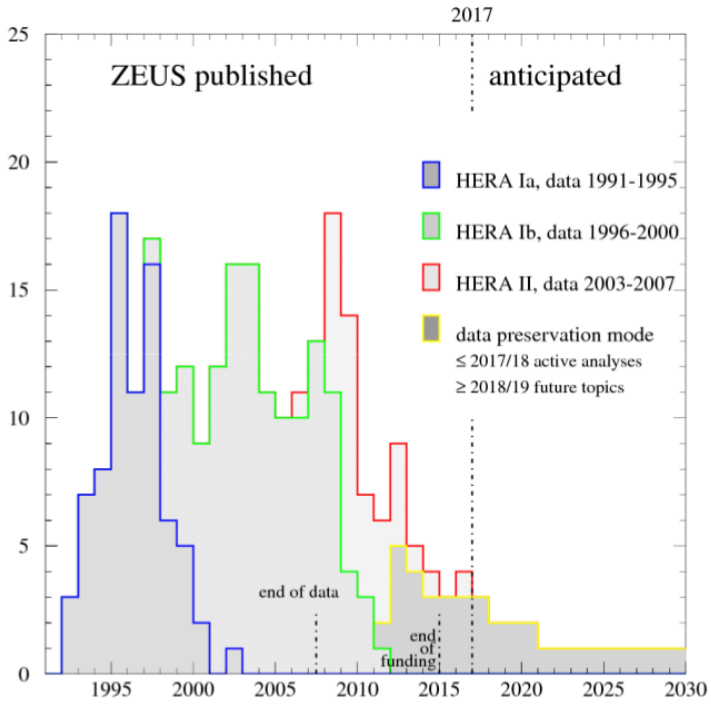
Preserving and reusing high-energy-physics data analyses

S. Dallmeier-Tiessen[2], R. Dasler[2], P. Fokianos[2], J. Kunčar[1], A. Lavasa[2], A. Mattmann[2], D. Rodríguez[1], T. Šimko[1], A. Trzcinska[2], I. Tsanaktsidis[2]

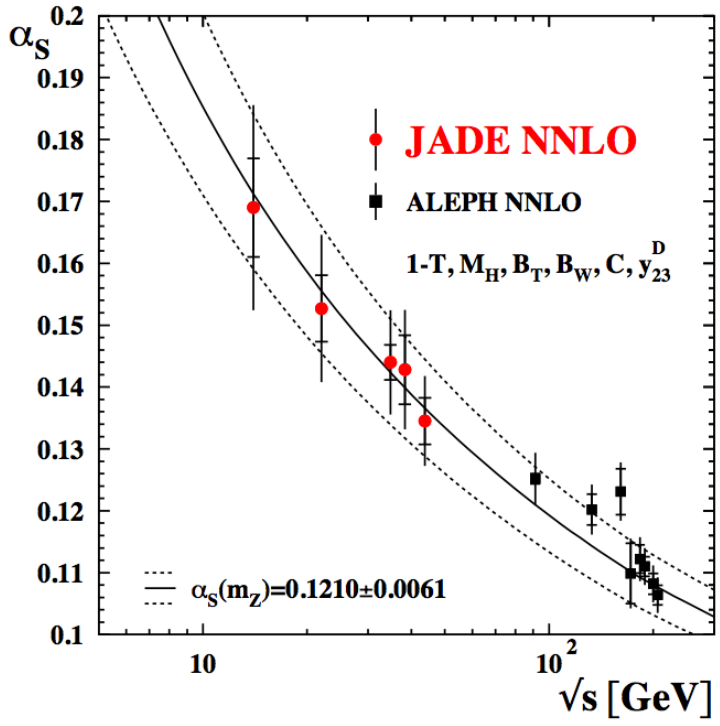
[1] *CERN Information Technology*
[2] *CERN Scientific Information Service*

Open Repositories 2017 · Brisbane, Australia · 26–30 June 2017

Long-term value of data!



