

# Reproducible high energy physics analyses

Diego Rodriguez Rodriguez  
CERN

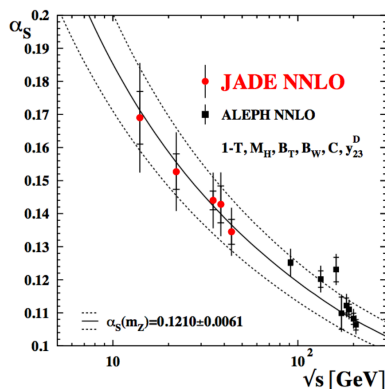
*Docker Containers for Reproducible Research Workshop, Cambridge*

# Outline

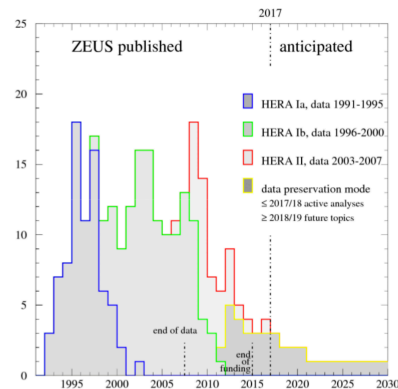
- Why?
- How?
  - CERN Analysis Preservation
  - REANA
- Challenges

# Long term value of data!

- Uniqueness of data
- Publications even ~15 years after data taking ends
- But data is not enough ...



