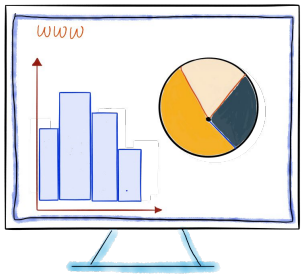
ESCAPE: update on RUCIO + JupyterLab + ATLAS Open Data integration



Arturo Sánchez Pineda

7-10th May 2021, LAPP

Overview

A series of exercises to perform during the ESCAPE DAC21 in November
https://docs.google.com/document/d/1mHZiVA7S2mgRKyMVUb39fwT9OUUfKZjHAlsTWc6hrs4/

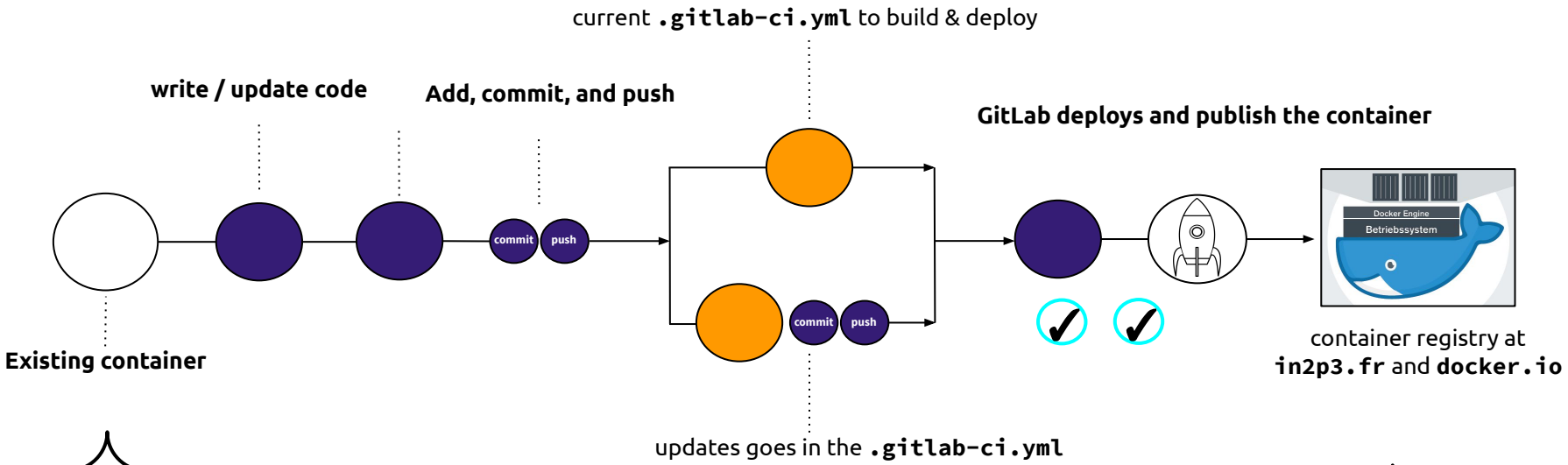
- Data “multiplication” where multiple version of the same data is generated, simulating a data-augmentation process
- Writing of such “multiplied” data back to the Datalake
- Exercises include the analysis of data stored in the Datalake
- Create clear instructions for other users that can be part of the challenge

The Jupyter client and RUCIO + ATLAS Open Data as a demo construction for final users
http://universidad.ch/escape-demo/suite/

- Activities relative to a integration and consolidation of the **Data Lake usage** via RUCIO CLI client and a friendly UI, JupyterLab + rucio-extension
- Efforts to consolidate those in a single collection of containers
- And how ATLAS Open Data is used as a analysis test pool for such integration for “normal” users
- Explore how this can be used no only for *single-user-laptop* or *single-user-vm* but integrated as a modular way to deliver containers in a multi-user platform like JupyterHub

