The ESCAPE Virtual Research Environment

Giovanni Guerrieri (CERN)

9 Dec 2025 - AstroORDAS workshop 2025

ESCAPE - The European Science Cluster of Astronomy & Particle Physics ESFRI Research Infrastructures has received funding from the European Union's Horizon 2020 research and innovation programme under the Grant Agreement n° 824064.

ESCAPE: European Science Cluster of Astronomy and Particle Physics

Consortium of 31 members, including:

- 10 ESFRI projects & landmarks: **CTA, EST, FAIR, HL-LHC, KM3NeT, SKA, LSST, VIRGO, ESO, JIVE**
- 2 pan-European International Organizations: **CERN and ESO**
- 2 European Research Infrastructures: **EGO and JIV-ERIC**
- 4 supporting European consortia: **APPEC, ASTRONET, ECFA and NuPECC**

Budget: **15.98 M€**

Duration: **48 months** (1/2/2019 -31/1/2023)

ESCAPE has received funding from the European Union's Horizon 2020 research and innovation programme under **Grant Agreement no. 824064**.

Since 2023, ESCAPE goes on as an Open Collaboration!

Digression on HEP sister sciences

SKAO

Credit: Rosie Bolton

SKAO is an intergovernmental organisation building the world's largest radio observatory

Membership

Members:
Australia, Canada, China, Germany, India, Italy, Netherlands, Portugal, South Africa, Spain, Sweden, Switzerland, UK

Accession stage:
France

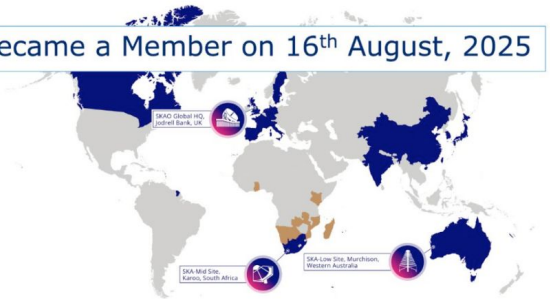
Negotiation stage:
South Korea

Early stages:
Japan

In conversation with several other potential member states

The Square Kilometre Array Observatory is an intergovernmental organisation, created in January 2021, to build and operate the world's two largest radio telescopes.

Sweden formally became a Member on 16th August, 2025



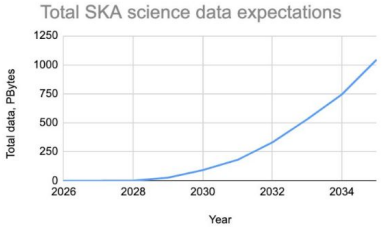
Digression on HEP sister sciences

SKAO

Credit: Rosie Bolton

SKA Science Data Expectations

Steady state, 300 PBytes per year of science data output, from 2035



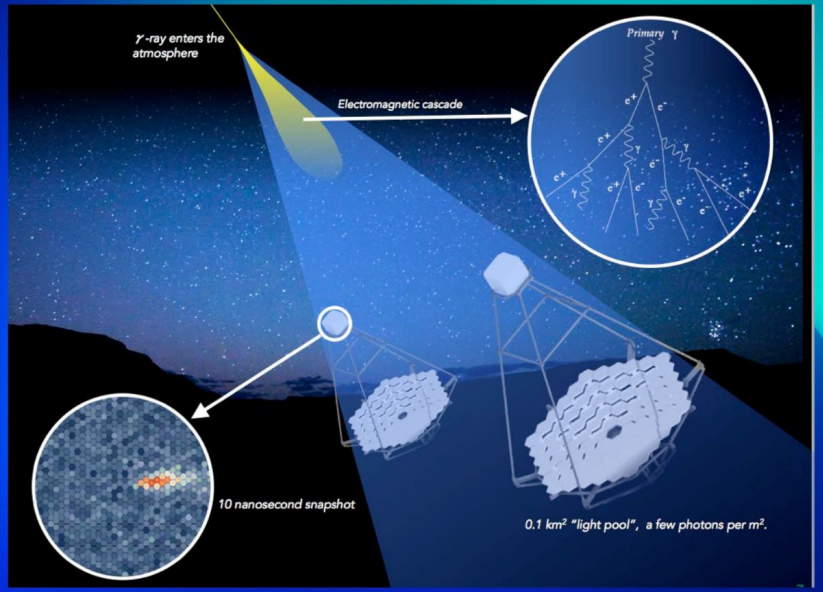
Science Data Volumes	Annual increase in new science data (PB)	Cumulative Total (nique) Science Data (PB)	notes - Low	notes - Mid
2026	-	-	no users	no users
2027	0.5	0.5	AA2 SV LOW	no users
2028	0.5	0.9	AA2 SV LOW	no users
2029	25.6	26.5	AA* SV LOW, increased data rate	AA2 SV MID
2030	65.4	91.9	LOW Cycle 0 starts	AA2 SV MID
2031	88.5	180.4	LOW Cycle 1	AA* SV MID, increased data rate
2032	150.0	330.4	LOW Cycle 2	MID Cycle 0 starts
2033	198.8	529.1	**LOW Cycle 3 Key Science Projects (KSP) start**	MID Cycle 1
2034	215.0	744.1		MID Cycle 2
2035	**300.0**	1044.1		**MID Cycle 3 KSPs start**

Digression on HEP sister sciences

CTAO

Credit: Max Linhoff

Imaging Atmospheric Cherenkov Telescopes



CTAO figures:

- Large collection areas (10^5 – 10^6 m^2)
- Excellent background rejection power
- Wide energy range 20 GeV – 300 TeV
- Good energy (15 to 7%) and angular resolution (0.15 – 0.02 deg)
- Reaction to external science alerts and fast repositioning (30 s to point towards any direction in the sky)
- Real-time analysis and internal science alerts

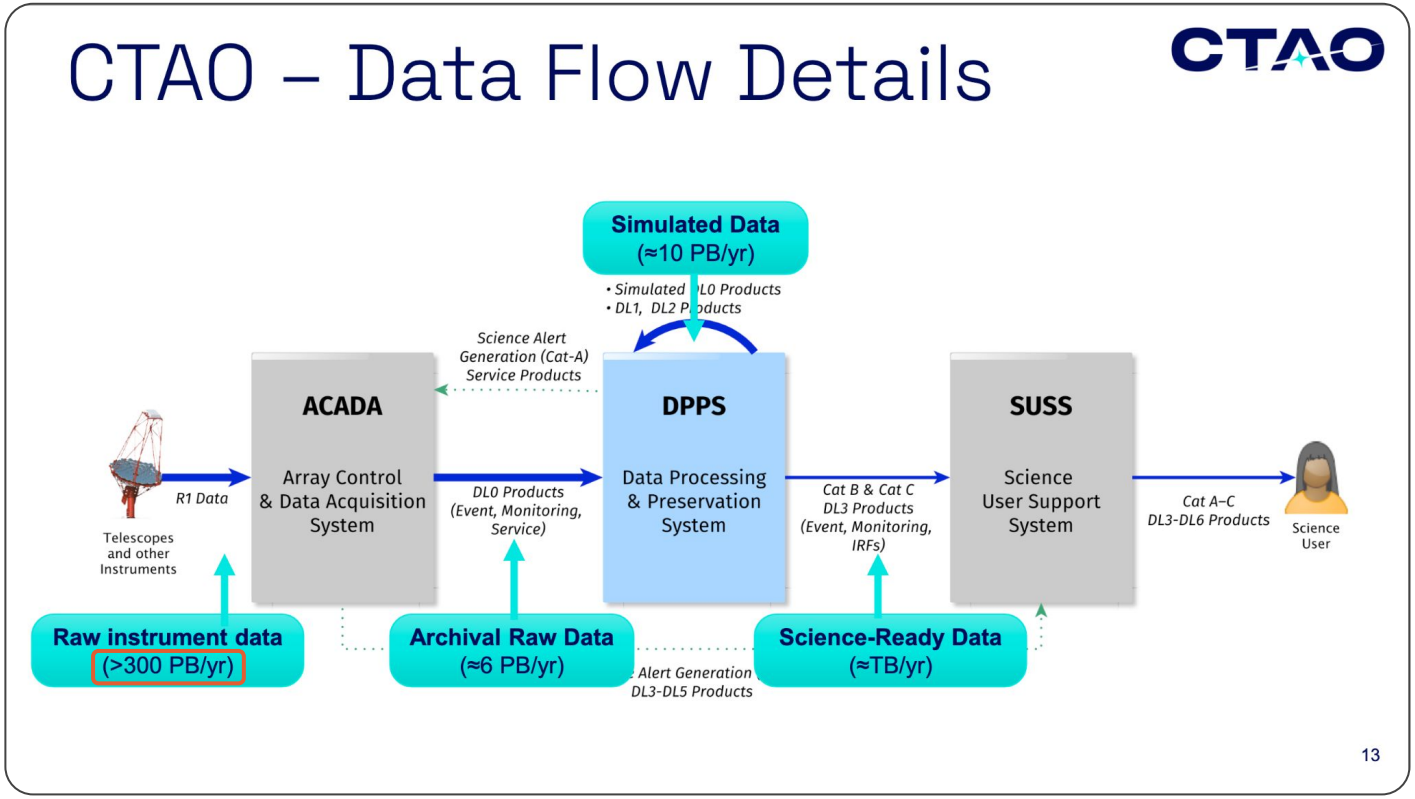
- The VHE sky is more populated than initially expected!