DPHEP <https://arxiv.org/abs/1205.4667>



Achim Geiser <https://indico.cern.ch/event/588219>

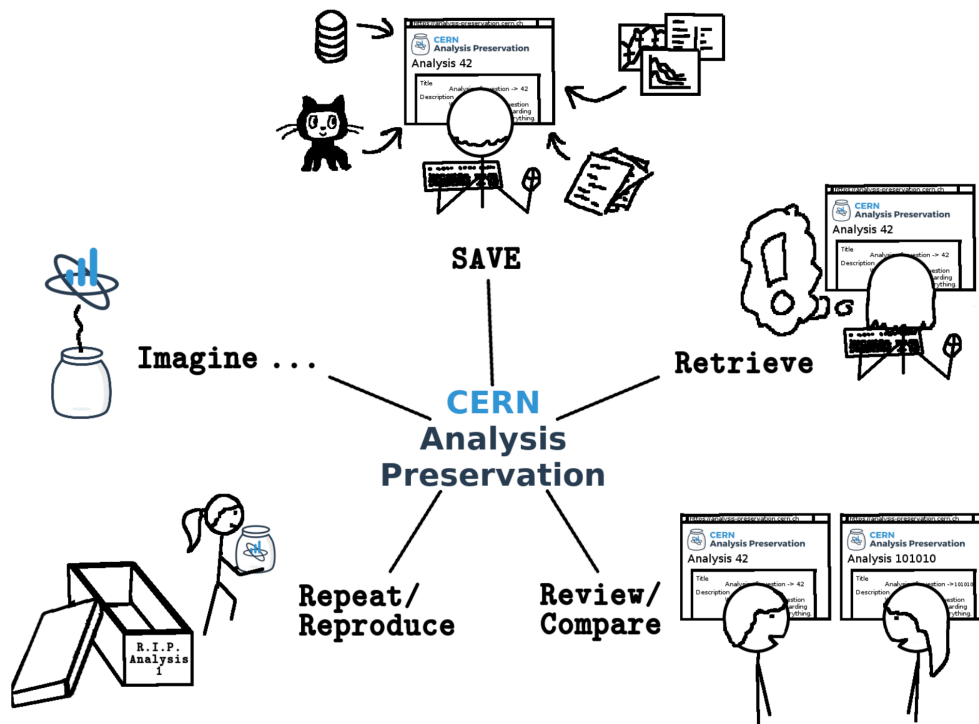
# CERN Analysis Preservation

# CERN Analysis Preservation (CAP)

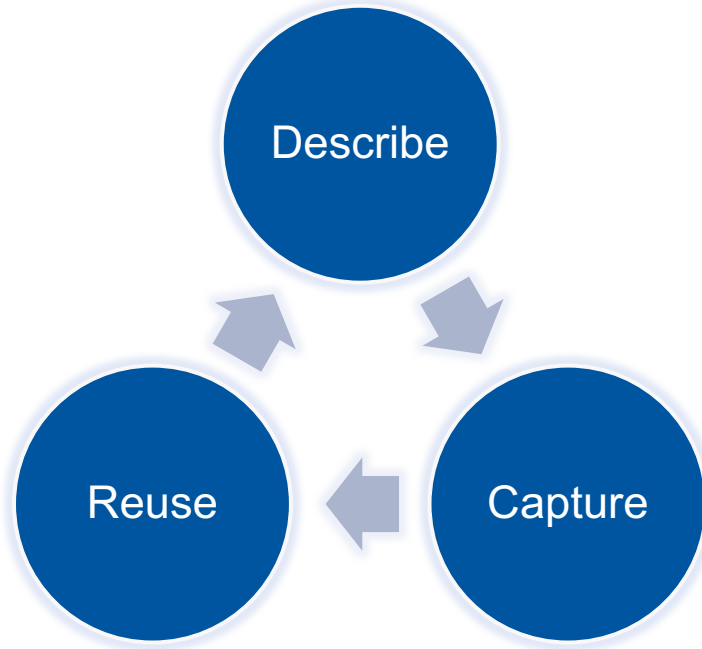
- A platform for **preserving knowledge** and **assets** of an individual physics analysis
- Capturing the elements needed to **understand** and **rerun** an analysis even several years later:
  - ✓ data
  - ✓ environment
  - ✓ context
  - ✓ software
  - ✓ workflow
  - ✓ documentation
- Advanced **search** for high-level physics information
- Applying standard **collaboration access restrictions**