Containers review

A reminder



Container CI / CD

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

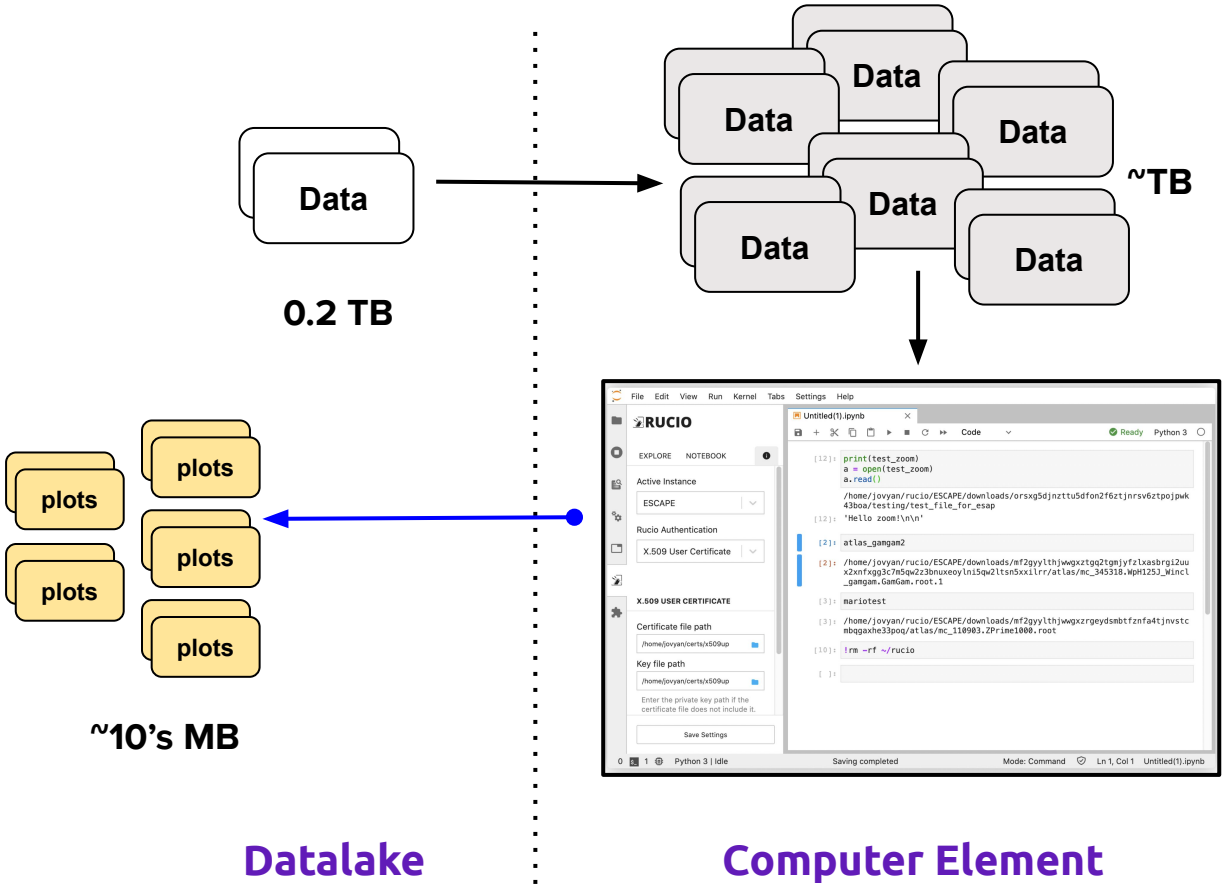
Several developments and deployment already in place

