Reusable data, reproducible analyses: today, tomorrow, next decade

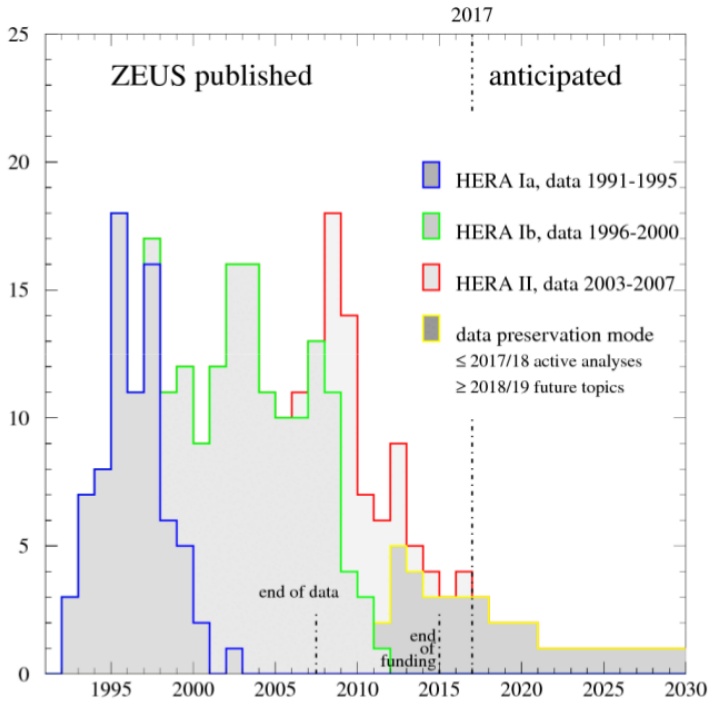
Tibor Šimko

CERN

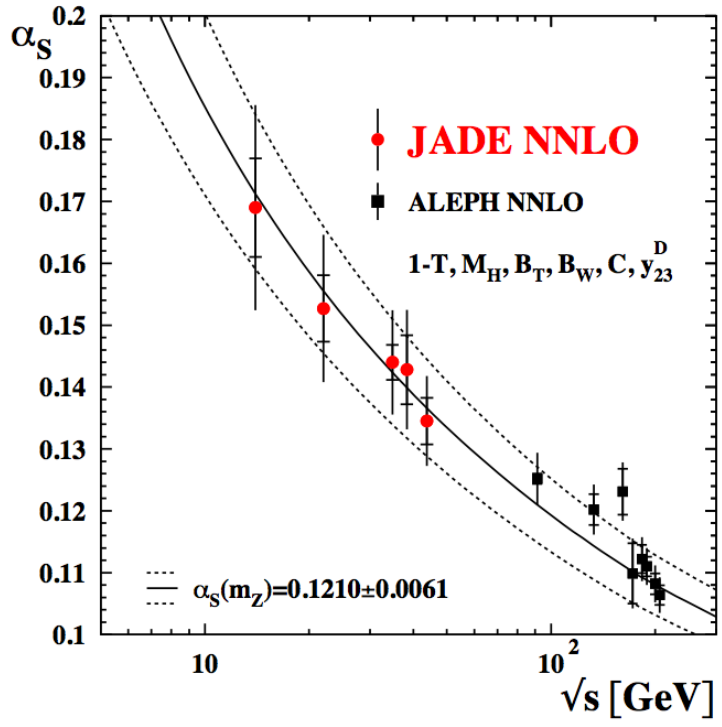
IEEE eScience2018 · Amsterdam, The Netherlands

29 October – 1 November 2018

Long-term value of data!



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



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

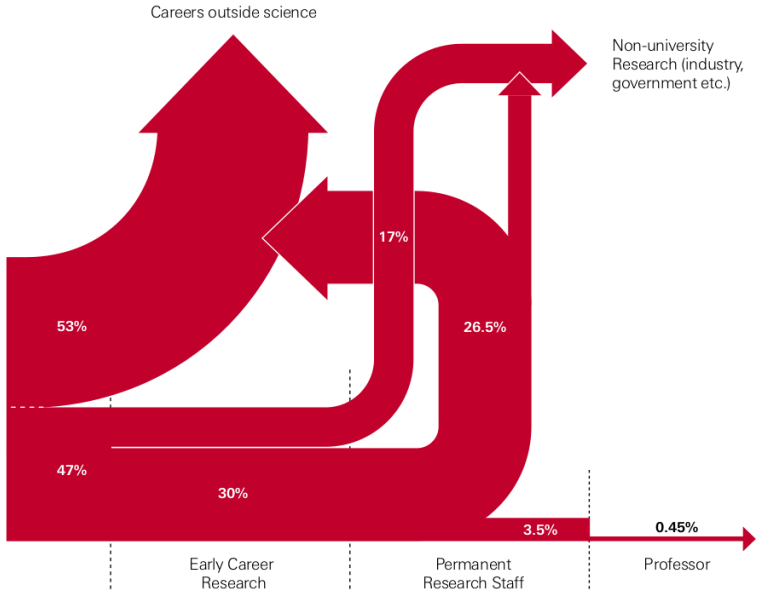
Collaborations publish papers even ∼15 years after data taking ends

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

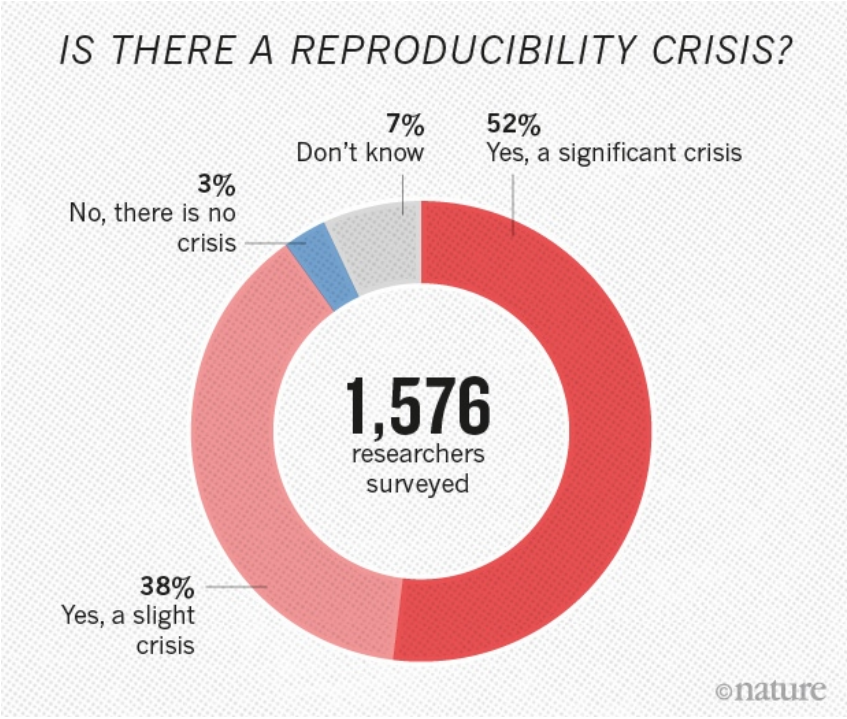


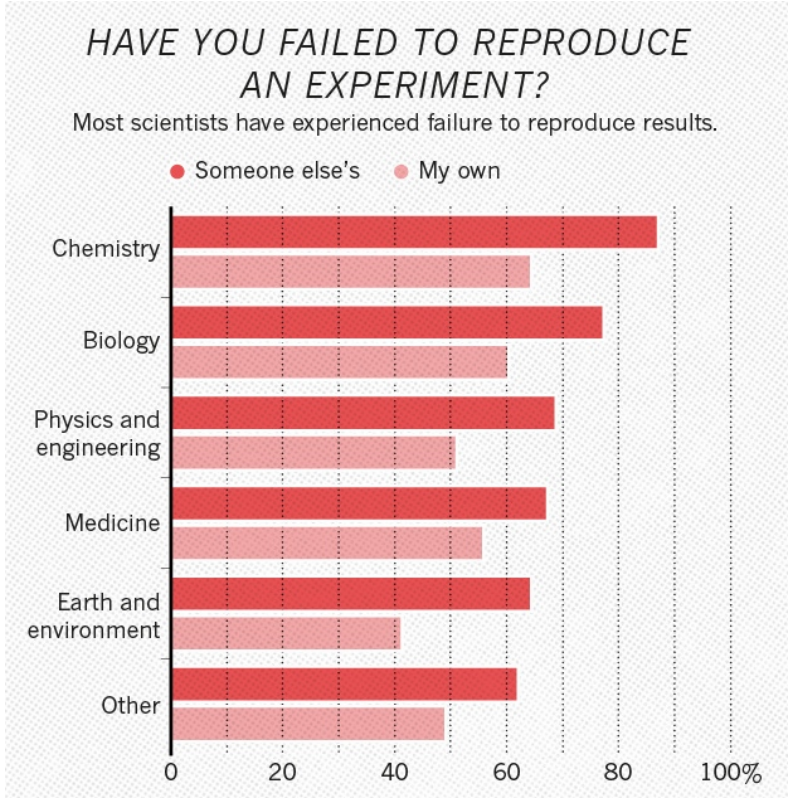
Career after PhD

THE ROYAL SOCIETY

High turnover of young researchers

Reusable and reproducible?





https://www.nature.com/news/1-500-scientists-lift-the-lid-on-reproducibility-1.19970

Half of researchers cannot reproduce their own experimental results

FAIR data principles

- **F**indable
  - persistent identifiers
  - rich metadata
  - indexed and searchable
- **A**ccessible
  - retrievable by identifiers
  - standard protocols
  - metadata vs data accessibility
- **I**nteroperable
  - knowledge representation language
  - common vocabularies
  - references to other metadata and data
- **R**eusable
  - domain-relevant attributes and community standards
  - clear licensing
  - provenance tracking

