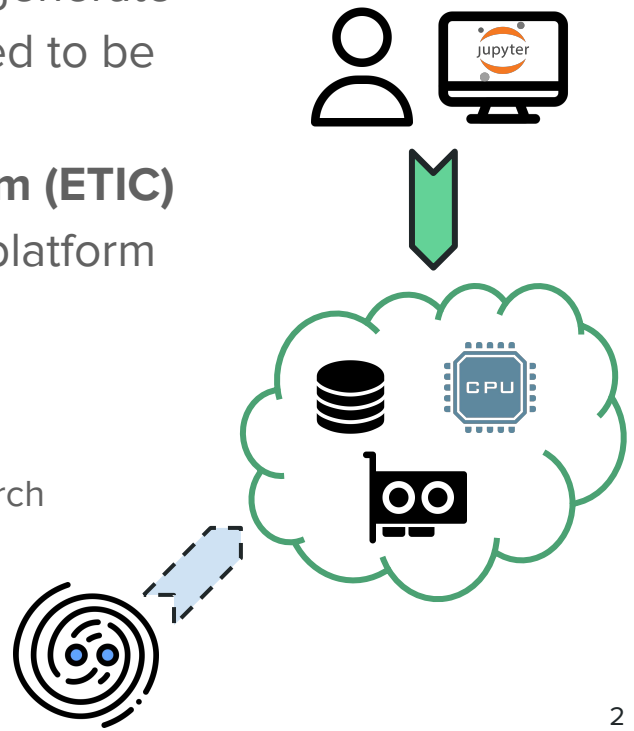
Building a Virtual Research Environment (VRE) for ET Bologna

Stefano Dal Pra, Daniele Bonacorsi, Tommaso Chiarusi, Elena Cuoco, Tommaso Diotalevi, Antonio Ghinassi, Marco Lorusso, Giovanni Mastropasqua, Andrea Paolucci, Riccardo Travaglini, Paolo Veronesi

Opatija (Croatia), 12/11/2025

Introduction

- Once operational, the **Einstein Telescope** (ET) will generate a large amount of experimental data, which will need to be efficiently managed and analyzed.
- As part of the national **ET Infrastructure Consortium (ETIC)** project, the main goal is to build a heterogeneous platform capable of connecting the analysts with a cloud infrastructure providing modern facilities:
  - Based on open-source standards (Jupyter, Kubernetes, ...);
  - High-level libraries and frameworks, for a easy-to-use research environment.



Einstein Telescope in Bologna

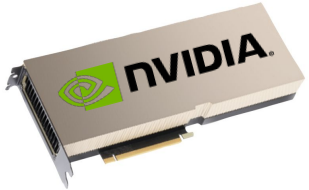
- The Bologna ET operational unit (BoET) is a joint effort between INFN and University of Bologna.
- **Scientific-technical goals**:
  - Development of a **computing cluster** (BETIF/DIFAET) for R&D through the setup of a heterogeneous infrastructure;
  - Development of electronic boards (in synergy with VIRGO) for super-attenuators;
  - DAQ test prototypes (developed for the VIRGO upgrade);
  - Time synchronization system based on White Rabbit;
  - Much more (theoretical studies, data analysis, …)

This talk!

Hardware specifications

- **CPUs**: 2 × Intel® Xeon® Platinum 8462Y+ (64C / 128T);

- **GPUs**: 2x NVidia® H100;



