

Boosting productivity of researches integrating IT services into an interactive analysis platform

D. Piparo (CERN EP-SFT) for the SWAN team



<https://cern.ch/swan>

Aug 31st, 2018

GridKa School





The Work Behind These Slides

- > I am here on behalf of the SWAN team, spanning two CERN Departments: Experimental Physics and Information Technology
- > The team: E. Tejedor, D. Castro, P. Mato, E. Bocchi, J. Moscicki, M. Lamanna, P. Kothuri, D. P.
- > **I'd like to thank Diogo Castro and Enric Tejedor for contributing so much to this talk.**

Introduction



Accelerating Science





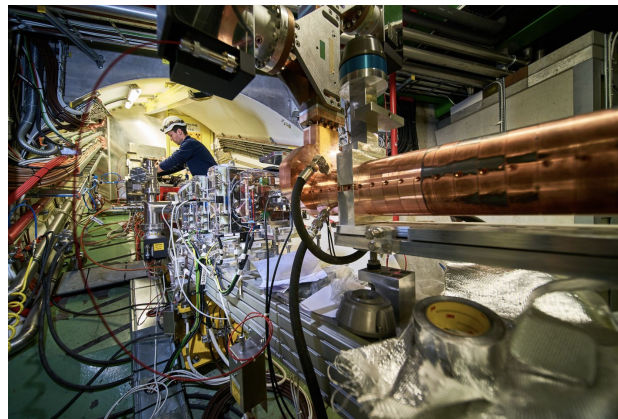
CERN Mission



<https://kt.cern/success-stories/cern-virtual-machine-file-system-euclid-science-data-centres>



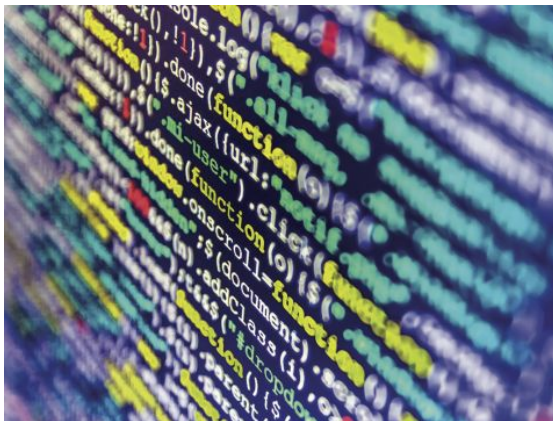
<https://root.cern.ch/code-snippet/learning-root>



<https://home.cern/about/experiments/awake>



CERN Mission



<https://kt.cern/success-stories/cern-virtual-machine-file-system-euclid-science-data-centres>