3

Digression on HEP sister sciences

CTAO

Credit: Max Linhoff

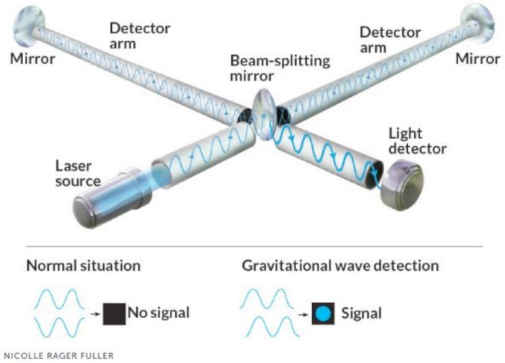


Digression on HEP sister sciences

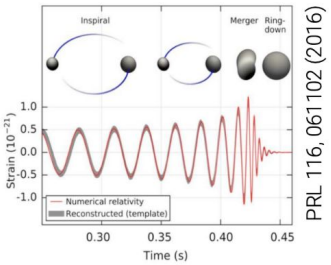
ET

Credit: Lia Lavezzi

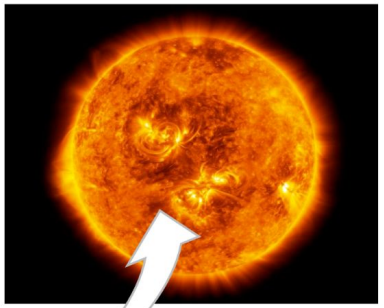
Gravitational Waves



To detect the passage of a GW, a **Michelson interferometer** is used.
The differential arm length, *a.k.a.* **strain**, is sampled in time



The signal is a **time series** of the **strain** variable



It is like measuring the change in the Earth-Sun distance by adding just **one atom** in–between!

Eight Rucio Community Workshop – Lia Lavezzi

5

Digression on HEP sister sciences

ET

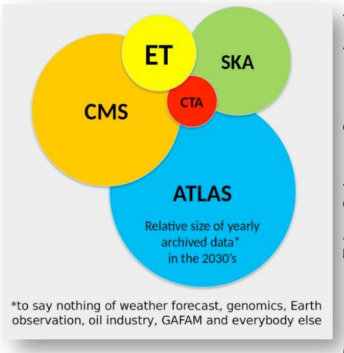
Credit: Lia Lavezzi

ET Data (Distribution) Requirements

As for LVK, there will be two kinds of data:

- **Strain** data (time series of defined duration)
  - **do not scale with sensitivity**
  - ET expected amount ~10 TB/year
- **Raw** data (from all the sensors)
  - **do scale with interferometer complexity**
  - ET expected amount ~50-100 PB/year

[See ET Preparatory Phase Project - D8.1 Computing and Data Requirements]



[S. Bagnasco, Erice Science Communication and Journalism International School 2023]

What grows is the useful scientific info in the data , thus the needed computing power.

- The strain time series structure probably will be similar to LIGO/Virgo
- We aim to eliminate the *metadata-in-the-name* choice and use real metadata to set those information
- We will need a way to retrieve the files based on metadata range selection
- We plan not to use a proprietary file format, but the coding has not yet been defined

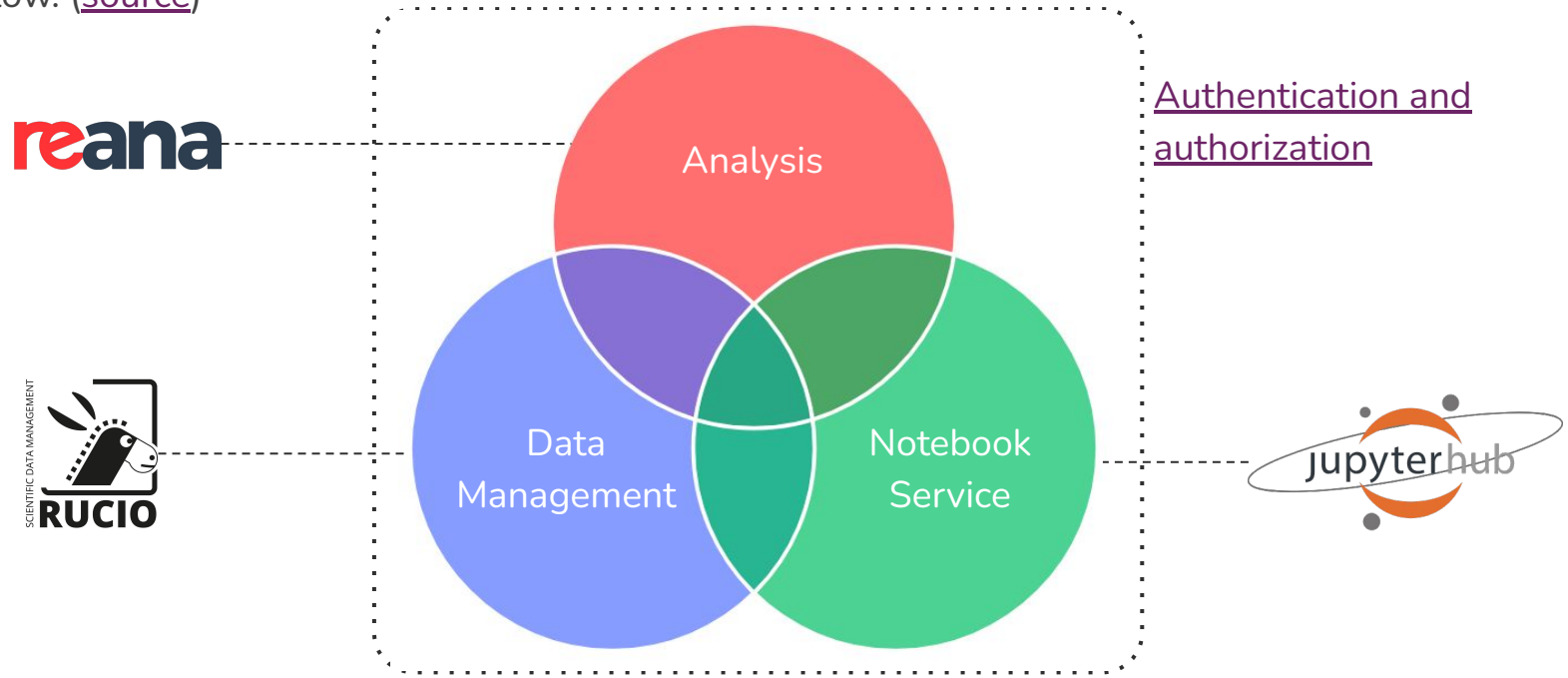
Eight Rucio Community Workshop – Lia Lavezzi 18

Key high-level ESCAPE project results

- Provided a full **Exabyte-scale Data Management** prototype (Storage, Transfer and Access) service with **common AAI** framework for distributed scientific computing
  - Validated through Data Challenges
- Developed a **catalogue** to publish digital scientific products of ESCAPE communities
  - ESCAPE research infrastructures onboarded
- Developed **interoperability standards** for astronomical data services
  - Virtual Observatory (VO) services prepared to be integrated with EOSC
- Produced a **reusable analysis toolkit** for integrating diverse service offerings, and a **Virtual Research Environment** prototype that integrates ESCAPE services
  - Adopted and replicated by a number of research infrastructures
- Several **Citizen Science** projects built upon ESCAPE developments
  - Attracted significant interest/involvement
- Dark Matter and Extreme Universe Science Projects as demonstrations of Open Science in ESCAPE
  - Integrated with EOSC platform through EOSC-Future

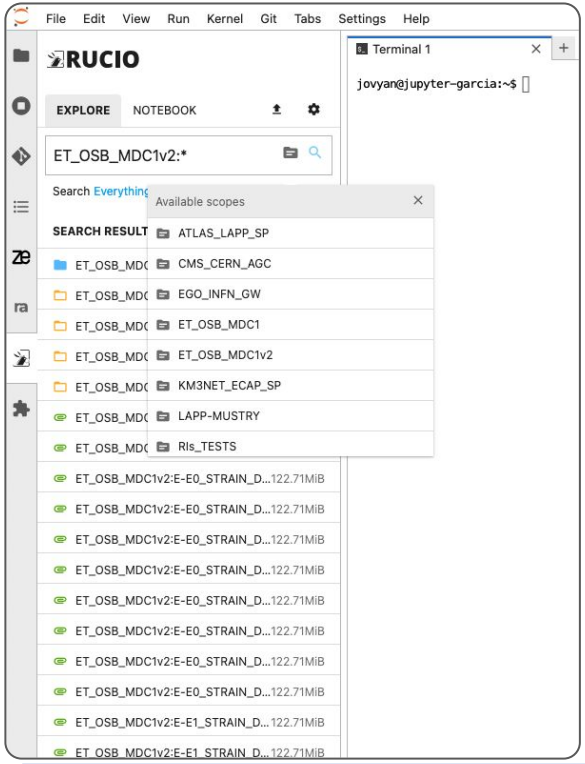
The ESCAPE Virtual Research Environment

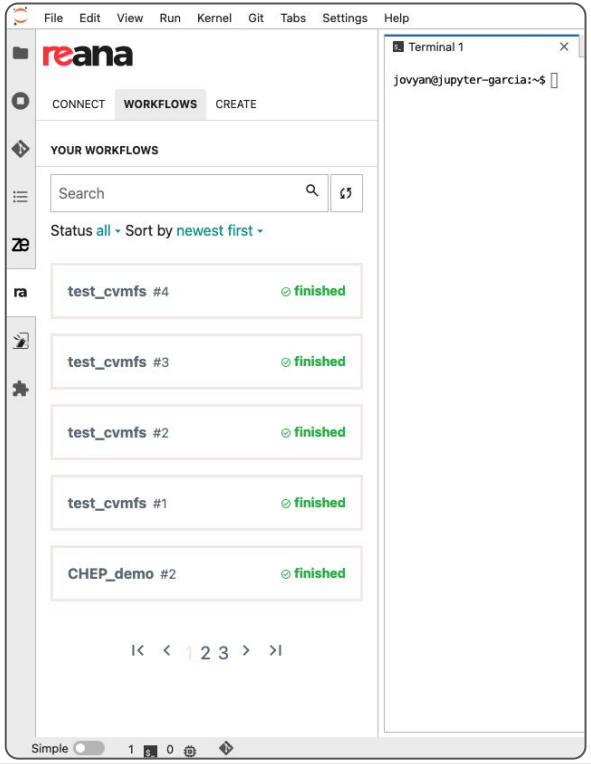
Virtual Research Environment (VRE): infrastructure and services that provide ***integrated*** data access, experiment's software and computational resources to execute one or more elements of an analysis workflow. (source)

