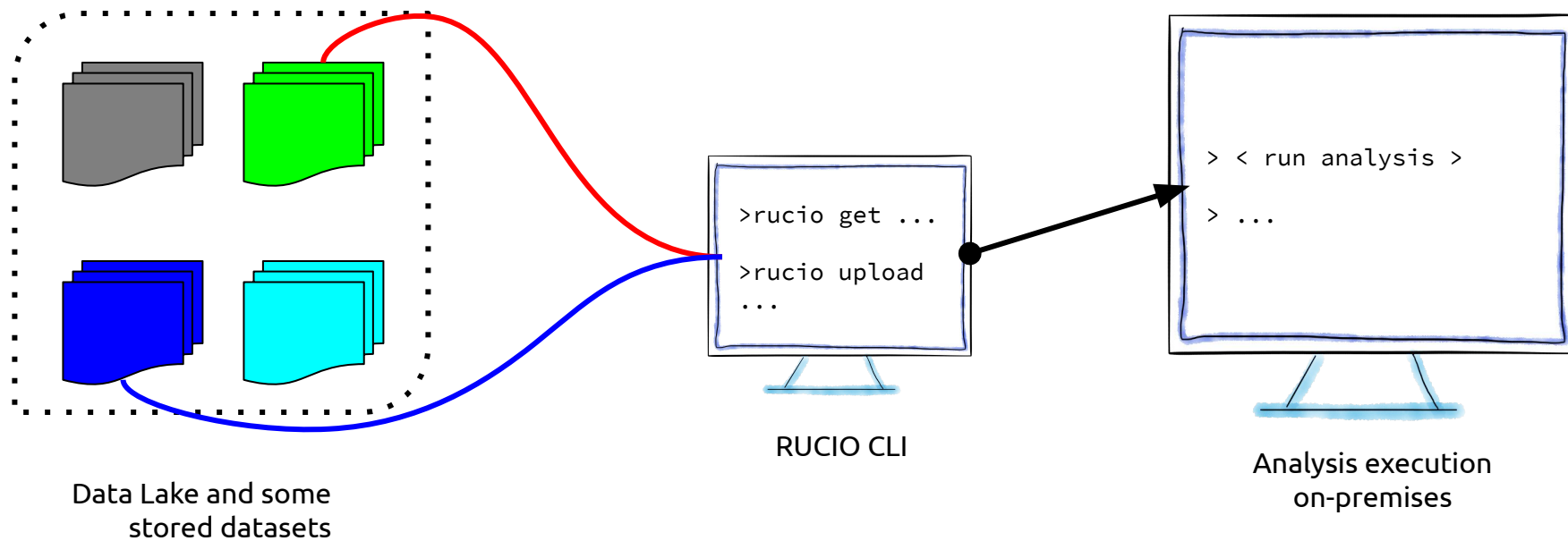
Much more computational complex particle physics analysis already exist and they will be used/converted and improved when needed so to be a proper set of analysis examples as close as possible to “real” analysis, using the existing publicly available data

More in [Opendata.atlas.cern - documentation 13 TeV - physics](https://opendata.atlas.cern/documentation/13TeV-physics).

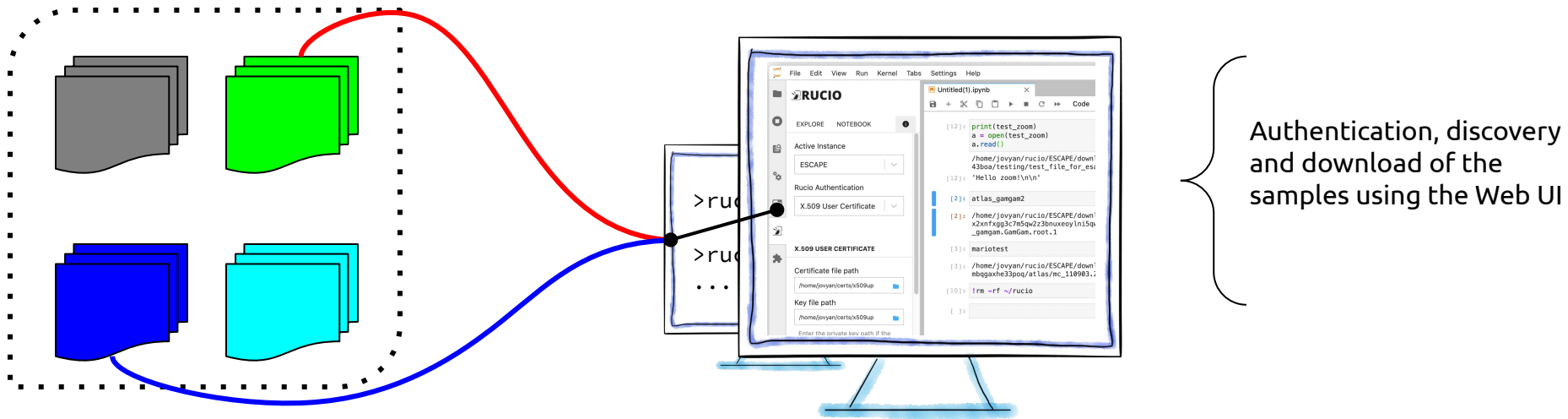
```
In [10]: show_image('histograms/hist_mYY_bin1.png')
```