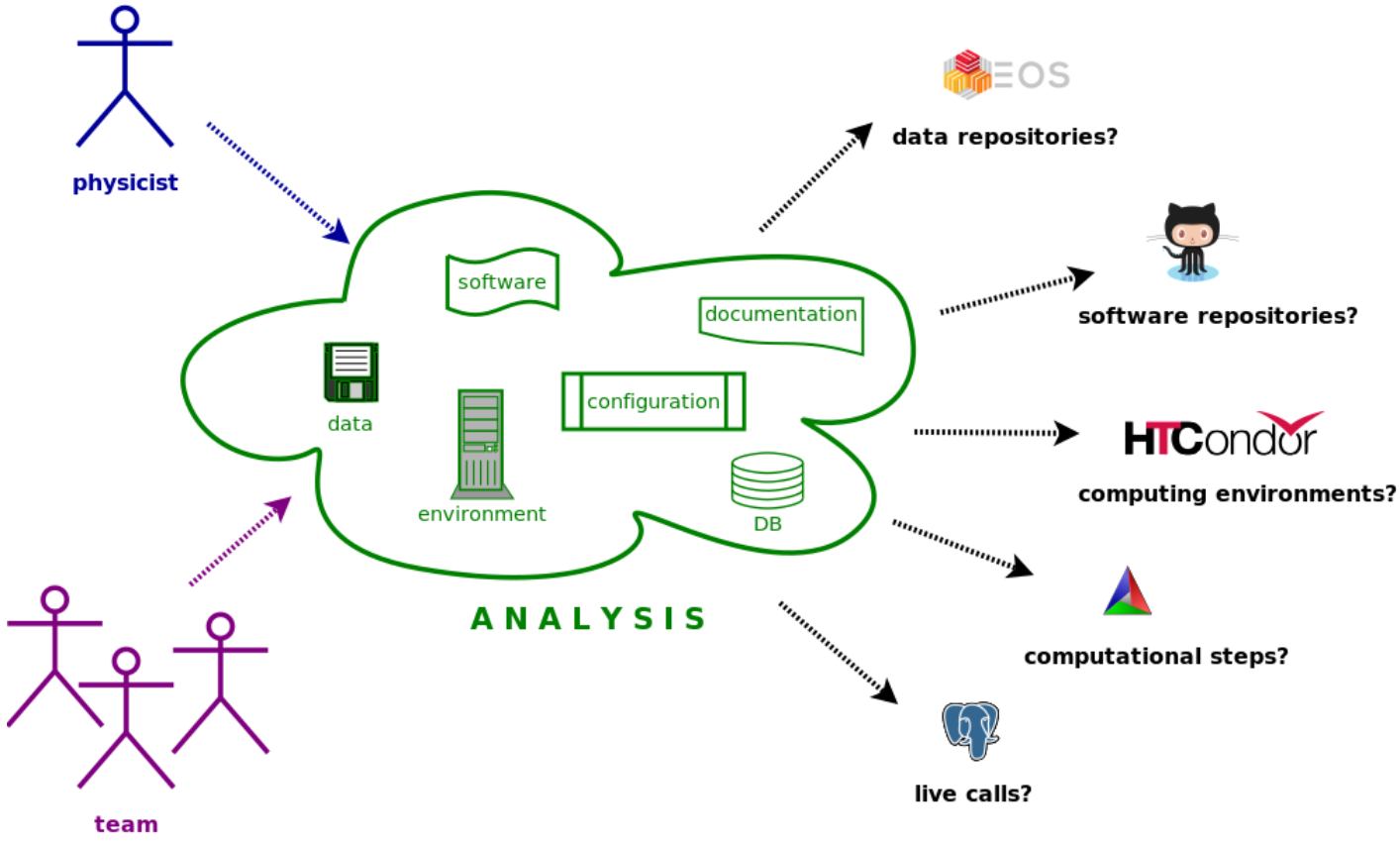
`https://www.nature.com/articles/sdata201618`

Ten rules for repr. comp. research

1. For every result, keep track of how it was produced
2. Avoid manual data manipulation steps
3. Archive the exact versions of all external programs used
4. Version control all custom scripts
5. Record all intermediate results, when possible in standardized formats
6. For analyses that include randomness, note underlying random seeds
7. Always store raw data behind plots
8. Generate hierarchical analysis output, allowing layers of increasing detail to be inspected
9. Connect textual statements to underlying results
10. Provide public access to scripts, runs, and results

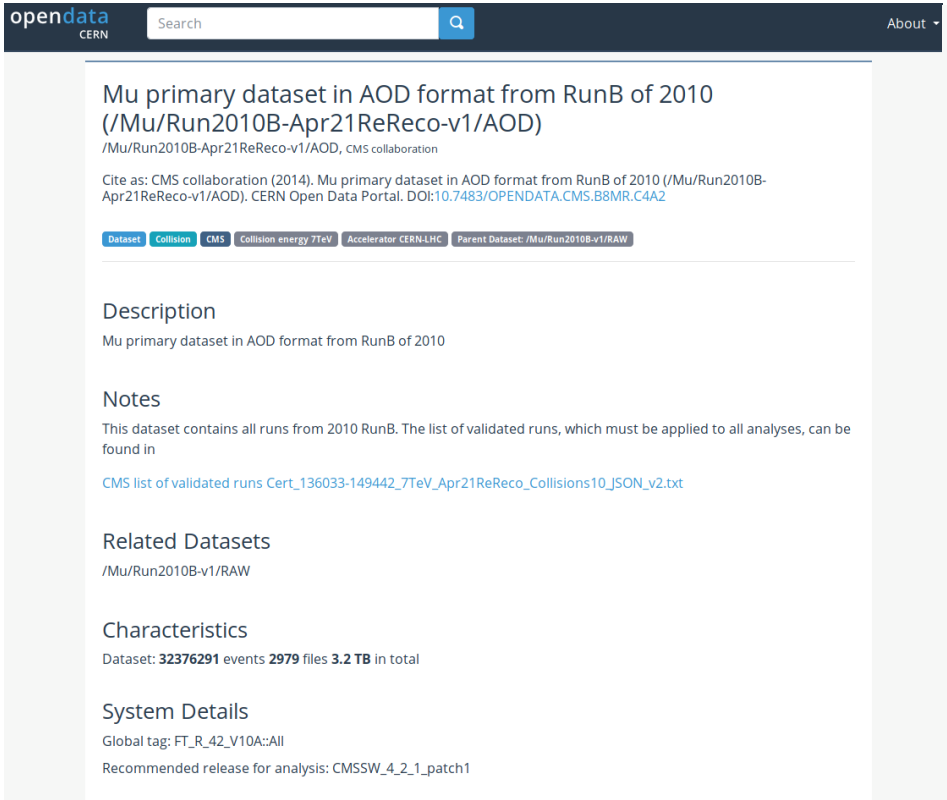
https://doi.org/10.1371/journal.pcbi.1003285

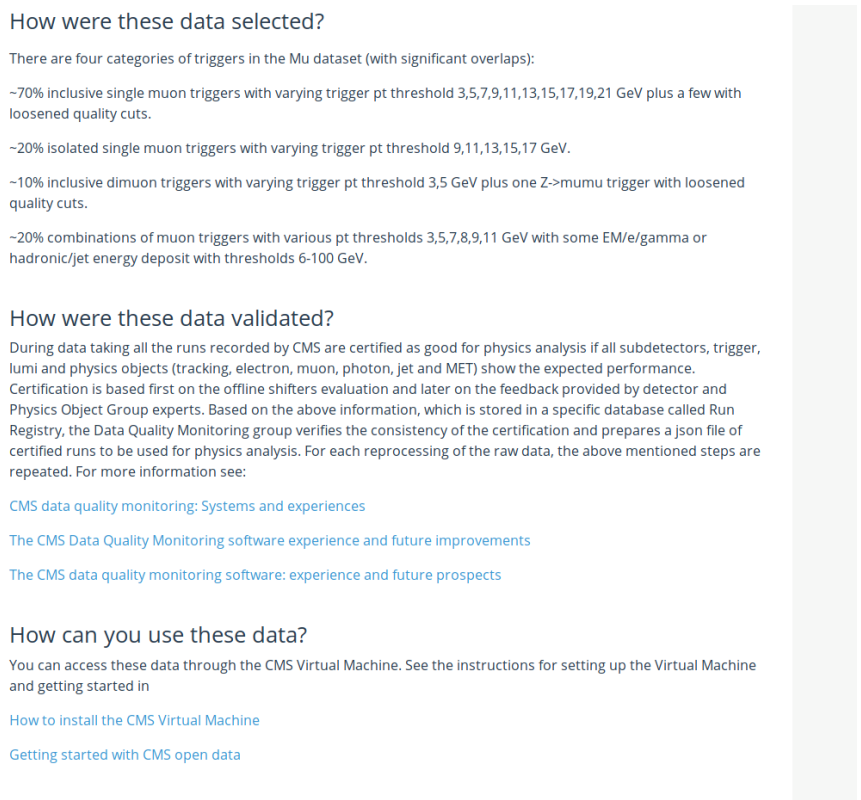
Preserving reusing data analyses



Capturing knowledge and assets of individual physics analyses to facilitate their future reuse