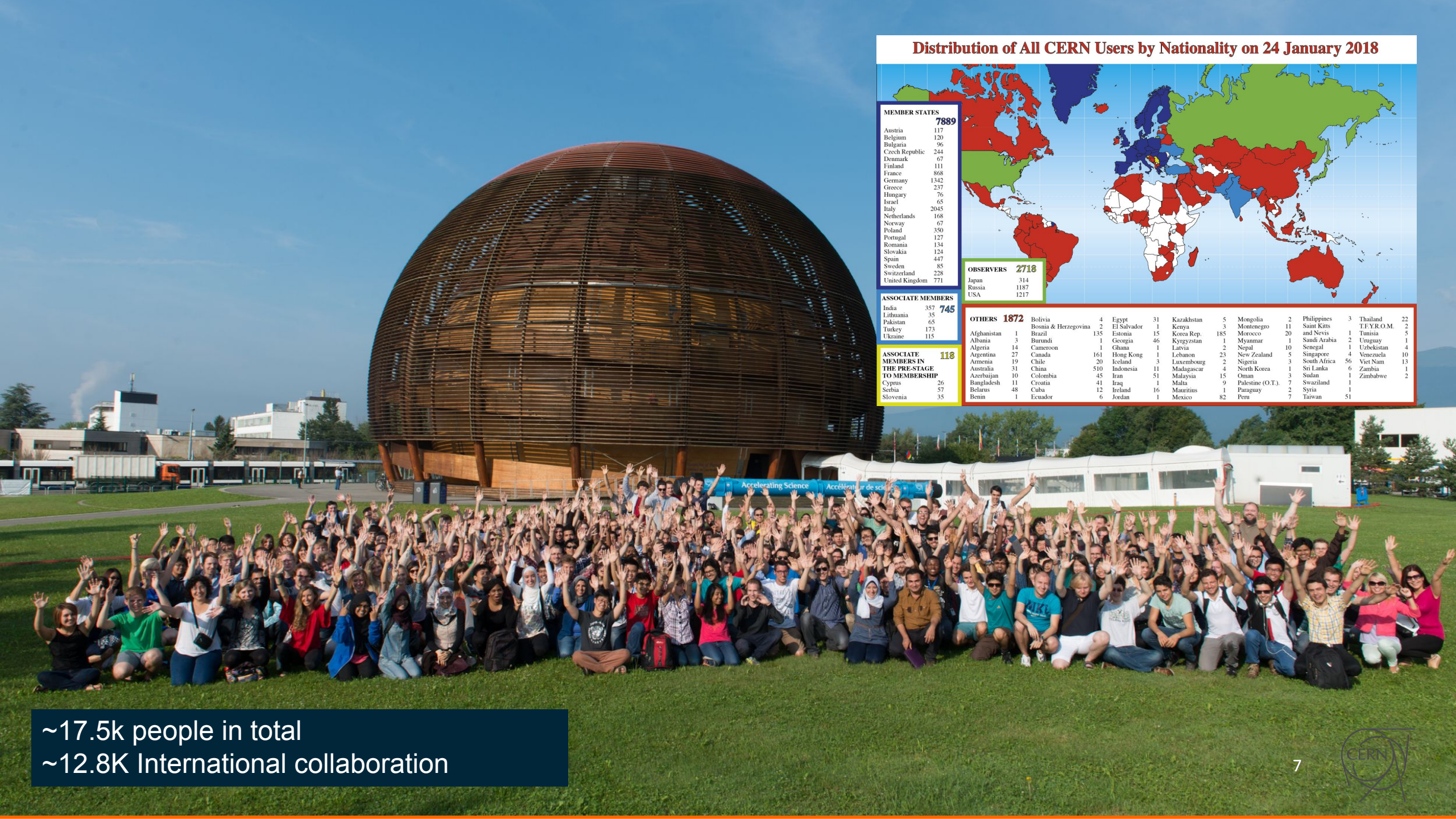


<http://root.cern.ch/code-snippet/learning-root>

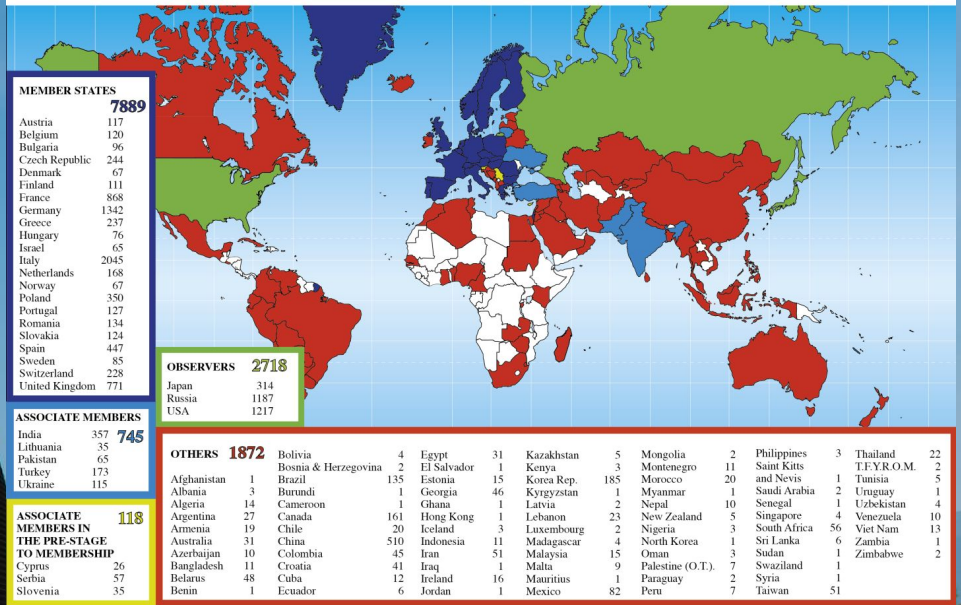
<http://careers.cern/>



<https://home.cern/about/experiments/awake>



Distribution of All CERN Users by Nationality on 24 January 2018

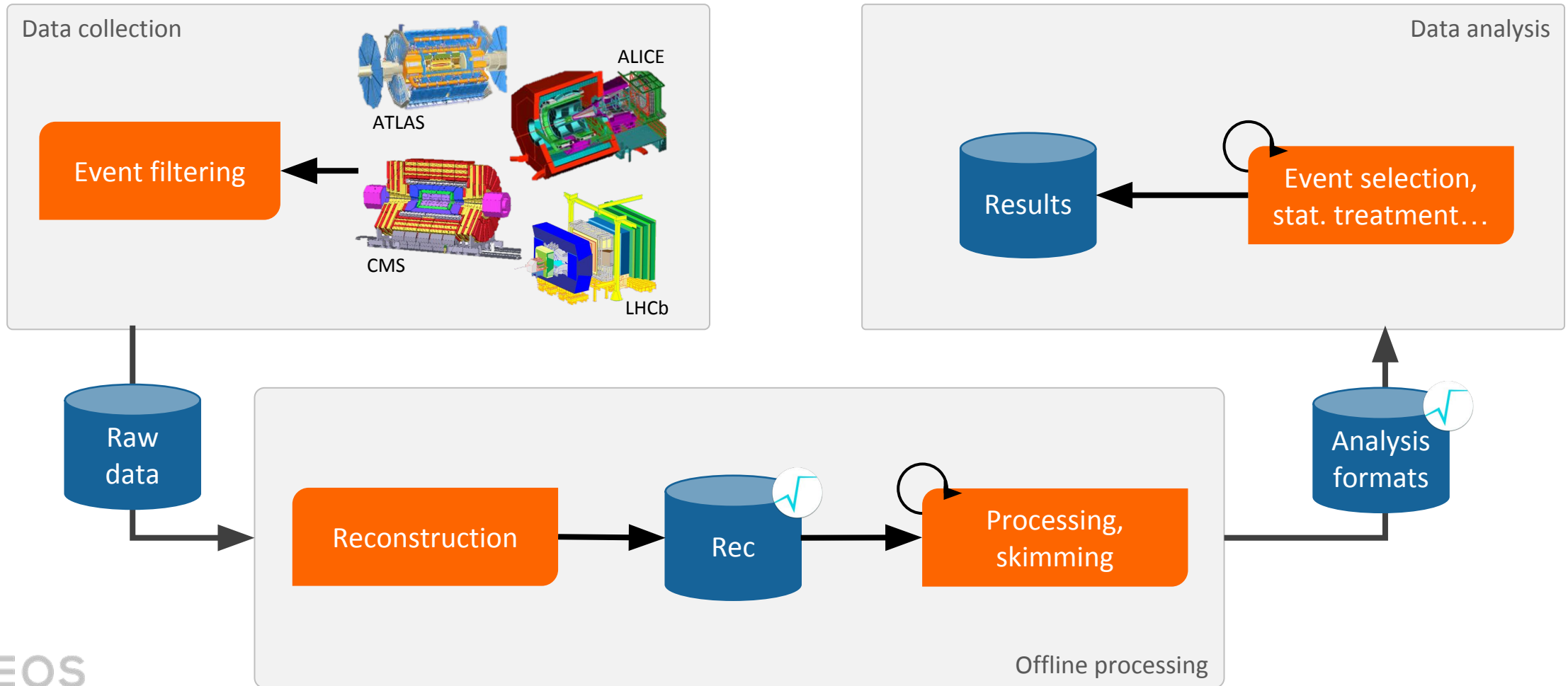


~17.5k people in total
~12.8K International collaboration



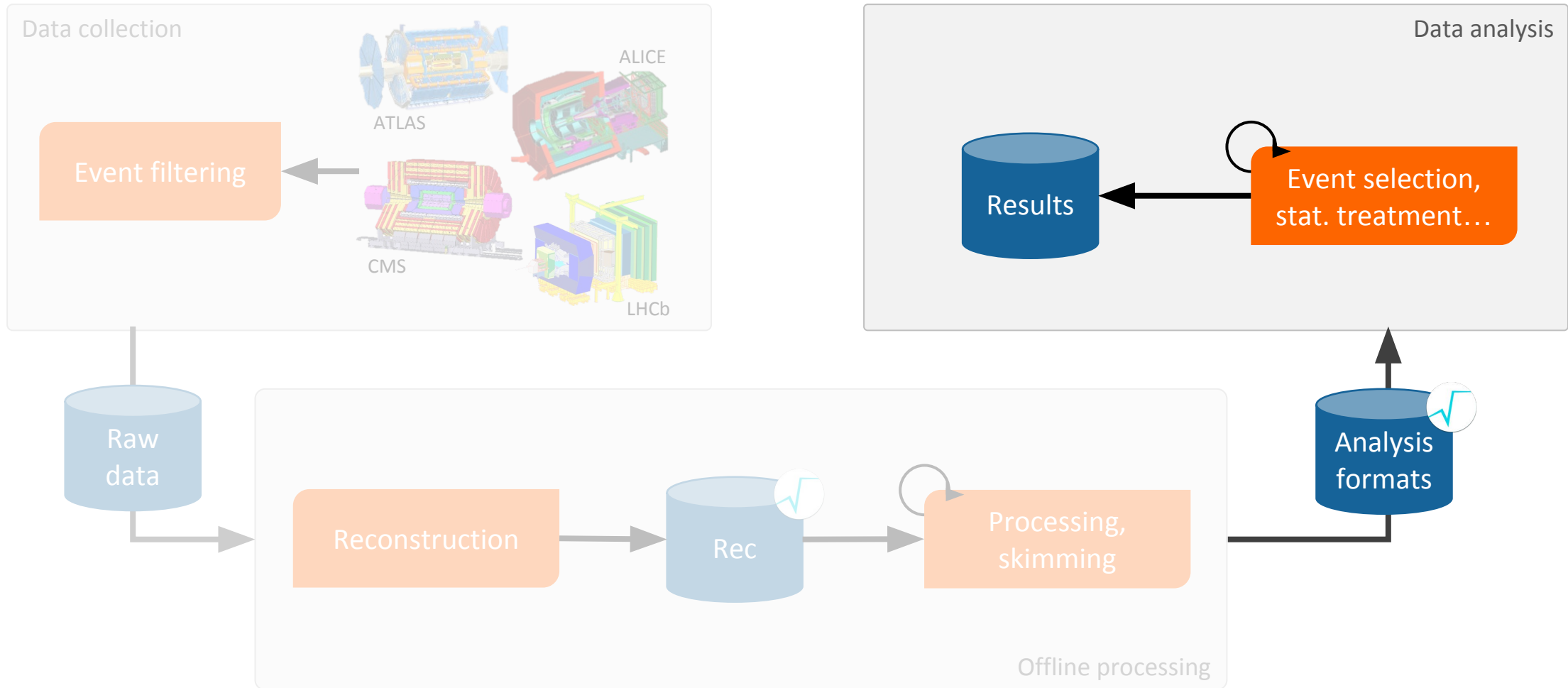


The steps towards data analysis





Service for Web based Analysis



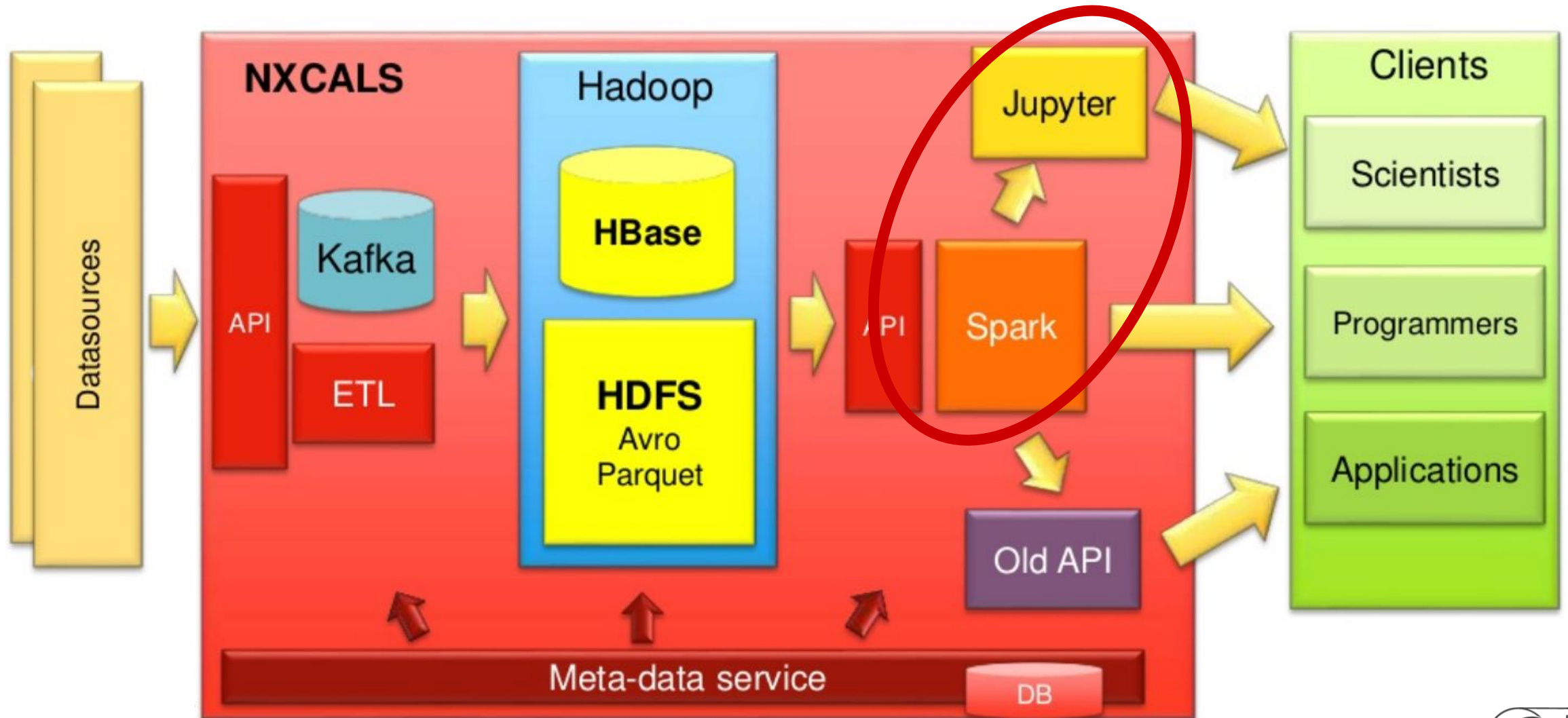


Not Only HEP Analysis: LHC Controls Data





Not Only HEP Analysis: LHC Controls Data



Service for **W**eb based **A**nalysis: **SWAN**



ROOT