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

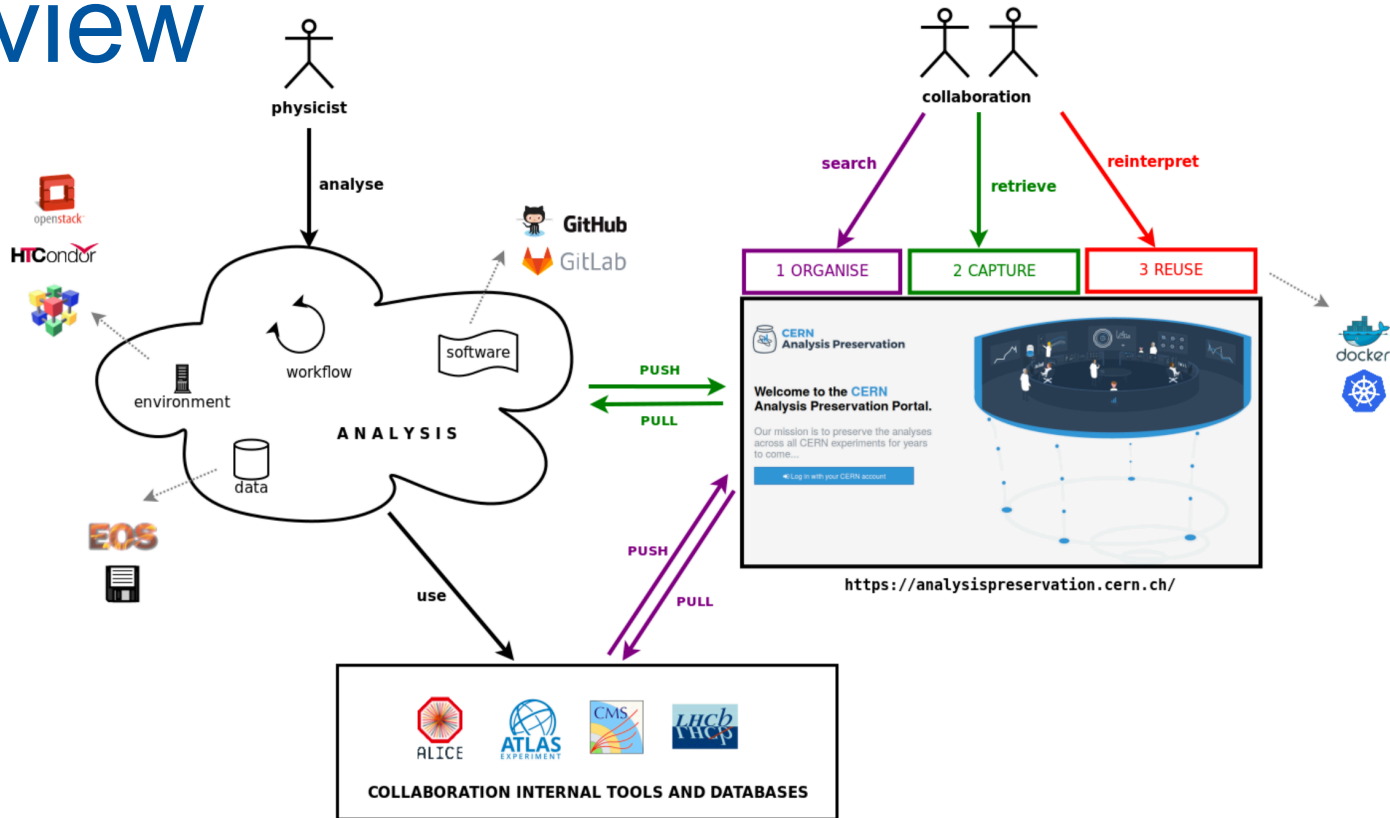
# Use cases



# Three pillars

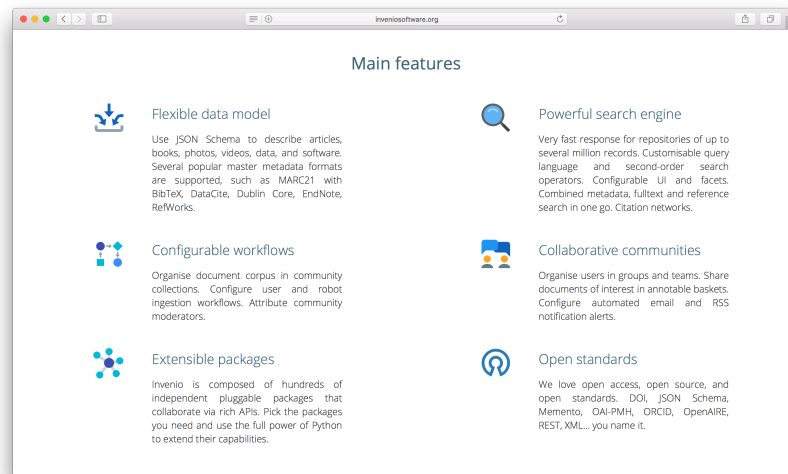
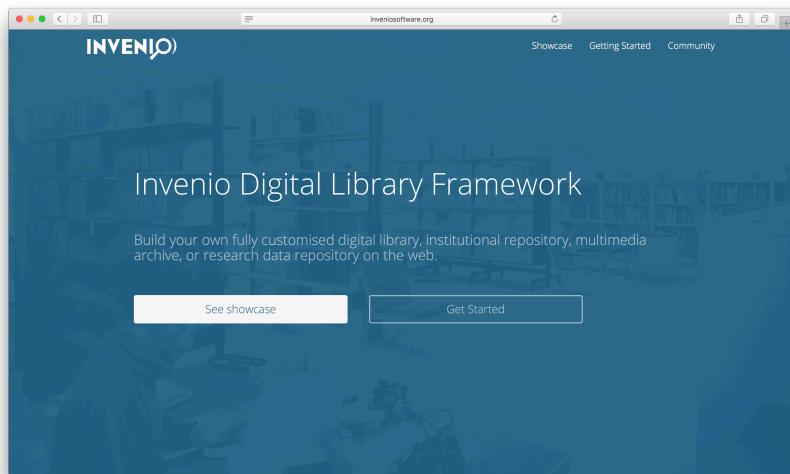