Preserving data 1/2





`http://opendata.cern.ch/record/14`

Context information about data selection, validation, use

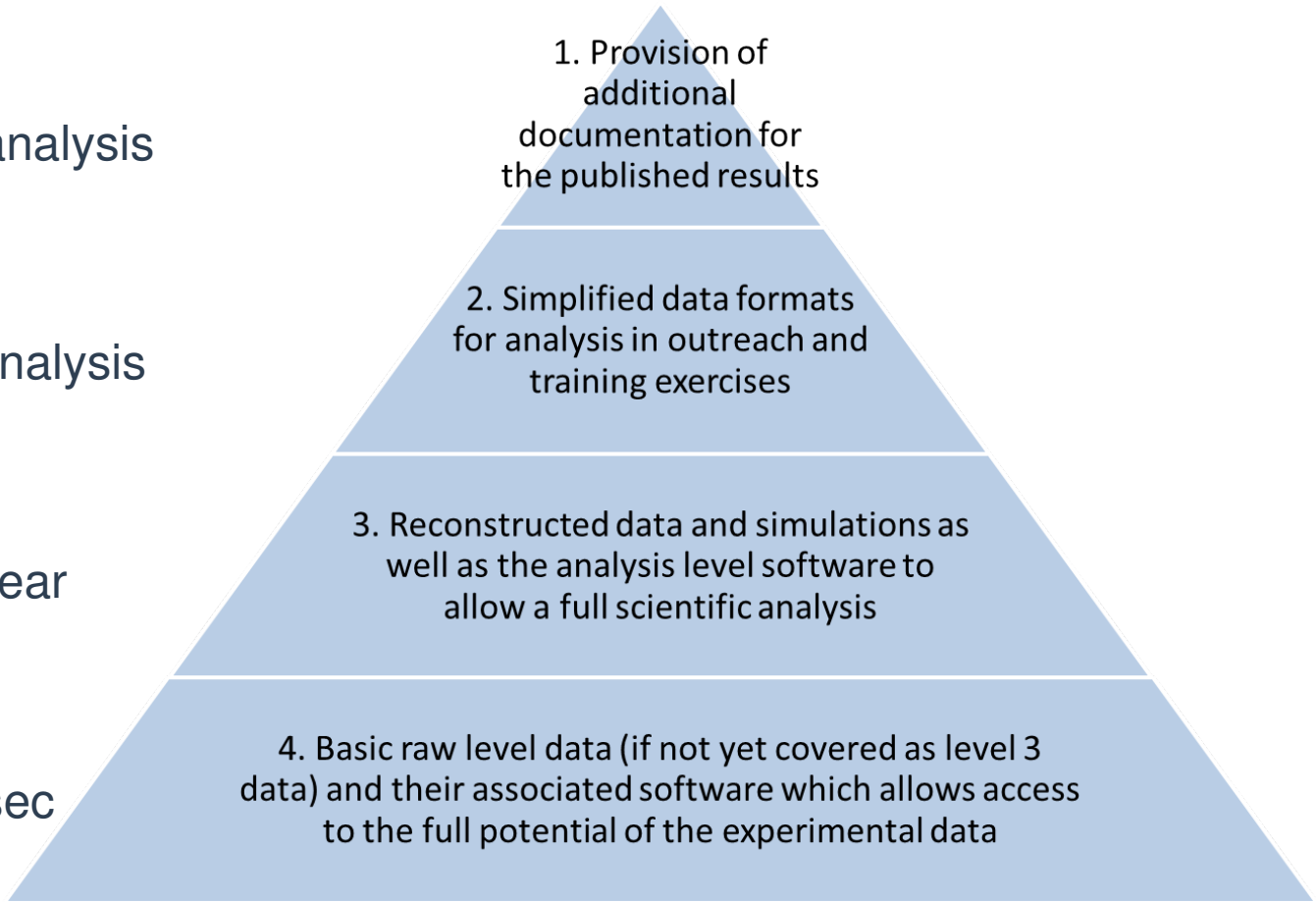
Preserving data 2/2

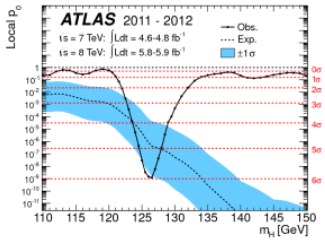
~GB/analysis

~TB/analysis

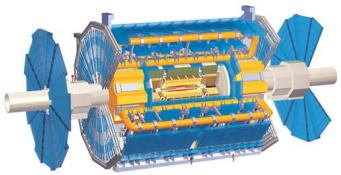
~PB/year

~GB/sec



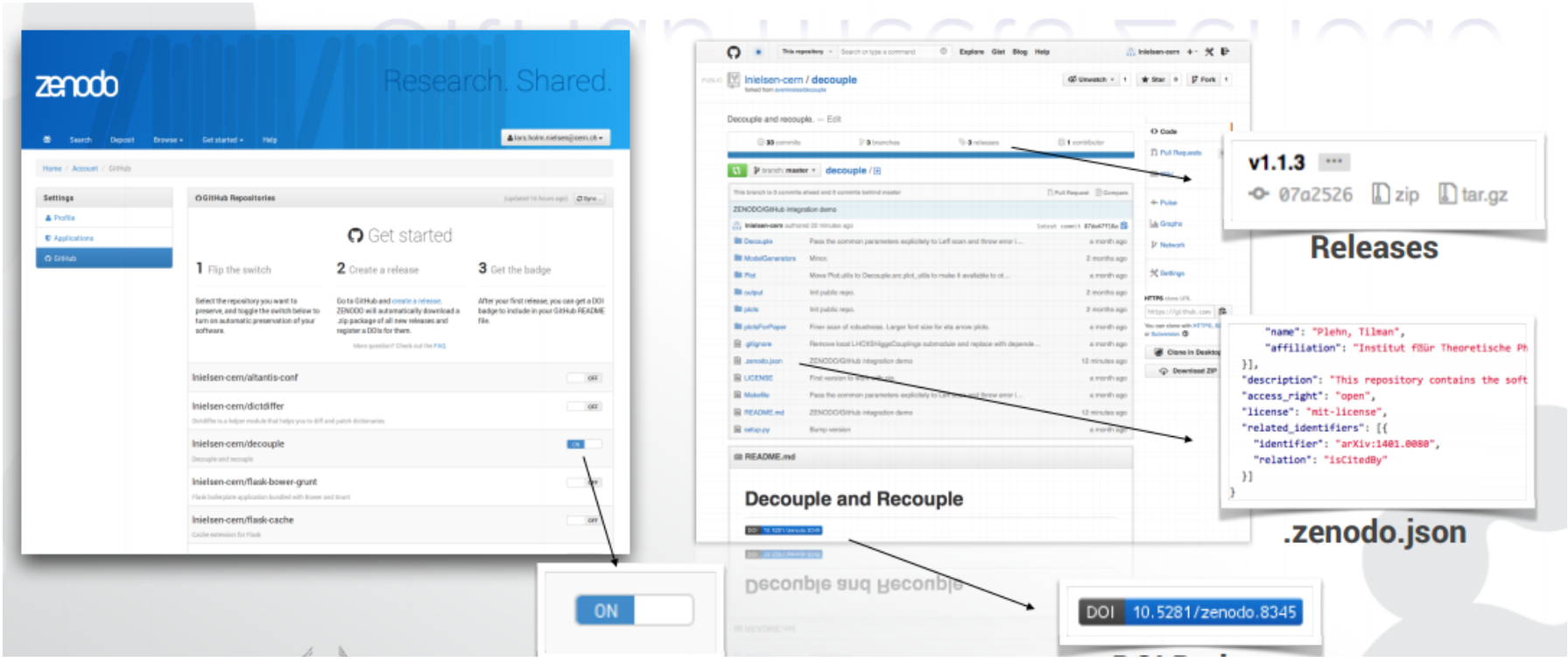


analysis



Slicing through LHC data pyramid

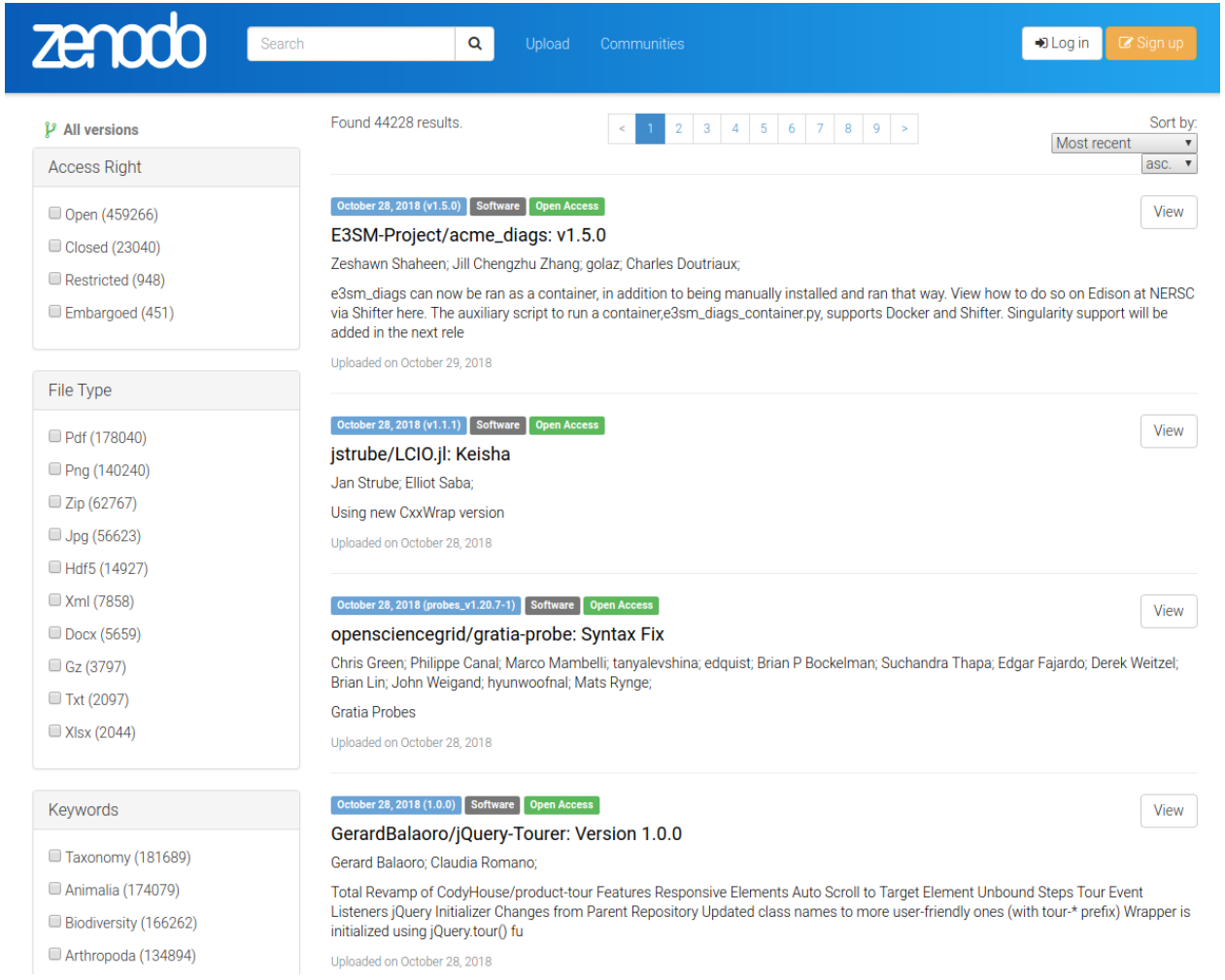
Preserving code 1/2



`https://guides.github.com/activities/citable-code`

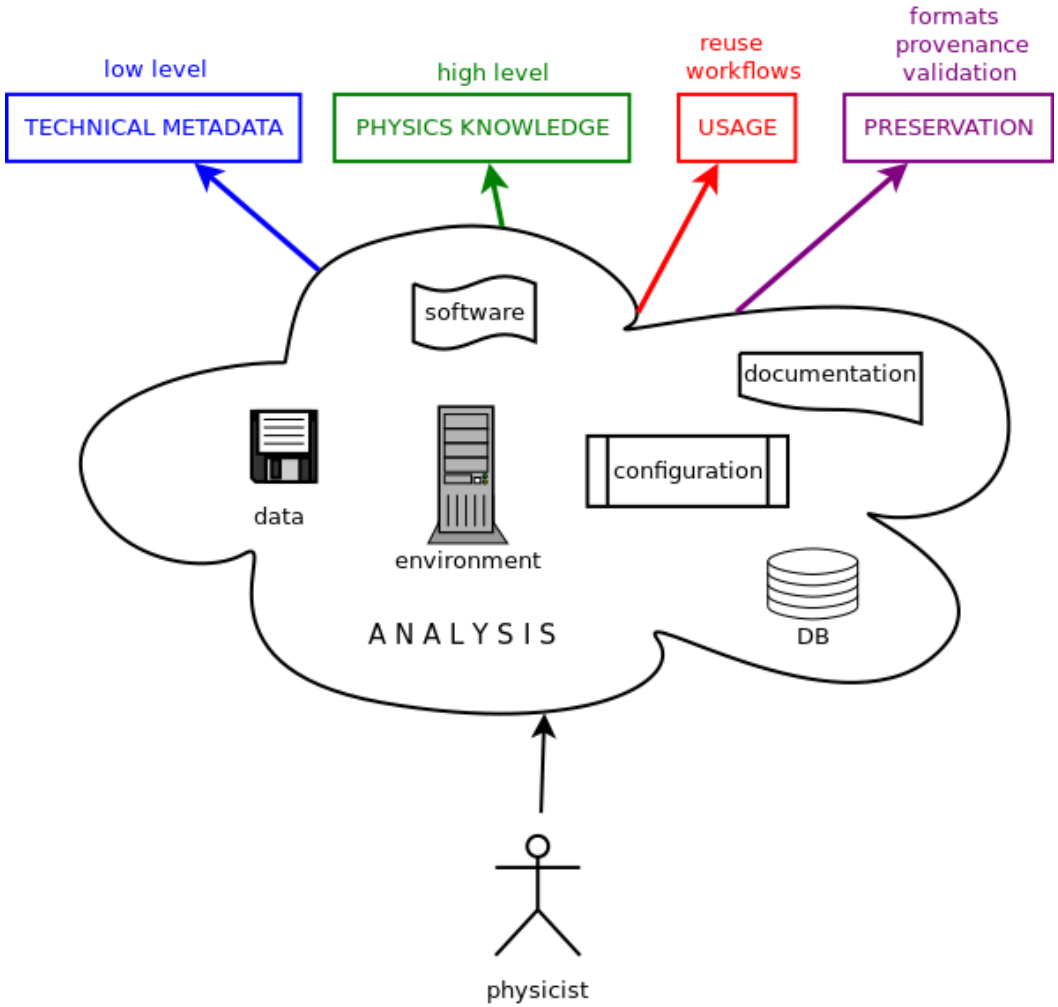
GitHub ↔ Zenodo bridge to automatically preserve releases