Achim Geiser `https://indico.cern.ch/event/588219`



DPHEP `https://arxiv.org/abs/1205.4667`

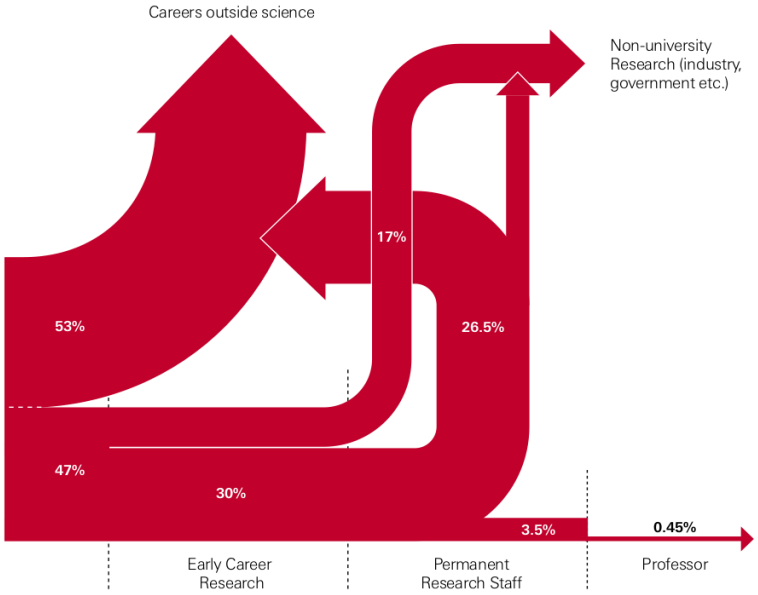
Collaborations publish papers even ∼15 years after data taking ends.

JADE data (1979–1986) still unique even ∼35 years later.

Long-term value of knowledge?

CMS collaboration

Experimental physics done by groups of ∼3000 physicists.



Career after PhD

THE ROYAL SOCIETY

High turnover of young researchers.

CERN Analysis Preservation

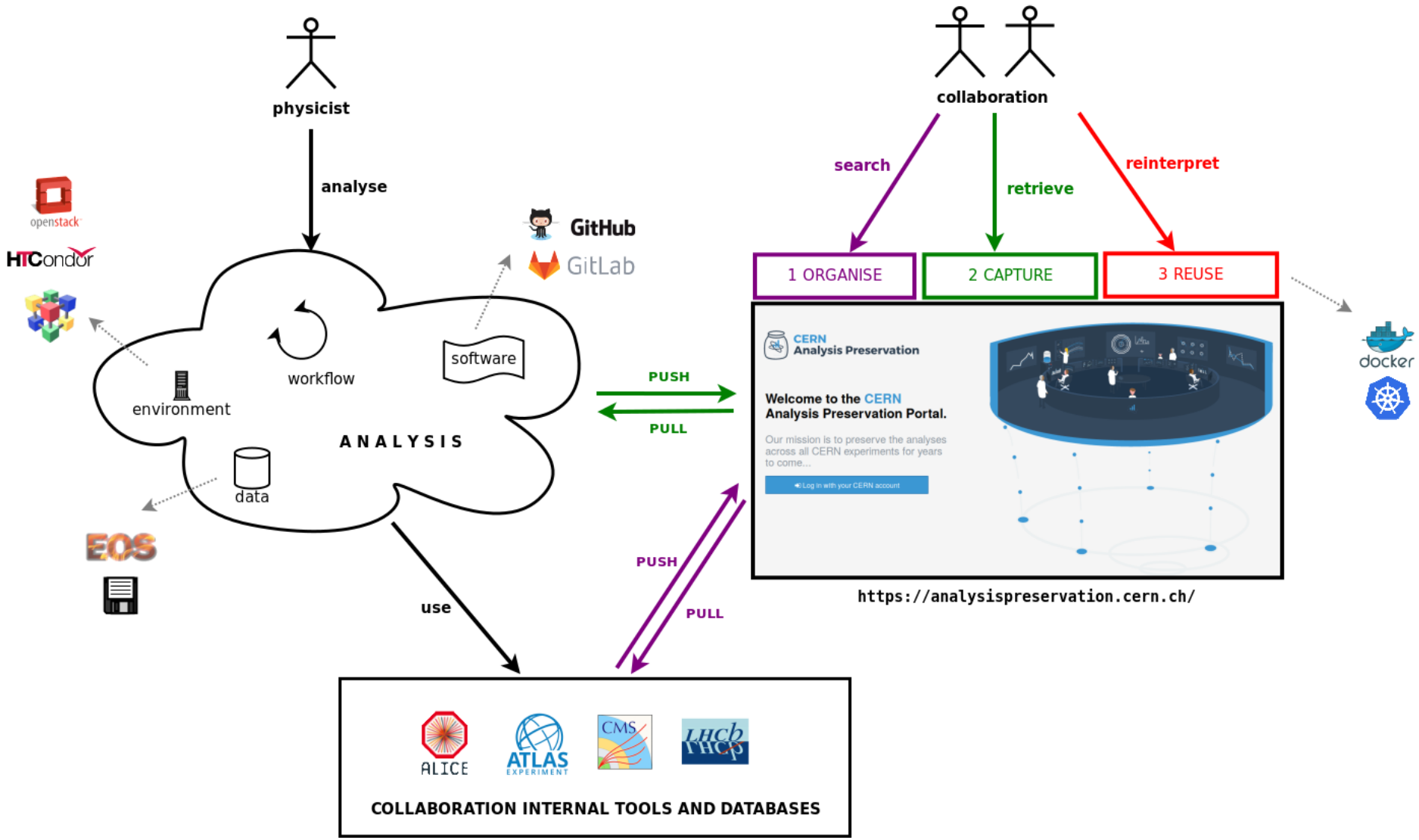
- A platform for **preserving knowledge** and **assets** of an individual physics analysis.
- Capturing the elements needed to **understand** and **rerun** an analysis even several years later:

  - ✔ data
  - ✔ software
  - ✔ environment
  - ✔ workflow
  - ✔ context
  - ✔ documentation

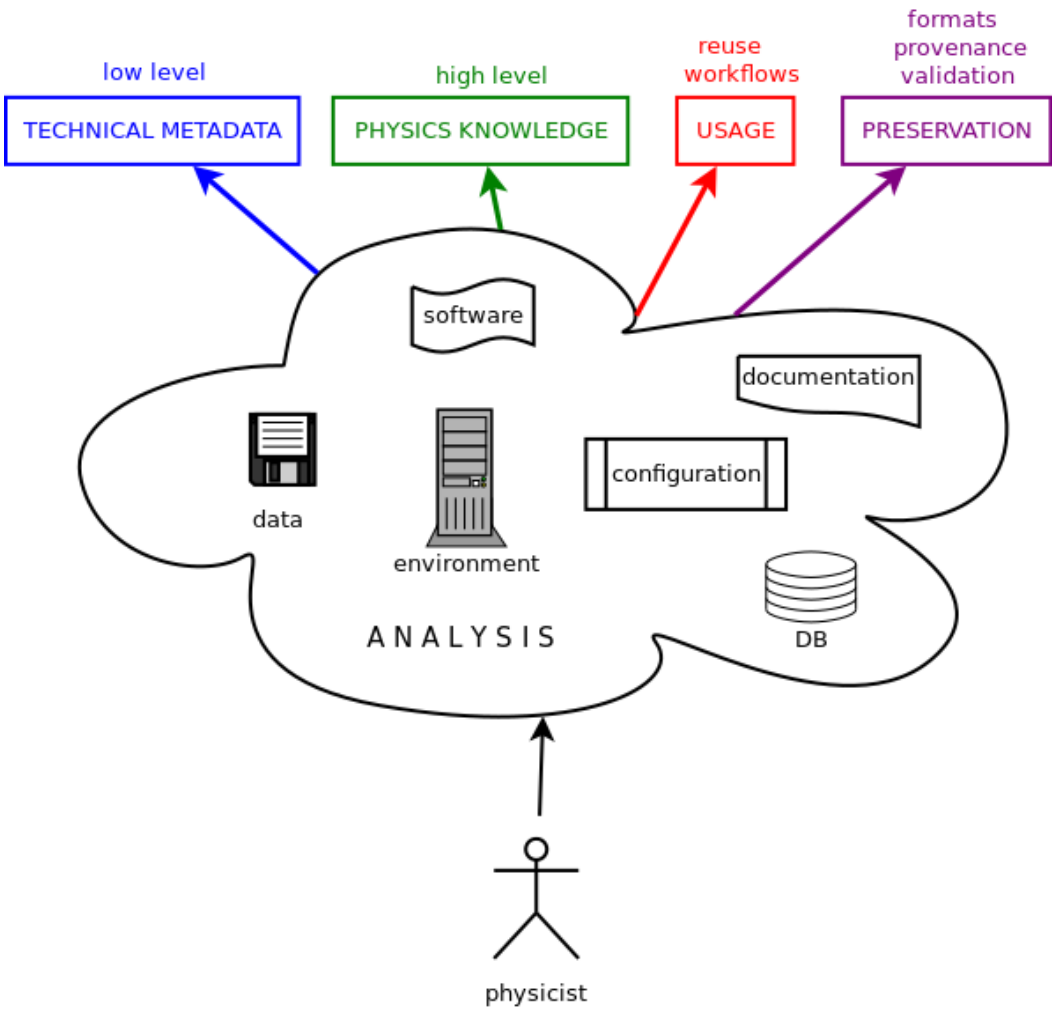
- Advanced **search** for high-level physics information
- Applying standard **collaboration access restrictions**