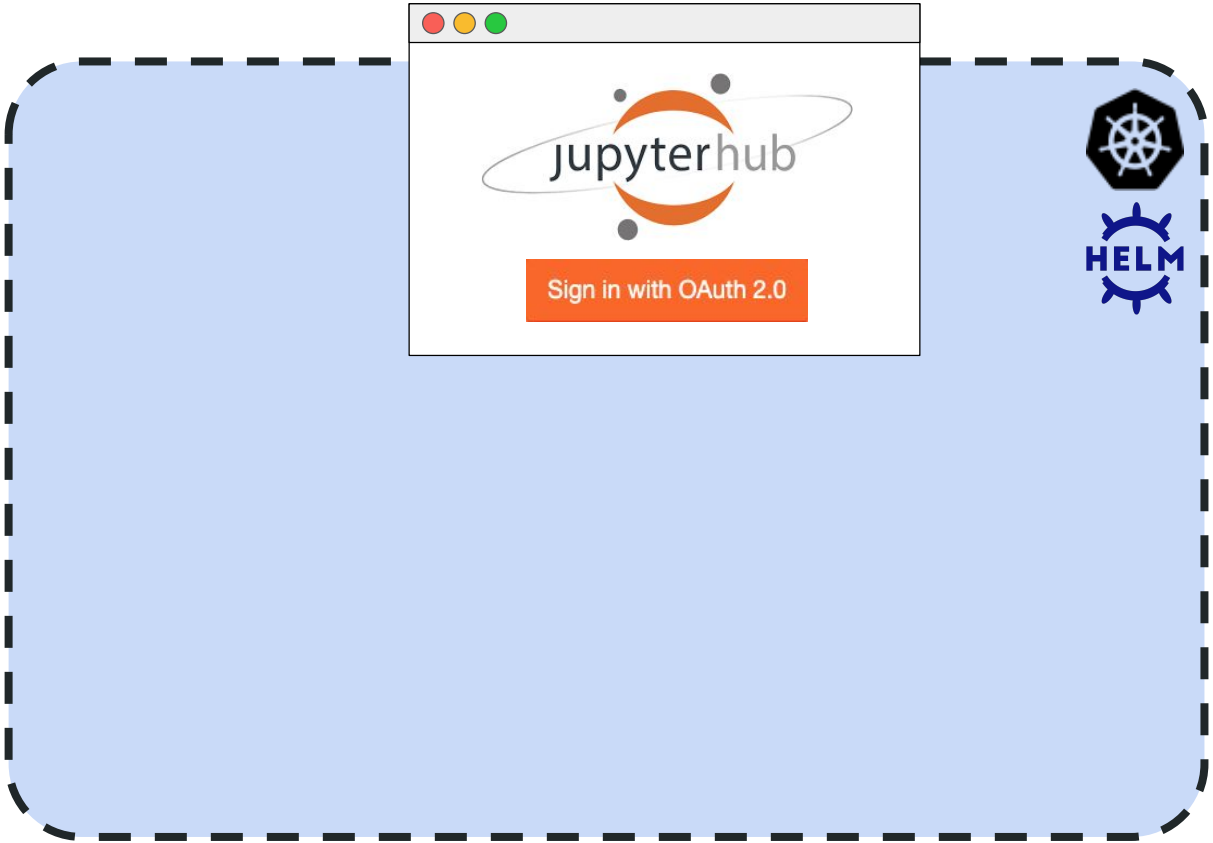
- **FPGAs**: 2x AMD® Alveo U55C, 3x AMD®Alveo V70,
  2x AMD® Alveo X3, 1x AMD® Alveo U250, 1x AMD® Alveo U45N.

- Memory: 2TB DDR5;
- Networking: 2x10 Gbps Ethernet + 2x40 Gbps Fiber Optic;
- Storage: 4x15TB PCIe NVMe + 1TB SSD. Configured as RAID 10 for redundancy.

Our prototypal Virtual Research Environment

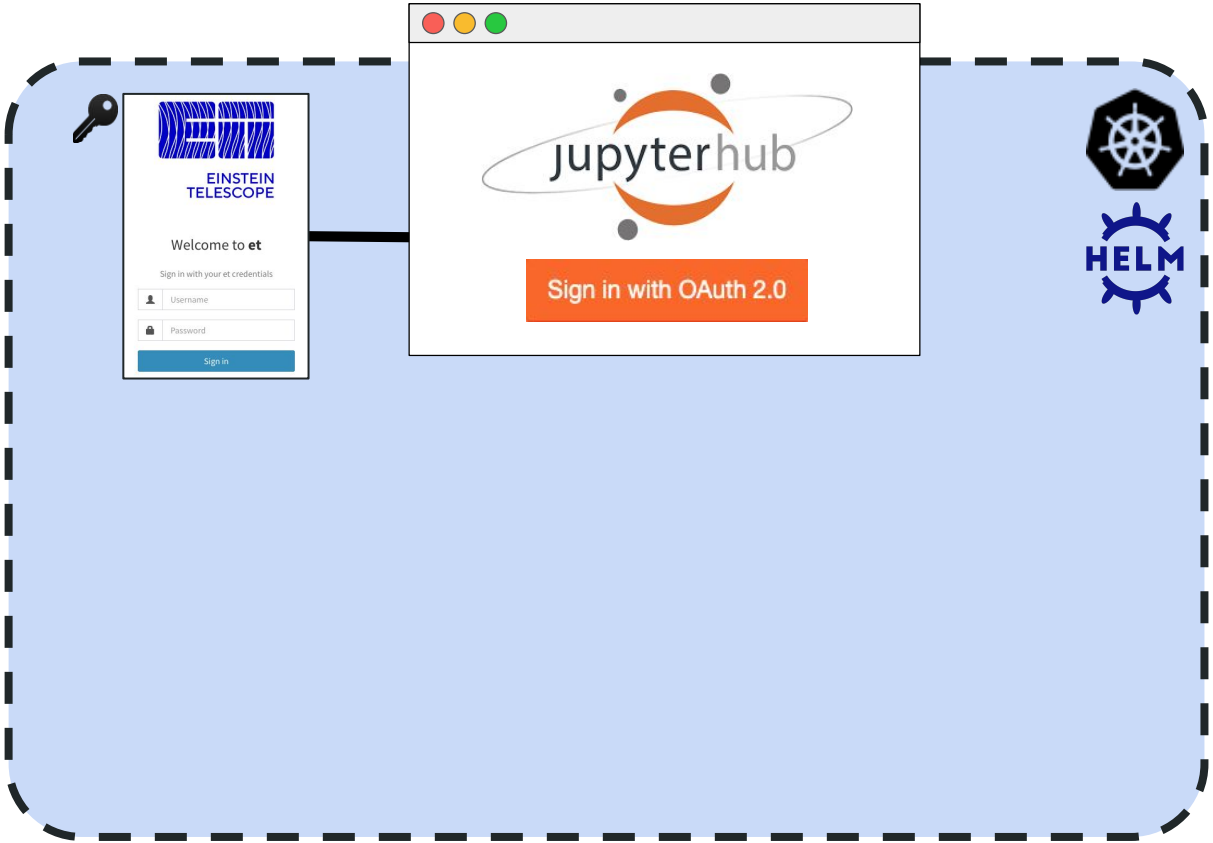
- Built directly from bare metal;
- Based on open-source **Kernel-based Virtual Machine** (KVM) virtualization module;
- Managed using **libvirt** API, creating a series of independent virtual nodes;
- GPU passthrough via **vfio-pci** dedicated driver
- Cluster managed with **Rancher Kubernetes Engine** (RKE2);
- Platform deployed using **HELM** package manager for K8s;
- User interface based on the **Jupyter** framework;
- Highly customizable user environment via **Docker** containers;
- Customizable python kernels, containerised and available using CERN's **CVMFS**;
- Data IO managed by **Rucio** directly in the platform interface.