Preserving code 2/2



Over 58,000 DOIs for software on Zenodo

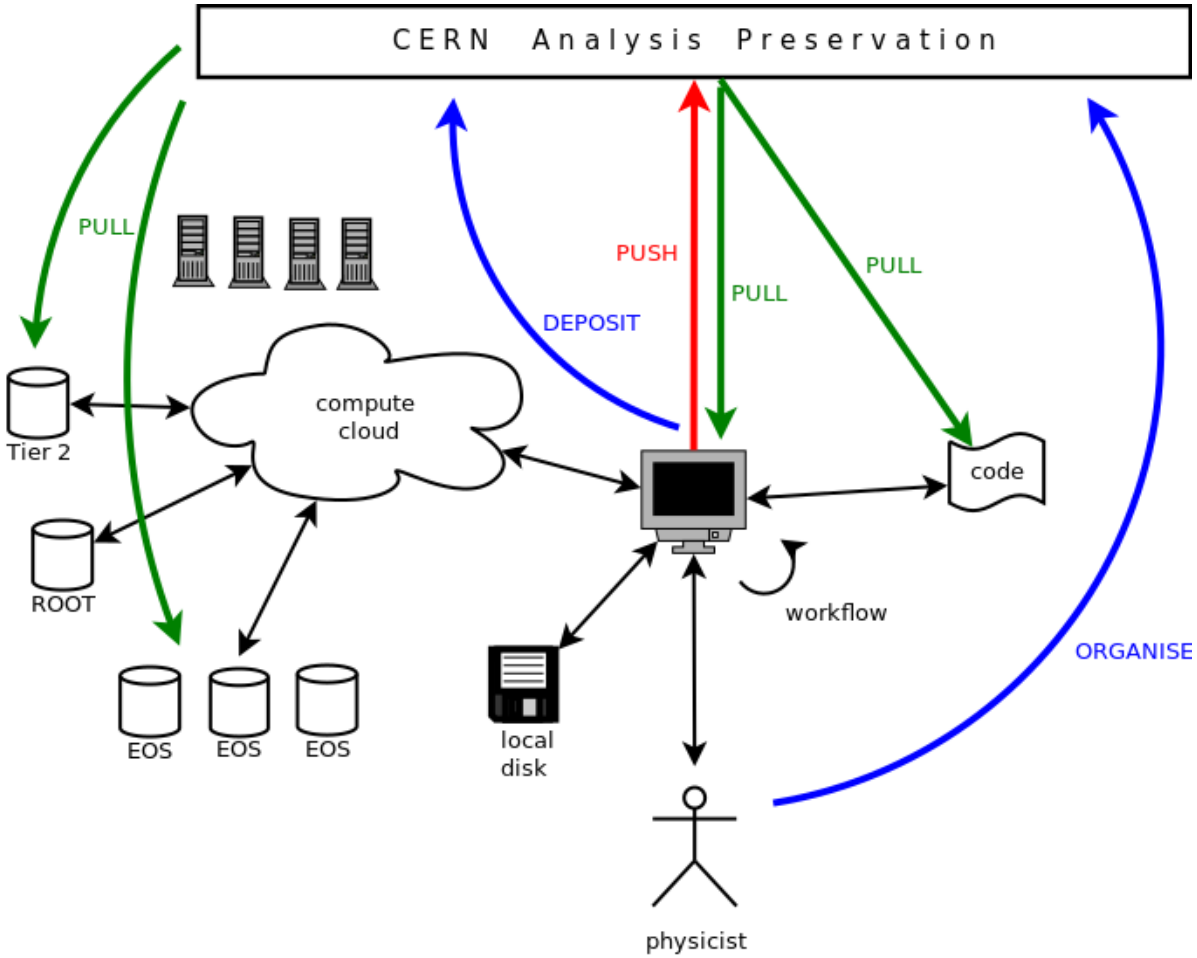
Preserving analyses 1/4



- JSON Schema
- W3C DCAT
- domain-specific fields

Structuring knowledge behind research data analysis

Preserving analyses 2/4

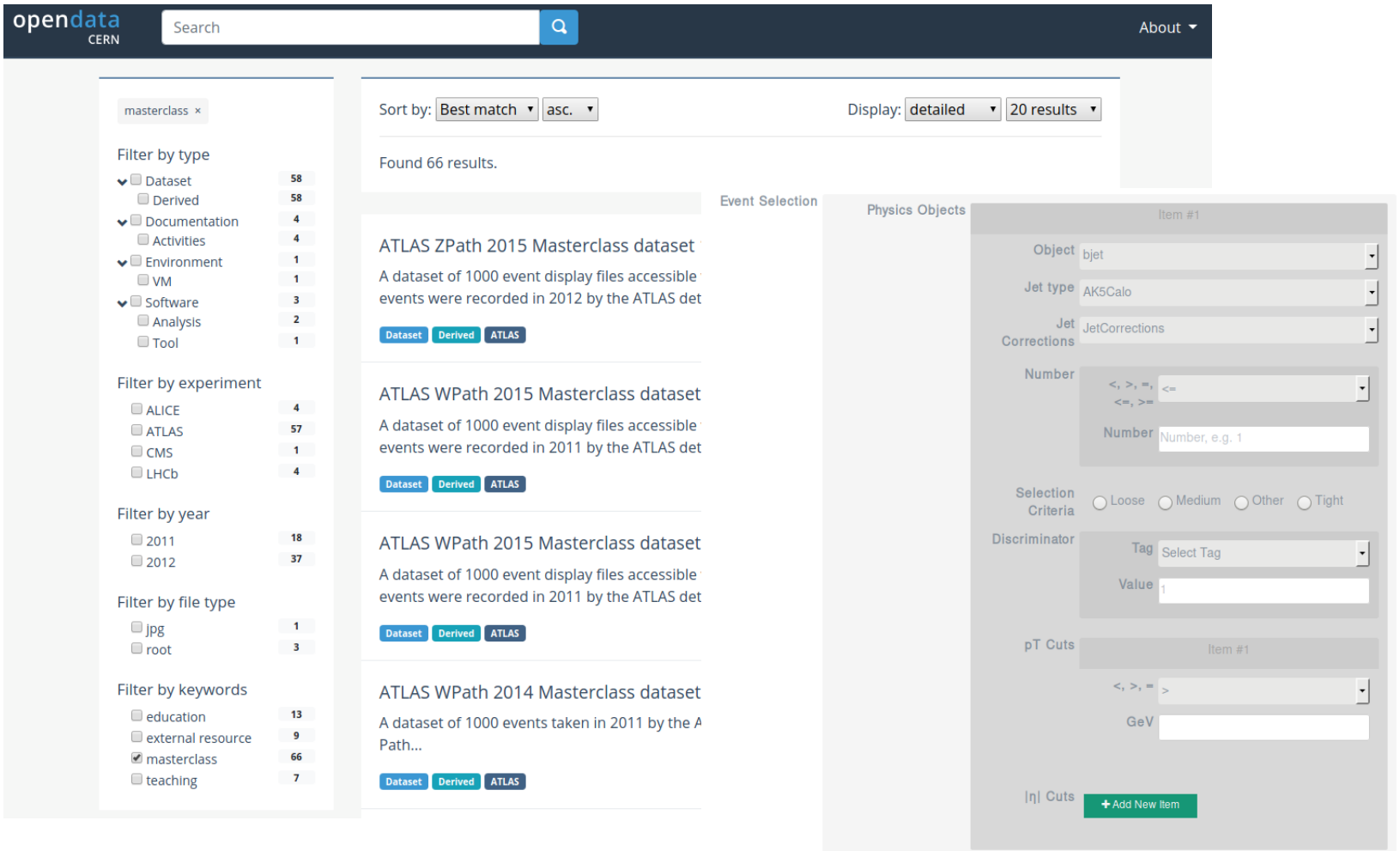


INVENIO

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

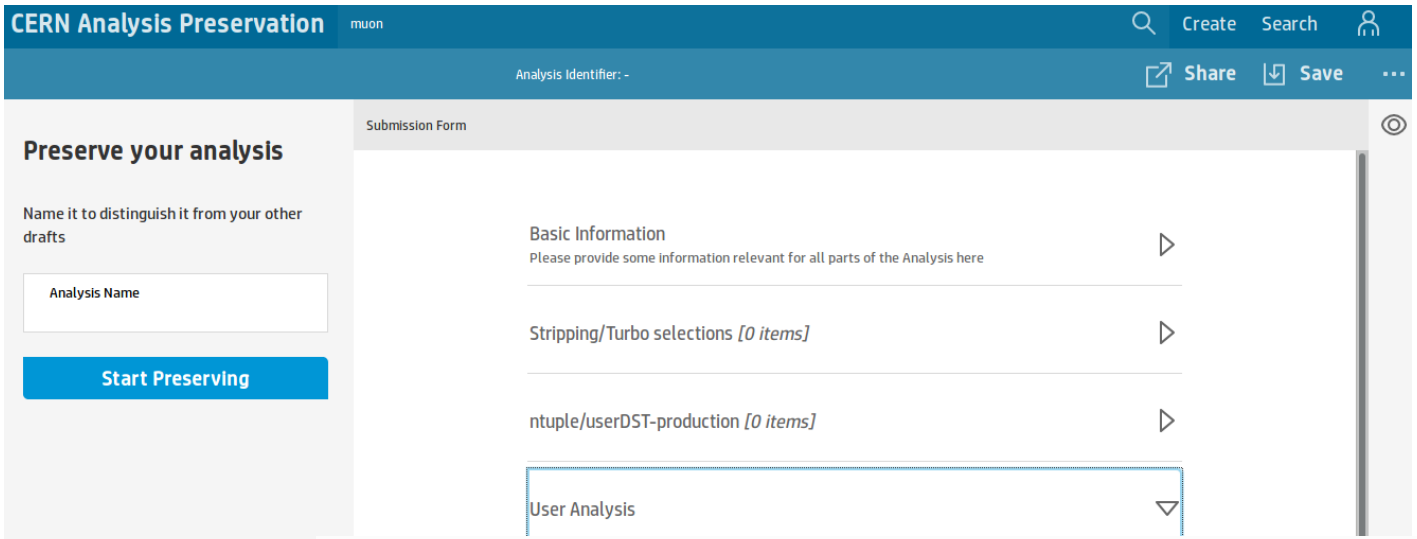
Taking consistent snapshot of analysis assets at a certain time

Preserving analyses 3/4



Information discovery through rich search syntax and facets

Preserving analyses 4/4




```
$ pip install cap-client
```

```
$ export CAP_SERVER_URL=https://analysispreservation.cern.ch/
$ export CAP_ACCESS_TOKEN=<your generated access token from server>
```

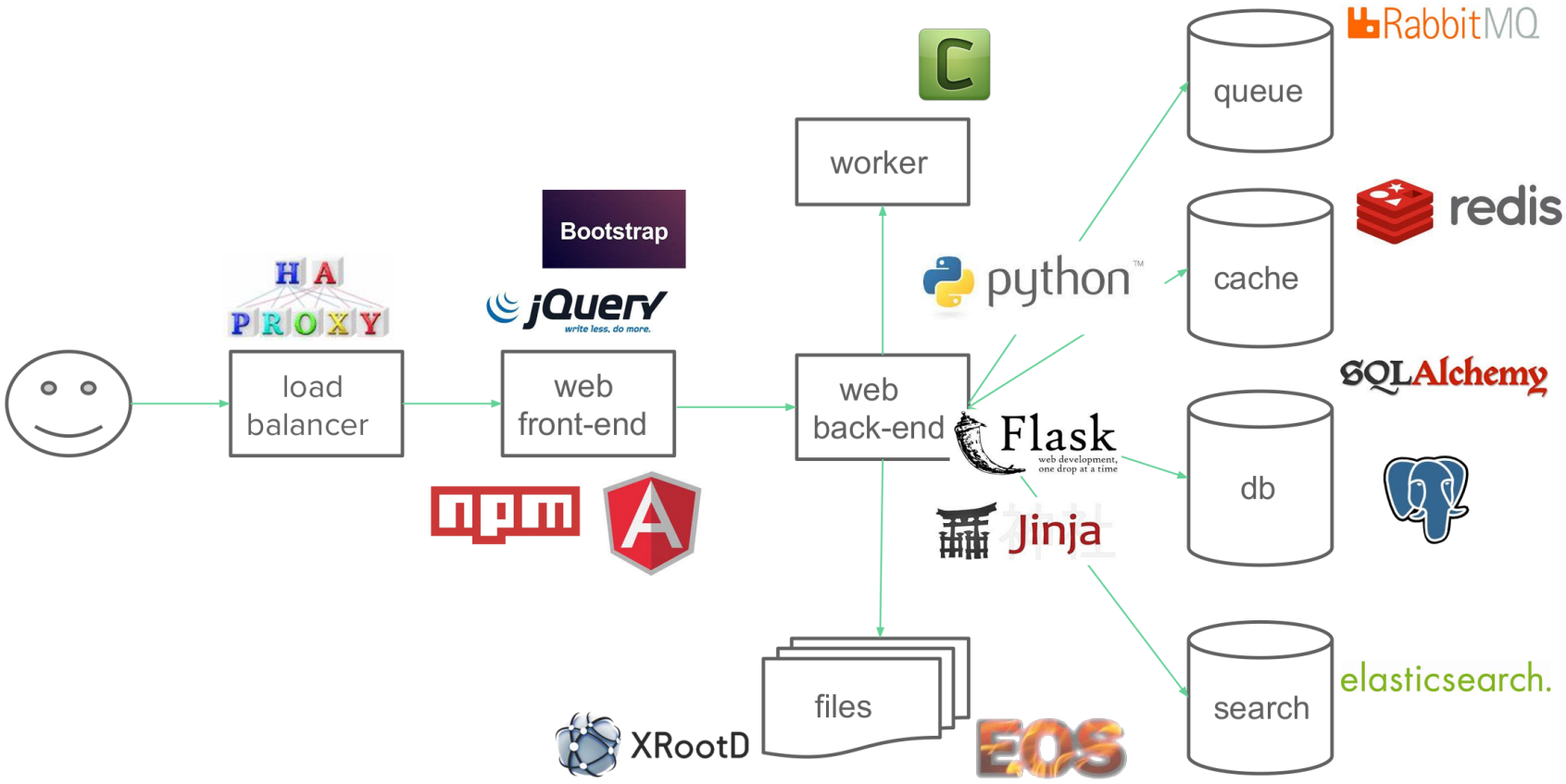
```
$ cap-client files upload <file path> --pid/-p <existing pid>

$ cap-client files upload file.json -p 89b593c498874ec8bcafc88944c458a7

File uploaded successfully.
```

