



Preserving and reusing high-energy-physics data analyses

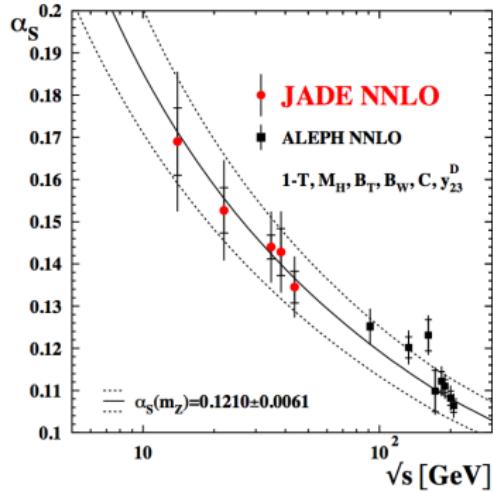
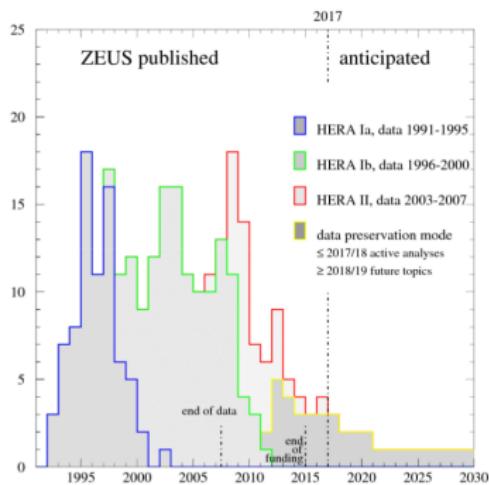
S. Dallmeier-Tiessen², R. Dasler², P. Fokianos², J. Kunčar¹,
A. Lavasa², A. Mattmann², D. Rodríguez¹, T. Šimko¹, A. Trzcinska²,
I. Tsanaktsidis²

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.

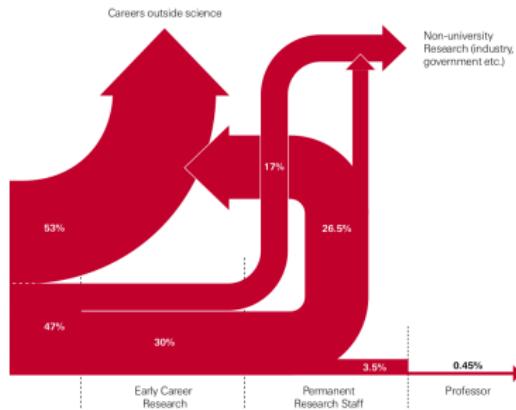
JADE data (1979–1986) still unique even ~ 35 years later.

Long-term value of knowledge?



CMS collaboration

Experimental physics done by groups of \sim 3000 physicists.