Architecture



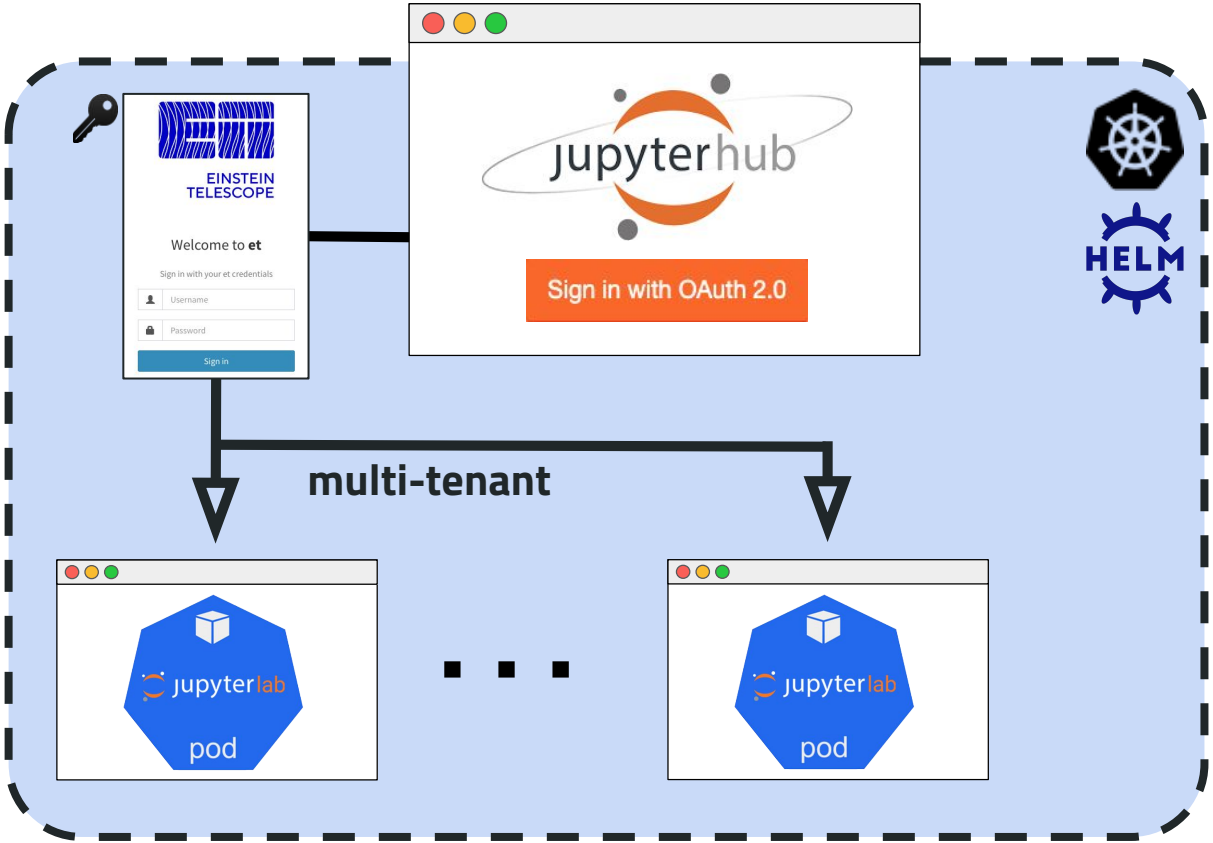
- Access to single hub entrypoint;

