# Reusable and Reproducible Data Analyses with REANA

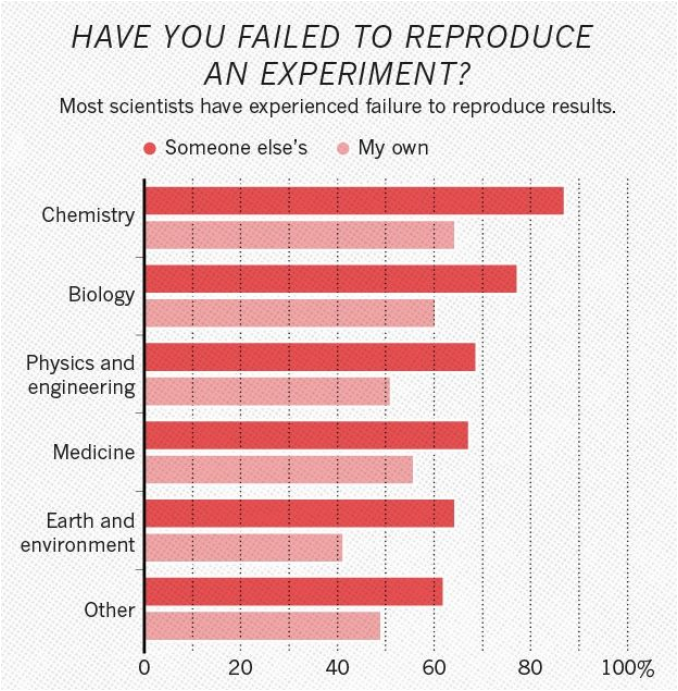
Tibor Šimko

CERN

2nd Big Data User Forum, CERN, December 10th 2024

https://indico.cern.ch/event/1468866/

# Reproducibility? Reusability?




https://www.nature.com/articles/533452a

| | | Data | |
|---|---|---|---|
| | | Same | Different |
| Analysis | Same | Reproducible | Replicable |
| | Different | Robust | Generalisable |

https://the-turing-way.netlify.app/reproducible-research/overview/overview-definitions.html

| Label | Data: Parameters | Data: Raw Data | Platform / Stack | Implementation | Method | Research Objective | Actor | Gain |
|---|---|---|---|---|---|---|---|---|
| **Repeat** | - | - | - | - | - | - | | Determinism |
| **Param. Sweep** | x | - | - | - | - | - | | Robustness / Sensitivity |
| **Generalize** | (x) | x | - | - | - | - | | Applicability across different settings |
| **Port** | - | - | x | - | - | - | | Portability across platforms, flexibility |
| **Re-code** | - | - | (x) | x | - | - | | Correctness of implementation, flexibility, adoption, efficiency |
| **Validate** | (x) | (x) | (x) | (x) | x | - | | Correctness of hypothesis, validation via different approach |
| **Re-use** | - | - | - | - | - | x | | Apply code in different settings, Re-purpose |
| **Independent x *(orthogonal)*** | | | | | | | x | Sufficiency of information, independent verification |

**Figure 1** PRIMAD Model: Categorizing the various types of reproducibility by varying the (P)latform, (R)esearch Objective, (I)mplementation, (M)ethod, (A)ctor and (D)ata, analyzing the gain they bring to computational experiments. x denotes the variable primed i.e. changed, (x) a variable that may need to be changed as a consequence, whereas – denotes no change.

https://drops.dagstuhl.de/opus/volltexte/2016/5817/pdf/dagrep_v006_i001_p108_s16041.pdf

Half of researchers cannot reproduce their own results

From “reproducibility” to “reusability”