Career after PhD

THE ROYAL SOCIETY

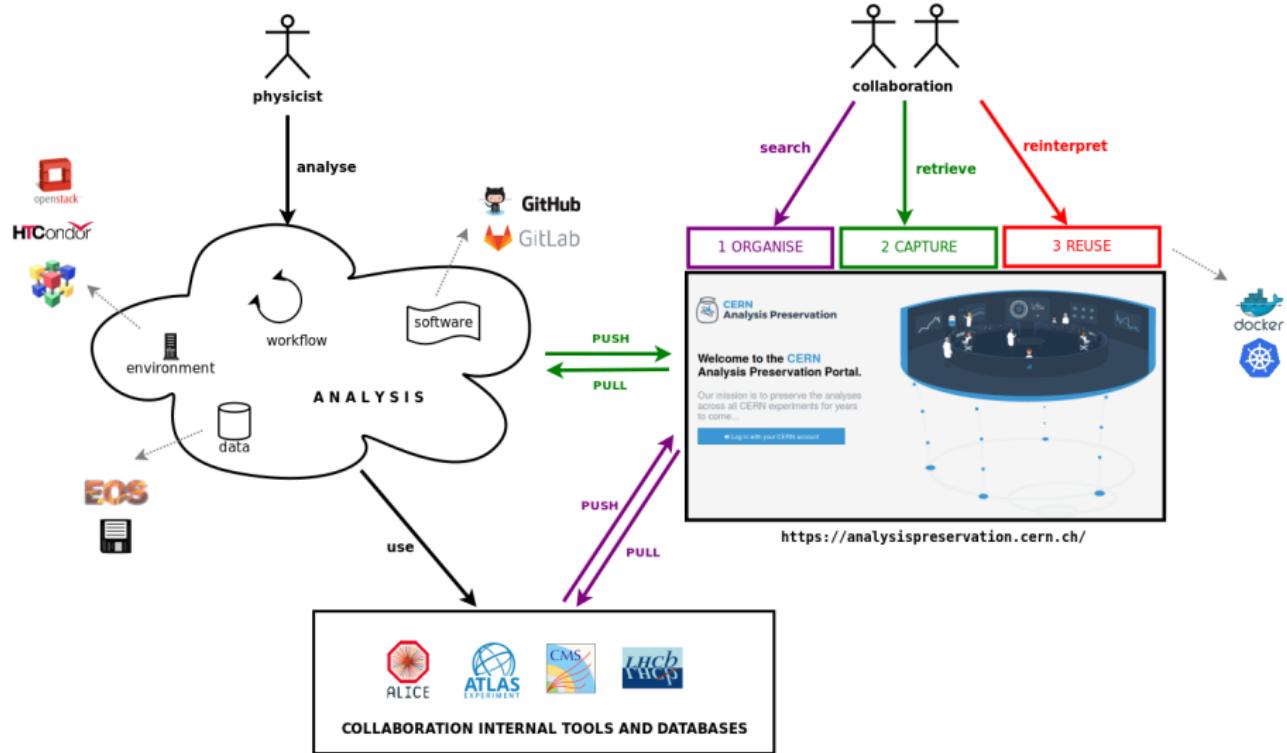
High turnover of young researchers.

CERN Analysis Preservation

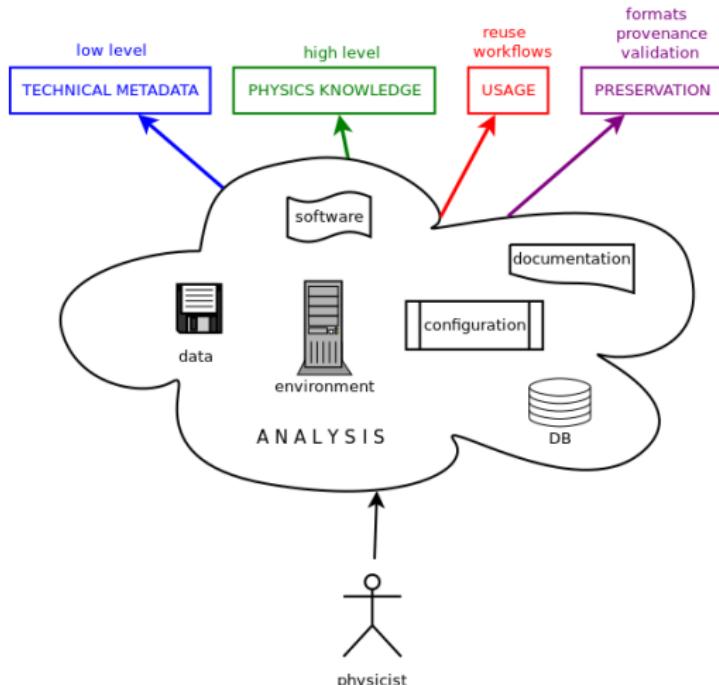
- A platform for **preserving knowledge** and **assets** of an individual physics analysis.
- Capturing the elements needed to **understand** and **rerun** an analysis even several years later:
 - ✓ data
 - ✓ software
 - ✓ environment
 - ✓ workflow
 - ✓ context
 - ✓ documentation
- Advanced **search** 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