Recap



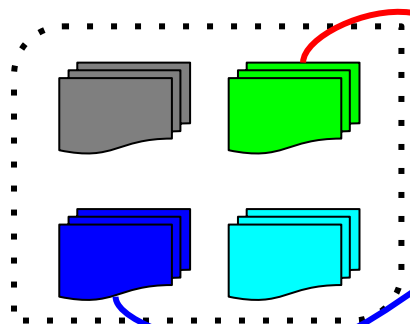
Traditional interaction with samples



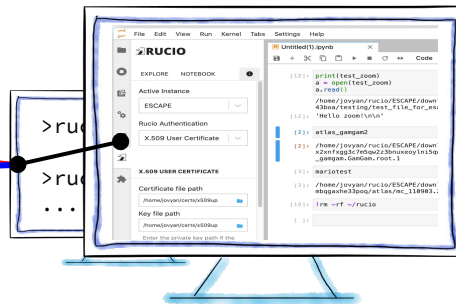
Data Lake and stored
Open Data datasets

RUCIO JupyterLab
extension

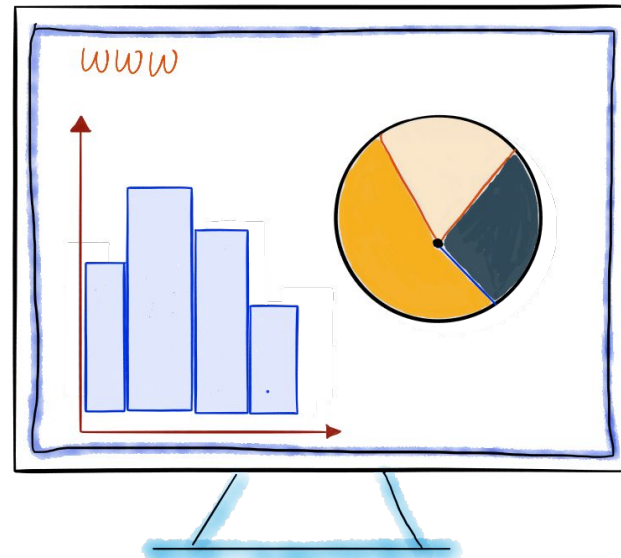
RUCIO+JupyterLab proposal for end users



Data Lake and stored
ATLAS OD datasets



RUCIO JupyterLab
extension on ATLAS OD
notebooks for testing



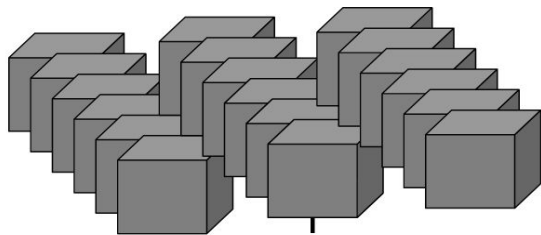
Analysis code, results and
visualisation

A view of the service

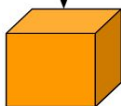
Summary

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.