Developed by CERN IT and CERN SIS in close collaboration with LHC experiments

System overview



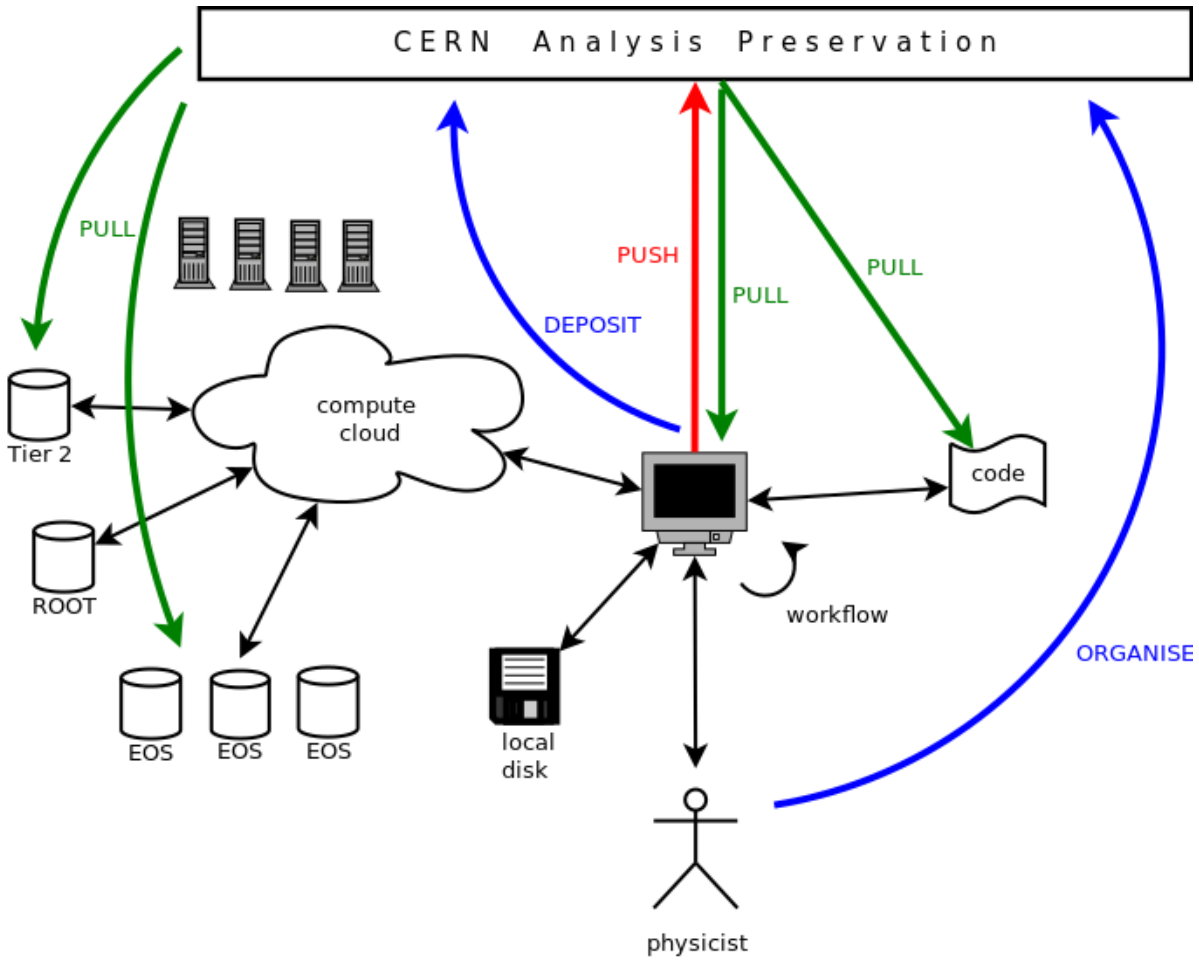
1. Describing an analysis



- JSON Schema
- W3C DCAT
- domain-specific fields

Structuring knowledge behind research data analysis.