Architecture



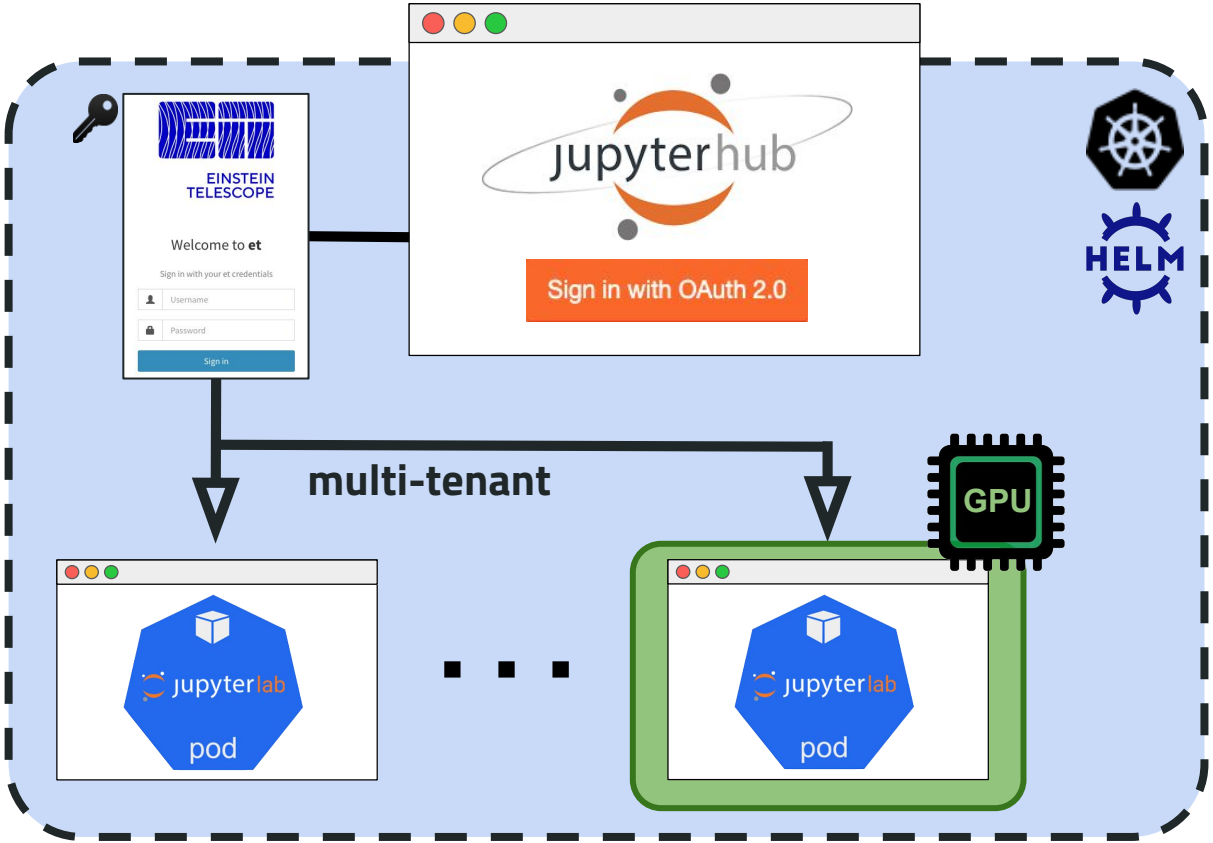
- Access to single hub entrypoint;
- Authn and authz token-based via INDIGO-IAM;