# Four pillars of reproducible computational research

**I. Where is your input data?**

- Input files
- Input parameters

**II. Where is your analysis code?**

- User code
- Software frameworks

**III. What is your environment?**

- Operating system
- Container images
- Live database calls

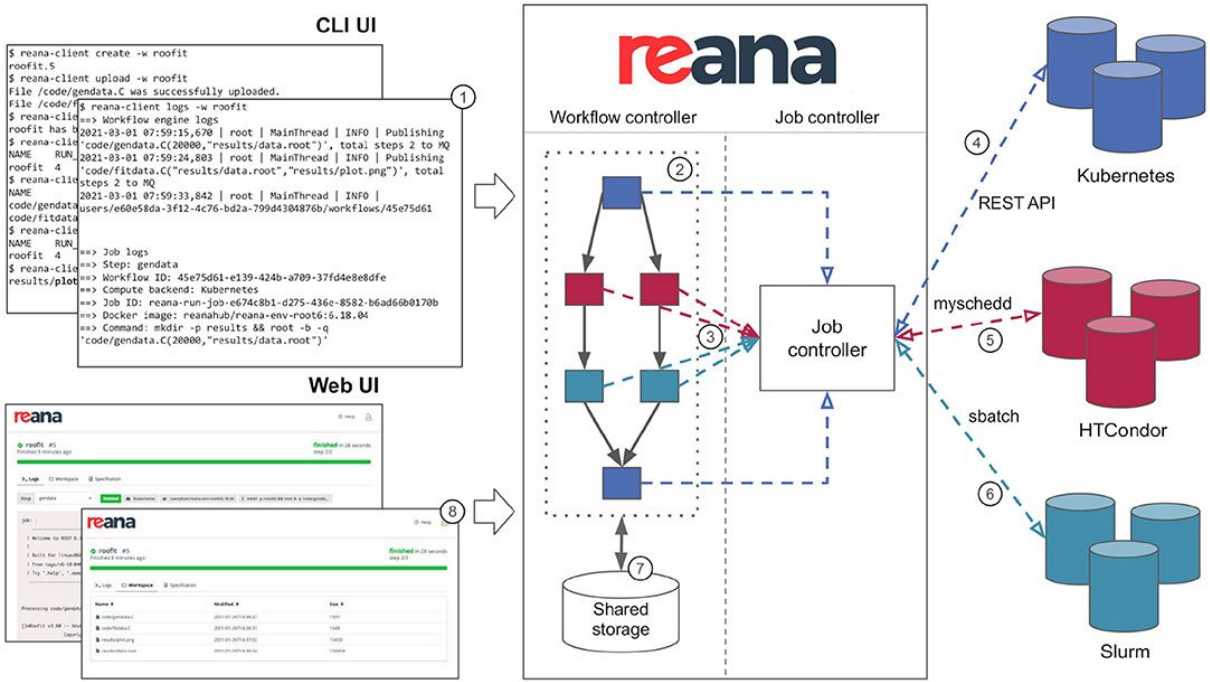
**IV. How to arrive at results?**

- Notebooks
- Workflows
- Shell scripts

Capturing essential information about an analysis in a structured actionable manner

# What is REANA?

Running containerised analysis workflows on the cloud




Multiple **compute backends**:

- Kubernetes
- HTCondor
- Slurm

Multiple **workflow languages**:

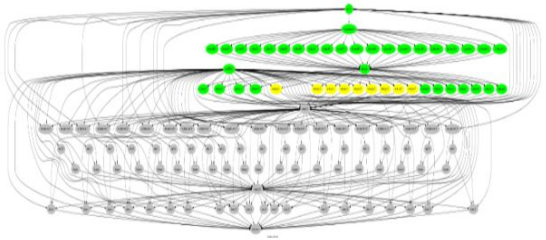
- CWL
- Serial
- Snakemake
- Yadage

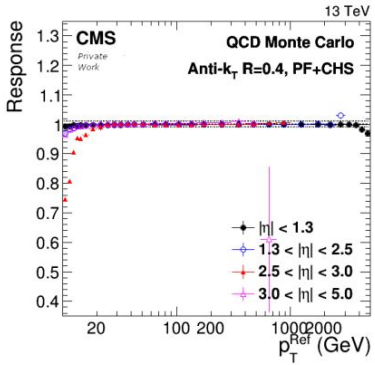
Multiple **means of use**:

- Command-line client
- Web UI

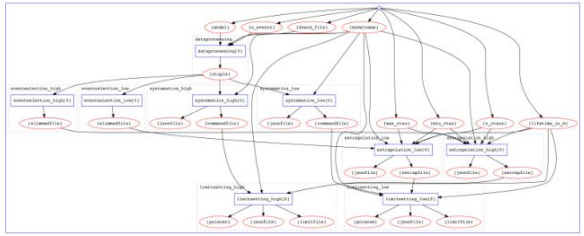
https://www.reana.io

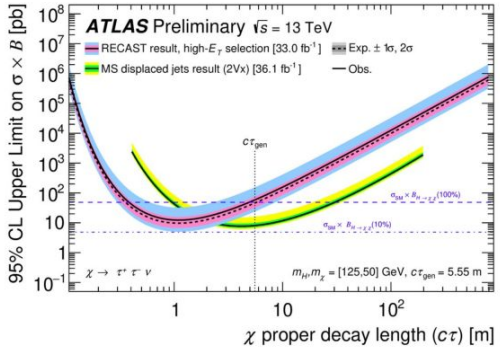
# Use cases: data production and data analyses