2. Capturing an analysis

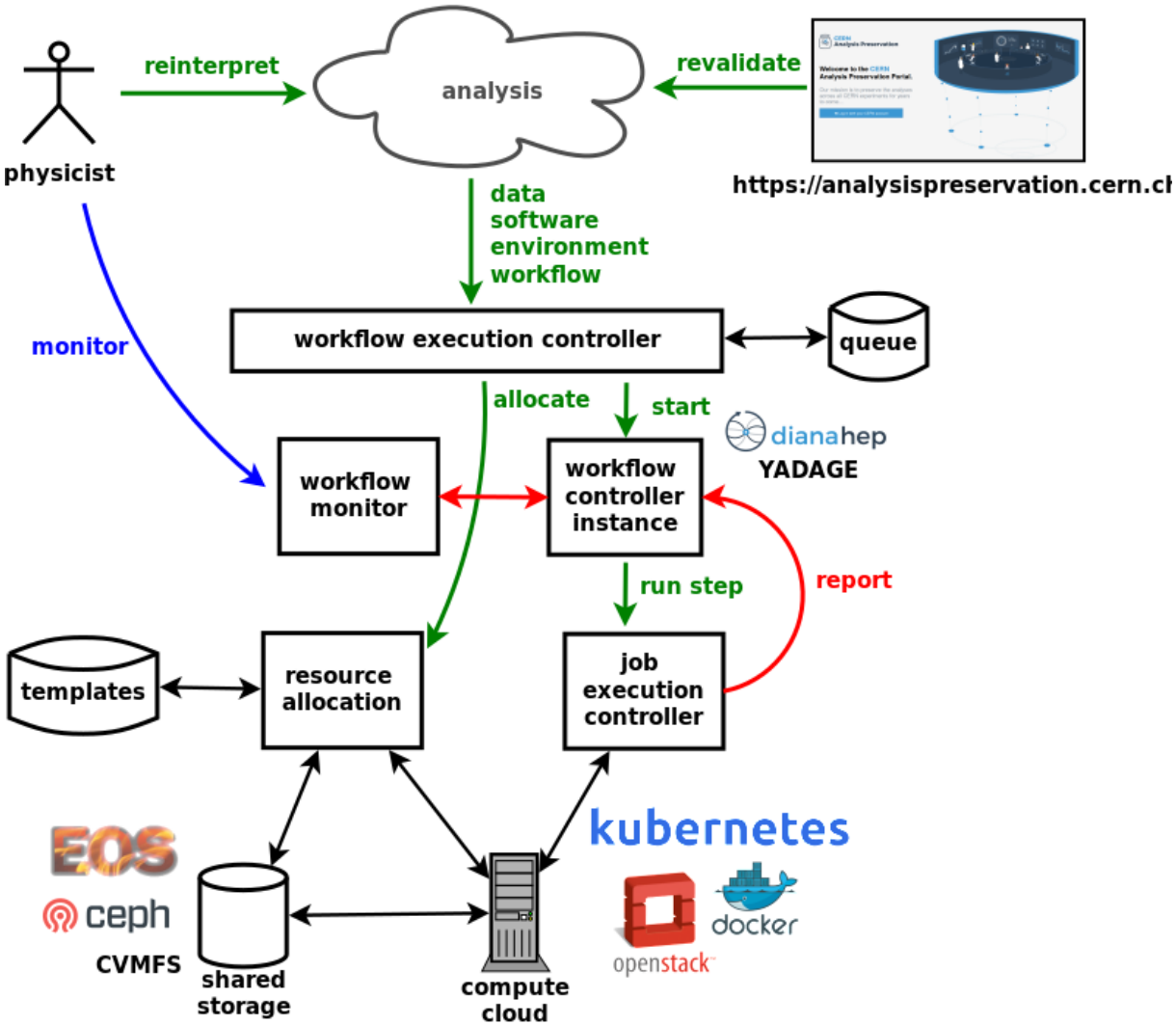


INVENIO

- datasets:
  local storage,
  cloud storage
- software:
  Git, SVN
- information:
  DBs, TWiki,
  SharePoint
- protocols:
  HTTP, XRootD

Taking consistent snapshot of analysis assets at a certain time.

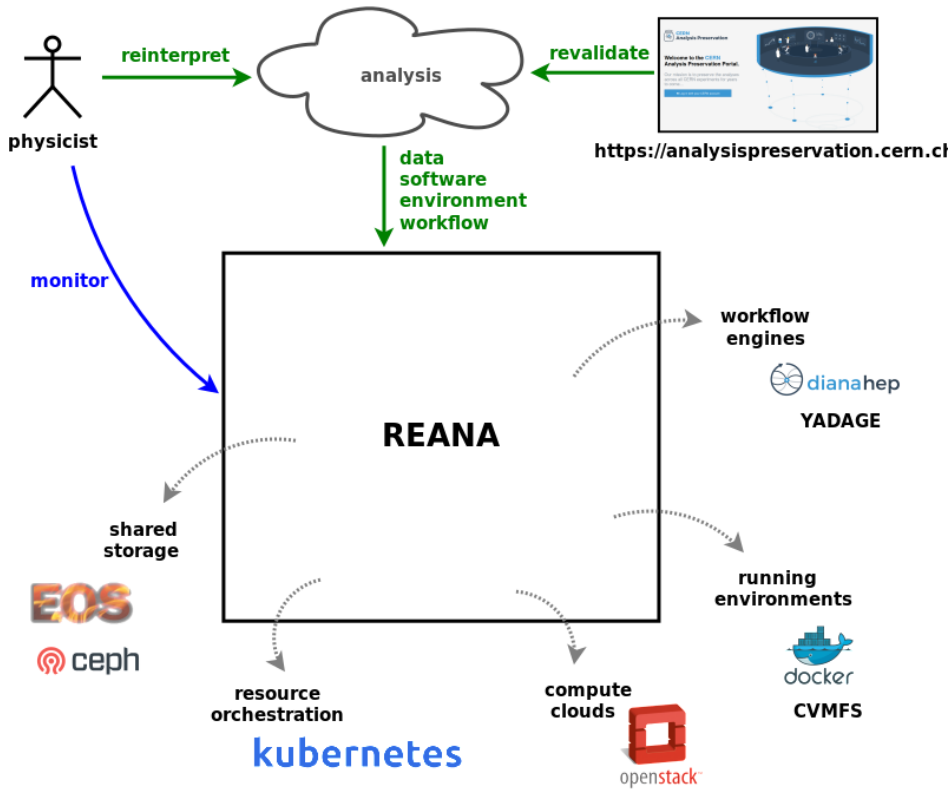
3. Reusing an analysis



Instantiating preserved analysis on the cloud.