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

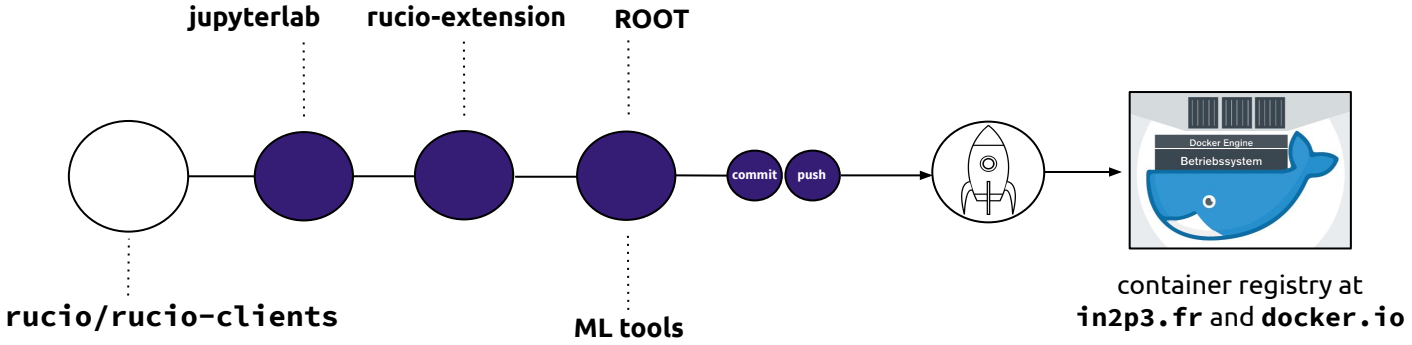
Analysis examples

We can also run the analysis examples over the “multiplied” data

- This can help to simulate longer analysis that can last several hours
- In case this kind of “stress” is useful in this challenge



BASE IMAGE = `rucio/rucio-clients`

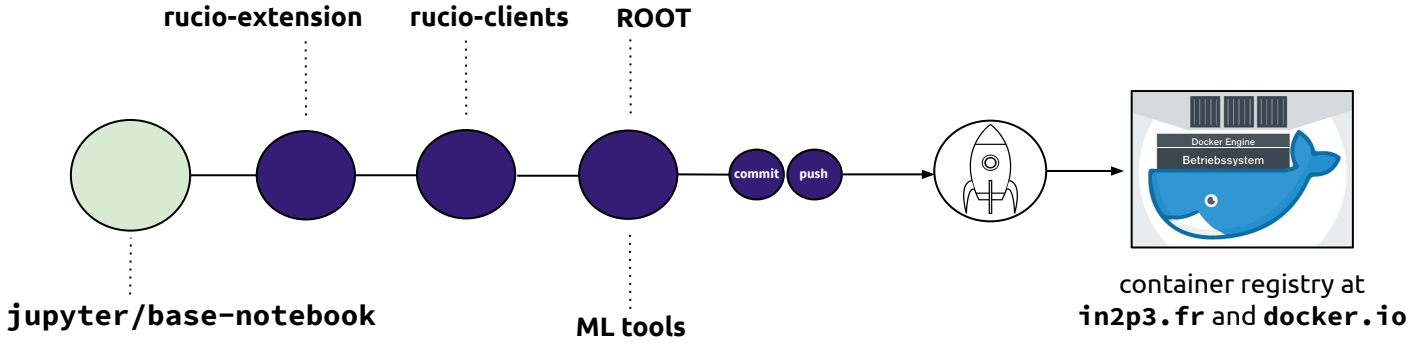


The containers start with a base image

- The one I have been using from CTA uses the rucio-clients as base image
- From there, I add extra HEP-related tools

- We are exploring different base container and software structure to create those to be used by the users with JupyterLab + rucio + other tools

BASE IMAGE = `jupyter/base-notebook`



The containers start with a base image

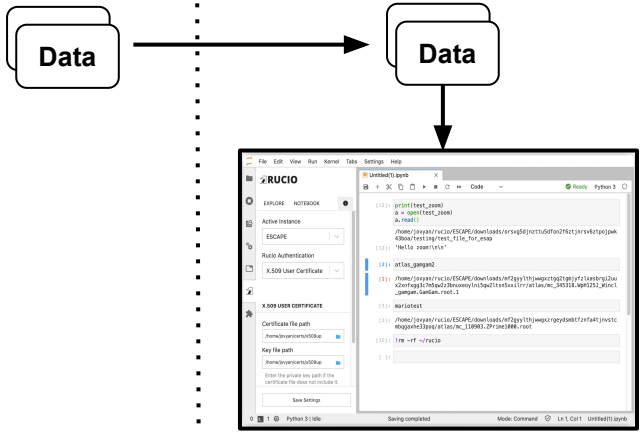
- In ATLAS Open Data we have being working also in containers that use jupyter as the base
- From there, I add extra HEP-related tools