Architecture



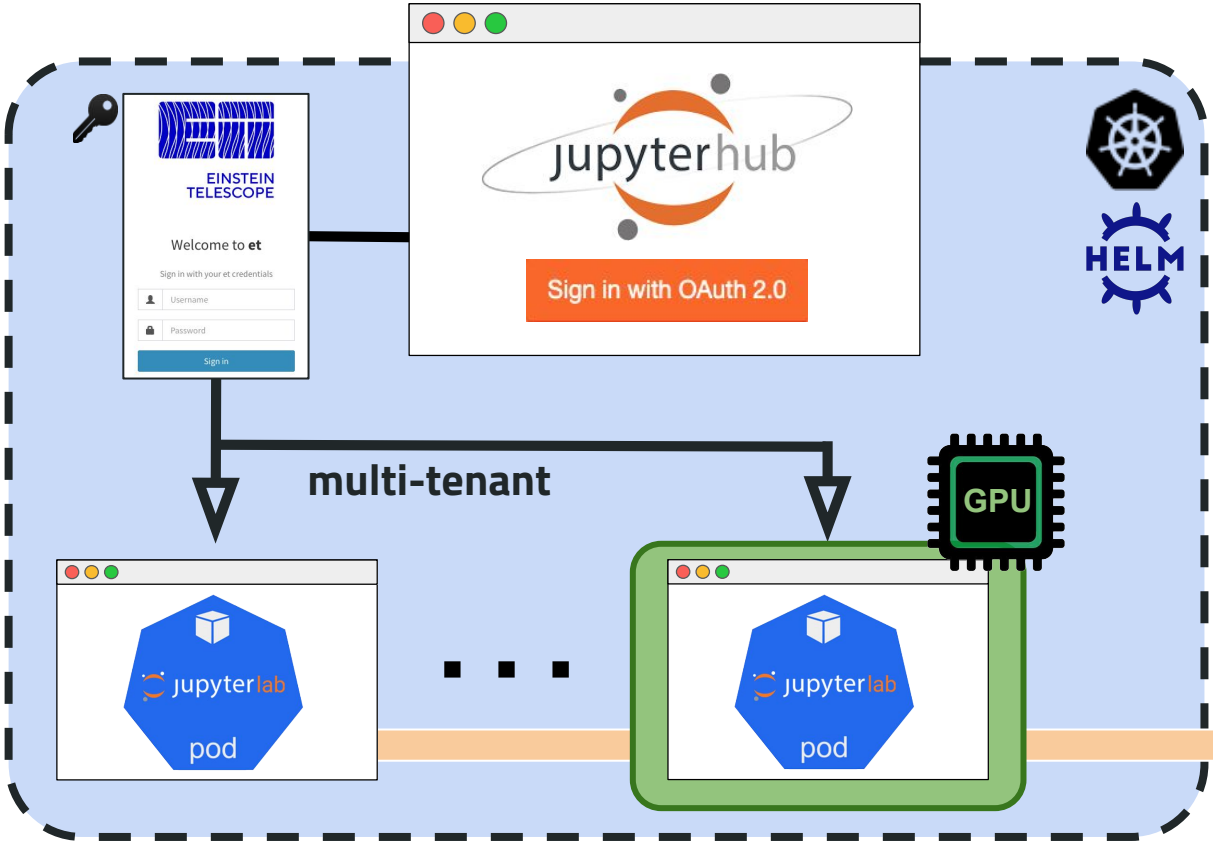
- Access to single hub entrypoint;
- Authn and authz token-based via INDIGO-IAM;
- Multi-user access to private JupyterLab sessions;

Architecture



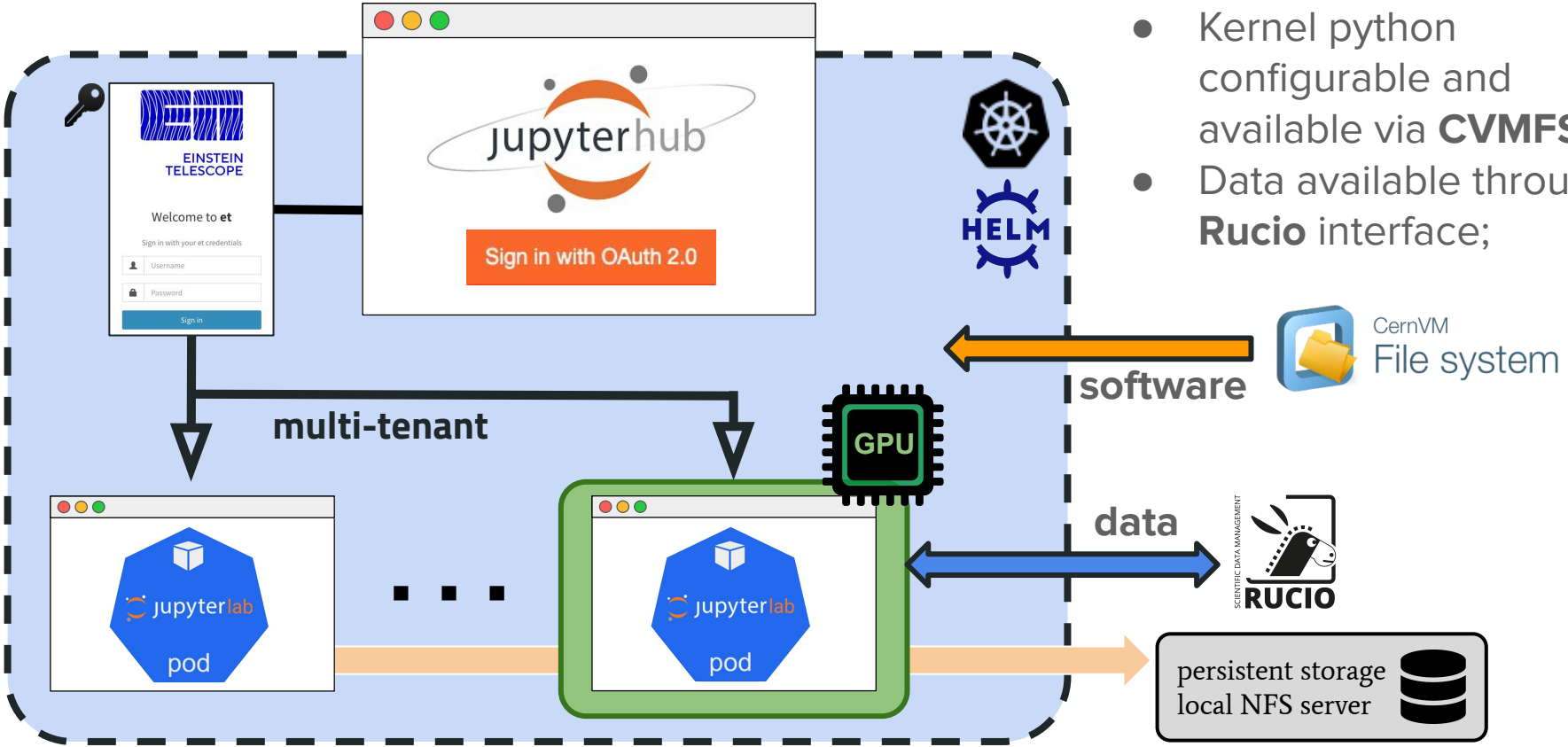
- Access to single hub entrypoint;
- Authn and authz token-based via INDIGO-IAM;
- Multi-user access to private JupyterLab sessions;
- GPU session available on demand;