Data production example: CMS jet energy resolutions and corrections
https://github.com/alintulu/reana-demo-JetMETAnalysis






Data analysis example: ATLAS displaced jet reinterpretations
https://cds.cern.ch/record/2714064

# ATLAS pMSSM searches

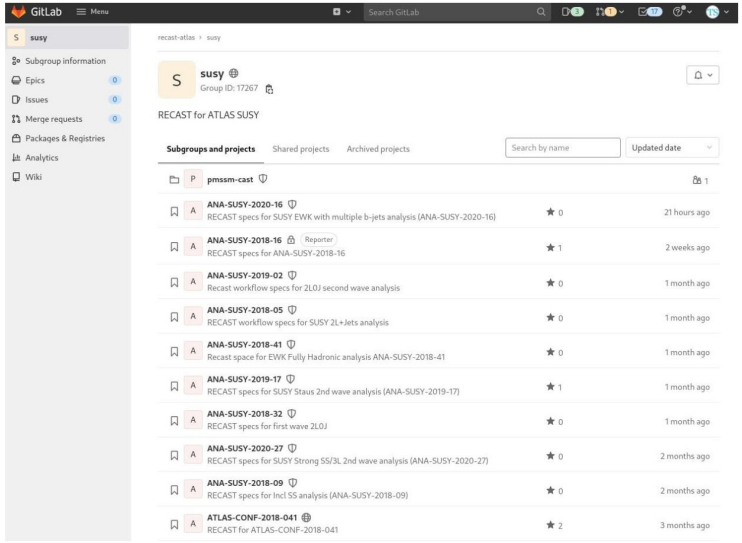



**Figure 1.** A screenshot of the ATLAS SUSY group analyses preserved on GitLab. Each repository is labeled with the internal ATLAS analysis identifier and contains both workflow files and additional data files needed for the computational processing.

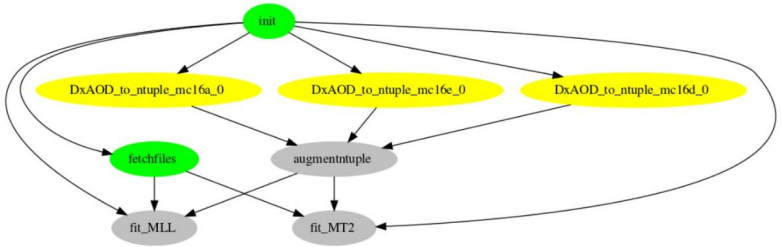



**Figure 2.** A typical pMSSM workflow. The computational runtime is about 10 minutes without systematics (test payload) and about 10 hours with all systematics (real payload).