- > **The** physics Data Analysis Framework
 - Created at CERN
- > Used for data analysis, I/O, ML, advanced graphics...
- > Integrated with Jupyter notebooks
 - First C++ kernel based on ROOT C++
- > The origin story of SWAN
- > <http://root.cern>





Service for Web based Analysis

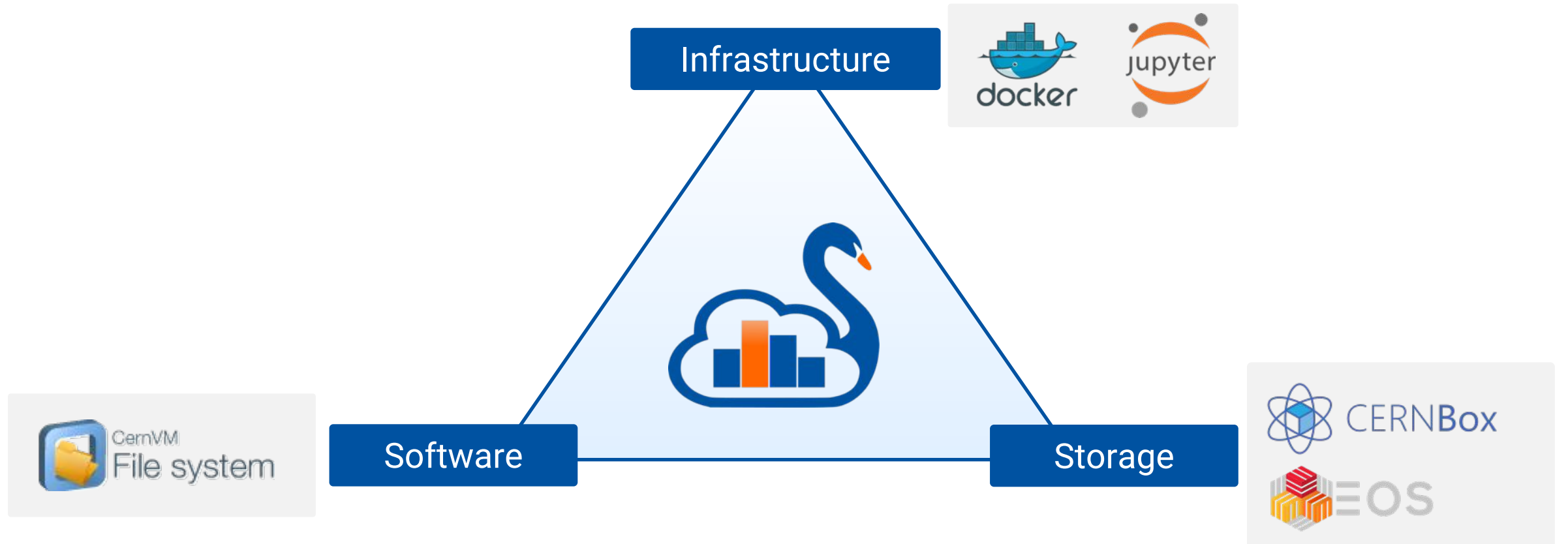
- > Analysis only with a web browser
 - Blur boundaries between web and local applications
 - Available everywhere and at anytime
 - Providing access also to Jupyter notebooks
- > Easy to use (but powerful)
 - No local installation and configuration needed
- > Create easily shareable scientific results: plots, data, code
 - Storage is crucial: mass & synchronized