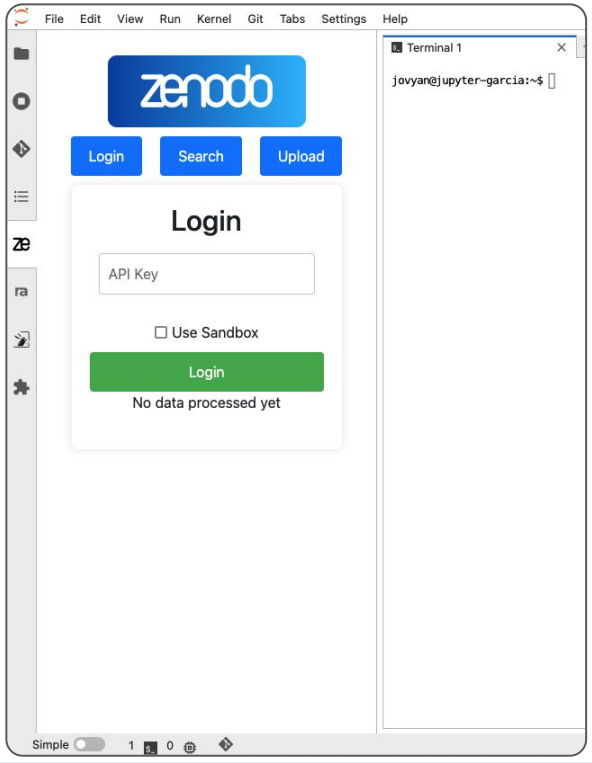


User's interactions with the VRE

What makes the VRE, the VRE







Example: from experiment's raw data to final results

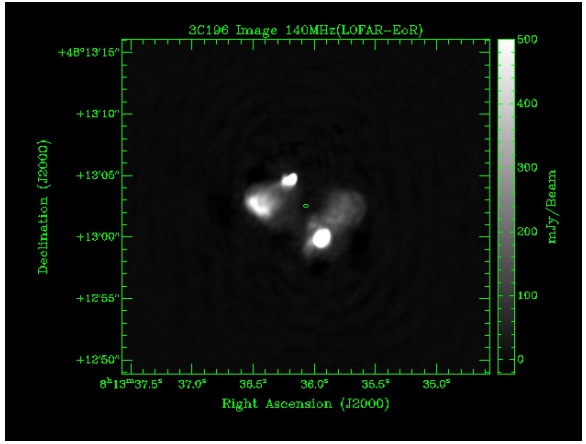
A multi-wavelength analysis in the VRE (3C196 quasar)

1. Data **injected** to the Data Lake from different radio sources from **external** locations

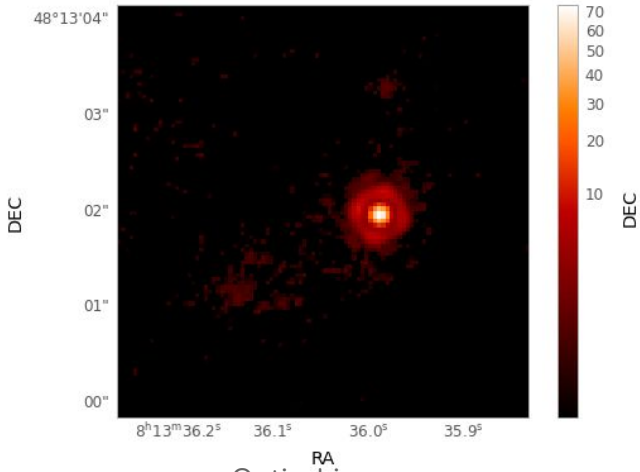
2. Download of data, **process** and **storage** of results back to the Data Lake

3. Combined **optical** data located via the VO plugin available in the **VRE**

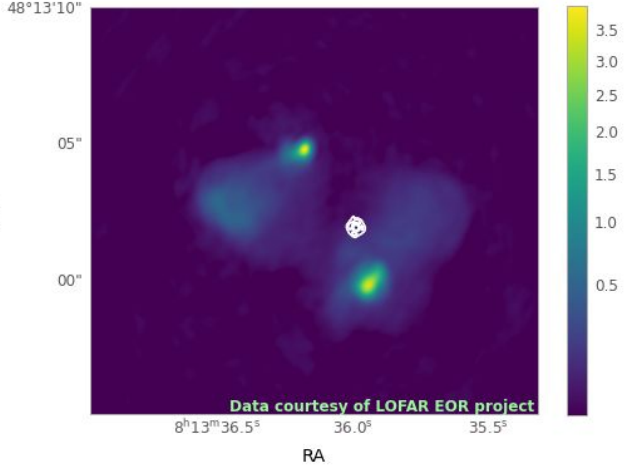
4. Optical and **radio** data combined.
Results uploaded back to the Data Lake.



Radio image



Optical image



Combined image

1. Data Management tools

2. Software distribution tools

3. Analysis facilities and reproducibility tools

The File Transfer Service

The underlying engine

- Orchestration of Third-Party Copies (TPCs)
- Streams transfers through itself if TPC is not supported
- Tape storage operations via the WLCG HTTP
- Tape REST API, SRM and XRootD
- Support for Cloud based storage
- Certificate and token authentication

FTS

The File Transfer Service

The underlying engine

- Simple **REST API** for submitting transfers.
- Convenient command-line tools and Python bindings
- Automated verification of destination file sizes and checksums
- Automated **retries** of failed file transfers
- **Multi-protocol** support: HTTP, SRM, XRootD, GridFTP, S3, ...
- Web-based real-time **monitoring**
- Web-based runtime configuration
- Runtime optimization to maximize throughput without overloading storages
- Activity based bandwidth reservation

FTS

What is Rucio

A Data Management system

- Rucio is a data management system meant to provide a **unified** way to manage data **distributed** across numerous, geographically separate, and **technologically different** storage systems.

- It was born within the ATLAS experiment, now is adopted by **several communities in HEP and beyond** (e.g. ESCAPE).

- Rucio is **declarative**: users can say what they want, and let Rucio figure out the details how to do it.

What is Rucio

A Data Management system

- Rucio is a data management system meant to provide a **unified** way to manage data **distributed** across numerous, geographically separate, and **technologically different** storage systems.

- It was born within the ATLAS experiment, now is adopted by **several communities in HEP and beyond** (e.g. ESCAPE).

- Rucio is **declarative**: users can say what they want, and let Rucio figure out the details how to do it.

Creation, transfer, and deletion of **replicas** of data

"I would like to delete all the datasets created for the MC23 winter campaign hosted at CNAF, but keep the ones at CERN. Then copy the ones at CERN to CNAF."

Management policies (usage, access, and data **lifetime**)

"I would like all the HIGGS samples to be read-only for users in this e-group, and I would like the replicas hosted at CNAF to have a lifetime of 180 days."

Workflow management systems integration

"Whenever DIRAC writes outputs in this EOS directory, I want to register these files in Rucio, keep 1 replica at CERN and have 2 replicas in Italy."

Automation of large-scale and **repetitive** operational tasks

"I'd like to resubmit all failing FTS jobs at INFN_TRIESTE for a maximum of 3 times. I would also like to delete all files that haven't been accessed in over 5 months."

What is Rucio

A Data Management system

- Rucio is a data management system meant to provide a **unified** way to manage data **distributed** across numerous, geographically separate, and **technologically different** storage systems.

- It was born within the ATLAS experiment, now is adopted by **several communities in HEP and beyond** (e.g. ESCAPE).

- Rucio is **declarative**: users can say what they want, and let Rucio figure out the details how to do it.