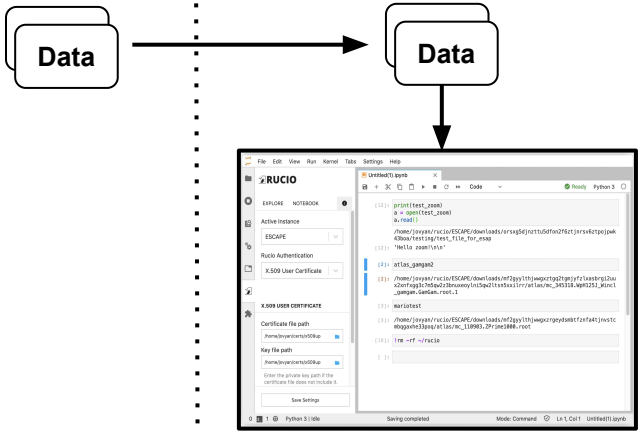
- This activity is done in synch with ongoing developments in ATLAS Open Data project to deliver containers for training in HEP

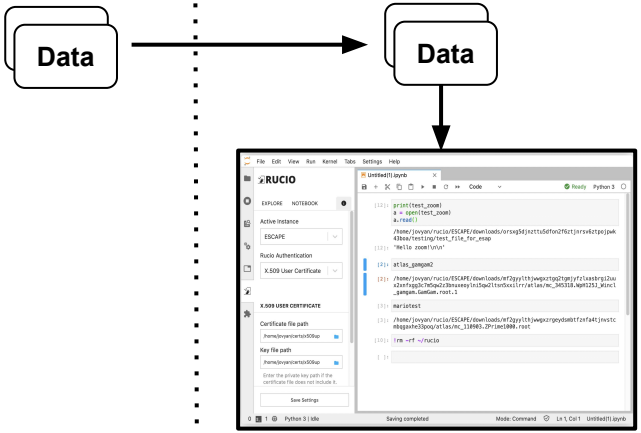
Analysis examples

We can use multiple computer elements performing analysis at the same time

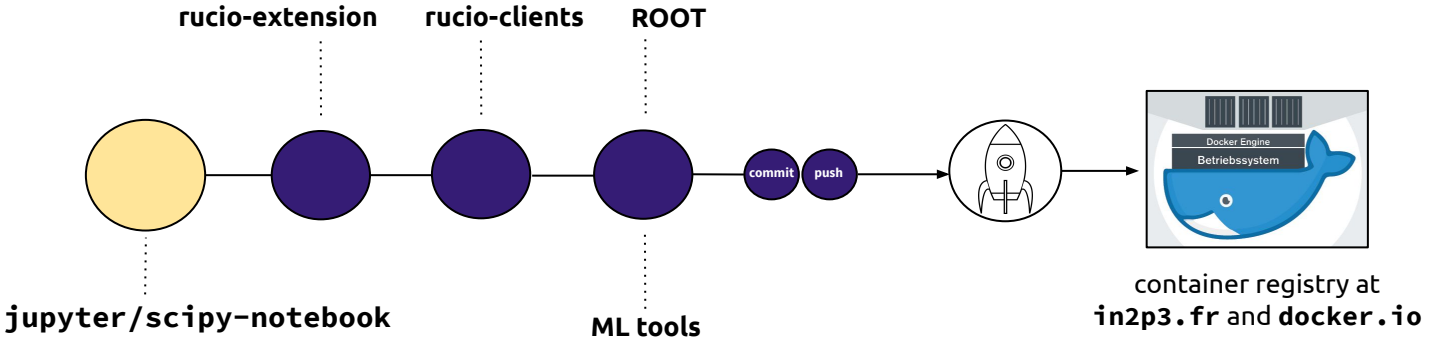
- multiple users with a single machine and/or a single user with multiple machines