Architecture



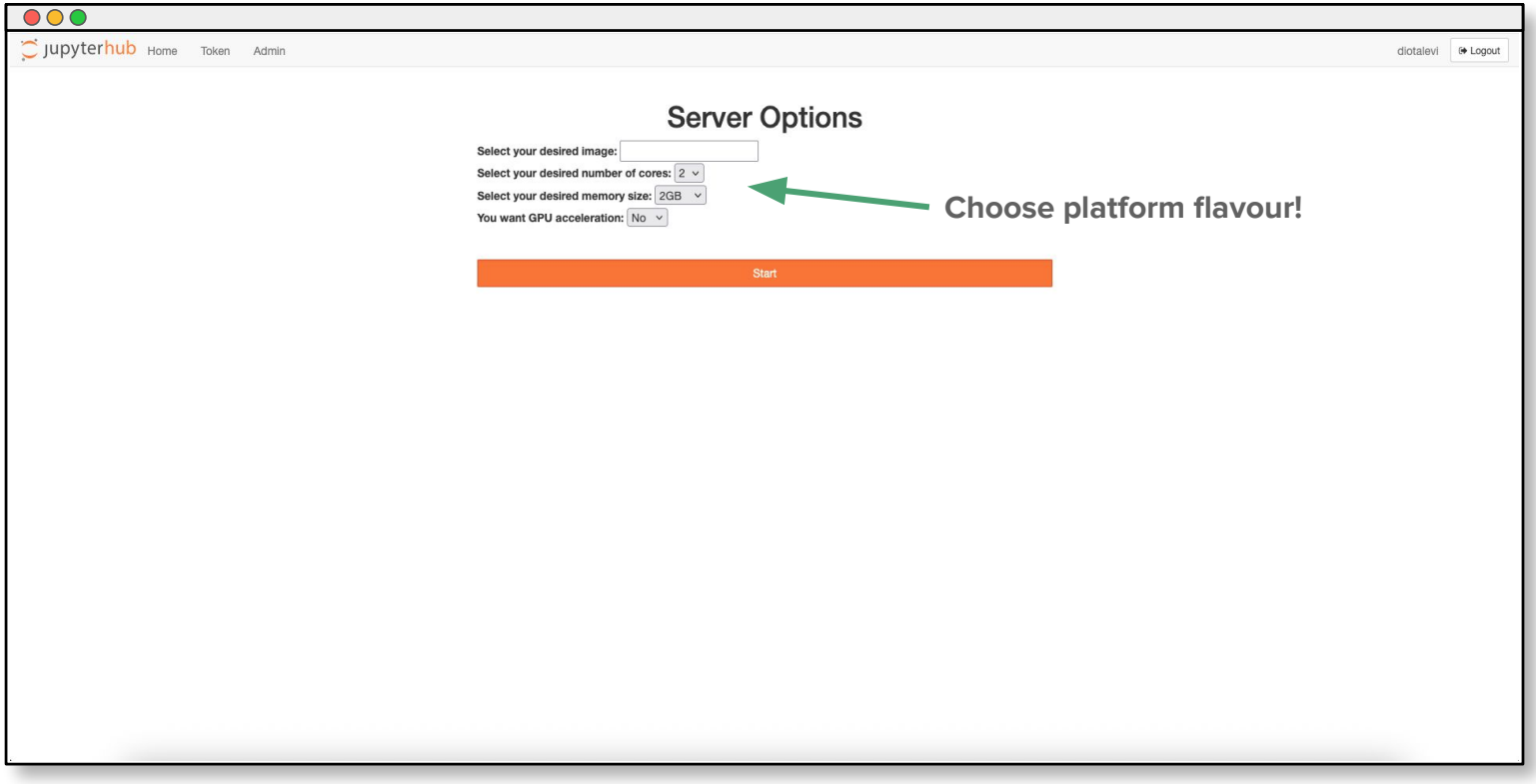
- Access to single hub entrypoint;
- Authn and authz token-based via INDIGO-IAM;
- Multi-user access to private JupyterLab sessions;
- GPU session available on demand;
- External storage for persistent user space;