- > Integration with CERN resources
 - Access software, user/experiments data, mass processing power
- > Synergy among analysis ecosystems : ROOT, R, Python, ...



<https://en.wikipedia.org/wiki/Argentina>



Integrating services





Open Source Technologies





Jupyter Notebooks

Text

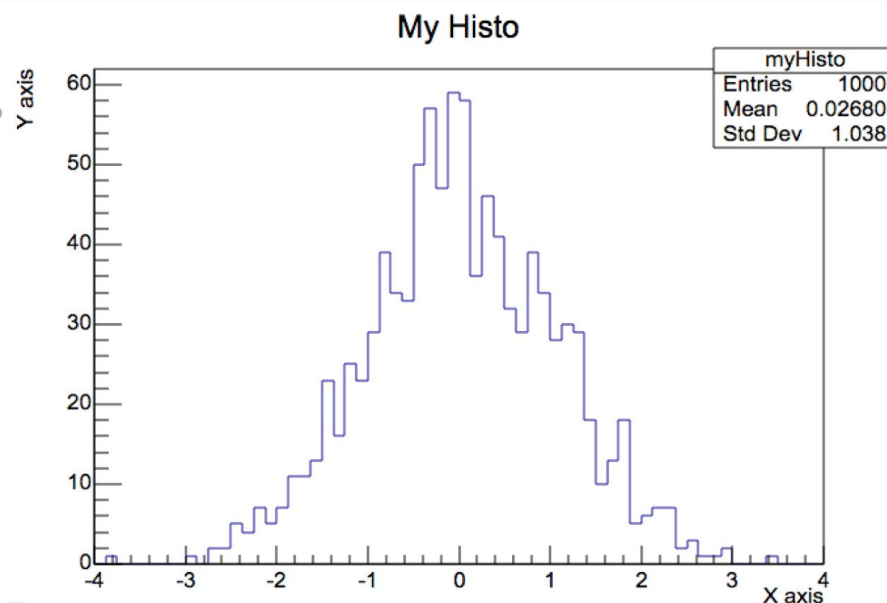
Code

Graphics

Displaying graphics

We can now draw the histogram. We will at first create a [canvas](#), the entity which in ROOT holds graphics primitives. Note that thanks to [JSROOT](#), this is not a static plot but an interactive visualisation. Try to play with it and save it as image when you are satisfied!

```
In [5]: c = ROOT.TCanvas()  
h.Draw()  
c.Draw()
```