Web based and command-line based deposit workflows

Digital repository technology



Technology stack using INVENIO digital repository

FAIR data principles

- **F**indable
  - persistent identifiers
  - rich metadata
  - indexed and searchable
- **A**ccessible
  - retrievable by identifiers
  - standard protocols
  - metadata vs data accessibility
- **I**nteroperable
  - knowledge representation language
  - common vocabularies
  - references to other metadata and data
- **R**eusable
  - domain-relevant attributes and community standards
  - clear licensing
  - provenance tracking

`https://www.nature.com/articles/sdata201618`

Reusing analyses

reana

Reproducible research data analysis platform

Flexible

Run many computational workflow engines.

Scalable

Support for remote compute clouds.

Reusable

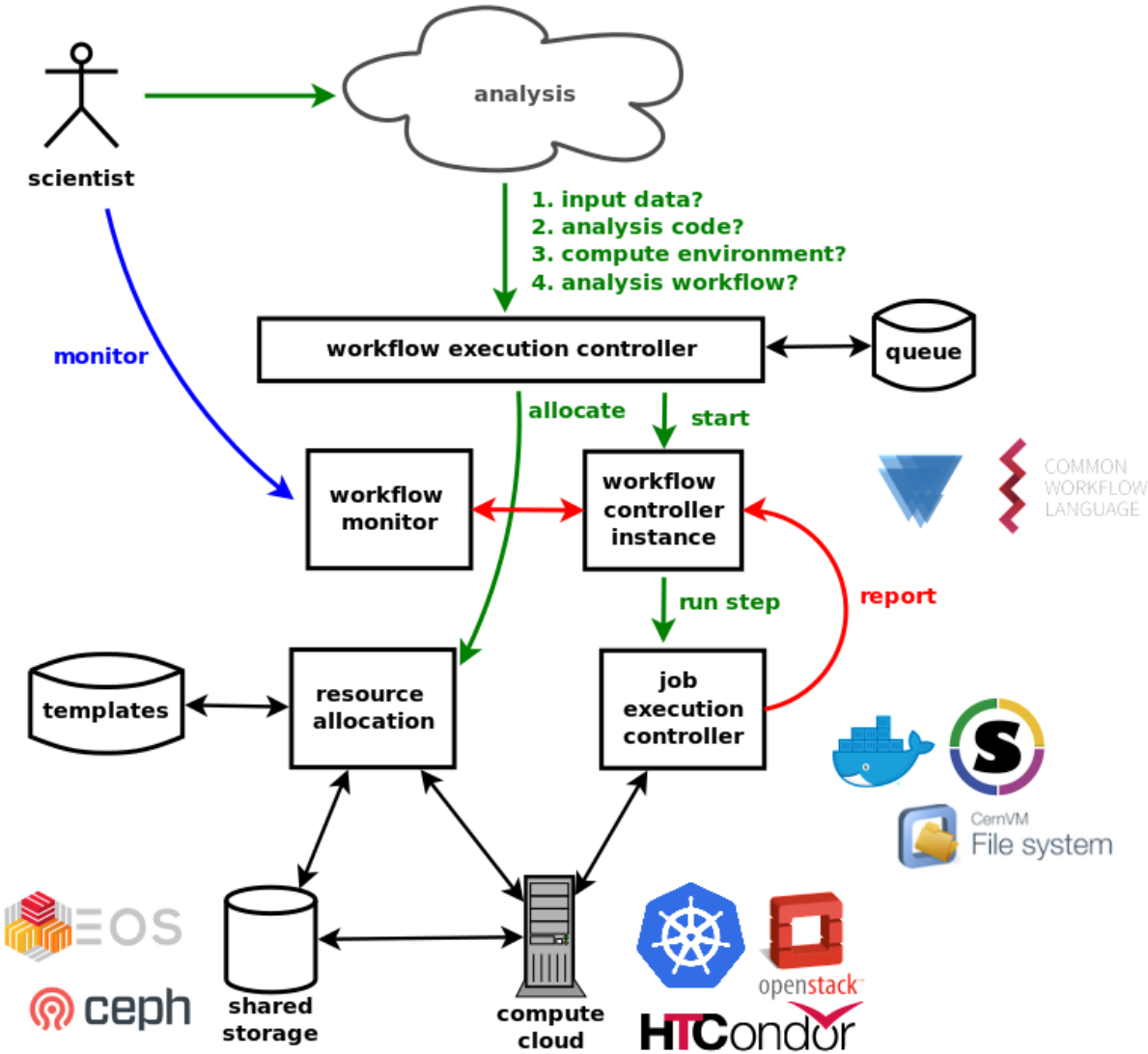
Containerise once, reuse elsewhere. Cloud-native.

Free

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

`http://www.reana.io/`

REANA architecture



REANA technology

- micro-services

python™ Flask web development, one drop at a time

- REST API

OPENAPI INITIATIVE Bravado

- services

RabbitMQ

- deployments

Four questions

1 Input data

What is your input data?

- input files
- input parameters

2 Analysis code

Which code analyses it?

- software frameworks
- user code

3 Compute environment

What is your environment?

- operating system
- database calls

4 Analysis workflow

Which steps did you take?

- single command
- complex workflows

Simple example

```
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

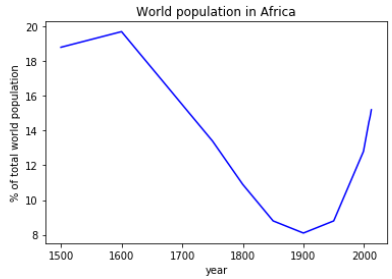
```
In [2]: # define input parameters
        input_file = "../data/World_historical_and_predicted_populations_in_percentage.csv"
        output_file = "../results/plot.png"
        region = 'Africa'
        year_min = 1500
        year_max = 2012

In [3]: # read input data file
        df = pd.read_csv(input_file)

In [4]: # add index
        df = df.set_index("Region", drop=False)

In [5]: # select region and years based on input parameters
        dfs = df.loc[region, str(year_min):str(year_max)]
        dft = pd.DataFrame({'year': dfs.index.astype(int), 'percentage': dfs.values}, columns=['year',
        'percentage'])

In [6]: # create output plot and save it to a file
        plot = plt.plot(dft['year'], dft['percentage'], color='blue')
        plt.title('World population in {0}'.format(region))
        plt.xlabel('year')
        plt.ylabel('% of total world population')
        plt.savefig(output_file)
```



2 **code**: Jupyter notebook

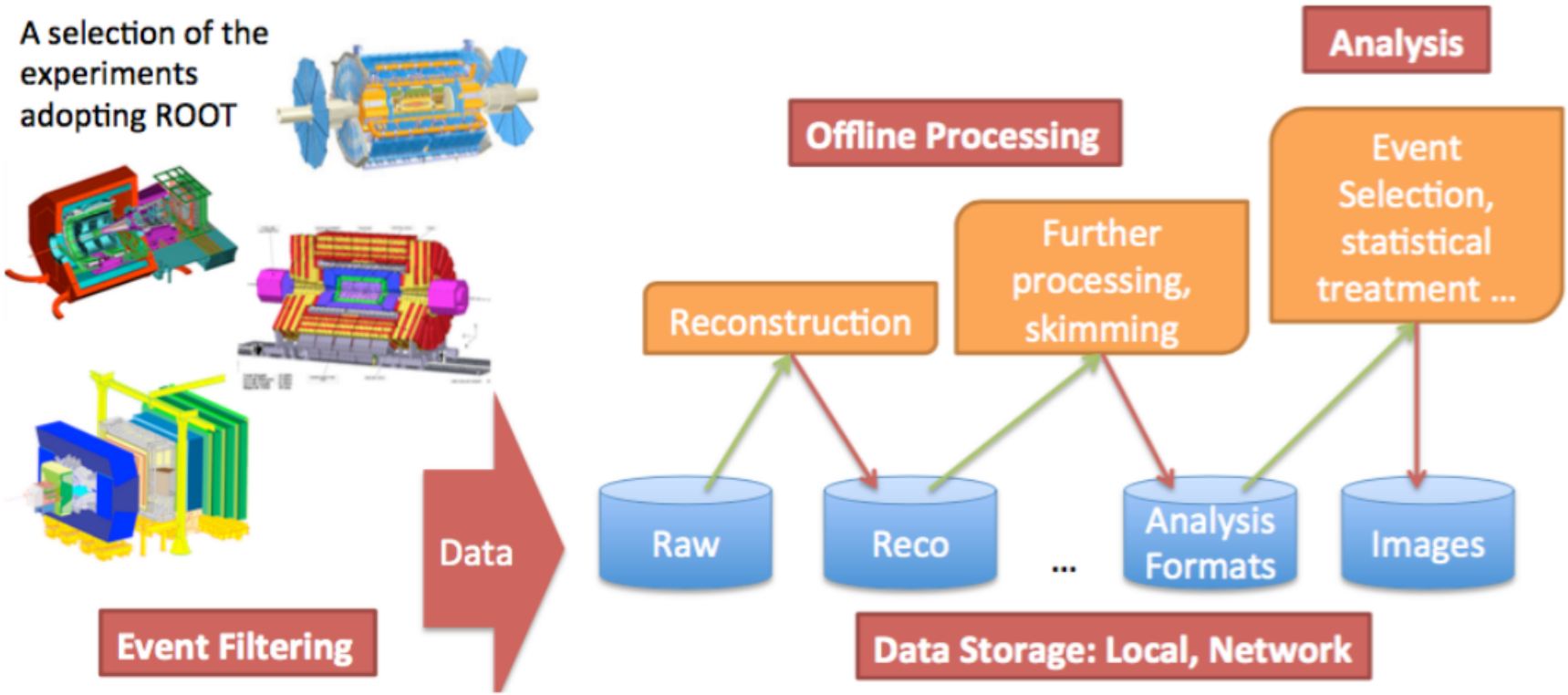
```
FROM centos:7
RUN yum install -y epel-release
RUN yum install -y \
    gcc \
    python-devel \
    python-pip
RUN pip install ipython==5.0.0 jupyter==1.0.0
ADD world_population_analysis.ipynb /code/
ADD World_historical_and_predicted_populations_in_percentage.csv /code/
WORKDIR /code
CMD ["jupyter",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