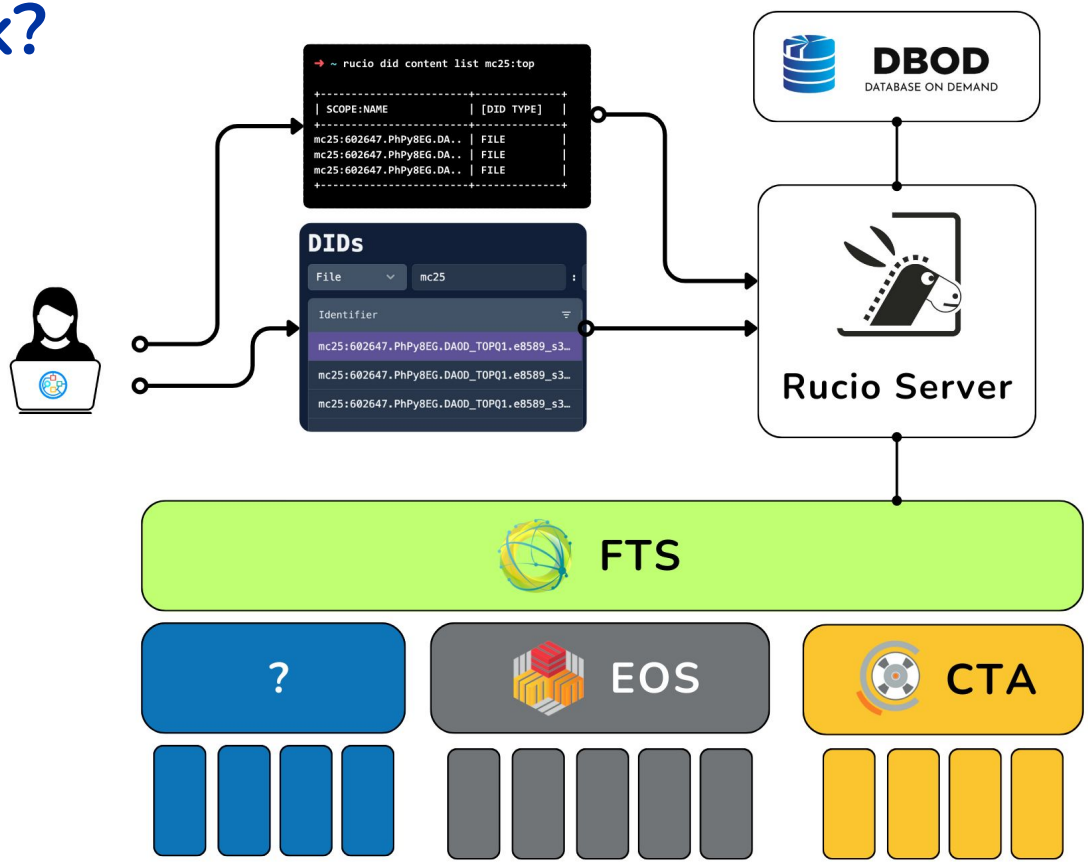
- ATLAS
- Belle II
- CTAO
- CMS
- DaFab
- DUNE
- EISCAT
- Einstein Telescope (ET)
- ePIC
- ESCAPE
- Fermilab
- HEP-CCE
- IHEP
- INFN
- InterTwin
- KM3NeT
- LIGO
- MAGIC
- RI-SCALE
- Rubin Observatory
- SKA
- XENON
- ...

How does Rucio work?

The architecture

- Rucio **shields** operators and users from the layered complexity of the underlying services (transfer jobs management, protocol migrations, credentials deprecation, lifetime policies, storage heterogeneity, etc.)

- All the technologies involved are **part of the same department**, so we have a good handle on reaction times / troubleshooting



What can Rucio do?
The metadata catalog

- Rucio accepts by default metadata from the WMS, a minimal set of condition data.
- It supports custom metadata fields.
- It can be integrated with **existing metadata platforms** to enrich the catalog

```
account:          tzero
adler32:          None
availability:     AVAILABLE
bytes:            2663168416
campaign:         data25_calib:tzero
closed_at:        2025-01-14 12:39:31
created_at:       2025-01-14 12:34:21
datatype:         RAW
deleted_at:       None
did_type:         DATASET
eol_at:           None
events:           4000
hidden:           False
is_archive:       None
is_new:           None
is_open:          False
length:           4
lumiblocknr:      None
md5:              None
monotonic:        False
name:             data25_calib.00490721.calibration.daq.RAW
obsolete:         False
panda_id:         None
phys_group:       None
prod_step:        daq
project:          data25_calib
provenance:       None
purge_replicas:   True
run_number:       490721
scope:            data25_calib
stream_name:      calibration_RAW
suppressed:       False
task_id:          150654
transient:        None
updated_at:       2025-01-14 12:39:31
version:          None
```

What can Rucio do?

Analysis help (from a user perspective)

- Users **don't need to worry about where the data is stored**. They only need to know what they are looking for.

```
> rucio did list mc25:*
+-----------------+--------------+
| SCOPE:NAME      | [DID TYPE]   |
|-----------------+--------------|
| mc25:top        | CONTAINER    |
| mc25:top_602647 | DATASET      |
+-----------------+--------------+

> rucio did content list mc25:top_602647
+-------------------------------------------------------------------------------+--------------+
| SCOPE:NAME                                                                    | [DID TYPE]   |
|-------------------------------------------------------------------------------+--------------|
| mc25:602647.PhPy8EG.DAOD_TOPQ1.e8589_s3889_r9364_p4514_nom.tt_truth_0.root | FILE         |
| mc25:602647.PhPy8EG.DAOD_TOPQ1.e8589_s3889_r9364_p4514_nom.tt_truth_1.root | FILE         |
| mc25:602647.PhPy8EG.DAOD_TOPQ1.e8589_s3889_r9364_p4514_nom.tt_truth_2.root | FILE         |
+-------------------------------------------------------------------------------+--------------+
```

The Rucio abstraction **trinity**:

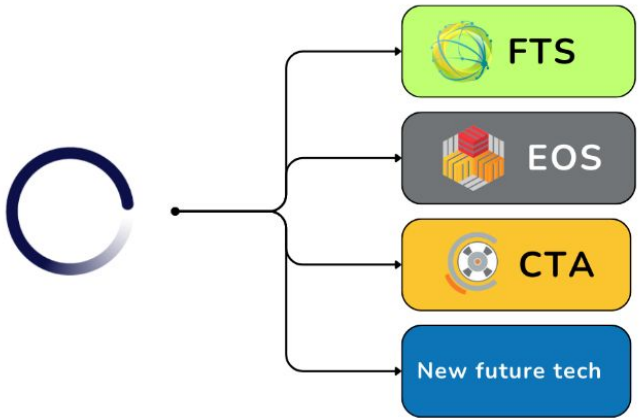
1. 📄 File: atomic unit
2. 🗂️ Dataset: set of 📄
3. 📦 Container: set of 📦, 🗂️

- Rucio integrates with **modern analysis facilities**, (e.g. Jupyter-based) to provide an interactive access to the data catalog.

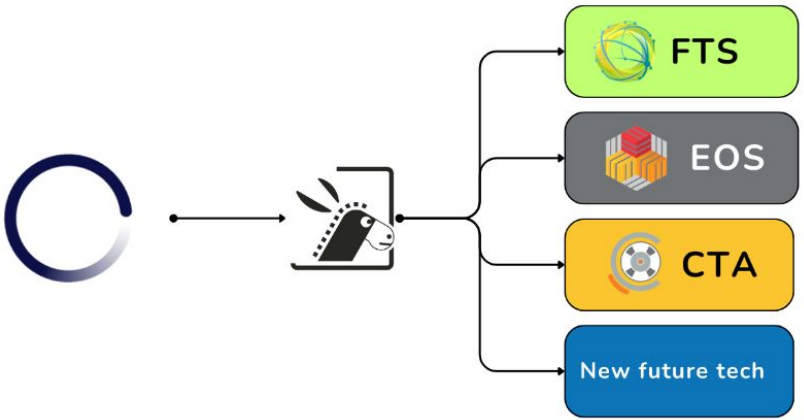
The reason

For which communities adopt Rucio

Before



After



~~1. Data Management tools~~

2. Software distribution tools

3. Analysis facilities and reproducibility tools

CVMFS
How to distribute software

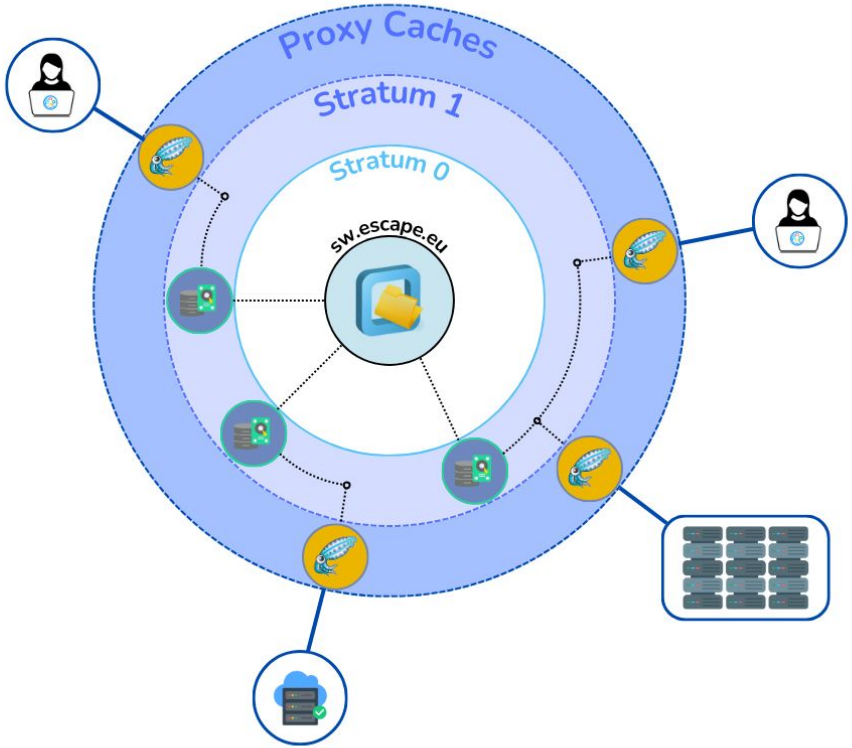
cernvm.cern.ch

The **CernVM-FS** is a **read-only file system** designed to distribute software, and more.

Software is updated and maintained on Stratum 0, synchronized to stratum 1

Data is cached on-demand

CVMFS mounted (via FUSE) as a local filesystem



CVMFS
How big is it?