# Overview





# Technology: Invenio



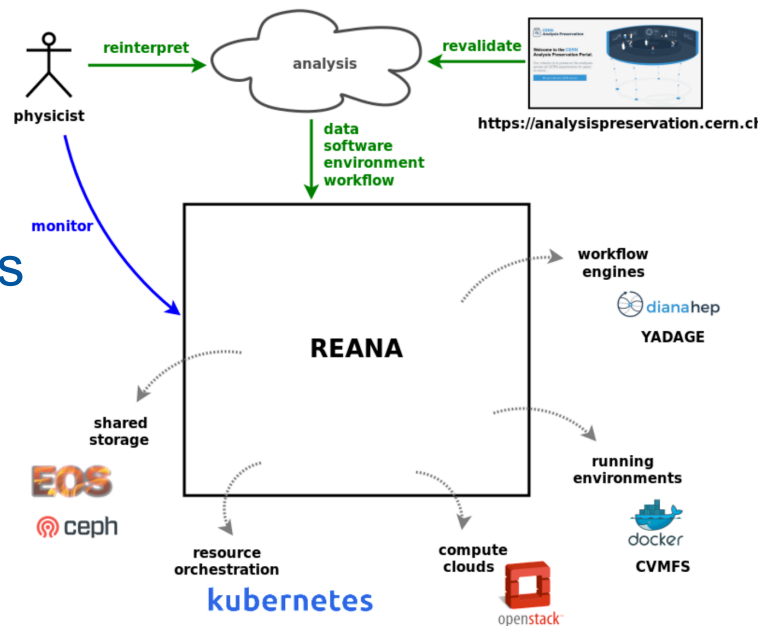
<http://inveniosoftware.org>

REANA = REusable ANAlyses

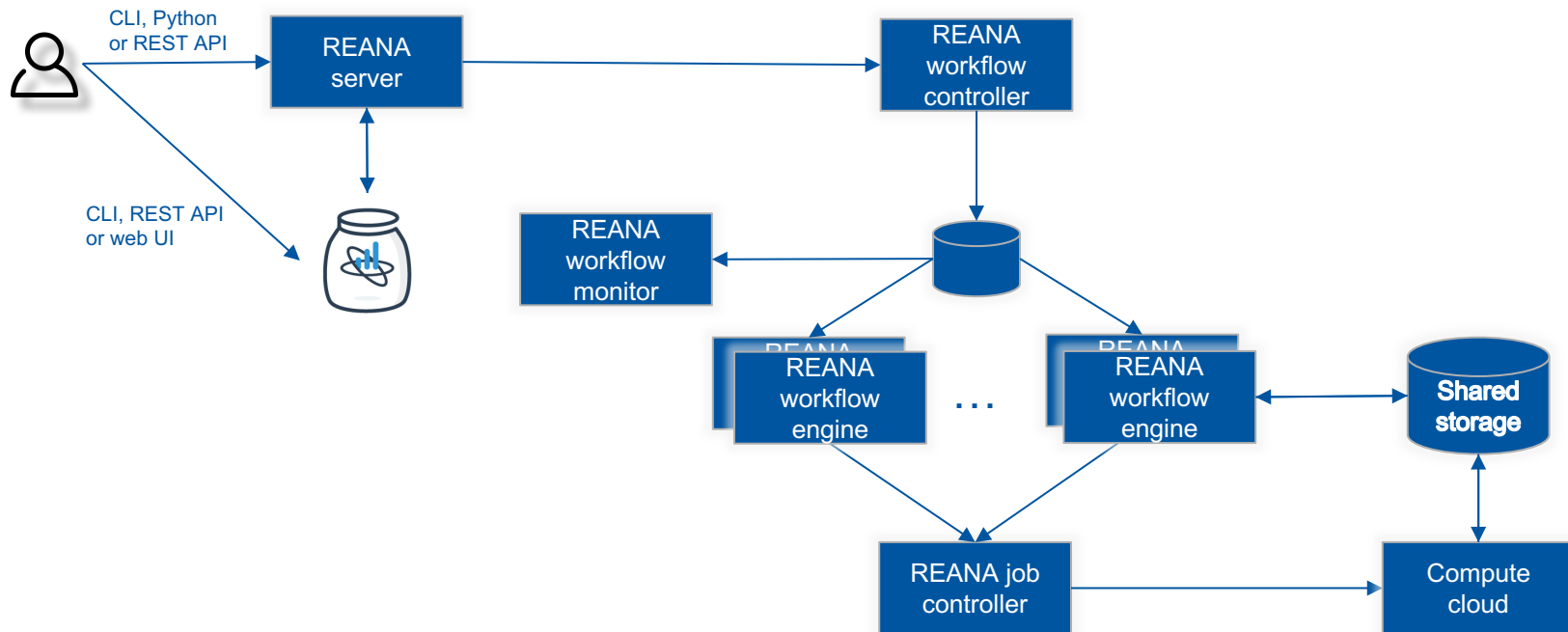
# REANA

- Building system to instantiate preserved analysis on the cloud  
<https://reanahub.io>
- Cloud native
- Aiming to support multiple scenarios
  - compute clouds
  - distributed storage systems
  - workflow engines
  - container technologies
- Close collaboration with:

DASPOS  reicast  dianahep



# REANA architecture



# Key concepts

- Data
- Software
- Environment
- Workflow

# Data

- Stored in CAP
- Ceph FS
- EOS
- ...