BASE IMAGE = `jupyter/scipy-notebook`

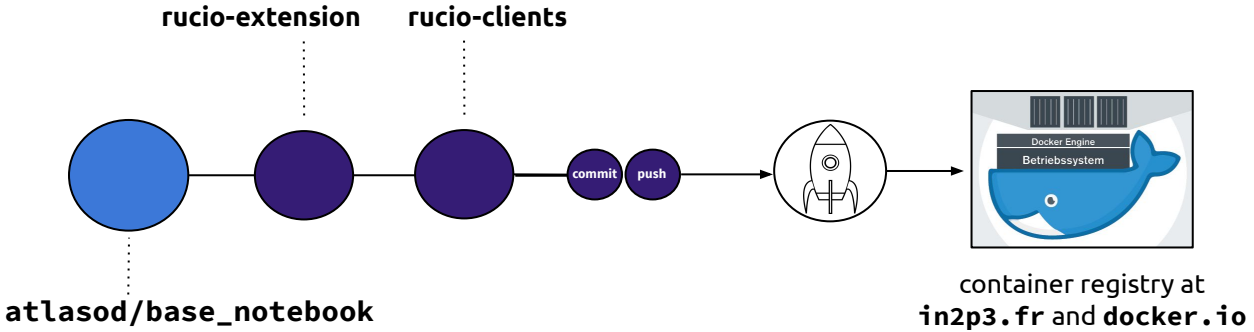


The containers start with a base image

- Jupyter has a family of “popular” containers that can be used to further decrease customizations from our side, and building times
- The idea is to the the flexibility of using different bases

- Allow us to use a combination of containers using JupyterHub Docker Spawner to deploy multiple-user JupyterHub, e.g. in already existing infrastructure

BASE IMAGE = `atlasod/base_notebook`



The containers start with a base image

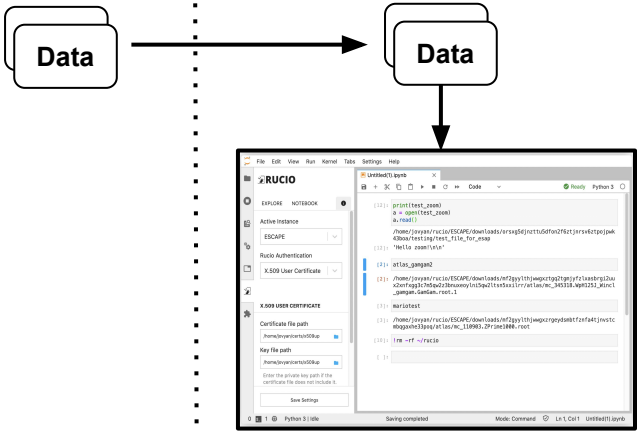
- In this approach, we can imaging adding the rucio functionalities to already established “experiment containers”
- In this examples, ATLAS Open Data has a container that is enhanced with ESCAPE-rucio tools.

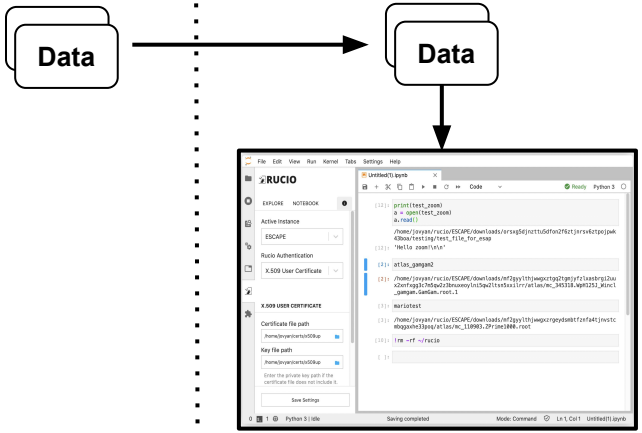
- This may reduce significantly the maintenance time and rely on well-supported tools (OS, jupyter, python, LCG, RUCIO, ROOT) as much as possible. Also, allow for an easier way to keep the container up to date