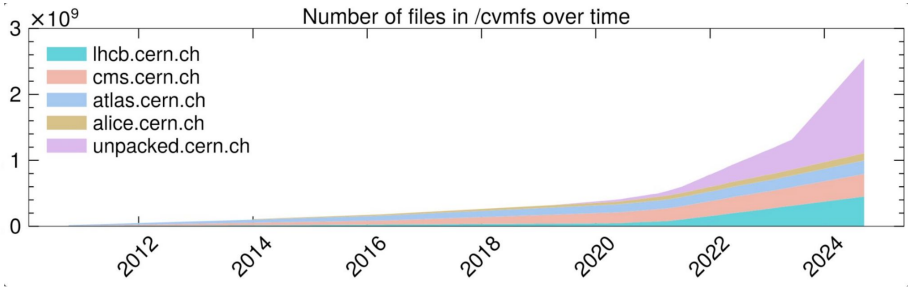
~ 15 Stratum 1s

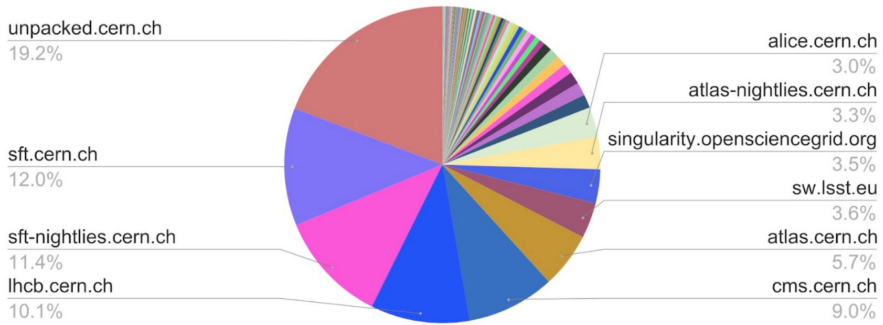
~ > 4 B files in the /cvmfs tree

~ 2 PB of data accessible through /cvmfs out of which ~1.5 PB in external files proven to scale up to 100 PB

~ > 4k container images

~ 260 repositories





CVMFS

Why should I bother?