[illegible]

*root file*

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
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, 60.6, 60.4, 60.3, 59.1, 57.1
Europe	18.3, 19.1, 18.3, 20.6, 20.8, 21.9, 24.7, 25.1, 27.2, 10.9, 10.7, 10.5, 7.8, 5.3
Latin America and the Caribbean	8.5, 1.7, 1.5, 2.2, 2.5, 3.4, 5.6, 6.8, 5.8, 6.6, 8.6, 9.1, 9.4
Northern America	0.7, 0.5, 0.3, 0.3, 0.3, 0.7, 2.1, 5.6, 8.5, 10.5, 5.4, 4.1
Oceania	0.7, 0.5, 0.3, 0.2, 0.2, 0.4, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5

CSV file

# Software

- Stored in CAP
- CernVM FS
- Preservation trusted *git* repositories
- ...

```
#ifndef __CINT__
#include "RooGlobalFunc.h"
#endif
#include "RooRealVar.h"
#include "RooDataSet.h"
#include "RooGaussian.h"
#include "RooChebychev.h"
#include "RooAddPdf.h"
#include "RooExtendPdf.h"
#include "TCanvas.h"
#include "TAxis.h"
#include "RooPlot.h"
using namespace RooFit ;

void fitdata(const char* input, const char* output)
{
    // Open input file with workspace (generated by rf14_wspacewrite)
    TFile *f = new TFile(input) ;

    // Retrieve workspace from file
    RooWorkspace* w = (RooWorkspace*) f->Get("w") ;

    // Retrieve x,model and data from workspace
    RooRealVar* x = w->var("x") ;
    RooAbsPdf* model = w->pdf("model") ;
    RooAbsData* data = w->data("modelData") ;

    // Fit model to data, extended ML term automatically included
    model->FitTo(*data) ;

    // Plot data and PDF overlaid
    RooPlot* xframe = x->frame(Title("Fit example")) ;
    data->plotOn(xframe) ;
    model->plotOn(xframe,Normalization(1.0,RooAbsReal::RelativeExpected)) ;

    // Overlay the background component of model with a dashed line
    model->plotOn(xframe,Components("bkg"),LineStyle(kDashed),Normalization(1.0,RooAbsReal::RelativeExpected)) ;

    // Draw the frame on the canvas
    TCanvas res("rf202_composite","rf202_composite",600,600) ;
    gPad->SetLeftMargin(0.15) ;
    xframe->GetYaxis()->SetTitleOffset(1.4) ;
    xframe->Draw();

    res.Update();
    res.SaveAs(output);
    res.Close();
}
```

# Environment

- Docker support, other technologies under investigation
- Encourage the usage of base images
  - i.e. *reanahub/reana-env-root6* for ROOT6 based analyses
- Take the most out of image layering
- Encourage collaboration and reusable images



# Environment – reana-env-root6

```
# Environment: ROOT6 on Ubuntu/Trusty:
FROM ubuntu:trusty
RUN apt-get update
RUN apt-get install --yes g++ cpp gcc gfortran git dpkg-dev make binutils libx11-dev libxpm-dev libxft-dev libxext-dev \
    libssl-dev libpcre3-dev xlibmesa-glu-dev libglew1.5-dev libftgl-dev libmysqlclient-dev \
    libfftw3-dev cfitsio-dev graphviz-dev libavahi-compat-libdnssd-dev libldap2-dev python-dev \
    libxml2-dev libkrb5-dev libgsl0-dev libqt4-dev libx11-dev libxpm-dev

ENV ROOTSYS /usr/local
RUN git clone --quiet http://root.cern.ch/git/root.git /code/root-v6-02-12 &&\
    cd /code/root-v6-02-12 &&\
    git checkout v6-02-12 &&\
    ./configure --all &&\
    make -j4 &&\
    make -j4 install &&\
    cd / &&\
    rm -rf /code
```