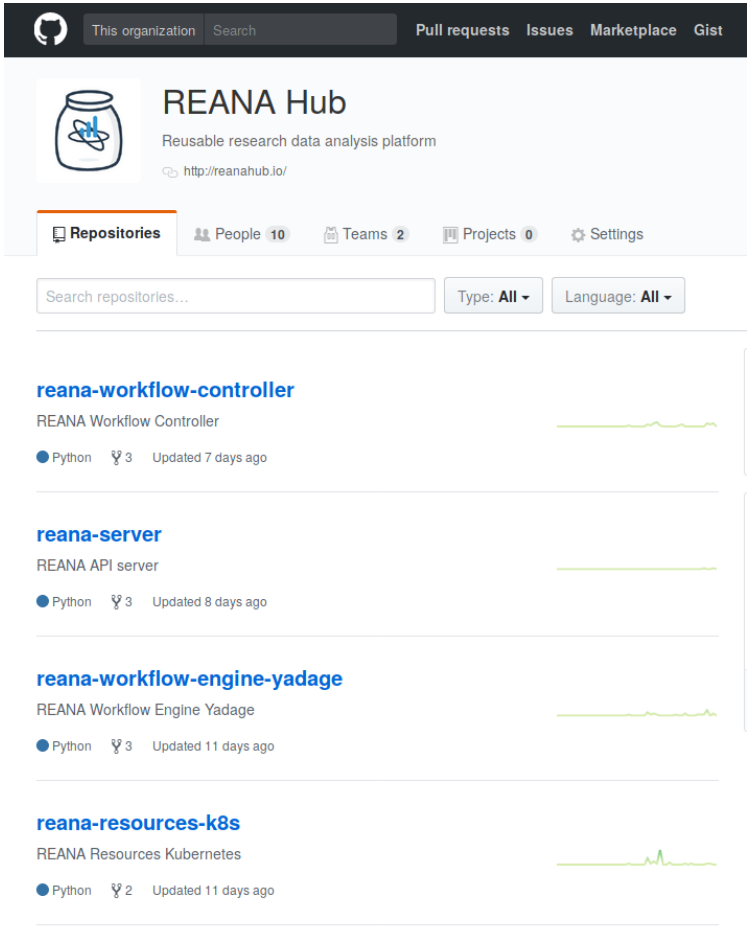
REANA = REusable ANAlyses

- a system for **reusable analysis** execution **on the cloud**
  - `https://reanahub.io`
- supporting **multiple scenarios**
  - multiple computing clouds
    → CERN OpenStack
  - multiple running environments
    → Docker with CVMFS
  - multiple resource orchestration
    → Kubernetes
  - multiple workflow engines
    → Yadage
  - multiple shared storage systems
    → Ceph, EOS



- close **collaboration** with DASPOS and recast

REANA is FOSS



REANA @ GitHub

REANA @ ReadTheDocs

Four questions

1 Input data

What is your input data?

- input files
- live DB calls

2 Analysis code

Which code analyses it?

- Jupyter notebook
- custom code

3 Compute environment

What is your environment?

- operating system
- software & libraries

4 Analysis workflow

Which steps did you take?

- single command
- complex workflows

Simple example: Jupyter

```
Region,1500,1600,1700,1750,1800,1850,1900,1950,1999,2008,2010,2012,2050,2150
World,100,100,100,100,100,100,100,100,100,100,100,100,100,100
Africa,18.8,19.7,15.5,13.4,10.9,8.8,8.1,8.8,12.8,14.5,14.8,15.2,19.8,23.7
Asia,53.1,58.4,63.9,63.5,64.9,64.1,57.4,55.6,60.8,60.4,60.4,60.3,59.1,57.1
Europe,18.3,19.1,18.3,20.6,20.8,21.9,24.7,21.7,12.2,10.9,10.7,10.5,7,5.3
Latin America and the Caribbean,8.5,1.7,1.5,2,2.5,3,4.5,6.6,8.5,8.6,8.6,8.6,9.1,9.4
Northern America,0.7,0.5,0.3,0.3,0.7,2.1,5,6.8,5.1,5,5,5,4.4,4.1
Oceania,0.7,0.5,0.4,0.3,0.2,0.2,0.4,0.5,0.5,0.5,0.5,0.5,0.5,0.5
```