pip install absl-py==1.4.0 accelerate==1.3.0 aiofile==3.9.0 aiohappyeyeballs==2.4.4 aiohttp==3.11.11 aiohttp-cors==0.7.0 aiohttp_retry==2.9.1 aiosignal==1.2.0 aiostream==0.4.5 alabaster==1.0.0 alembic==1.13.3 altair==5.2.0 annotated-types==0.6.0 antlr4-python3-runtime==4.13.1 anyio==4.6.1 appdirs==1.4.4 archspec==0.1.4 argon2-cffi==23.1.0 argon2-cffi-bindings==21.2.0 arrow==1.3.0 asciitree==0.3.3 asdf==3.5.0 asdf_standard==1.1.1 asdf_transform_schemas==0.5.0 asn1crypto==0.24.0 asteval==0.9.31 astor==0.8.1 astroid==3.3.8 astropy==6.1.4 astropy-iers-data==0.2024.10.21.0.33.21 asttokens==2.4.1 astunparse==1.6.3 async_generator==1.10 async-lru==2.0.4 async-timeout==4.0.2 asyncache==0.3.1 atomicwrites==1.3.0 attrs==25.1.0 auth-get-sso-cookie==2.4.0 Authlib==0.15.5 autograd==1.7.0 autopep8==2.3.1 avro==1.12.0 awkward==2.8.1 awkward_cpp==45 babel==2.17.0 backcall==0.2.0 backoff==2.2.1 bcrypt==4.1.3 beautifulsoup4==4.10.0 benIget==0.4.2.post1 black==24.10.0 bleach==3.3.1 blosc==1.11.3 blosc2==2.7.1 bokeh==3.1.1 boost_histogram==1.3.2 boto3==1.35.48 botocore==1.35.48 build==0.8.0 cabinetry==0.5.2 cachetools==5.5.1 caio==0.9.21 calmjs.parse==1.3.1 capturer==3.0 Cartopy==0.24.1 ccorp-yaml-include-relative-path==0.0.4 certifi==2024.8.30 cffi==1.17.1 cfgv==3.3.1 chardet==5.2.0 charset-normalizer==3.4.0 click==8.1.8 click-default-group==1.2.4 click-didyoumean==0.3.0 clikit==0.6.2 cloudpickle==3.1.1 cma==3.4.0 coffea==2025.3.0 colorama==0.4.6 colorcet==3.1.0 colored==2.2.3 coloredlogs==15.0.1 colorlog==6.7.0 comm==0.2.2 commonmark==0.9.1 configobj==5.0.9 configparser==5.3.0 confluent-kafka==2.6.1 confuse==2.0.1 cons==0.4.6 contourpy==1.0.7 control==0.9.1 contur==3.1.1 correctionlib==2.6.4 CouchDB==1.2 coverage==7.4.1 cppy==1.2.1 cpymad==1.15.0 cramjam==2.8.3 crashtest==0.3.1 crc32c==2.7.1 crypto==1.4.1 cryptography==44.0.3 cx_Oracle==8.3.0 cycler==0.12.1 Cython==3.0.12 dask==2025.2.0 dask-awkward==2025.3.0 dask-histogram==2025.2.0 dask-jobqueue==0.9.0 dask-kubernetes==2025.1.1 dask_lxplus==0.3.3 dataclasses-json==0.5.8 debugpy==1.8.1 decayalgebra==0.18.5 decorator==5.2.1 defusedxml==0.7.1 Deprecated==1.2.13 deprecation==2.1.0 dill==0.3.8 distlib==0.3.9 distributed==2025.2.0 distro==1.5.0 dm-tree==0.1.8 docopt==0.6.2 docstring-parser==0.15 docutils==0.21.2 donfig==0.8.1.post1 dotmap==1.3.30 elasticsearch==6.3.0 entrypoints==0.4 et_xmlfile==1.1.0 etuples==0.3.9 Events==0.5 exceptiongroup==1.2.2 execnet==1.9.0 executing==2.1.0 extension-helpers==1.0.0 faiss==1.11.0 fastjsonschema==2.17.1 filelock==3.16.1 fire==0.6.0 flake8==7.1.1 flake8-bugbear==24.8.19 flake8-builtins==2.5.0 flatbuffers==25.2.10 fonttools==4.40.0 fpdf2==2.7.9 fqdn==1.5.1 frozendict==2.3.8 frozenlist==1.4.1 fs==2.4.16 fsspec==2025.2.0 fsspec_xrootd==0.5.1 func_adl==3.3.3 func_adl_xAOD==2.1 funcsigs==1.0.2 future==1.0.0 gast==0.6.0 GaudiTesting==1.0.0 gcovr==7.2 GDAL==3.10.2 Genshi==0.7.7 globus-sdk==3.9.0 google-api-core==2.10.1 google-api-python-client==1.12.11 google-auth==2.38.0 google-auth-oauthlib==1.0.0 google-pasta==0.2.0 googleapis-common-protos==1.56.4 googledrivedownloader==0.4 gosam==2.1.2+c307997 graphviz==0.20.1 greenlet==3.0.3 grpcio==1.65.2 grpcio-status==1.56.2 gssapi==1.8.3 h11==0.14.0 h5py==3.13.0 HeapDict==1.0.1 hepdata-converter==0.3.1 hepdata_lib==0.16.0 hepdata_validator==0.3.6 hepstats==0.9.2 hepunits==2.3.5 hf-xet==1.1.5 hist==2.8.0 histoprint==2.6.0 horovod==0.28.1 htcondor==24.0.7 html5lib==1.1 httpcore==1.0.7 httpstan==4.13.0 httpx==0.28.1 httpx-sse==0.4.0 httpx-ws==0.7.1 huggingface_hub==0.33.0 humanfriendly==10.0 humanize==0.0.0 hyperopt==0.2.7 hypothesis==6.135.1 identify==2.5.24 idna==3.10 imageio==2.37.0 imagesize==1.4.1 iminuit==2.30.1 imjoy-rpc==0.5.59 imjoy-utils==0.1.2 importlib_metadata==8.6.1 importlib_resources==6.5.2 iniconfig==0.0.0 ipydatawidgets==4.3.2 ipykernel==6.29.5 ipympl==0.9.3 ipyparallel==9.0.0 ipython==8.32.0 ipython_genutils==0.2.0 ipywidgets==8.1.5 iso8601==1.0.2 isodate==0.7.2 isoduration==20.11.0 isort==5.12.0 itk==5.4.0 itk-core==5.4.0 itk-filtering==5.4.0 itk-io==5.4.0 itk-meshtopolydata==0.11.0 itk-numerics==5.4.0 itk-registration==5.4.0 itk-segmentation==5.4.0 itkwasm==1.0b185 itkwasm-downsample==1.4.3 itkwasm-downsample-wasi==1.5.0 itkwidgets==1.0a53 jacobi==0.9.2 jax==0.6.1 jaxlib==0.6.1 jedi==0.19.1 Jinja2==3.1.5 jmespath==1.0.1 joblib==1.4.2 json5==0.9.6 jsonpatch==1.33 jsonpointer==2.0 jsonschema==4.23.0 jsonschema-path==0.3.3 jsonschema-specifications==2023.7.1 jupyter==1.0.0 jupyter_client==7.4.9 jupyter-console==6.6.3 jupyter_contrib_core==0.4.2 jupyter_contrib_nbextensions==0.7.0 jupyter_core==5.7.2 jupyter-events==0.10.0 jupyter_highlight_selected_word==0.2.0 jupyter_latex_envs==1.4.6 jupyter-lsp==2.2.5 jupyter_nbextensions_configurator==0.6.4 jupyter_packaging==0.12.3 jupyter_server==2.14.2 jupyter_server_terminals==0.5.3 jupyterlab==4.3.0 jupyterlab_pygments==0.3.0 jupyterlab_rise==0.42.0 jupyterlab_server==2.27.3 jupyterlab_widgets==3.0.13 keras==3.7.0 Keras_Applications==1.0.8 Keras_Preprocessing==1.1.2 keras-tuner==1.4.7 kfp==2.8.0 kfp-pipeline-spec==0.3.0 kfp-server-api==2.0.3 kiwisolver==1.4.5 kopf==1.37.2 kr8s==0.17.4 kt-legacy==1.0.5 kubernetes==23.3.0 kubernetes_asyncio==21.7.1 langchain==0.3.14 langchain-community==0.3.14 langchain-core==0.3.29 langchain-text-splitters==0.3.5 langsmith==0.2.10 lap==0.5.12 lark==1.1.5 lazy_loader==0.4 lazy-object-proxy==1.10.0 lhereader==1.0.11 libclang==18.1.1 lightning-utilities==0.12.0 linkify-it-py==2.0.2 llvmlite==0.44.0 lmfit==1.2.2 locket==1.0.0 logical-unification==0.4.6 logilab-common==1.8.1 looseversion==1.3.0 lxml==5.3.1 lz4==4.0.0 M2Crypto==0.41.0 make-it-sync==2.0.0 Mako==1.3.5 Markdown==3.7 markdown-it-py==2.2.0 MarkupSafe==2.1.5 marshmallow==3.16.0 marshmallow-enum==1.5.1 matplotlib==3.10.3 matplotlib-inline==0.1.7 mccabe==0.7.0 mdit-py-plugins==0.4.0 mdurl==0.1.2 memory_profiler==0.61.0 memray==1.17.2 meson==1.3.2 metakernel==0.30.2 mimesis==14.0.0 miniKanren==1.0.3 minio==7.2.15 miniopy-async==1.21.1 minrpc==0.0.11 mistune==3.0.1 ml_dtypes==0.5.1 mock==3.0.5 more-itertools==10.6.0 mpi4py==3.1.5 mplhep==0.3.31 mplhep_data==0.0.0 mpmath==1.2.1 msgpack==1.0.5 multidict==6.4.4 multipledispatch==1.0.0 multiprocess==0.70.16 mypy==1.16.1 mypy_extensions==1.0.0 mysql-connector-python==8.0.18 Naked==0.1.32 namex==0.0.8 nanobind==2.7.0 narwhals==1.9.4 nbclassic==1.2.0 nbclient==0.5.3 nbconvert==7.16.4 nbformat==5.9.1 ndindex==1.8 nest_asyncio==1.6.0 networkx==3.4.2 ngff-zarr==0.10.2 nltk==3.8.1 nodeenv==1.7.0 nose-py3==1.7.0 notebook==6.5.7 notebook_shim==0.2.4 numba==0.61.2 numba-stats==1.10.1 numcodecs==0.16.0 numdifftools==0.9.41 numexpr==2.10.1 numpy==2.1.3 oauthlib==3.0.1 octave_kernel==0.32.0 odfpy==1.4.1 onnx==1.16.0 onnxruntime==1.22.0 openai==0.27.8 openapi-core==0.19.4 openapi-schema-validator==0.6.3 openapi-spec-validator==0.7.1 opencv-python==4.10.0.84 openpyxl==3.1.5 openqasm3==1.0.1 opensearch-py==2.8.0 opt_einsum==3.3.0 optree==0.14.1 optuna==4.4.0 oracledb==3.1.1 ordered-set==4.1.0 orjson==3.9.15 outcome==1.3.0 overrides==7.7.0 OWSLib==0.17.1 packaging==24.2 pandas==2.2.3 pandocfilters==1.5.1 panel==1.1.0 param==1.13.0 paramiko==2.9.2 parse==1.20.2 parsl==1.2.0 parso==0.8.4 partd==1.4.2 particle==0.25.2 pastel==0.2.1 pathable==0.4.4 pathos==0.2.3 pathspec==0.11.1 patsy==0.5.6 pbr==6.0.0 pep517==0.13.1 pexpect==4.9.0 pickleshare==0.7.5 pillow==11.2.1 pip==23.3.1 pipdeptree==2.9.0 pipenv==2024.4.0 platformdirs==4.3.6 plotext==5.2.7 plotly==5.16.1 pluggy==1.5.0 plumbum==1.7.2 ply==3.11 polars==0.18.13 pooch==1.6.0 pox==0.2.5 ppft==1.6.4.9 pre_commit==3.8.0 prettytable==3.0.0 prometheus_client==0.21.1 prompt_toolkit==3.0.47 propcache==0.2.1 protobuf==4.25.4 psutil==5.9.8 ptyprocess==0.7.0 pure_eval==0.2.3 py==1.11.0 py-cpuinfo==9.0.0 py4j==0.10.9.7 pyarrow==20.0.0 pyasn1==0.6.1 pyasn1_modules==0.4.1 pybind11==2.11.1 pycairo==1.28.0 pycodestyle==2.12.1 pycparser==2.22 pycryptodome==3.21.0 pyct==0.5.0 pydantic==2.11.7 pydantic_core==2.33.2 pydantic-settings==2.6.1 pyDigitalWaveTools==1.1 pydot==1.4.1 pydot_ng==2.0.0 pyeda==0.29.0 pyerfa==2.0.1.4 pyflakes==3.2.0 Pygments==2.18.0 pygraphviz==1.11 PyHEADTAIL==1.16.4 pyHepMC3==3.3.1 pyHepMC3.rootIO==3.3.1 pyHepMC3.search==3.3.1 pyhf==0.7.6 PyJWT==2.0.1 pykube-ng==23.6.0 pylev==1.4.0 pylhe==0.9.0 pylint==3.3.7 PyNaCl==1.5.0 pyparsing==2.4.7 pypdf==3.11.0 pypng==0.20220715.0 pyproj==3.7.1 PyQt-builder==1.18.1 PyQt5==5.15.10 PyQt5_sip==12.15.0 pyroscope-io==0.8.5 pyrsistent==0.18.0 pyserial==3.4 pyshp==2.3.1 pyshtools==4.13.1 pysimdjson==7.0.1 pyslha==3.2.6 pyspark==3.5.6 pystan==3.10.0 pytensor==2.31.3 pytest==8.3.2 pytest-cov==4.1.0 pytest-runner==6.0.1 pytest-xdist==3.3.1 python-box==7.2.0 python-dateutil==2.9.0.post0 python-dotenv==1.0.1 python-gitlab==1.8.0 python-json-logger==2.0.7 python-jsonpath==0.7.1 python-prctl==1.8.1 python-vxi11==0.9 pythran==0.17.0 pytorch-lightning==2.5.1.post0 pytz==2022.7.1 pyviz_comms==2.0.1 PyWavelets==1.6.0 PyYAML==6.0.2 pyzmq==26.2.1 qastle==0.18.0 qibo==0.2.18 qibojit==0.1.7 qrcode==7.4.2 qtconsole==5.6.1 QtPy==2.4.2 qulacs==0.6.4.1 railroad-diagrams==1.1.1 rdflib==6.1.1 redis==4.3.3 referencing==0.35.1 regex==2023.5.5 requests==2.32.3 requests-gssapi==1.2.3 requests-oauthlib==1.2.0 requests-toolbelt==1.0.0 requests-unixsocket==0.2.0 resample==1.10.1 retry==0.9.2 retrying==1.3.3 rfc3339_validator==0.1.4 rfc3986_validator==0.1.1 rich==13.9.4 rich-argparse==1.6.0 rise==5.7.2.dev2 rpds-py==0.18.0 rq==1.10.1 rsa==4.0 ruamel.yaml==0.18.10 ruamel.yaml.clib==0.2.12 ruff==0.12.1 s3transfer==0.10.3 safetensors==0.4.5 scandir==1.10.0 scikit-build==0.18.1 scikit-hep==2025.4.1 scikit-image==0.25.2 scikit-learn==1.5.2 scikit-rf==1.1.0 scipy==1.15.3 seaborn==0.13.2 selenium==4.28.0 semantic_version==2.10.0 Send2Trash==1.8.3 sentence-transformers==3.4.1 sentencepiece==0.2.0 servicex==3.1.0 setuptools==75.8.0.post20250710 setuptools-scm==8.2.0 shapely==2.0.5 shellescape==3.8.1 shellingham==1.5.4 shortuuid==1.0.13 simplegeneric==0.8.1 simplejson==3.16.1 singledispatch==3.4.0.3 sip==6.8.3 six==1.16.0 sniffio==1.3.1 snowballstemmer==2.2.0 sortedcontainers==2.4.0 soupsieve==2.4.1 spey==0.2.3 Sphinx==8.1.3 sphinx_rtd_theme==3.0.2 sphinxcontrib-applehelp==2.0.0 sphinxcontrib-devhelp==2.0.0 sphinxcontrib-htmlhelp==2.1.0 sphinxcontrib-jquery==4.1 sphinxcontrib-jsmath==1.0.1 sphinxcontrib-qthelp==2.0.0 sphinxcontrib-serializinghtml==2.0.0 SQLAlchemy==2.0.16 stack_data==0.6.3 statsmodels==0.14.5 stevedore==3.3.0 stomp.py==6.1.0 storm==0.23 strip-hints==0.1.10 svg.path==6.3 svgpathtools==1.6.1 svgwrite==1.4.3 swandaskcluster==3.1.0 sympy==1.14.0 synr==0.5.0 tables==3.10.2 tabulate==0.9.0 tblib==1.7.0 tcut_to_qastle==0.7 tenacity==9.0.0 tensorboard==2.19.0 tensorboard-data-server==0.7.2 tensorboard-plugin-wit==1.8.1 tensorboardX==2.6.2.2 tensorflow_cpu==2.19.0 tensorflow-io-gcs-filesystem==0.37.1 tensorflow-probability==0.25.0 termcolor==2.4.0 terminado==0.18.1 testpath==0.6.0 tex2pix==0.3.1 texttable==1.7.0 textual==0.74.0 tf_keras==2.19.0 tf2onnx==1.16.1 TFPWA==0.2.3.dev66+g5a8b7e2 tfs-pandas==4.0.0 threadpoolctl==3.5.0 tifffile==2023.4.12 tinycss2==1.2.1 tinydb==4.8.2 tokenizers==0.21.0 toml==0.10.2 tomli==2.0.1 tomlkit==0.11.8 toolz==1.0.0 torch==2.7.1 torch_cluster==1.6.3 torch-geometric==2.6.1 torch_scatter==2.1.2 torch_sparse==0.6.18 torchmetrics==1.7.2 torchvision==0.22.1 tornado==6.4.1 tqdm==4.67.1 traitlets==5.14.3 traittypes==0.2.1 transformers==4.52.4 trio==0.28.0 trio-websocket==0.11.1 typeguard==2.13.3 typer==0.15.1 types-python-dateutil==2.9.0.20241206 types-PyYAML==6.0.12.20241230 typing_extensions==4.12.2 typing_inspect==0.9.0 typing-inspection==0.4.0 tzdata==2024.1 tzlocal==2.0.0 uc-micro-py==1.0.2 uhi==0.3.0 ultralytics==8.3.58 ultralytics-thop==2.0.14 uncertainties==3.1.7 uproot==5.6.0 uproot_browser==0.6.7 uri-template==1.3.0 uritemplate==3.0.1 urllib3==2.2.3 vcversioner==2.16.0.0 vector==1.6.1 verboselogs==1.7 versioneer==0.29 virtualenv==20.28.0 virtualenv-clone==0.5.4 virtualenvwrapper==4.8.4 wasmtime==28.0.0 watchdog==2.1.7 wcwidth==0.2.5 webargs==8.4.0 webcolors==24.11.1 webencodings==0.5.1 webio_jupyter_extension==0.1.0 websocket-client==1.8.0 websockets==14.1 Werkzeug==3.1.3 wheel==0.45.0 widgetsnbextension==4.0.13 wrapt==1.16.0 wsproto==1.2.0 xarray==2025.4.0 xenv==1.0.1 xgboost==2.1.3 xlrd==2.0.1 xrootd==5.8.3 xxhash==3.5.0 xyzservices==2023.5.0 yacs==0.1.8 yarl==1.18.3 zarr==3.0.0 zfit==0.26.0 zfit_interface==0.0.3 zfit_physics==0.8.1 zict==3.0.0 zipp==3.20.2 zstandard==0.22.0