Does it Have a Terminal?



```
bash-4.1$ root -b
```

```
-----  
| Welcome to ROOT 6.12/06                http://root.cern.ch |  
|                                     (c) 1995-2017, The ROOT Team |  
| Built for linuxx8664gcc                |  
| From tag v6-12-06, 9 February 2018    |  
| Try '.help', '.demo', '.license', '.credits', '.quit'/'.'q' |  
-----
```

```
root [0] .q
```

```
bash-4.1$ ls /cvmfs
```

```
alice.cern.ch      alice-ocdb.cern.ch  atlas-condb.cern.ch  fcc.cern.ch  lhcb.cern.ch  sft-nightlies.cern.ch
```

```
alice-nightlies.cern.ch  atlas.cern.ch      cms.cern.ch      geant4.cern.ch  sft.cern.ch
```

```
bash-4.1$ ls /eos
```

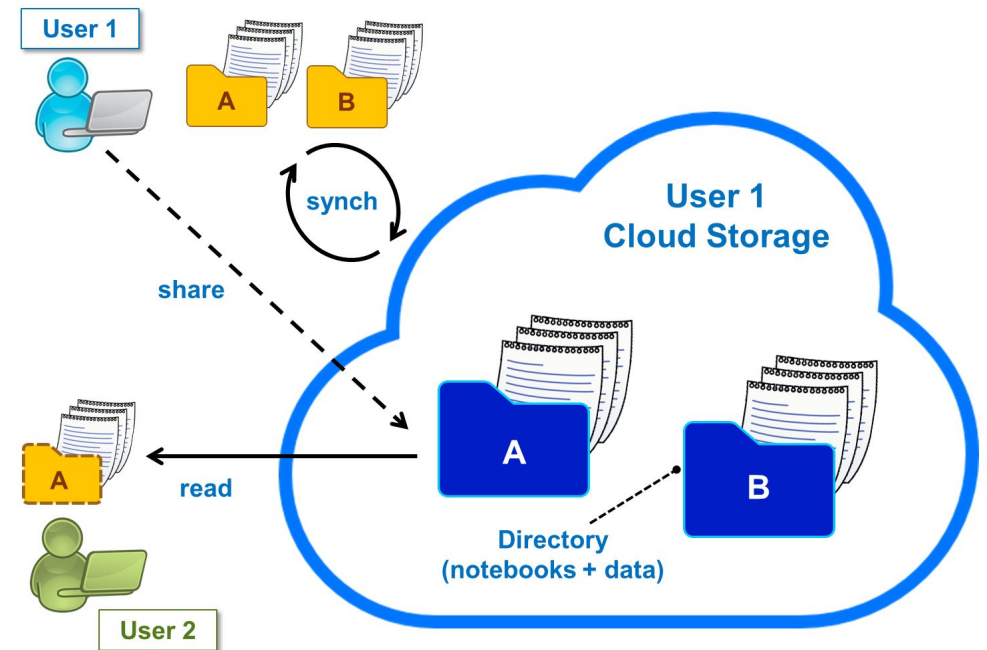
```
experiment  fcc  hepdata  opendata  project  user
```

```
bash-4.1$
```



Storage

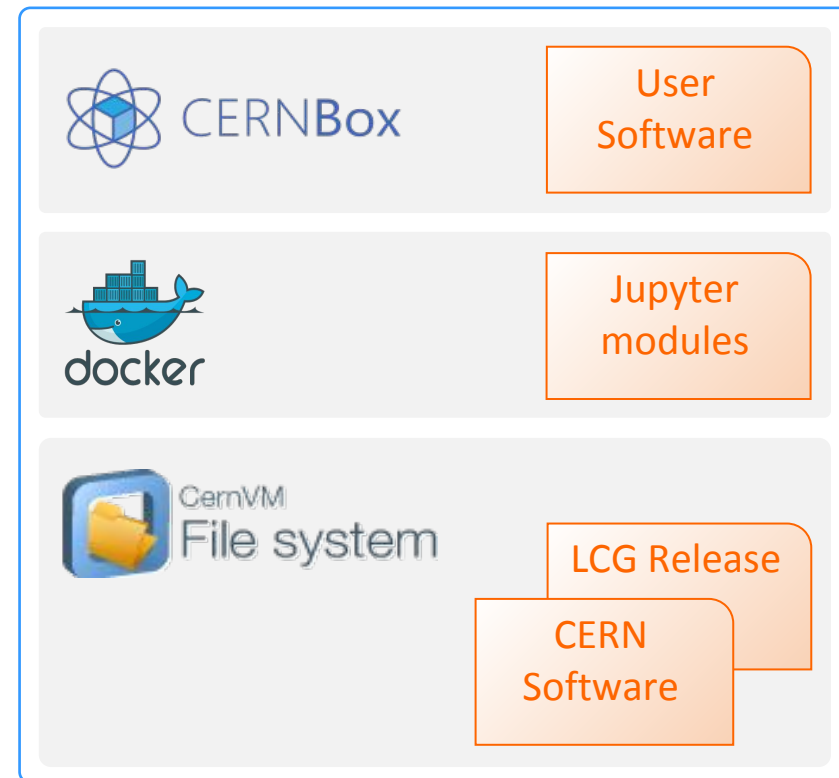
- > Uses EOS disk storage system
 - All experiment data potentially available
- > CERNBox is SWAN's home directory
 - Storage for your notebooks and data
- > Sync&Share
 - Files synced across devices and the Cloud
 - Collaborative analysis