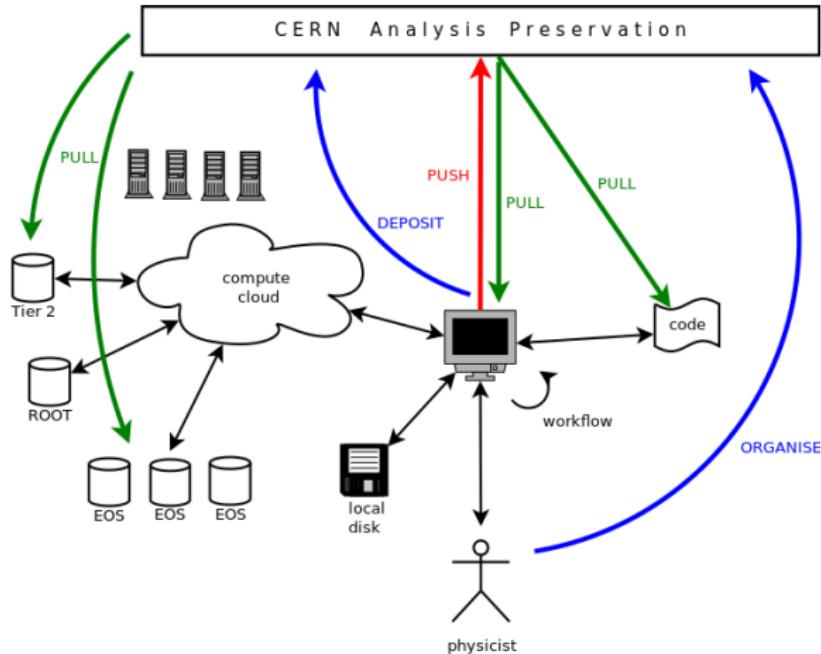


INVENIO

- JSON Schema
- W3C DCAT
- domain-specific fields

Structuring knowledge behind research data analysis.

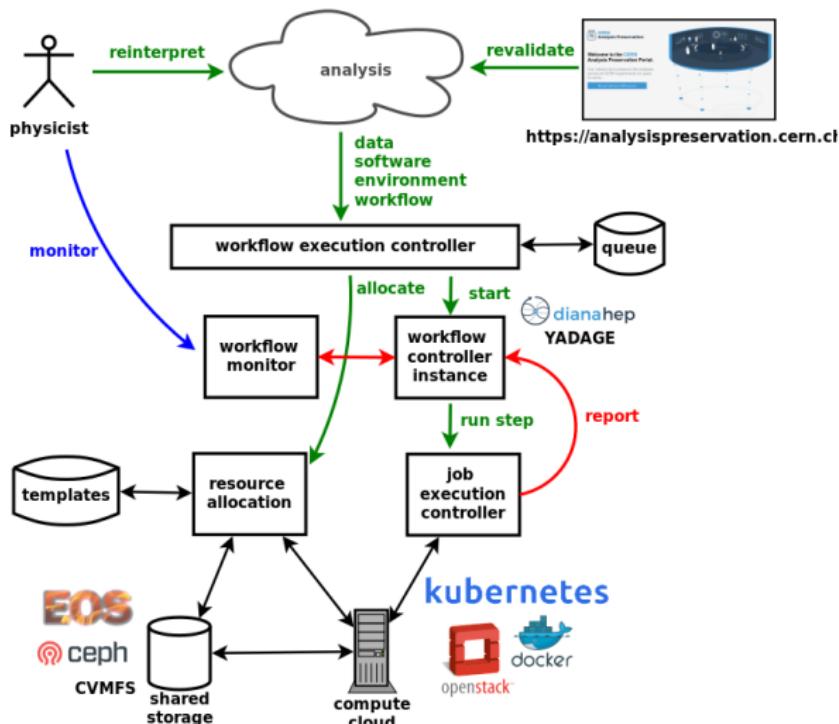
2. Capturing an analysis



- 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.