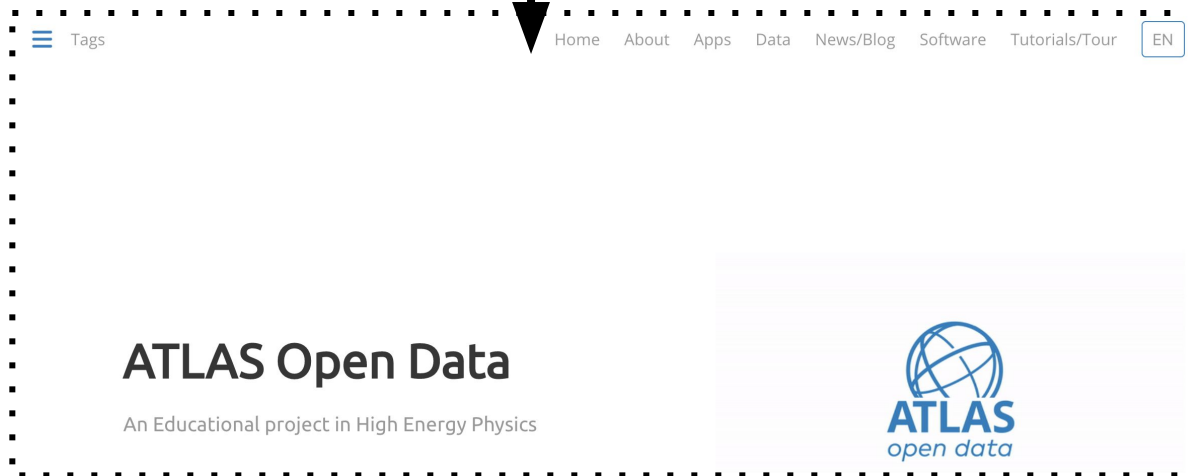
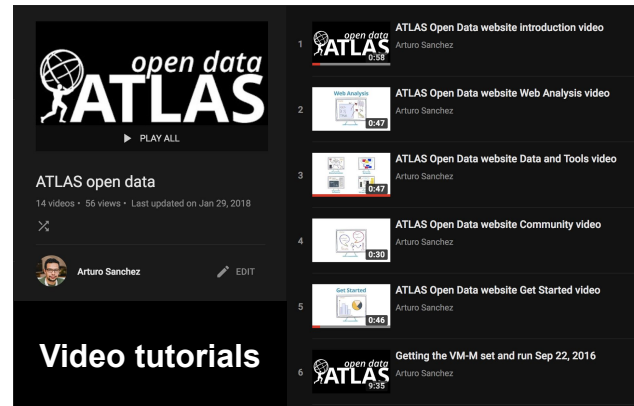
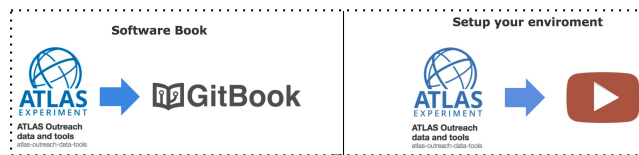
Backup



datasets



The ATLAS Open Data project ([more here](#)) aims to release real and simulated Data, together with Open Source software resources to analyse those samples. As well as documentation in several useful formats!