Software and data access



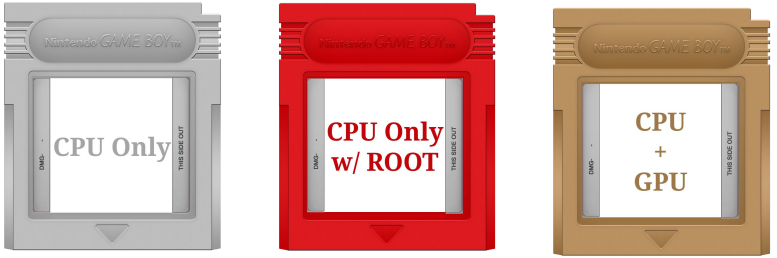
- Kernel python configurable and available via **CVMFS**;
- Data available through **Rucio** interface;

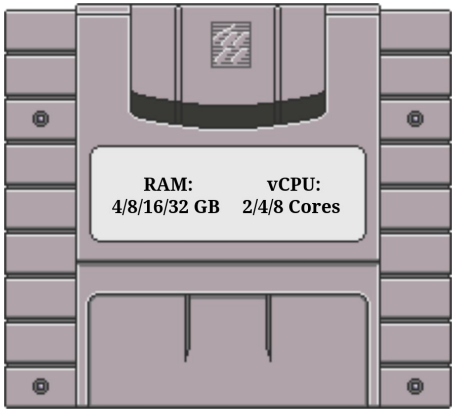
Welcome page (after authentication)