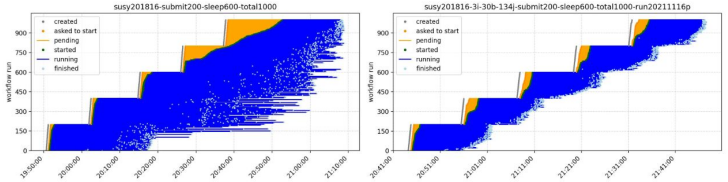
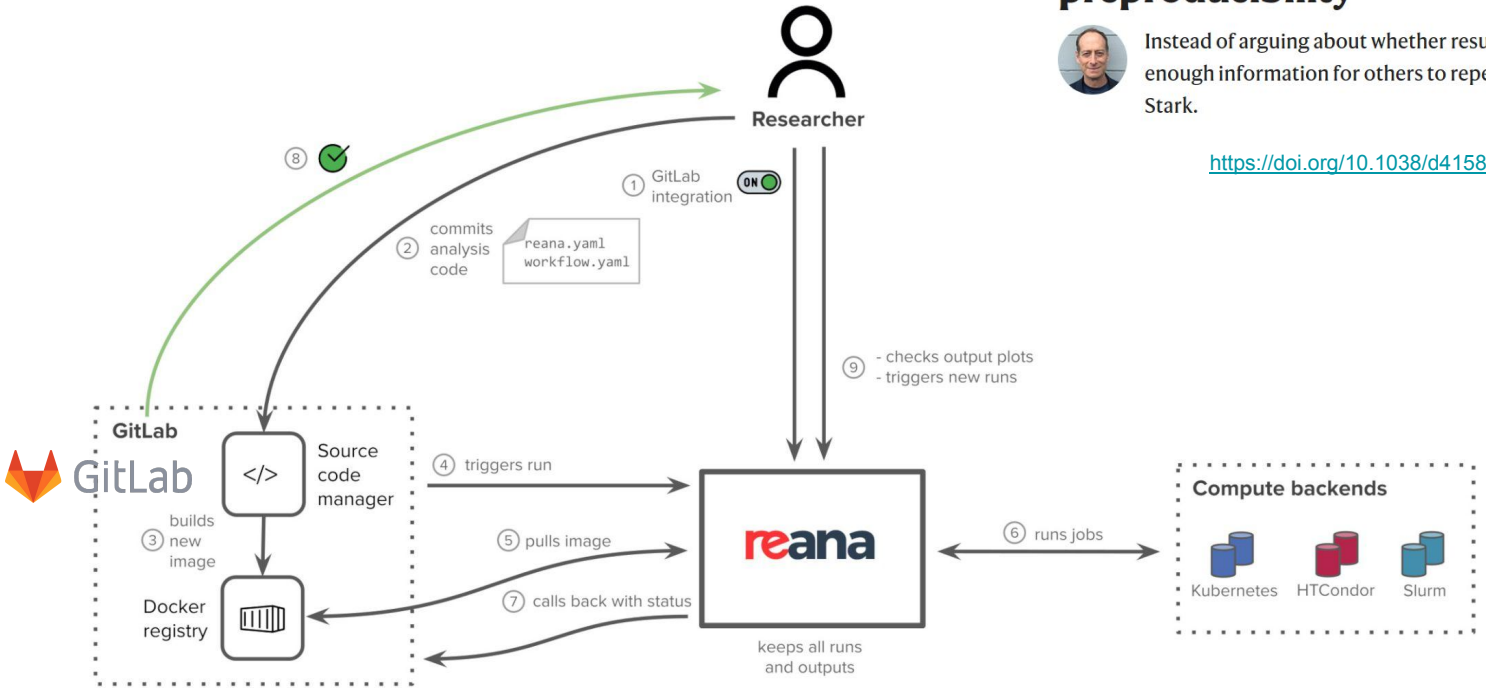



**Figure 8.** A scalability test submitting 200 workflows every 10 minutes. A cluster with 448 cores (left) cannot keep up with the load. A cluster with 1072 cores (right) can comfortably hold the incoming workload.

https://arxiv.org/abs/2403.03494

Streamlining the execution of thousands of reinterpretation workflows at scale

# Driving future reproducibility




WORLD VIEW · 24 MAY 2018

## Before reproducibility must come preproducibility

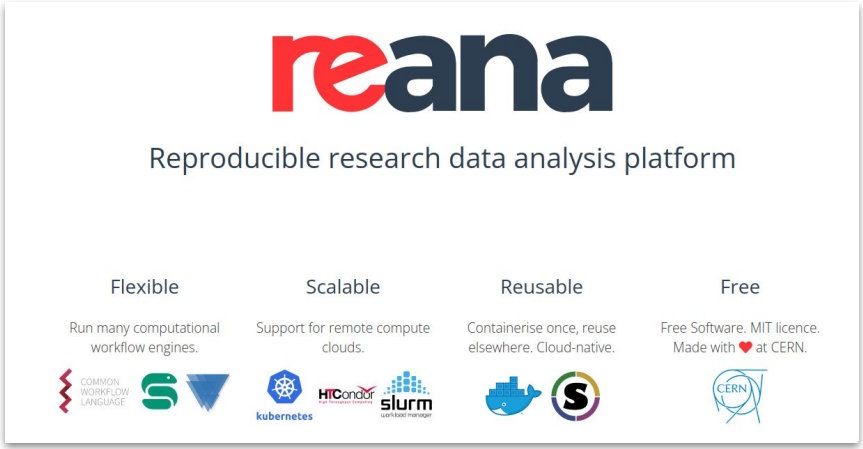


Instead of arguing about whether results hold up, let's push to provide enough information for others to repeat the experiments, says Philip Stark.