# Workflow

- Workflow specifications over difficult to reproduce READMEs
- Testable approach
- *yadage* workflows support
- Alternatives such as *snakemake* under investigation

```
stages:  
- name: hello_world  
  dependencies: [init]  
  scheduler:  
    scheduler_type: singlestep-stage  
  parameters:  
    parone: {stages: init, output: par, unwrap: true}  
    outputfile: '{workdir}/hello_world.txt'  
  step:  
    process:  
      process_type: 'string-interpolated-cmd'  
      cmd: 'echo Hello {parone} | tee {outputfile}'  
    publisher:  
      publisher_type: 'frompar-pub'  
      outputmap:  
        outputfile: outputfile  
    environment:  
      environment_type: 'docker-encapsulated'  
      image: busybox
```

*yadage* hello world example

# Workflow – simple roofit example

```
stages:
- name: gendata
  dependencies: ['init']
  scheduler:
    scheduler_type: singlestep-stage
  parameters:
    events: {stages: init, output: events, unwrap: true}
    outfilename: '{workdir}/data.root'
  step:
    process:
      process_type: 'interpolated-script-cmd'
      script: root -b -q 'gendata.C({events},{outfilename})'
    publisher:
      publisher_type: 'frompar-pub'
      outputmap:
        data: outfilename
    environment:
      environment_type: 'docker-encapsulated'
      image: johndoe/reana-demo-root6-roofit
- name: fitdata
  dependencies: ['gendata']
  scheduler:
    scheduler_type: singlestep-stage
  parameters:
    data: {stages: gendata, output: data, unwrap: true}
    outfile: '{workdir}/plot.png'
  step:
    process:
      process_type: 'interpolated-script-cmd'
      script: root -b -q 'fitdata.C("{data}", "{outfile}")'
    publisher:
      publisher_type: 'frompar-pub'
      outputmap:
        plot: outfile
    environment:
      environment_type: 'docker-encapsulated'
      image: johndoe/reana-demo-root6-roofit
```



Lukas Heinrich <https://github.com/diana-hep/yadage>

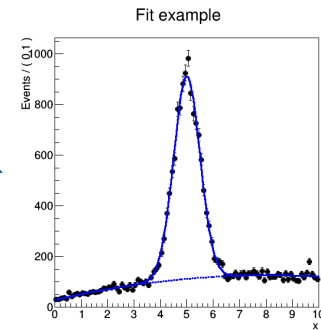
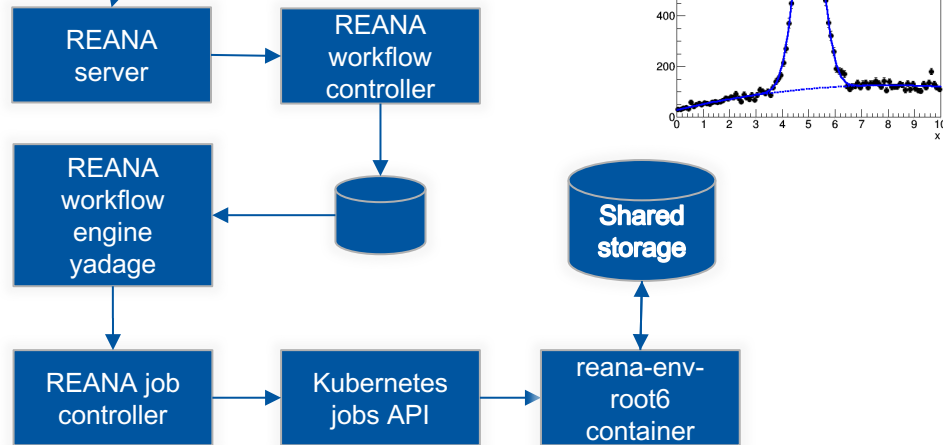
# REANA example

```
stages:
- name: gendata
  dependencies: ['init']
  scheduler:
    scheduler_type: singlestep-stage
  parameters:
    events: {stages: init, output: events, unwrap: true}
    outfile: '{workdir}/data.root'
  step:
    process:
      process_type: 'interpolated-script-cmd'
      script: root -b -q 'gendata.C({events},{outfile})'
    publisher:
      publisher_type: 'frompar-pub'
      outputmap:
        data: outfile
    environment:
      environment_type: 'docker-encapsulated'
      image: johndoe/reana-demo-root6-roofit
- name: fitdata
  dependencies: ['gendata']
  scheduler:
    scheduler_type: singlestep-stage
  parameters:
    data: {stages: gendata, output: data, unwrap: true}
    outfile: '{workdir}/plot.png'
  step:
    process:
      process_type: 'interpolated-script-cmd'
      script: root -b -q 'fitdata.C(\"{data}\",\"{outfile}\")'
    publisher:
      publisher_type: 'frompar-pub'
      outputmap:
        plot: outfile
    environment:
      environment_type: 'docker-encapsulated'
      image: johndoe/reana-demo-root6-roofit
```