3 **environment**: CentOS7, IP5

4 **workflow**: `papermill ...`

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

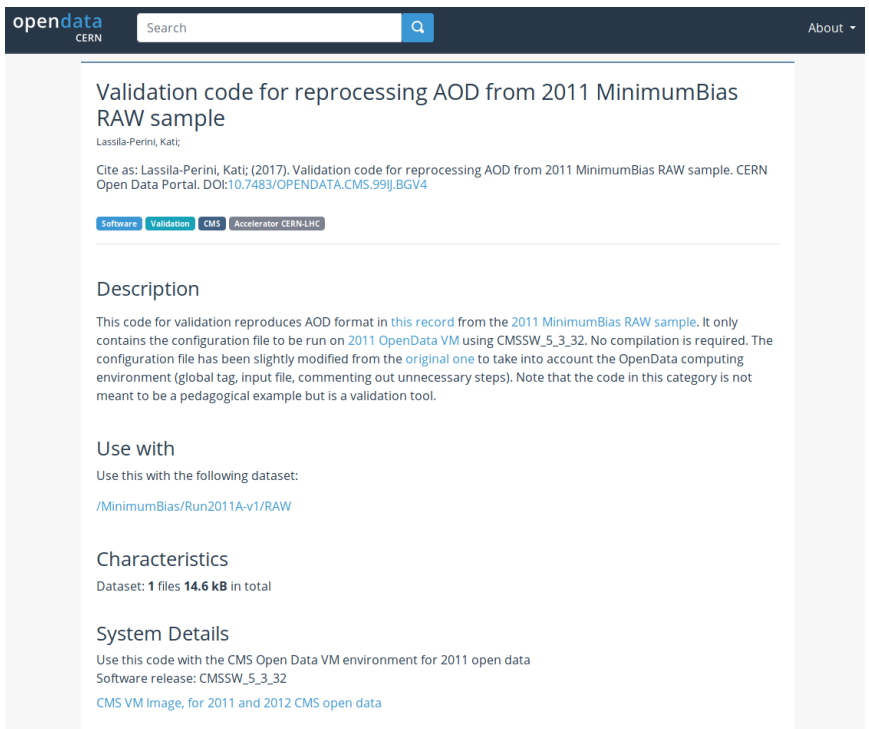
HEP data analyses

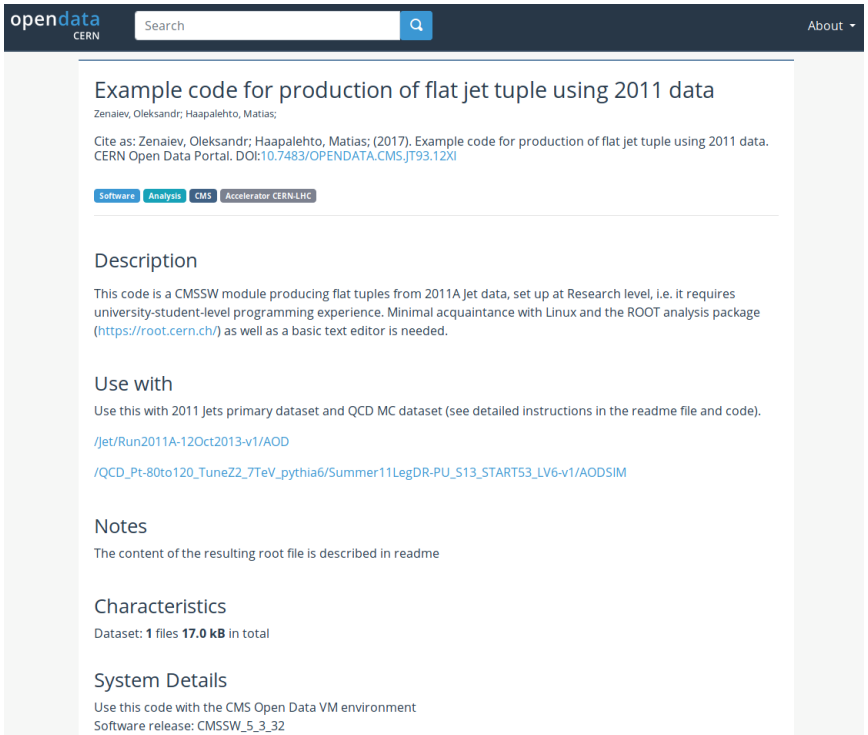


D. Krücker *et al* `https://indico.desy.de/indico/event/18343`

Targeting both data production and data analysis stages

Data production



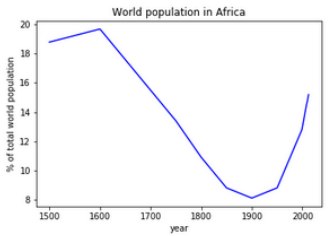


Reconstruction and flat jet tuple production from CMS Open Data

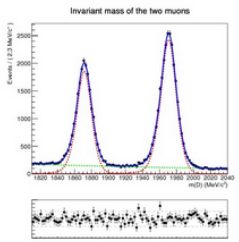
Data analyses

```
$ cat inputs/names.txt
  Jane Doe
  Joe Bloggs
$ reana-client start
$ reana-client download
$ cat outputs/greetings.txt
  Hello Jane Doe!
  Hello Joe Bloggs!
```

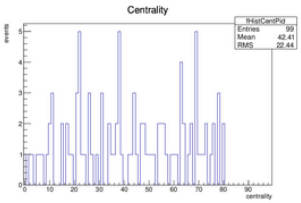
"Hello world"



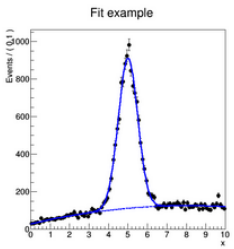
Parametrised Jupyter notebook



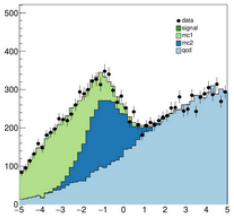
LHCb rare charm decay search



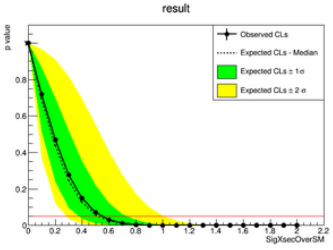
ALICE LEGO train test run



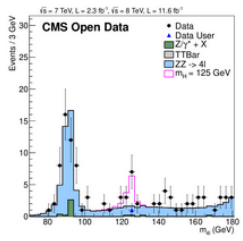
ROOT/RooFit physics analysis



ATLAS BSM search



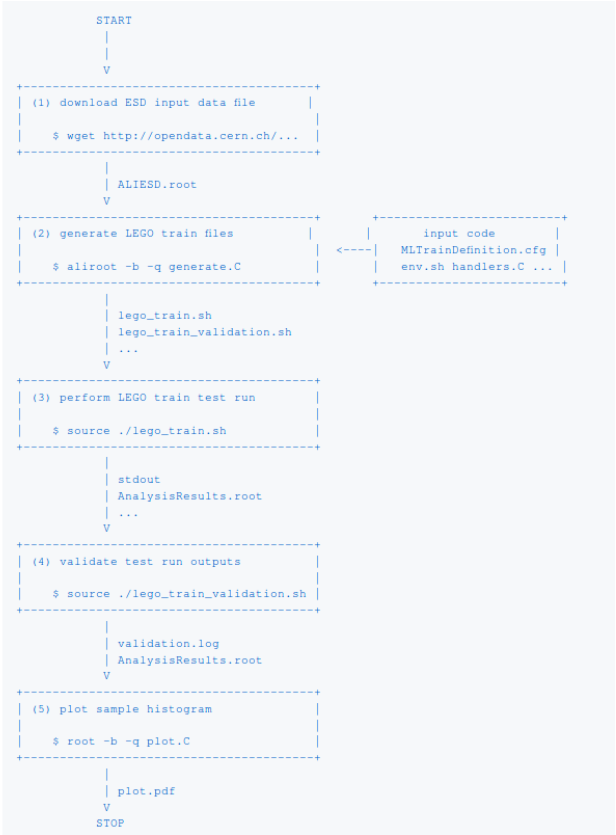
ATLAS RECAST

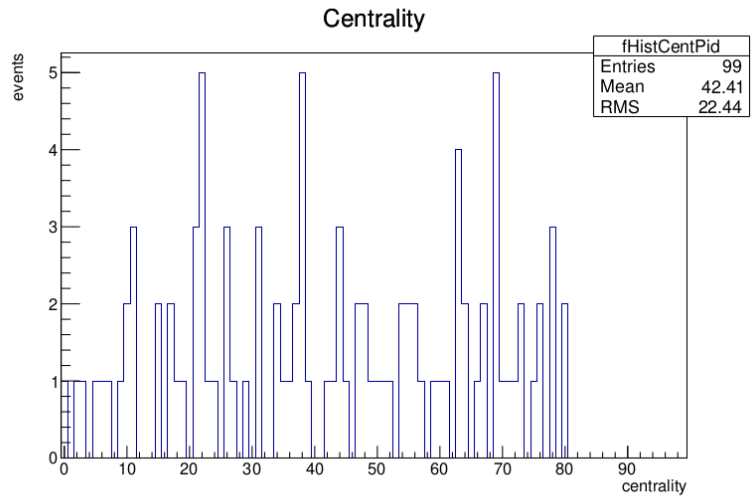


CMS Higgs-to-four-leptons

Several physics analysis examples

Example: ALICE analysis train

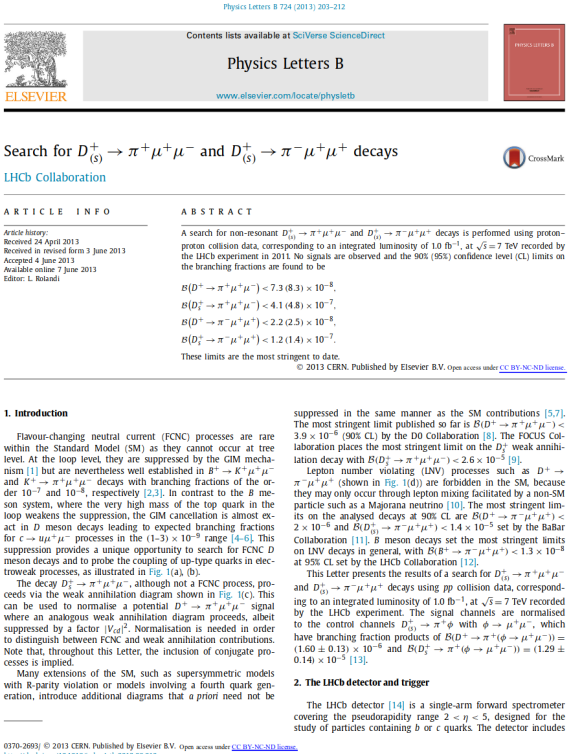




https://github.com/reanahub/reana-demo-alice-lego-train-test-run/

Using ALICE production software for train test run and validation

Example: LHCb rare charm decay



Physics Letters B 724 (2013) 203–212

Contents lists available at SciVerse ScienceDirect

Physics Letters B

www.elsevier.com/locate/physletb

Search for $D^+_{(s)} \to \pi^+\mu^+\mu^-$ and $D^+_{(s)} \to \pi^-\mu^+\mu^+$ decays

LHCb Collaboration

ARTICLE INFO

Article history:
Received 24 April 2013
Received in revised form 3 June 2013
Accepted 4 June 2013
Available online 7 June 2013
Editor: L. Rolandi

ABSTRACT

A search for non-resonant $D^+_{(s)} \to \pi^+\mu^+\mu^-$ and $D^+_{(s)} \to \pi^-\mu^+\mu^+$ decays is performed using proton–proton collision data, corresponding to an integrated luminosity of 1.0 fb^{-1}, at $\sqrt{s} = 7$ TeV recorded by the LHCb experiment in 2011. No signals are observed and the 90% (95%) confidence level (CL) limits on the branching fractions are found to be

$\mathcal{B}(D^+ \to \pi^+\mu^+\mu^-) < 7.3\,(8.3) \times 10^{-8}$,
$\mathcal{B}(D_s^+ \to \pi^+\mu^+\mu^-) < 4.1\,(4.8) \times 10^{-7}$,
$\mathcal{B}(D^+ \to \pi^-\mu^+\mu^+) < 2.2\,(2.5) \times 10^{-8}$,
$\mathcal{B}(D_s^+ \to \pi^-\mu^+\mu^+) < 1.2\,(1.4) \times 10^{-7}$.