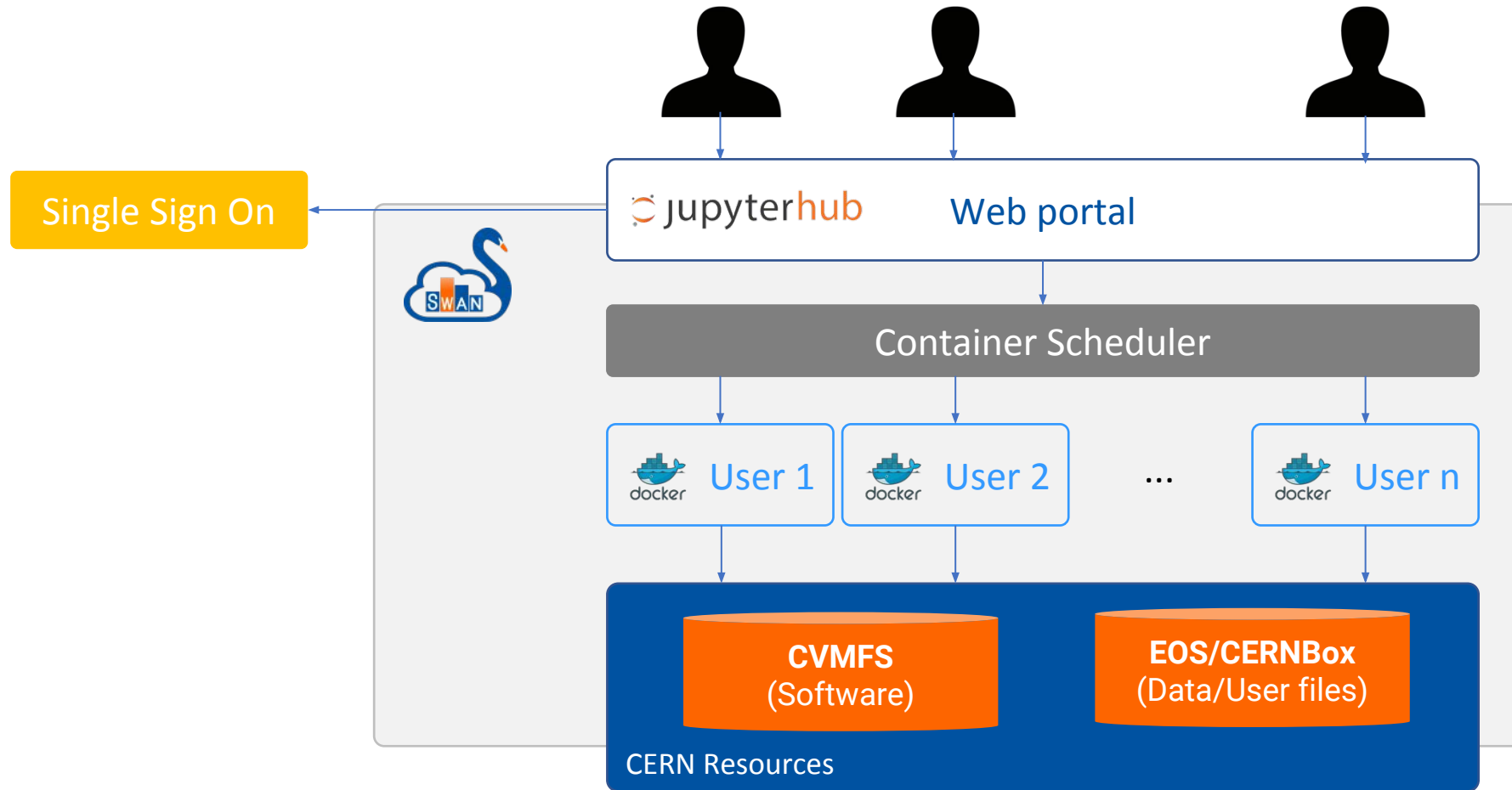
Software

- > Software distributed through CVMFS
 - “LCG Releases” - pack a series of compatible packages
 - **Single Docker image, many software stacks**
 - Reduced Docker Image size
 - Lazy fetching of software
- > Possibility to install libraries in user cloud storage
 - Good way to use custom/not mainstream packages
 - Configurable environment
- > Multiple “kernels” (“links” between Jupyter and a language) available
 - Python 2 or 3, ROOT C++, R, Octave