CVMFS
Why should I bother?

```
source /cvmfs/sft.cern.ch/lcg/views/LCG_108/x86_64-el9-gcc13-opt/setup.sh
```

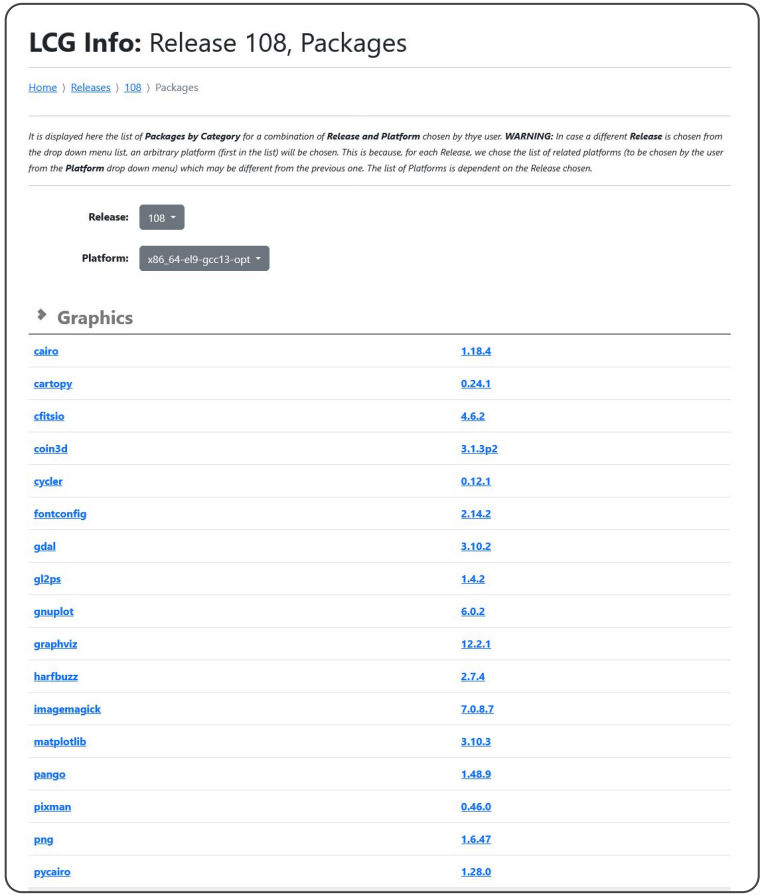
CVMFS
The catalog

https://lcginfo.cern.ch/

Provides an extensive list of packages, architectures and versions available in the repository.

Doesn't provide an entrypoint to get that stack of software in a terminal. There's no pointer to **`source /cvmfs/sft.cern.ch/lcg/views...`**

Navigating CVMFS at the moment is close to **navigating the dark web**.



CVMFS
The struggle

6 hits of tab to get a stack of software, provided that I know where to go.

Could be better.

We're working on fixing that.

```
[gguerrie@jupyter-gguerrie ~]$ source /cvmfs/sft.cern.ch/
.cvmfsdirtab              .cvmfsdirtab-17Jun2019   lcg/
.cvmfsdirtab-09May2019    .cvmfsdirtab~            test-fab-alma9-deleteme
.cvmfsdirtab-16Aug2018    README.md
[gguerrie@jupyter-gguerrie ~]$ source /cvmfs/sft.cern.ch/lcg/
README.md          dev/               lastUpdate         releases/
app/               etc/               latest/            sources/
contrib/           external/          mapfile.txt        testing/
contrib_backup/    hepsoft/           nightlies/         views/
contrib_test/      k8s_build/         pkgs_view_ml_1.0
[gguerrie@jupyter-gguerrie ~]$ source /cvmfs/sft.cern.ch/lcg/views/
Display all 454 possibilities? (y or n)
[gguerrie@jupyter-gguerrie ~]$ source /cvmfs/sft.cern.ch/lcg/views/LCG_108
LCG_108/                        LCG_108_ATLAS_8/
LCG_108_ATLAS_1/                LCG_108_cuda/
LCG_108_ATLAS_2/                LCG_108_geant4ext20250725/
LCG_108_ATLAS_3/                LCG_108_geant4ext20250725-qt6/
LCG_108_ATLAS_4/                LCG_108_swan/
LCG_108_ATLAS_5/                LCG_108a/
LCG_108_ATLAS_6/                LCG_108a_ATLAS_1/
LCG_108_ATLAS_7/                LCG_108a_ATLAS_2/
[gguerrie@jupyter-gguerrie ~]$ source /cvmfs/sft.cern.ch/lcg/views/LCG_108/
.cvmfscatalog                arm64-mac15-clang170-opt/    x86_64-el9-gcc14fp-opt/
aarch64-el9-gcc14-opt/       x86_64-el9-clang19-opt/      x86_64-el9-gcc15-dbg/
aarch64-el9-gcc15-opt/       x86_64-el9-gcc13-opt/        x86_64-el9-gcc15-opt/
arm64-mac13-clang150-opt/    x86_64-el9-gcc14-dbg/        x86_64-ubuntu2204-gcc11-opt/
arm64-mac14-clang160-opt/    x86_64-el9-gcc14-opt/        x86_64-ubuntu2404-gcc13-opt/
[gguerrie@jupyter-gguerrie ~]$ source /cvmfs/sft.cern.ch/lcg/views/LCG_108/x86_64-el9-gcc13-opt/
.cvmfscatalog  fonts/         lib64/         plugins/       share/
bin/           icons/         libexec/       python/        src/
cmake/         include/       macros/        scripts/       test/
[gguerrie@jupyter-gguerrie ~]$ source /cvmfs/sft.cern.ch/lcg/views/LCG_108/x86_64-el9-gcc13-opt/setup.sh
```

CVMFS

The solution (is it really?)

Within the ATLAS experiment (and likely within other experiments as well), software teams have developed a custom way of deploying resources.

This addresses the problem, but doesn't solve it.

```
[gguerrie@lxplus923 ~]$ setupATLAS
lsetup               lsetup <tool1> [ <tool2> ...] (see lsetup -h):
 lsetup asetup        (or asetup) to setup an Athena release
 lsetup astyle        ATLAS style macros
 lsetup atlantis      Atlantis: event display
 lsetup centralpage   Find samples
 lsetup eiclient      Event Index
 lsetup emi           EMI: grid middleware user interface
 lsetup lcgenv        lcgenv: setup tools from cvmfs SFT repository
 lsetup panda         Panda: Production ANd Distributed Analysis
 lsetup pyami         pyAMI: ATLAS Metadata Interface python client
 lsetup root          ROOT data processing framework
 lsetup rucio         distributed data management system client
 lsetup scikit        python data analysis ecosystem
 lsetup views         Set up a full LCG release
 lsetup xcache        XRootD local proxy cache
 lsetup xrootd        XRootD data access
advancedTools        advanced tools menu
diagnostics          diagnostic tools menu
helpMe               more help
installPip           install relocatable pip modules locally
installRpm           install relocatable rpms locally
printMenu            show this menu
queryC               find / whatis container query
showVersions         show versions of installed software

[gguerrie@lxplus923 ~]$ lsetup "root 6.36.02-x86_64-el9-gcc14-opt"
************************************************************************
Requested:  root ...
 Setting up root 6.36.02-x86_64-el9-gcc14-opt ...
  ROOT is from lcgenv -p LCG_108_ATLAS_2 x86_64-el9-gcc14-opt ROOT
>>>>>>>>>>>>>>>>>>>>>>>> Information for user <<<<<<<<<<<<<<<<<<<<<<<<
 root:
   Tip for _this_ standalone ROOT and grid (ie prun) submission:
    avoid --athenaTag if you do not need athena
    use --rootVer=6.36.02 --cmtConfig=x86_64-el9-gcc14-opt
 root:
   Note this ROOT from lcgenv sets up a long list of library paths.
   This may slow down executables; they look at each library path item.
   A faster alternative - setup the entire LCG stack with views:
    lsetup "views LCG_108_ATLAS_2 x86_64-el9-gcc14-opt"
************************************************************************
```