These limits are the most stringent to date.

1. Introduction

Flavour-changing neutral current (FCNC) processes are rare within the Standard Model (SM) as they cannot occur at tree level. At the loop level, they are suppressed by the GIM mechanism [1] but are nevertheless well established in $B^+ \to K^+\mu^+\mu^-$ and $K^+ \to \pi^+\mu^+\mu^-$ decays with branching fractions of the order 10^{-7} and 10^{-8}, respectively [2,3]. In contrast to the B meson system, where the very high mass of the top quark in the loop weakens the suppression, the GIM cancellation is almost exact in D meson decays leading to expected branching fractions for $c \to u\mu^+\mu^-$ processes in the (1–3) $\times\, 10^{-9}$ range [4–6]. This suppression provides a unique opportunity to search for FCNC D meson decays and to probe the coupling of up-type quarks in electroweak processes, as illustrated in Fig. 1(a), (b).

The decay $D^+_{(s)} \to \pi^+\mu^+\mu^-$, although not a FCNC process, proceeds via the weak annihilation diagram shown in Fig. 1(c). This can be used to normalise a potential $D^+ \to \pi^+\mu^+\mu^-$ signal where an analogous weak annihilation diagram proceeds, albeit suppressed by a factor $|V_{cd}|^2$. Normalisation is needed in order to distinguish between FCNC and weak annihilation contributions. Note that, throughout this Letter, the inclusion of conjugate processes is implied.

Many extensions of the SM, such as supersymmetric models with R-parity violation or models involving a fourth quark generation, introduce additional diagrams that *a priori* need not be suppressed in the same manner as the SM contributions [5,7]. The most stringent limit published so far is $\mathcal{B}(D^+ \to \pi^+\mu^+\mu^-) < 3.9 \times 10^{-6}$ (90% CL) by the D0 Collaboration [8]. The FOCUS Collaboration places the most stringent limit on the D_s^+ weak annihilation decay with $\mathcal{B}(D_s^+ \to \pi^+\mu^+\mu^-) < 2.6 \times 10^{-5}$ [9].

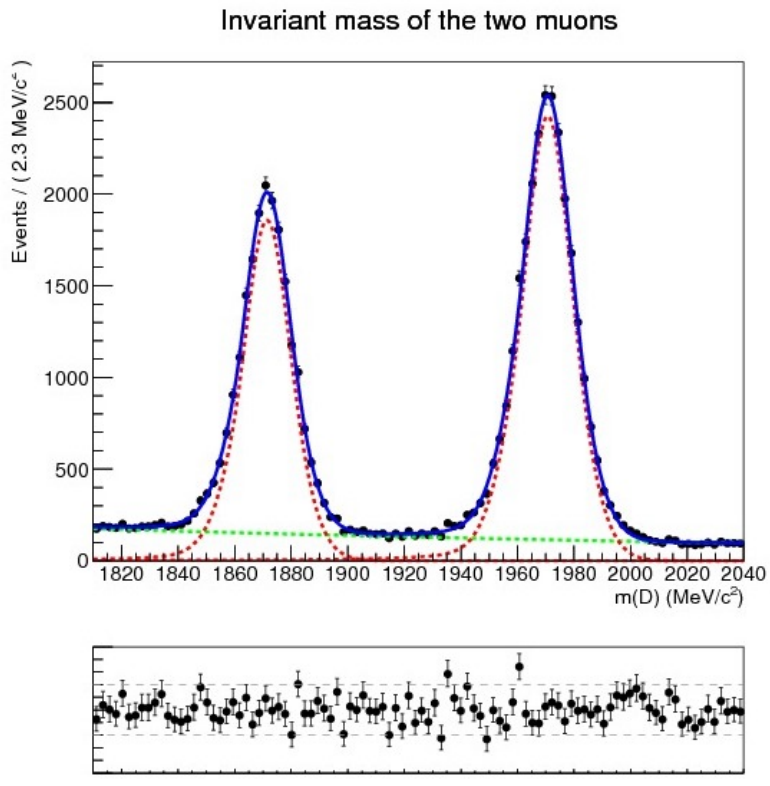
Lepton number violating (LNV) processes such as $D^+ \to \pi^-\mu^+\mu^+$ (shown in Fig. 1(d)) are forbidden in the SM, because they may only occur through lepton mixing facilitated by a non-SM particle such as a Majorana neutrino [10]. The most stringent limits on the analysed decays at 90% CL are $\mathcal{B}(D^+ \to \pi^-\mu^+\mu^+) < 2 \times 10^{-6}$ and $\mathcal{B}(D_s^+ \to \pi^-\mu^+\mu^+) < 1.4 \times 10^{-5}$ set by the BaBar Collaboration [11]. B meson decays set the most stringent limits on LNV decays in general, with $\mathcal{B}(B^+ \to \pi^-\mu^+\mu^+) < 1.3 \times 10^{-8}$ at 95% CL set by the LHCb Collaboration [12].

This Letter presents the results of a search for $D^+_{(s)} \to \pi^+\mu^+\mu^-$ and $D^+_{(s)} \to \pi^-\mu^+\mu^+$ decays using pp collision data, corresponding to an integrated luminosity of 1.0 fb^{-1}, at $\sqrt{s} = 7$ TeV recorded by the LHCb experiment. The signal channels are normalised to the control channels $D^+_{(s)} \to \pi^+\phi$ with $\phi \to \mu^+\mu^-$, which have branching fraction products of $\mathcal{B}(D^+ \to \pi^+(\phi \to \mu^+\mu^-)) = (1.60 \pm 0.13) \times 10^{-6}$ and $\mathcal{B}(D_s^+ \to \pi^+(\phi \to \mu^+\mu^-)) = (1.29 \pm 0.14) \times 10^{-5}$ [13].

2. The LHCb detector and trigger

The LHCb detector [14] is a single-arm forward spectrometer covering the pseudorapidity range $2 < \eta < 5$, designed for the study of particles containing b or c quarks. The detector includes

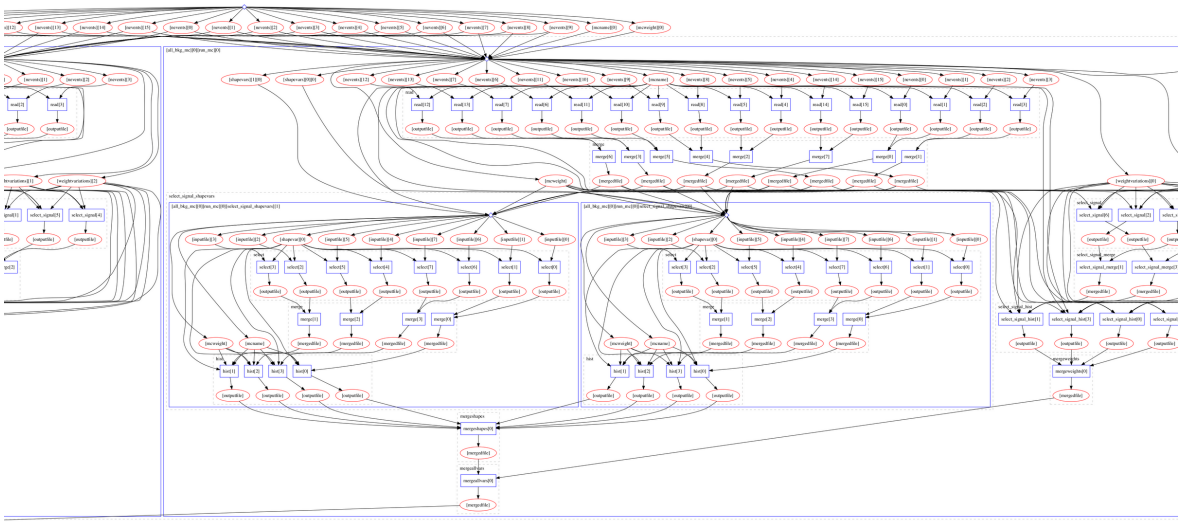
0370-2693/ © 2013 CERN. Published by Elsevier B.V. Open access under CC BY-NC-ND license.
http://dx.doi.org/10.1016/j.physletb.2013.06.010

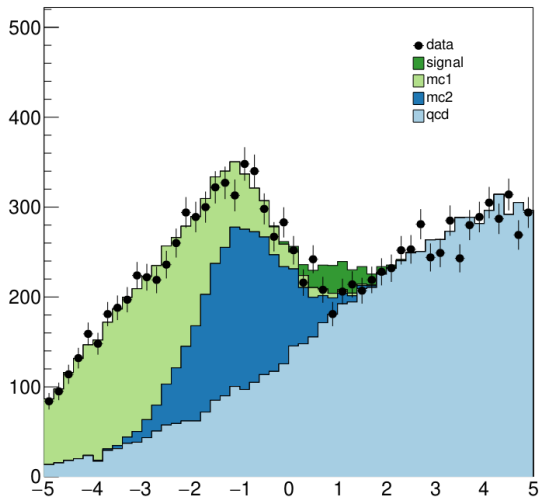


https://github.com/reanahub/reana-demo-lhcb-d2pimumu/

Search for $D^+_{(s)} \to \pi^+\mu^+\mu^-$ and $D^+_{(s)} \to \pi^-\mu^+\mu^+$ decays