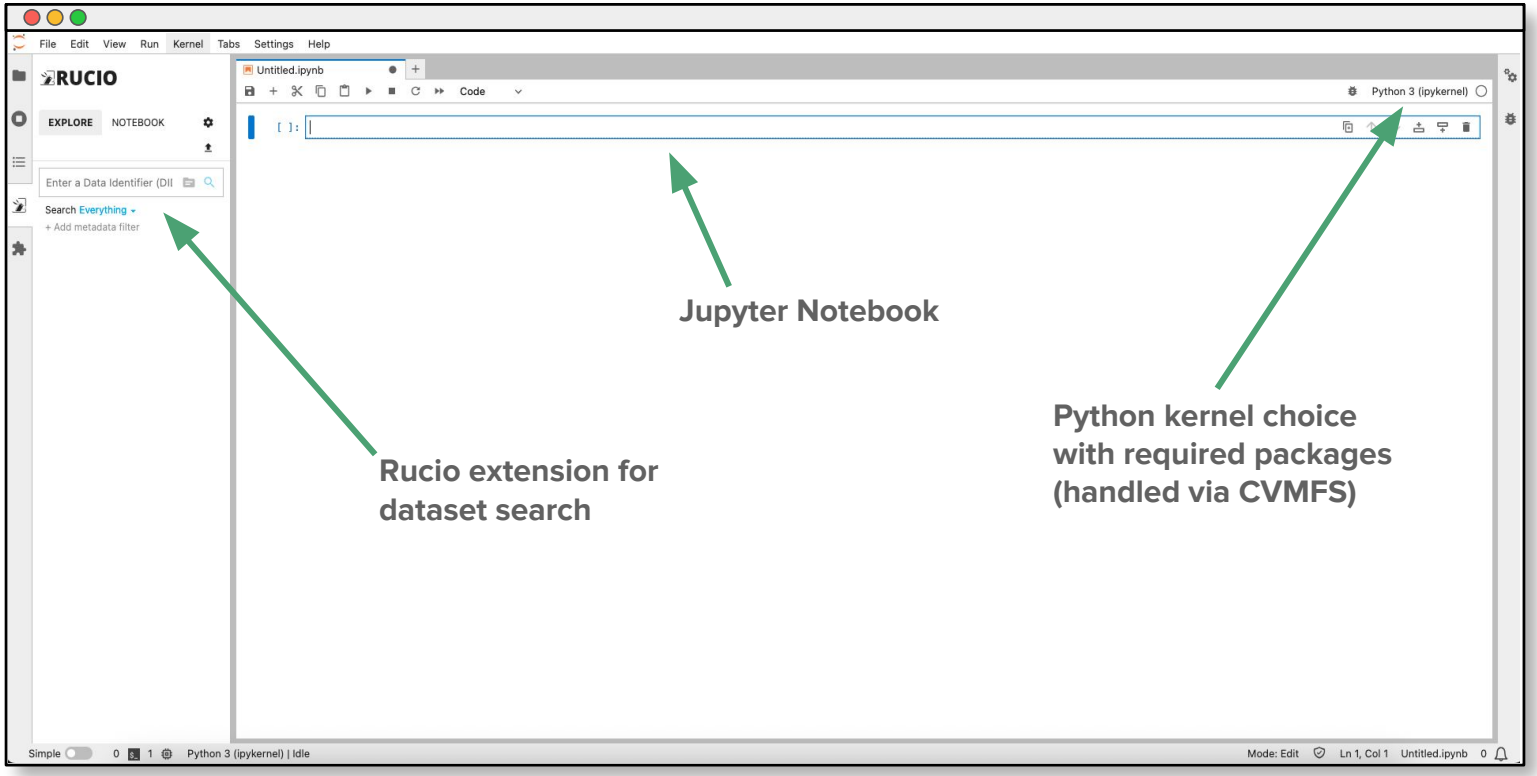
Platform flavours

- Three default images:
  - CPU only
  - CPU+GPU with CUDA toolkit installed;
  - Base images: only python 3.11. Fully customizable.
- Number of vCPU cores:
  - 2,4,8
- Memory size:
  - 2,4,8,16,32 GB
- GPU acceleration: yes or no (in that case is necessary the CPU+GPU image).



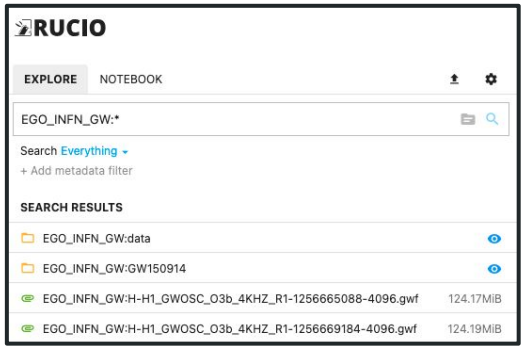


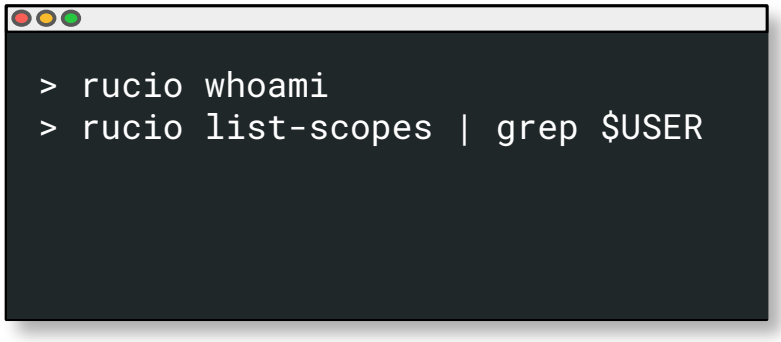
JupyterLab user session



Rucio data access

- Rucio client available on-platform, with the JupyterLab extension;
- Checks available copies of a chosen dataset:
  - provides access to its content directly, through a reference on a Jupyter notebook cell
- Authentication to Rucio Storage Elements (RSE) is possible via x509 proxies;
  - Currently, the ESCAPE RSE is used, waiting for the ET one.
- Available also via the CLI, for additional access and management.





Work in progress and outlook

- Define the storage areas for data, and enforce group access policies;
- Fix some auth* issues with Rucio labextension (CLI counterpart works fine);
- Add FPGA access inside the k8s cluster, for multi-user access:
  - Understand how partitioning the board works, some tools already available from AMD!
- Enable workflow management (i.e. REANA, HTCondor jobs, ...)

Outlook:

- Steadily progress towards a production-level prototype.

Conclusions

- The Bologna ET group is actively committed to the development of a **Virtual Research Environment**, using cutting-edge hardware and open-source software/tools/technologies:
    - Heterogeneous computing resources completely transparent for the user.
      Currently working on **CPU/GPU** (and soon enough FPGAs);
    - Full customization for software & libraries for analysis, thanks to containerised configurations loaded through CVMFS;
    - Enforced **AAA** (Any Data, Anytime, Anywhere) **data access** with Rucio and network protocols (like XRootD and WebDAV), for stage-in/out operations;
    - The platform will be available for authenticated (and authorized) ET users, via **IAM** groups.
- R&D will also be done on the available hardware: testing GPUs and FPGAs capabilities, with the future objective of expanding the cluster!