workflow.yml

*reana-client  
not released yet!*

```
$ pip install reana-client
$ export REANA_SERVER_URL=https://reana.cern.ch
$ reana-client run workflow.yml
[INFO] Starting reana-demo-root6-roofit analysis...
[...]
[INFO] Done. You can see the results in the 'output/' directory.
```



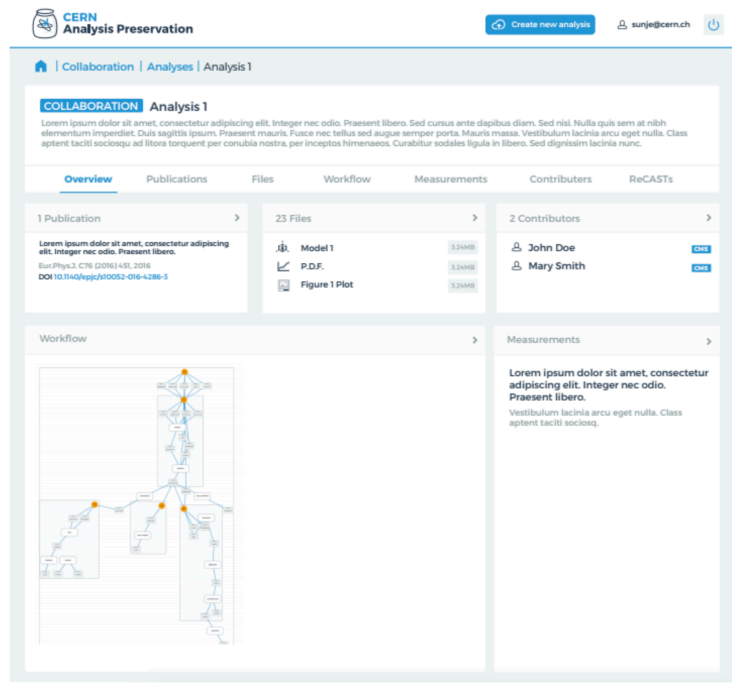
more at <https://github.com/reanahub/reana-demo-root6-roofit>

# Current status

- Local development environment
- Multiorganisation setup
- REANA environments (Docker base images)
- First physics example well received by community
- User testing for the CAP portal
- Pilot with four experiments

# What is next?




- Real world physics analysis coming
- Minimal but well documented API
- Release CLI and Python clients
- Minimal integration of both projects



# Challenges

- Social challenges
  - publish or perish
  - scientific benefit vs cost of preservation
- Data
  - Ever-increasing data size?
- Software
  - Ever-changing computing technology?

CERN Analysis Preservation  <http://analysispreservation.cern.ch/>  
 <http://github.com/cernanalysispreservation>

REANA  <http://reanahub.io/>  
 <http://github.com/reanahub>  
 @reanahub

**CERN IT** H. Hirvonsalo, D. Rodríguez, T. Šimko **CERN SIS** S. Dallmeier-Tiessen, R. Dasler, S. Feger, P. Fokianos, A. Lavasa, A. Mattmann, I. Tsanaktsidis, A. Trzcinska **ALICE** M. Gheata, C. Grigoras, M. Zimmermann **ATLAS** K. Cranmer, L. Heinrich, A. Sanchez Pineda, D. Rousseau, F. Socher CMS A. Calderon, A. Geiser, A. Huffman, K. Lassila-Perini, T. McCauley, A. Rao, A. Rodriguez Marrero **LHCb** S. Amerio, B. Couturier, S. Neubert, A. Trisovic **CERN CernVM** J. Blomer **CERN Kubernetes** R. Rocha **CERN EOS** L. Mascetti **DASPOS** M. Hildreth, H. Meng, D. Thain, A. Vyushkov **DPHEP** J. Shiers





Questions?