Architecture

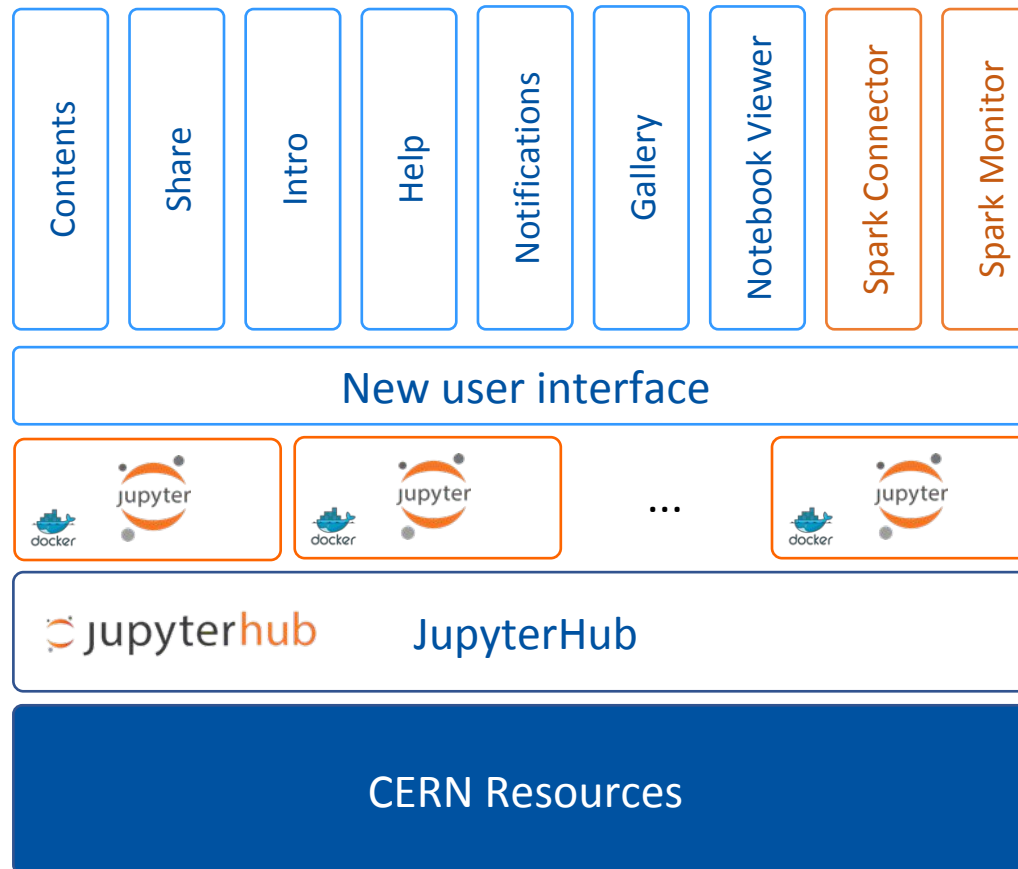


Leveraging the power of Jupyter

Service for Web based Analysis



Extensions, extensions everywhere





New User Interface



Configure Environment

Specify the parameters that will be used to contextualise the container which is created for you. See [the online SWAN guide](#) for more details.

Software stack [more...](#)
91

Platform [more...](#)
x86_64-slc6-gcc62-opt

Environment script [more...](#)
e.g. \$CERNBOX_HOME/MySWAN/myscript.sh

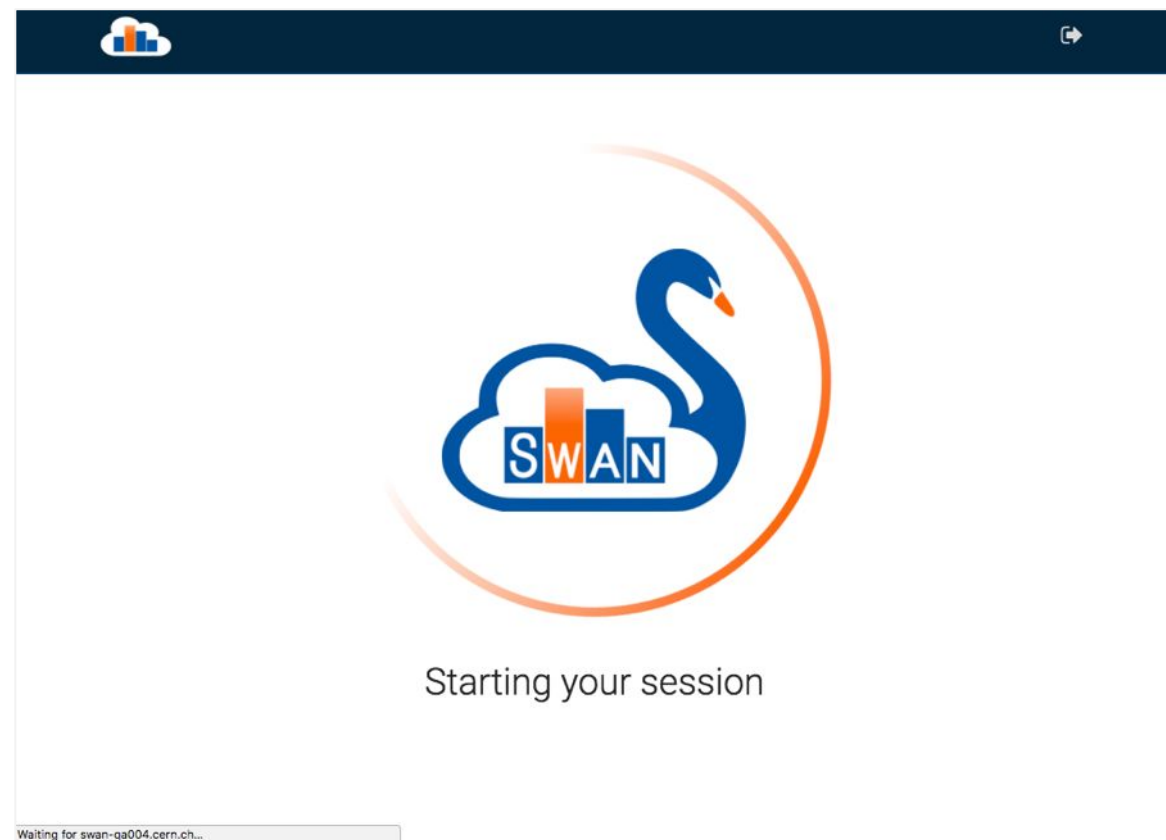
Number of cores [more...](#)
2

Memory [more...](#)
8 GB

Spark cluster [more...](#)
Hadalytic

☐ Always start with this configuration

Start my Session





New User Interface



ProjectsShareCERNBox

SWAN > My Projects

My Projects

☐ NAME

STATUS

MODIFIED

Proj1		5 days ago
Proj2		15 days ago
Project		21 days ago
Project 1		2 months ago
Project 2		4 months ago
ProjTest		15 days ago
Spark		7 days ago
SWAN-Spark_NXCALS_Example		20 days ago
teste		19 days ago

SWAN © Copyright CERN 2017. All rights reserved.
Home | Contacts | Support | Report a bug | Imprint



Spark > physics_analysis_using_swan_spark_template
(autosaved)

FILEEDITVIEWINSERTCELLKERNELHELPNot TrustedPython 2



Integration of SWAN with Spark clusters

This notebook demonstrates the functionality provided by a SWAN prototype machine that allows to offload computations to an external Spark cluster. The Spark version we are going to use is 2.1.0 and we are going to connect to the analytix cluster (as previously selected in the SWAN web form).