~~1. Data Management tools~~

~~2. Software distribution tools~~

3. Analysis facilities and reproducibility tools

What is an analysis facility?
A biased definition

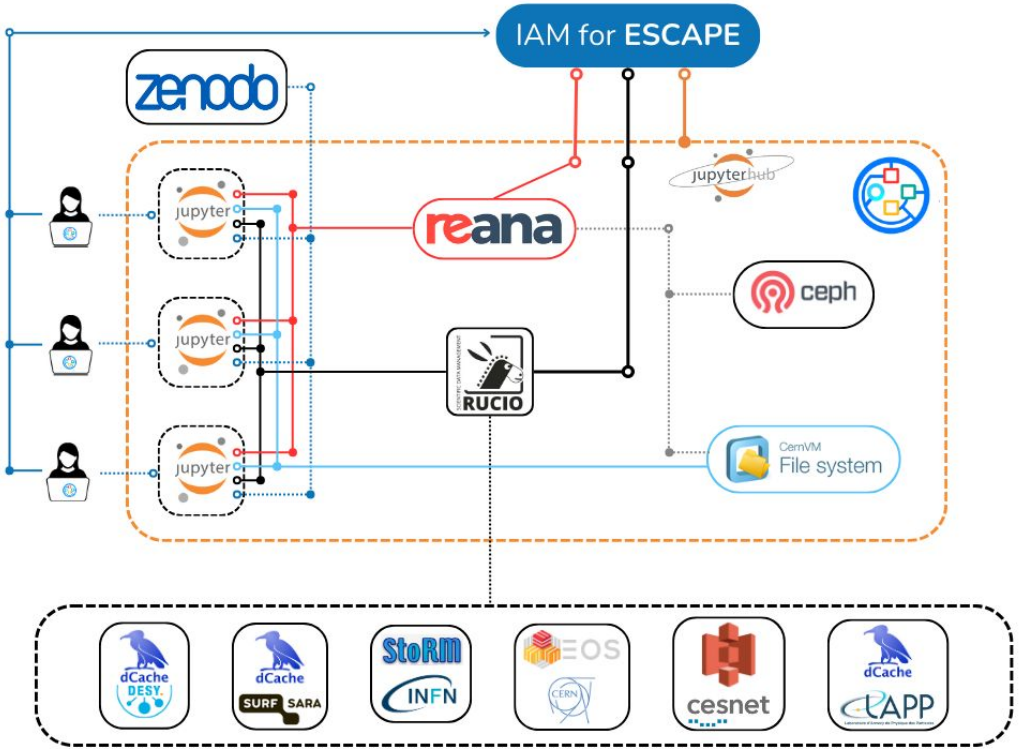
Analysis Facilities Whitepaper

Analysis facility (AF): infrastructure and services that provide integrated data, software and computational resources to execute one or more elements of an analysis workflow.

- Perform fast research iterations on large datasets interactively.
- Convert interactive to batch-schedulable workloads.
- Interact with the WLCG and scale outside of the facility on occasion.
- Reproducibly instantiate desired software stack.
- Collaborate in a multi-organisational team.
- Efficiently train machine learning models for HEP.

The ESCAPE Virtual Research Environment

1. AAI: A federated and reliable **Authentication and Authorization** layer
2. The Rucio Data Lake: A **federated distributed storage** solution, providing functionalities for data injection and replication through a Data Management framework (Rucio)
3. REANA: A **computing** cluster supplying the processing power to run full analyses with REANA, a re-analysis service
4. CVMFS: A **read-only file system** designed to distribute software, and more.
5. JupyterHub: A **notebook interface** with containerised environments to hide the infrastructure's complexity from the user.



reana

Reproducible research data analysis platform

Flexible

Run many computational workflow engines.

Scalable

Support for remote compute clouds.

Reusable

Containerise once, reuse elsewhere. Cloud-native.

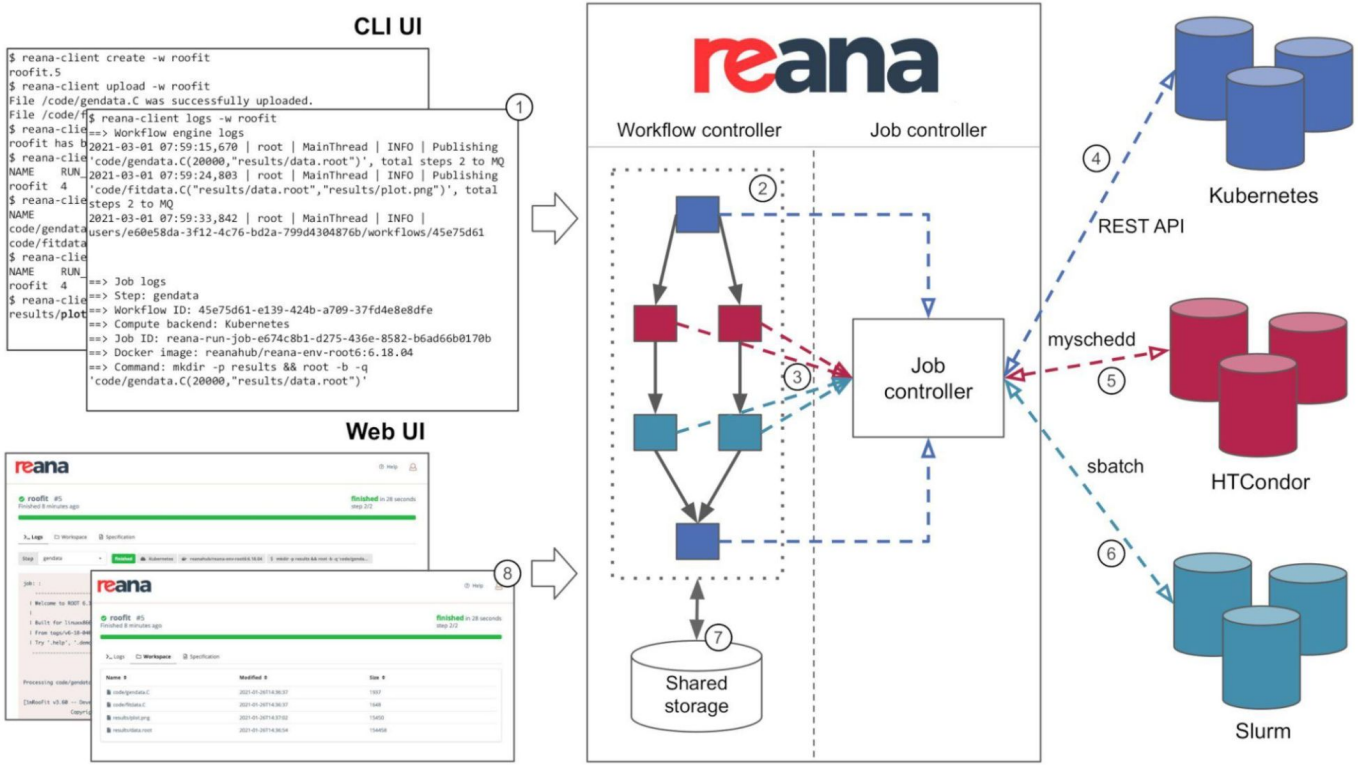
Free

Free Software. MIT licence.
Made with ❤ at CERN.

REANA

Reproducible research data analysis platform

ref.



Multiple compute backends:

- Kubernetes
- HTCondor
- Slurm

Multiple workflow languages:

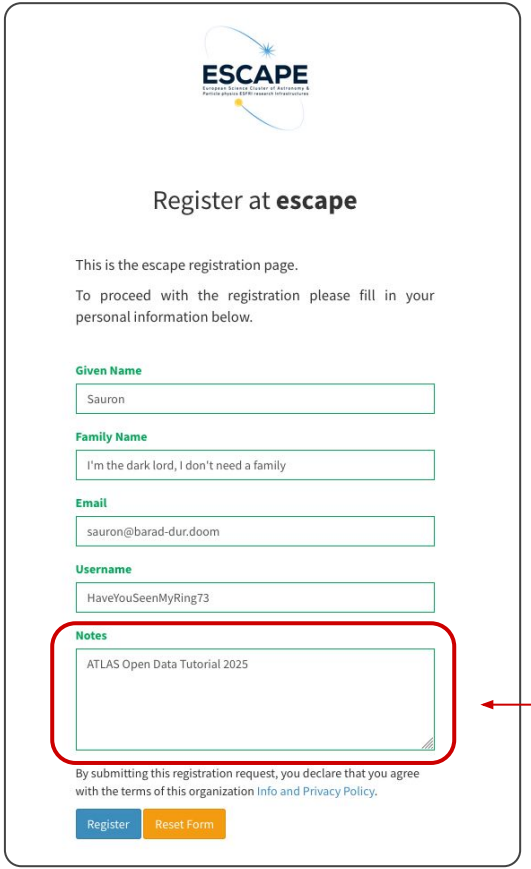
- CWL
- Serial
- Snakemake
- Yadage

Multiple means of use:

- Command-line client
- Web UI

How to get an account

- Connect to https://jhub-vre.cern.ch
- Wait until we (I) grant you permission



DEMO

A few examples from our repos

ATLAS Open Data

- Stored with Rucio in a public endpoint
- Find examples at https://github.com/atlas-outreach-data-tools/notebooks-collection-opendata
- We're going to go through this: metadata notebook

REANA

- Example

RUCIO

- We're gonna go through a basic set of commands