https://doi.org/10.1038/d41586-018-05256-0

REANA as a continuous integration engine for source code management systems

# Community



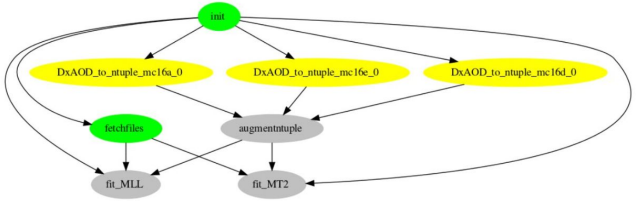

REANA reproducible analysis platform
https://www.reana.io


Workshop on workflow languages for HEP (May 2024)
https://indico.cern.ch/event/1380367/

Seeing synergies with astronomy, life sciences, *etc*

Seeking synergies across experiments

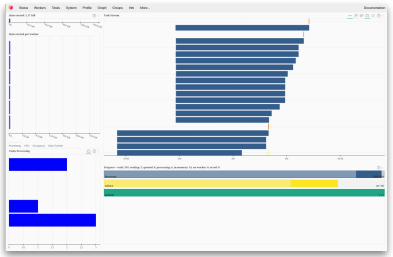
# Supporting Dask workflows




```
inputs:
  files:
    - myanalysis.py
workflow:
  type: serial
  resources:
    dask:
      image: mydaskenv:2023.10.1
      cores: 100
   specification:
     steps:
       - environment: mydaskenv:2023.10.1
         commands:
           - python myanalysis.py
outputs:
  files:
    - myhistogram.png
```

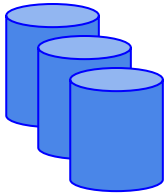
```
$ kubectl get pods
NAME                                                              STATUS   AGE
reana-run-batch-0ecf0402-b0fe-41f0-885e-8c88bb094fe0-zckqb        Running  40s
reana-run-dask-0ecf0402-b0fe-41f0-885e-8c88bb094fe0-defaul2hxx9   Running  10s
reana-run-dask-0ecf0402-b0fe-41f0-885e-8c88bb094fe0-defaul5rxqx   Running  10s
reana-run-dask-0ecf0402-b0fe-41f0-885e-8c88bb094fe0-defaul764b9   Running   9s
reana-run-dask-0ecf0402-b0fe-41f0-885e-8c88bb094fe0-defaulk7flr   Running  10s
reana-run-dask-0ecf0402-b0fe-41f0-885e-8c88bb094fe0-defaulkq8gm   Running  10s
reana-run-dask-0ecf0402-b0fe-41f0-885e-8c88bb094fe0-defaulm24vx   Running  10s
reana-run-dask-0ecf0402-b0fe-41f0-885e-8c88bb094fe0-defaulpnjk5   Running  25s
reana-run-dask-0ecf0402-b0fe-41f0-885e-8c88bb094fe0-defaulvjgs2   Running  10s
reana-run-dask-0ecf0402-b0fe-41f0-885e-8c88bb094fe0-schedusf4bg   Running  40s
reana-run-job-4ac45b36-1f8e-47c5-80a3-c73cf03954dd-hnpvt          Running  29s
```

A corresponding Dask cluster is spawned on-demand at the start of the workflow run



A need to “revive” past Dask versions

# Compute on Kubernetes



**infrastructure nodes** (API server, workflow controller, message queue)

**batch orchestration nodes** (running workflows and notebooks)

**job nodes** (running main jobs)

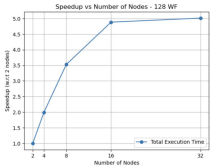



Figure 5: Speedup of experiment execution time when running 128 test workflows on clusters with progressively increasing number of nodes.

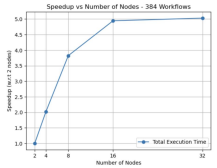
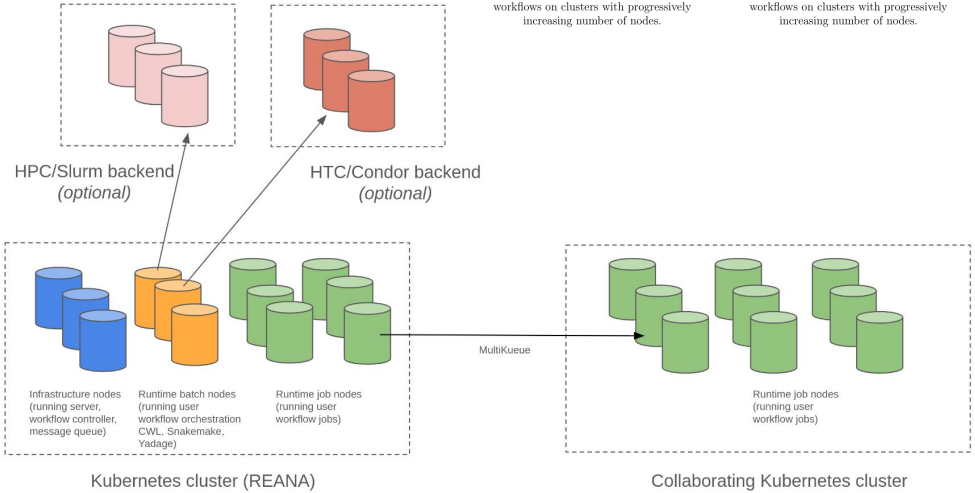