Example: BSM search

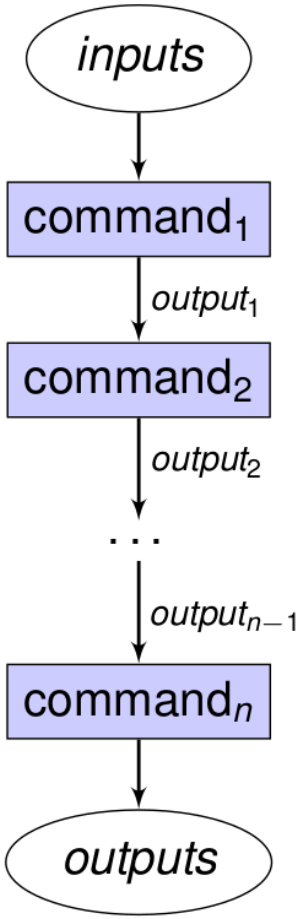


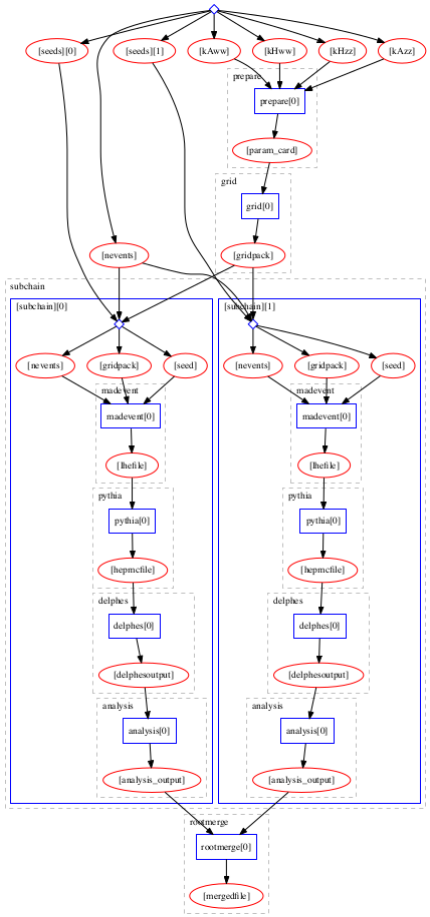


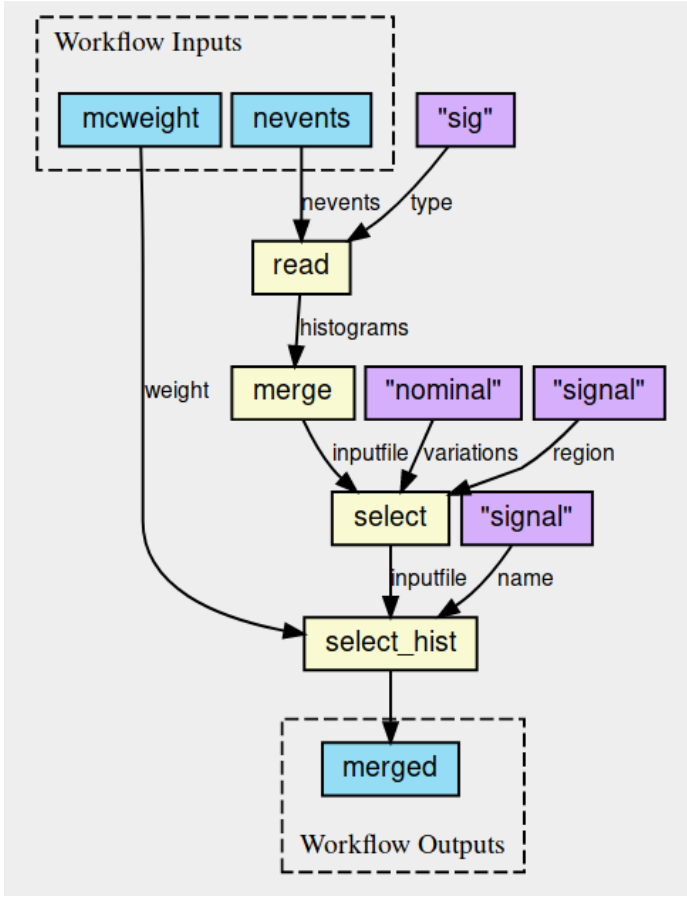
https://github.com/reanahub/reana-demo-bsm-search/

Complex computational workflows typical in particle physics analyses

Computational workflows







Serial Yadage CWL

Running workflows

1

Structure your analysis

```
version: 0.2.0
code:
   files:
   - code/mycode.py
inputs:
   files:
   - inputs/mydata.csv
   parameters:
      myparameter: myvalue
environments:
   - type: docker
      image: johndoe/mypython:1.0
workflow:
   type: cwl
   file: workflow/myworkflow.cwl
outputs:
   files:
   -outputs/myplot.png
```

more

2

Select a REANA cluster...

```
$ export
REANA_SERVER_URL=https://reana.cern.ch/
```

...or install your own

```
# install kubectl 1.9.1 and minikube
0.23.0
$ sudo dpkg -i kubectl*.deb minikube*.deb
$ minikube start --kubernetes-
version="v1.6.4"
# install reana-cluster utility
$ mkvirtualenv reana-cluster
$ pip install reana-cluster
# deploy new cluster and check progress
$ reana-cluster init
$ reana-cluster status
# set environment variables for reana-
client
$ eval $(reana-cluster env)
```

more

3

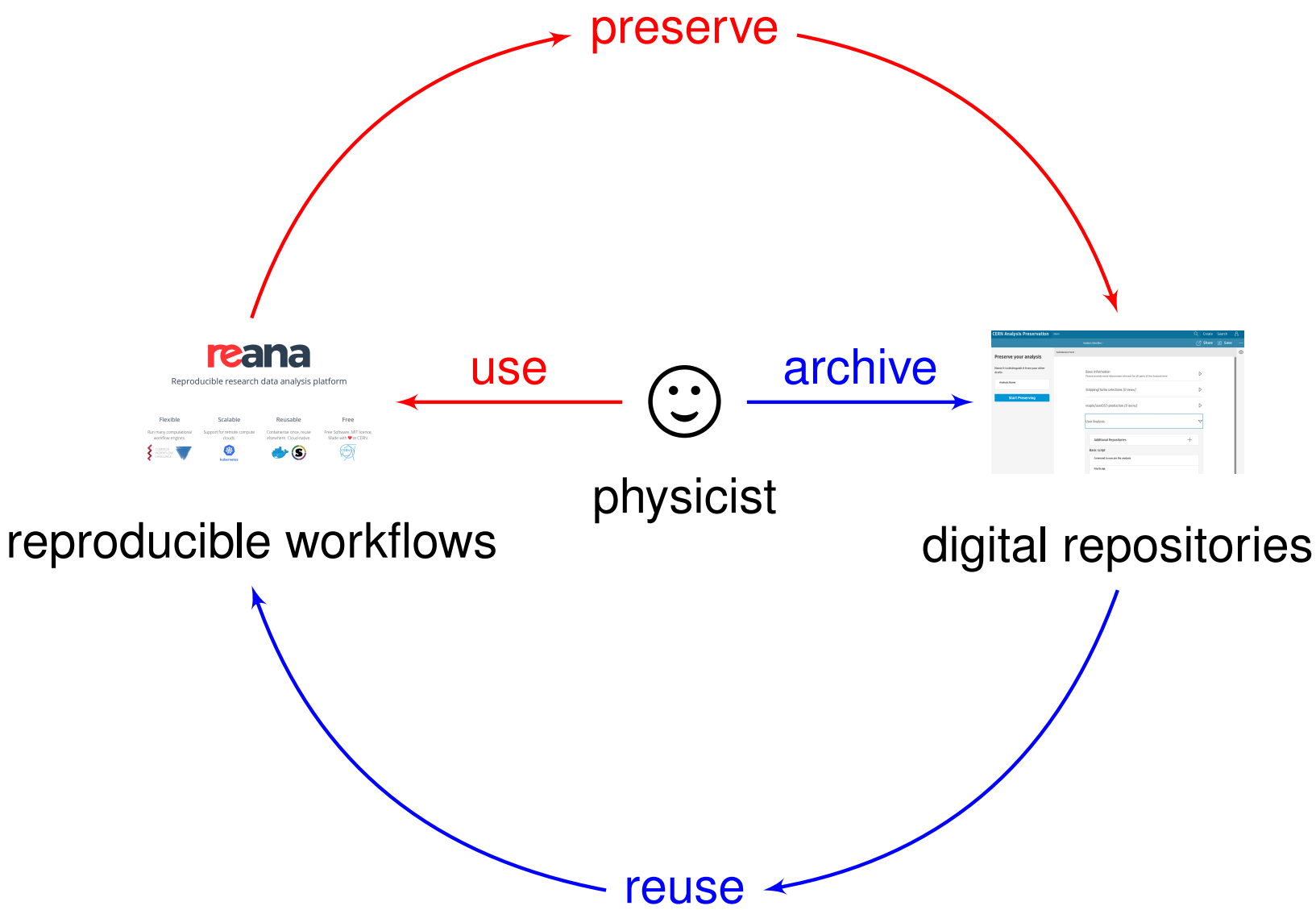
Run your analysis

```
# install reana-client
$ mkvirtualenv reana-client -p /usr/bin
/python2.7
$ pip install reana-client
$ reana-client ping
# create new workflow
$ export REANA_WORKON=$(reana-client
workflow create)
# upload runtime code and inputs
$ reana-client code upload ./code/*
$ reana-client inputs upload ./inputs/*
# start workflow and check progress
$ reana-client workflow start
$ reana-client workflow status
# download outputs
$ reana-client outputs list
$ reana-client outputs download
myplot.png
```

more

Rich command-line user client

Reproducibility ⇌ Preservation

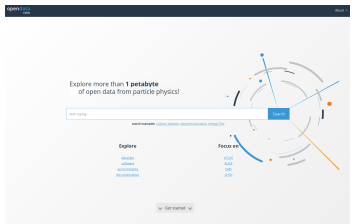


Conclusions

- reusable data, reproducible analyses
  - structure knowledge → JSON schema
  - capture assets → push and pull
  - actionable processes → “workflow is the new data”
- leverage on synergies
  - diverse scientific domains → DAG
  - industry standards → containers, “distributed Linux”
- overcoming challenges
  - technological → scaling out, O(10GB) containers, O(10k) steps
  - sociological → culture change, smooth integration into daily work

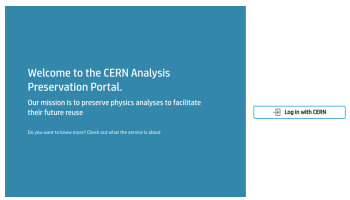
CERN IT D. Kousidis, R. Maciulaitis, J. Okraska, D. Rodriguez, T. Šimko · **CERN SIS** S. Dallmeier-Tiessen, S. Feger, P. Fokianos, A. Lavasa, S. van de Sandt, I. Tsanaktsidis, A. Trzcinska · **ALICE** Y. Foka, M. Gheata, C. Grigoras, M. Zimmermann · **ATLAS** K. Cranmer, L. Heinrich, A. Sanchez Pineda, D. Rousseau, F. Socher · **CMS** H. Bittencourt, A. Calderon, E. Carrera, A. Geiser, A. Huffman, C. Lange, K. Lassila-Perini, L. Lloret, T. McCauley, A. Rao, A. Rodriguez Marrero · **LHCb** S. Amerio, C. Burr, B. Couturier, S. Neubert, C. Parkes, S. Roiser, A. Trisovic · **OPERA** G. De Lellis, S. Dmitrievsky · **CERN CernVM** J. Blomer · **CERN EOS** L. Mascetti, H. Rousseau · **CERN Kubernetes** R. Rocha · **CERN OpenShift** A. Lossent, A. Peon

References



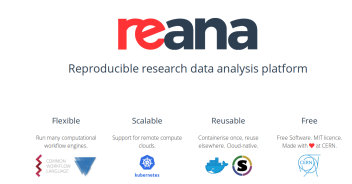
CERN Open Data

http://opendata.cern.ch
http://github.com/cernopendata
cernopendata



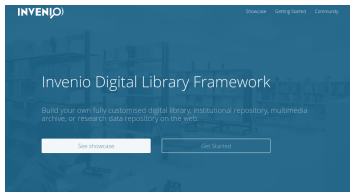
CERN Analysis Preservation

http://analysispreservation.cern.ch
http://github.com/cernanalysispreservation
analysispreserv



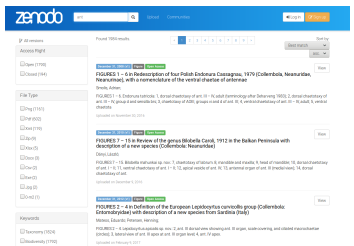
REANA

http://www.reanahub.io
http://github.com/reanahub
reanahub



Invenio

http://inveniosoftware.org
http://github.com/inveniosoftware
inveniosoftware



Zenodo

https://zenodo.org
http://github.com/zenodo
zenodo_org