1 **input**: CSV file

Regional Analysis

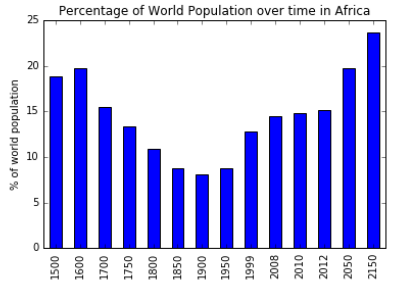
We'll start with a histogram depicting the evolution of a specific region's portion of the world population, in percentage.

```
In [6]: def histogram_by_region(region):
            local_pop=pop[['Region', str(region)]].groupby('Region').sum()

            plot=local_pop.plot(kind='bar', legend=None, title='Percentage of World Population over tim
        e in '+ str(region))

            plot.set_ylabel('% of world population')
            plot.set_xlabel('')
```

```
In [7]: histogram_by_region('Africa')
```



2 **code**: Jupyter notebook

```
FROM centos:7
RUN yum install -y epel-release
RUN yum install -y \
    gcc \
    python-devel \
    python-pip
RUN pip install ipython==5.0.0 jupyter==1.0.0
ADD world_population_analysis.ipynb /code/
ADD World_historical_and_predicted_populations_in_percentage.csv /code/
WORKDIR /code
CMD ["jupyter", "nbconvert","world_population_analysis.ipynb"]
```

3 **environment**: CentOS7, IP5

4 **workflow**: `jupyter nbconvert`

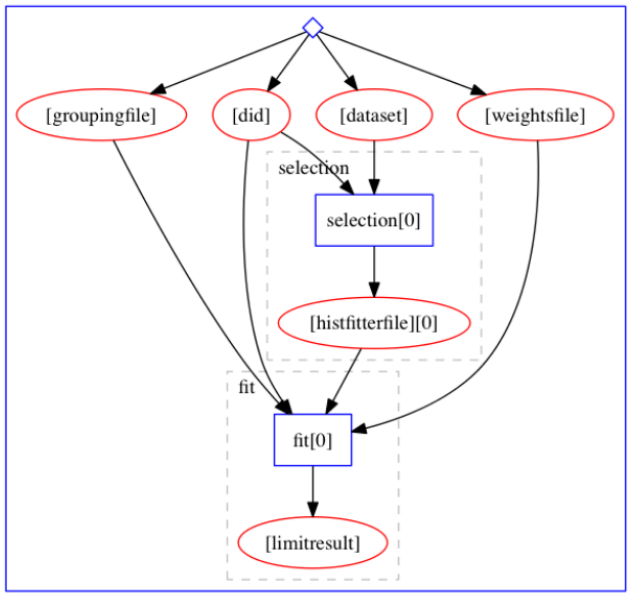
`https://github.com/reanahub/reana-demo-worldpopulation`

Complex example: DAG workflows

- **case studies** in high-energy-physics with LHC collaborations
  - ALICE AliPhysics post-LEGO train analysis
  - ATLAS multi-B-jets analysis
  - LHCb Lb2LcD0K analysis and data production