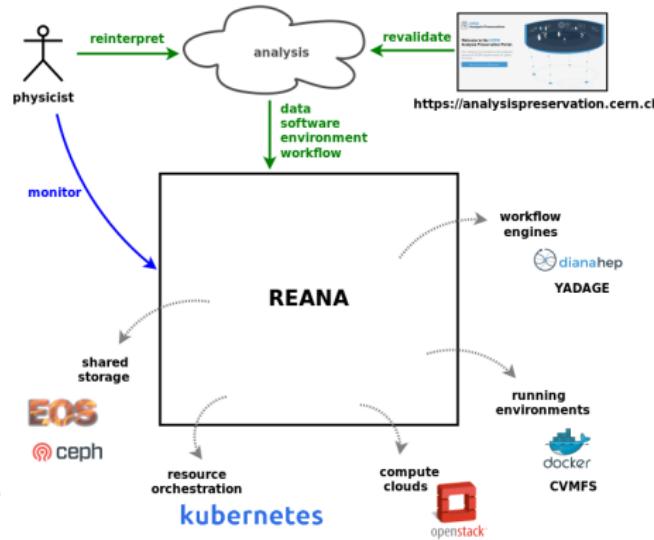
REANA = REusable ANAlyses

- 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



REANA is FOSS

This organization Search Pull requests Issues Marketplace Gist

REANA Hub

Reusable research data analysis platform
<http://reanahub.io/>

Repositories People 10 Teams 2 Projects 0 Settings

Search repositories... Type: All Language: All

reana-workflow-controller

REANA Workflow Controller

Python 3 Updated 7 days ago

reana-server

REANA API server

Python 3 Updated 8 days ago

reana-workflow-engine-yadage

REANA Workflow Engine Yadage

Python 3 Updated 11 days ago

reana-resources-k8s

REANA Resources Kubernetes

Python 2 Updated 11 days ago



REANA - Reusable Analyses

Navigation

- 1. Introduction
- 2. Installation
- 3. Getting started
- 4. Examples
- 5. Architecture
- 6. Components
- 7. Contributing
- 8. Changes
- 9. License
- 10. Authors

REANA@DockerHub

REANA@GitHub

Quick search

Go

REANA - Reusable Analyses

build passing coverage 100% docs latest issues ready for work 2 gitter join chat

license

GNU General Public License v2.0

REANA is a system that permits to instantiate research data analyses on the cloud. It uses container-based technologies and was born to target the use case of particle physics analyses in LHC collaborations. The system paves the way to reusing and reinterpreting preserved data analyses even several years after the original analysis.

- [1. Introduction](#)
 - [1.1. About](#)
 - [1.2. Features](#)
- [2. Installation](#)
 - [2.1. Installing REANA client](#)
 - [2.2. Installing REANA cloud](#)
 - [2.3. Configuring cluster](#)
 - [2.4. Initialising cloud](#)
- [3. Getting started](#)
 - [3.1. About](#)
 - [3.2. Install minikube](#)
 - [3.3. Start minikube](#)
 - [3.4. Install REANA](#)
 - [3.5. Initialise REANA cloud](#)
 - [3.6. Run "hello world" example application](#)
 - [3.7. Run "word population" example analysis](#)
 - [3.8. Washing our bowl](#)
- [4. Examples](#)
 - [4.1. Hello world](#)
 - [4.2. Jupyter notebook](#)
 - [4.3. ROOT and RooFit](#)
- [5. Architecture](#)
 - [5.1. Overview](#)
 - [5.2. Technology](#)

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,100,100,100,100,100,100,100,100,100,100,100,100,100,100,100
 Africa,18,8,19,7,15,5,13,4,18,9,8,8,8,1,8,8,12,8,14,5,14,8,15,2,19,8,23,7
 Asia,53,1,58,4,63,9,63,5,64,9,64,1,57,4,55,6,60,6,68,4,64,4,60,3,59,1,57,1
 Europe,18,3,19,1,18,3,20,6,28,8,21,9,24,7,21,7,12,2,10,9,10,10,5,7,5,3
 Latin America and the Caribbean,8,5,1,7,1,5,2,2,5,3,4,5,6,6,8,5,8,6,8,6,8,6,8,6,9,1,9,4
 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

```
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"]
```