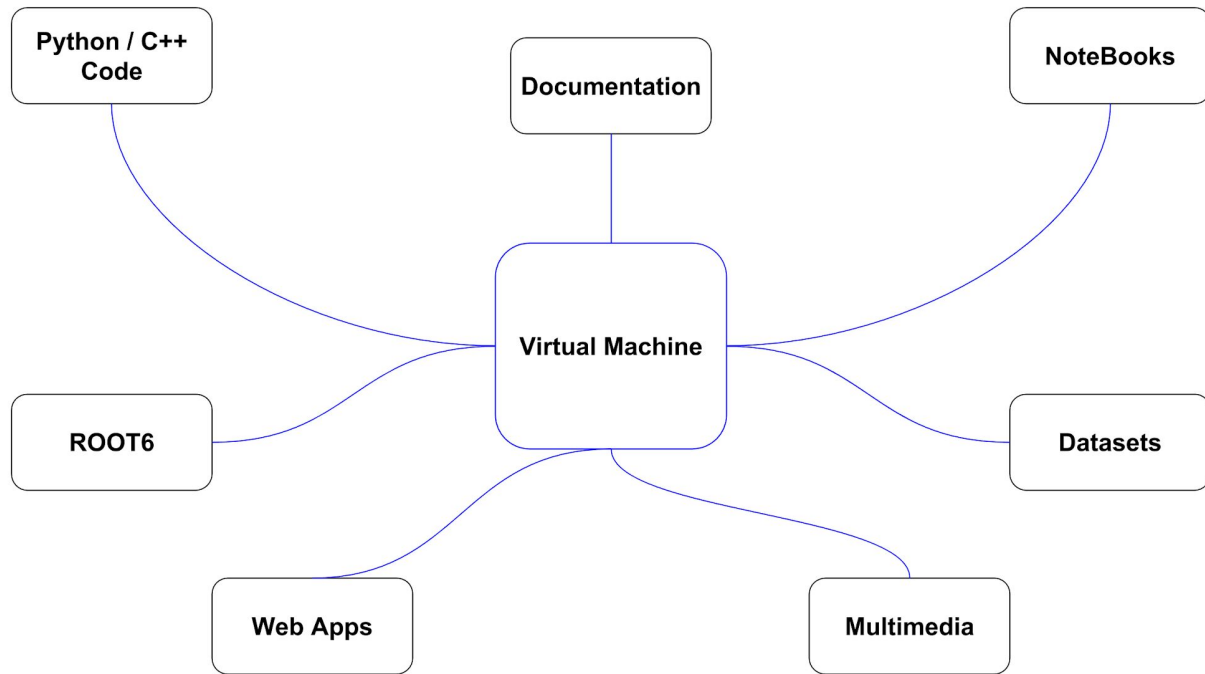


<http://opendata.atlas.cern/new>

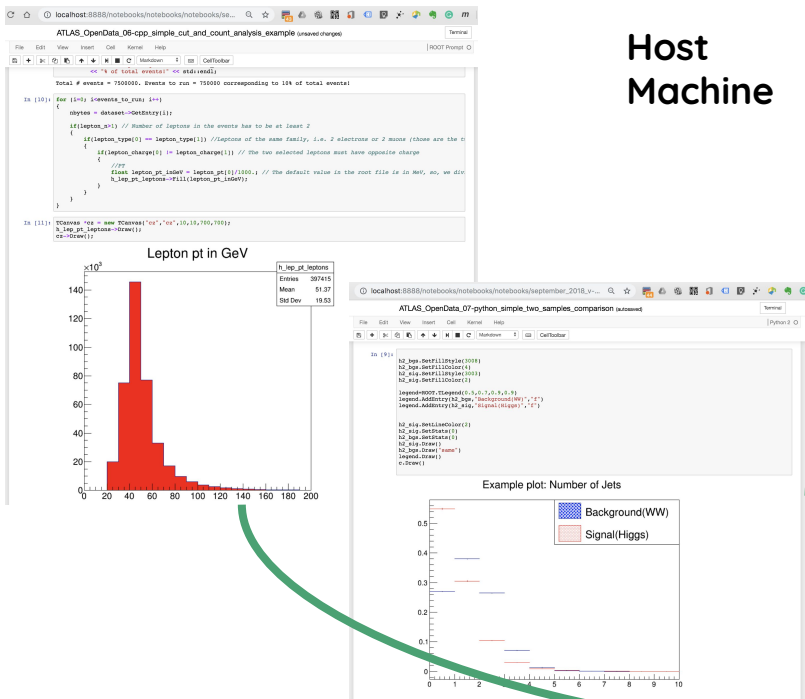
Virtual Machines as Servers

Quick view of the anatomy of the ATLAS Open Data VM

- Based in a Linux-kind OS with standard graphical UI
- The OS is enhanced with *all* ROOT's needed libraries and dependencies
- ROOT5 or ROOT6 analysis framework and IPython
- The Open **Datasets and Software** analysis frameworks
- Jupyter-notebook technology and Examples Notebooks
- Documentation in form of PDFs and Video tutorials.

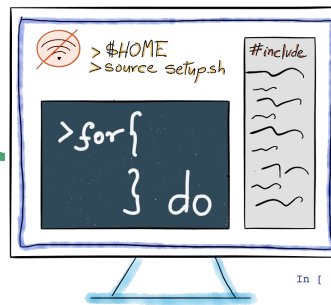
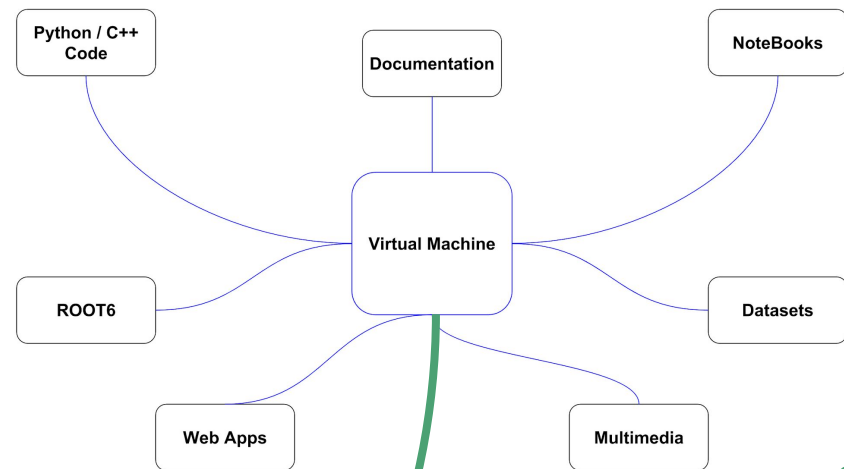


Host Machine



Jupyter notebooks can run ROOT commands and other Python libraries and tools

- The notebooks use both the Python and the C++ ROOT kernel to produce results using the VM as a server, teaching as well the principles of Cloud and Distributed Computing.



Uses as a Server

```

In [ ]: import sys
        !{sys.executable} -m pip install --upgrade --user pip
        !{sys.executable} -m pip install -U numpy pandas uproot matplotlib keras scikit-learn --user

In [ ]: import os
        import csv
        import pandas as pd
        import numpy as np
        import matplotlib.pyplot as plt
        from pandas import read_csv
        from matplotlib import pyplot
  
```