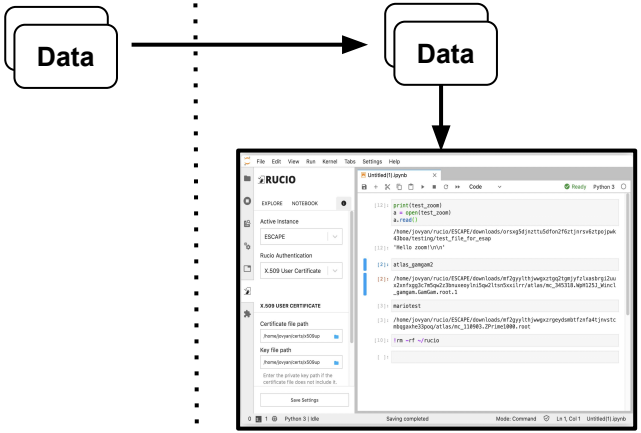
Multiple-user analysis

We can also:

- Create a volatile a JupyterHub that for a multiple-user usage of the datalake
  - CERN OpenStack resources are a feasible option.
- Try to integrate the exercises with volunteer computer services like MyBinder and Google Colab (?)

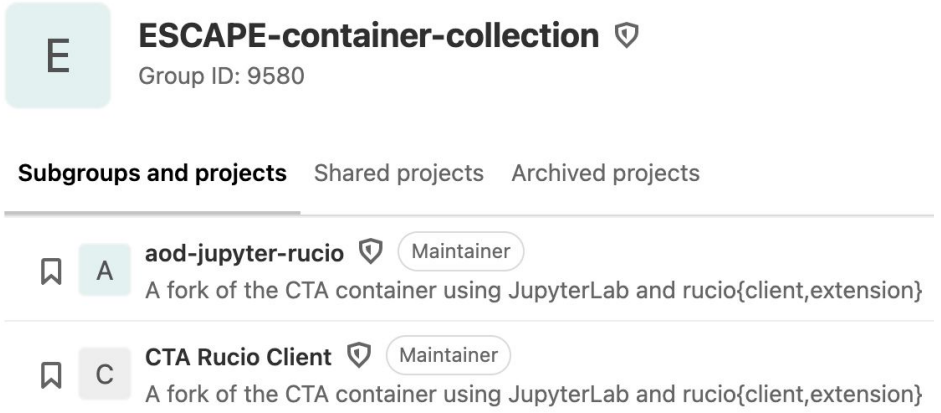




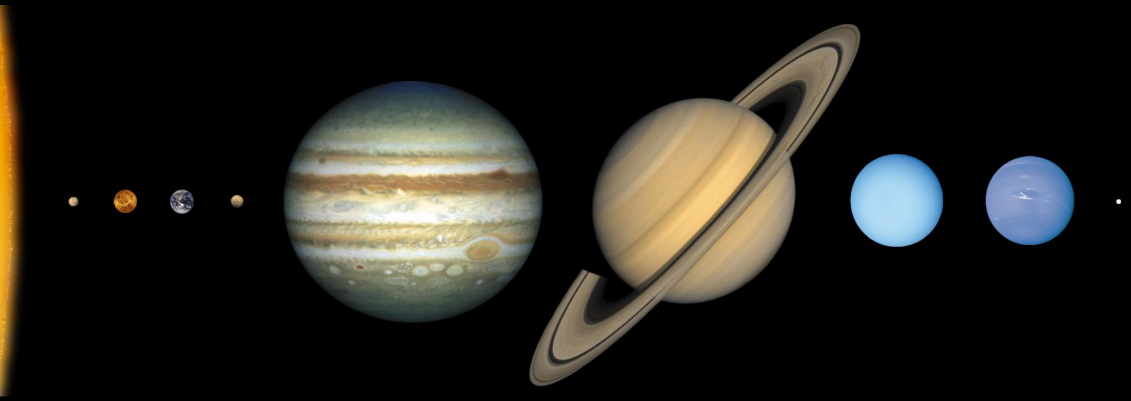


A containers collection

Such tests ancontaines are growing in https://gitlab.in2p3.fr/container-collection



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**?**