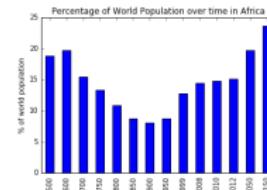
3 environment: CentOS7, IP5

Regional Analysis

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

```
In [6]: def histogram_world_population(region):
    local_pop = pd.read_csv('world_population.csv')[['Region', 'Year']]
    local_pop = local_pop.groupby('Region').sum()
    plt.figure()
    local_pop.plot(kind='bar', legend=None, title='Percentage of World Population over time by region')
    plt.xlabel('Year')
    plt.ylabel('% of world population')
    plt.title('Percentage of World Population over time by region')
    plt.show()
```

To [7]: histogram by region('Africa')



2 **code**: Jupyter notebook

4 workflow: jupyter nbconvert

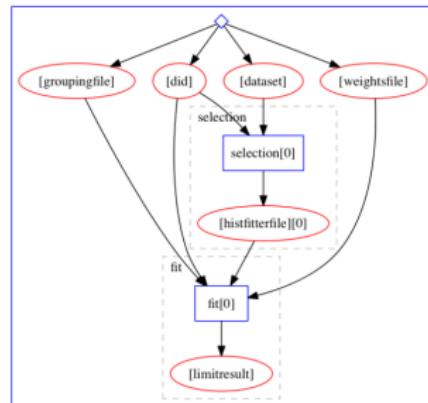
Q <https://github.com/reanahub/reana-demo-worldpopulation>

Complex example: DAG workflows

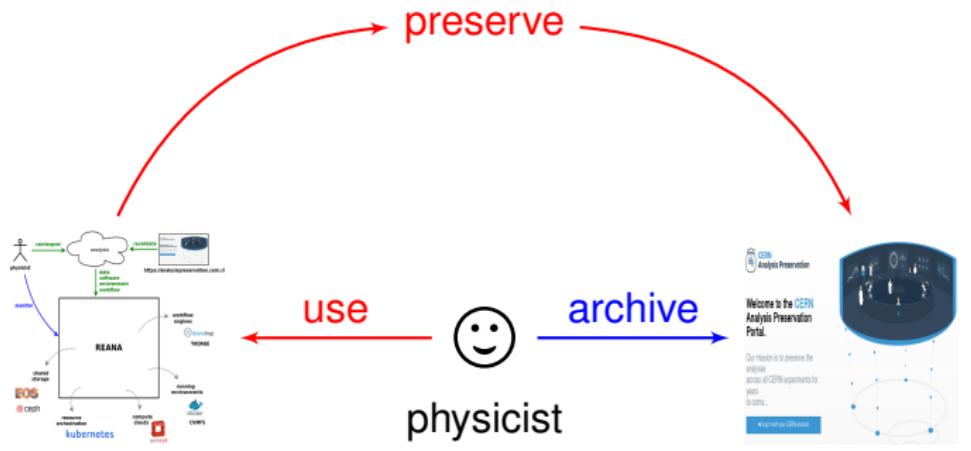
- **case studies** in high-energy-physics with LHC collaborations
 - ALICE AliPhysics post-LEGO train analysis
 - ATLAS multi-B-jets analysis
 - LHCb Lb2LcD0K analysis and data production
- **yadage** parametrised workflow engine



```
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#'}
```



Reusability \rightleftharpoons Preservation



REANA

CERN Analysis Preservation

reuse

Conclusions



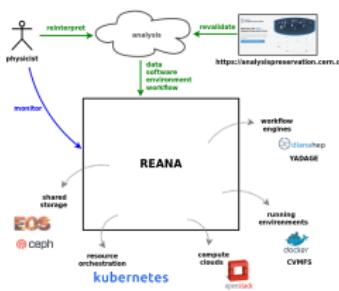
CERN Analysis Preservation

- 🌐 <http://analysispreservation.cern.ch>
- ⌚ <http://github.com/cernanalysispreservation>
- ✉️ analysis-preservation-support@cern.ch



Invenio

- 🌐 <http://inveniosoftware.org>
- ⌚ <http://github.com/inveniosoftware>
- 🐦 [inveniosoftware](https://twitter.com/inveniosoftware)
- ✉️ info@inveniosoftware.org



REANA

- 🌐 <http://reanahub.io>
- ⌚ <http://github.com/reanahub>
- 🐦 [reanahub](https://twitter.com/reanahub)
- ✉️ info@reanahub.io