Figure 6: Speedup of experiment execution time when running 384 test workflows on clusters with progressively increasing number of nodes.

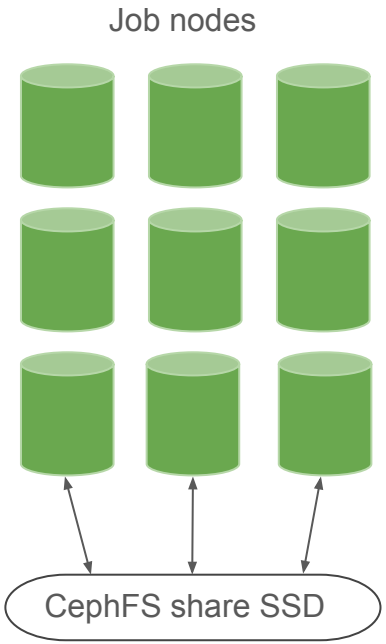



Typical node flavour: 8 vCPUs, 15 GB RAM

Typical number of nodes: 50-100

Testing Kueue/MultiKueue for job dispatch

# Shared storage




Workflow jobs use shared scratch space mounted as Kubernetes persistent volume

```
from snakemake.remote.XRootD import RemoteProvider as XRootDRemoteProvider

XRootD = XRootDRemoteProvider(stay_on_remote=True)

file_numbers = XRootD.glob_wildcards("root://eosuser.cern.ch//eos/user/j/johndoe/mydata_{n}.csv").n

rule all:
    input:
        expand("mylocaldata/myfile_{n}.csv", n=file_numbers)

rule process_data:
    input:
        XRootD.remote("root://eosuser.cern.ch//eos/user/j/johndoe/mydata_{n}.csv")
    output:
        "mylocaldata/myfile_{n}.csv"
    container:
        "docker://docker.io/johndoe/myanalysis:1.0"
    shell:
        "process_data {input[0]} {output[0]}"
```

Data can be accessed “live” via XRootD

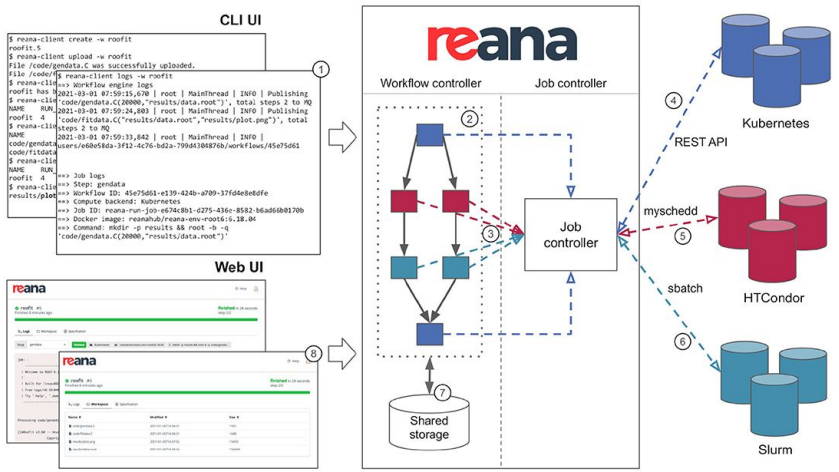
```
rule mystep:
  container:
    "docker://docker.io/johndoe/myanalysis:1.0.0"
  resources:
    voms_proxy=True,
    rucio=True
  shell:
    "rucio get my_rucio_scope:my_rucio_file"
```

… or via Rucio

# Conclusions

Ultimate goal: facilitate future reuse of current research data analyses

- driving reuse through preproducibility
- technology challenges: large containers, complex computational workflows
- sociology challenges: declarative programming paradigm, publish-or-perish culture
- synergies with computational reproducibility needs in astronomy, life sciences




**Data + Code + Environment + Services + Workflow = Reusable Analyses**