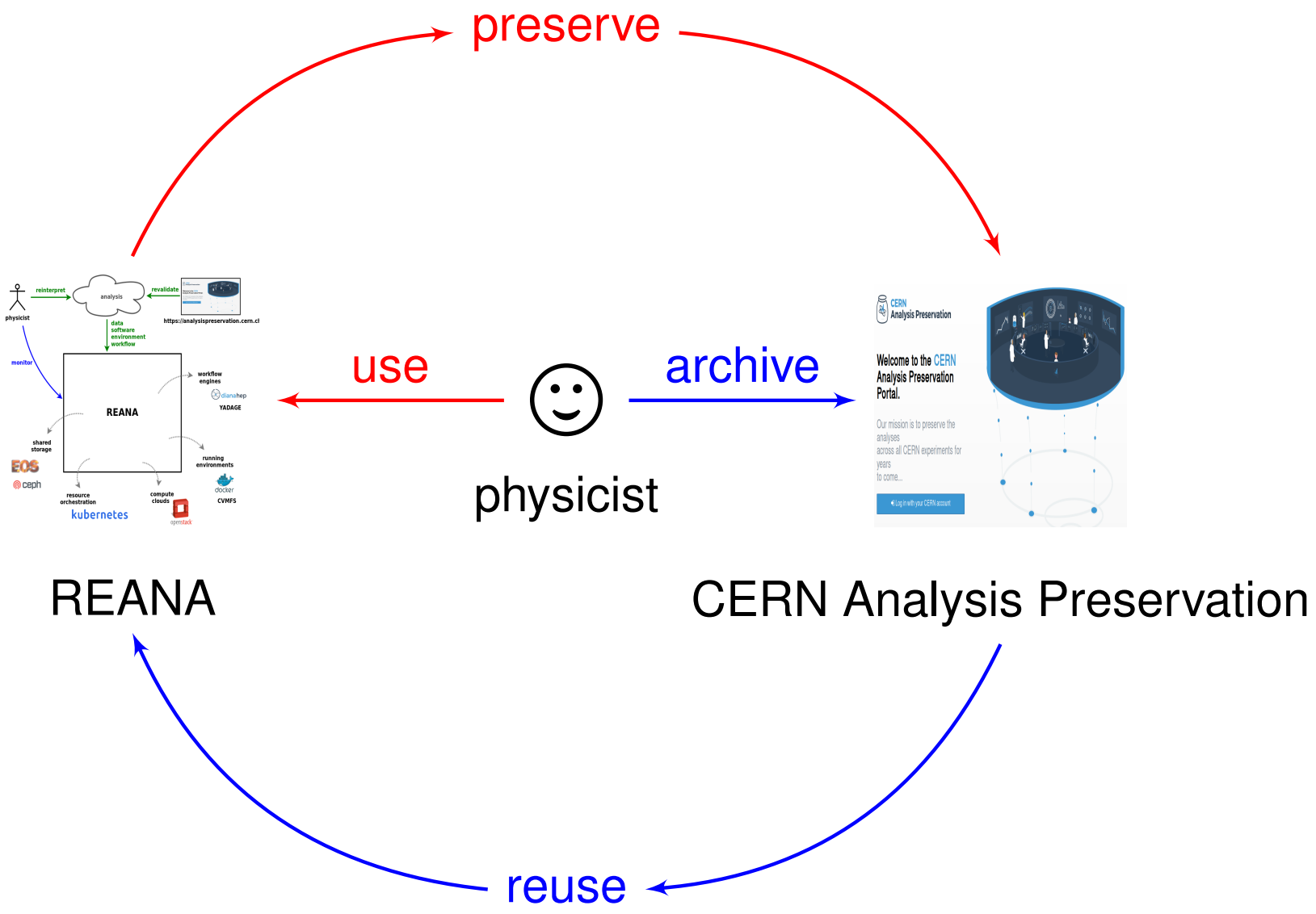
- **yadage** parametrised workflow engine


```yaml
stages:
  - name: selection
    dependencies: ['init']
    scheduler:
      scheduler_type: singlestep-stage
      parameters:
        dataset: {stages: init, output: dataset, unwrap: true}
        submitdir: '{workdir}/submitdir'
        outputprefix: '{workdir}/histfitter.root'
        did: {stages: init, output: did, unwrap: true}
      step: {$ref: 'selscript.yml#'}
  - name: fit
    dependencies: ['selection']
    scheduler:
      scheduler_type: singlestep-stage
      parameters:
        bkgtree: 'root://eosuser.cern.ch///eos/project/r/recast/Bkg_2.4.15-2-0_merged.root'
        datatree: 'root://eosuser.cern.ch///eos/project/r/recast/Data_2.4.15-2-0.root'
        outputjson: '{workdir}/fitoutput.json'
        selectionoutput: {stages: selection, output: histfitterfile, unwrap: true}
        weightsfile: {stages: init, output: weightsfile, unwrap: true}
        did: {stages: init, output: did, unwrap: true}
      step: {$ref: 'fitscript.yml#'}
```



Lukas Heinrich `http://github.com/diana-hep/yadage`

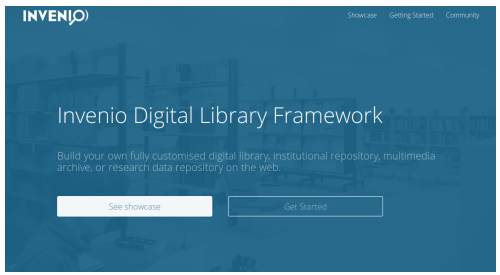
Reusability ⇌ Preservation



Conclusions

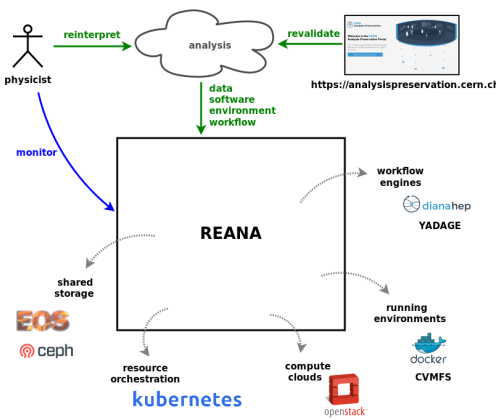
CERN Analysis Preservation

- http://analysispreservation.cern.ch
- http://github.com/cernanalysispreservation
- analysis-preservation-support@cern.ch



Invenio

- http://inveniosoftware.org
- http://github.com/inveniosoftware
- inveniosoftware
- info@inveniosoftware.org



REANA

- http://reanahub.io
- http://github.com/reanahub
- reanahub
- info@reanahub.io