Step 1 - Acquire the necessary credentials to access the Spark cluster.

```
In [1]: import getpass
import os, sys, re

print("Please enter your password")
ret = os.system("echo \"%s\" | kinit" % re.escape(getpass.getpass()))

if ret == 0: print("Credentials created successfully")
else:      sys.stderr.write('Error creating credentials, return code: %s\n' % ret)
```

Collaborative Analysis

Service for Web based Analysis



Sharing made easy

- > Sharing from inside SWAN interface
 - Integration with CERNBox
- > Users can share “Projects”
 - Special kind of folder that contains notebooks and other files, like input data
 - Self contained

The screenshot displays the SWAN web interface. At the top, there's a navigation bar with the SWAN logo, 'Projects', and 'Share' tabs. The main content area shows a breadcrumb trail: 'SWAN > My Projects > Super Real Analysis with TOTEM data'. Below this, the project title 'Super Real Analysis with TOTEM data' is followed by an upward arrow. A list of files is shown: 'DistillDistribution.ipynb' and 'dataset.root'. A search bar with the placeholder 'Start typing to add names...' is present. To the right, a 'Share Project' modal is open, showing the project name and a list of users to share with: 'Danilo Piparo (danilo)' and 'Enric Tejedor Saavedra (enric)'. At the bottom of the modal are 'Stop Sharing' and 'Update' buttons. The footer of the SWAN interface contains copyright information: 'SWAN © Copyright CERN 2016-2018. All rights reserved. Home | Contact | Support | Report a bug'.



The Share tab

- > Users can list which projects...
 - they have shared
 - others have shared with them

The screenshot shows the 'Share' tab in the SWAN application. The top navigation bar includes 'Projects', 'Share' (highlighted), and 'CERNBox'. The main content area is titled 'SWAN > Share' and contains two sections: 'Projects shared with me' and 'Projects shared by me'. Each section has a table of shared projects.

Projects shared with me

NAME	SIZE	SHARED BY	DATE
UP2University Pilot	Empty	jupytercon2	7 minutes ago

Projects shared by me

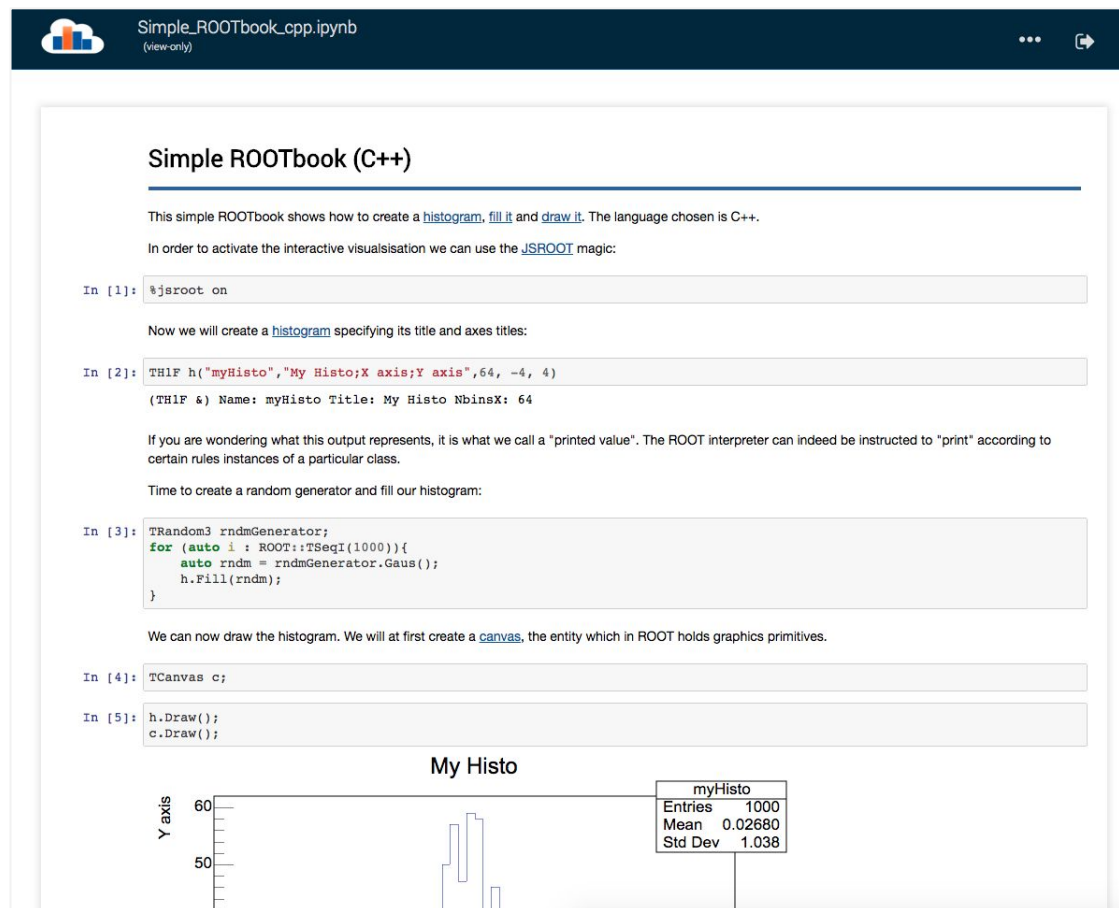
NAME	SHARED WITH	DATE
Higgs Boson discovery	2 people/groups	18 hours ago
Super Real Analysis with TOTEM data	diogo	19 hours ago

SWAN © Copyright CERN 2016-2018. All rights reserved.
Home | Contact | Support | Report a bug



Inspecting a Project

- > Users can inspect shared project contents
 - Browsing of the files
 - Static rendering of notebooks
- > Useful to decide whether to accept or not the shared project





Accepting a Shared Project

- > If accepted, project is cloned to the receiver's CERNBox
 - The receiver will work on his own copy
- > Concurrent editing not (yet?) supported by Jupyter
 - Safer to clone

The screenshot shows the 'Share' tab in the SWAN interface. At the top, there's a navigation bar with 'Projects', 'Share' (active), and 'CERNBox'. Below this, the breadcrumb 'SWAN > Share' is visible. The main content area is divided into two sections: 'Projects shared with me' and 'Projects shared by me'. Each section contains a table of shared projects.

Projects shared with me

NAME	SIZE	SHARED BY	DATE
UP2University Pilot	Empty	jupytercon2	7 minutes ago

Projects shared by me

NAME	SHARED WITH	DATE
Higgs Boson discovery	2 people/groups	18 hours ago
Super Real Analysis with TOTEM data	diogo	19 hours ago

At the bottom of the interface, there is a footer with the text: 'SWAN © Copyright CERN 2016-2018. All rights reserved. Home | Contact | Support | Report a bug'. The CERN logo is also present in the bottom right corner of the interface.



Access to Computing Resources

Spark integration



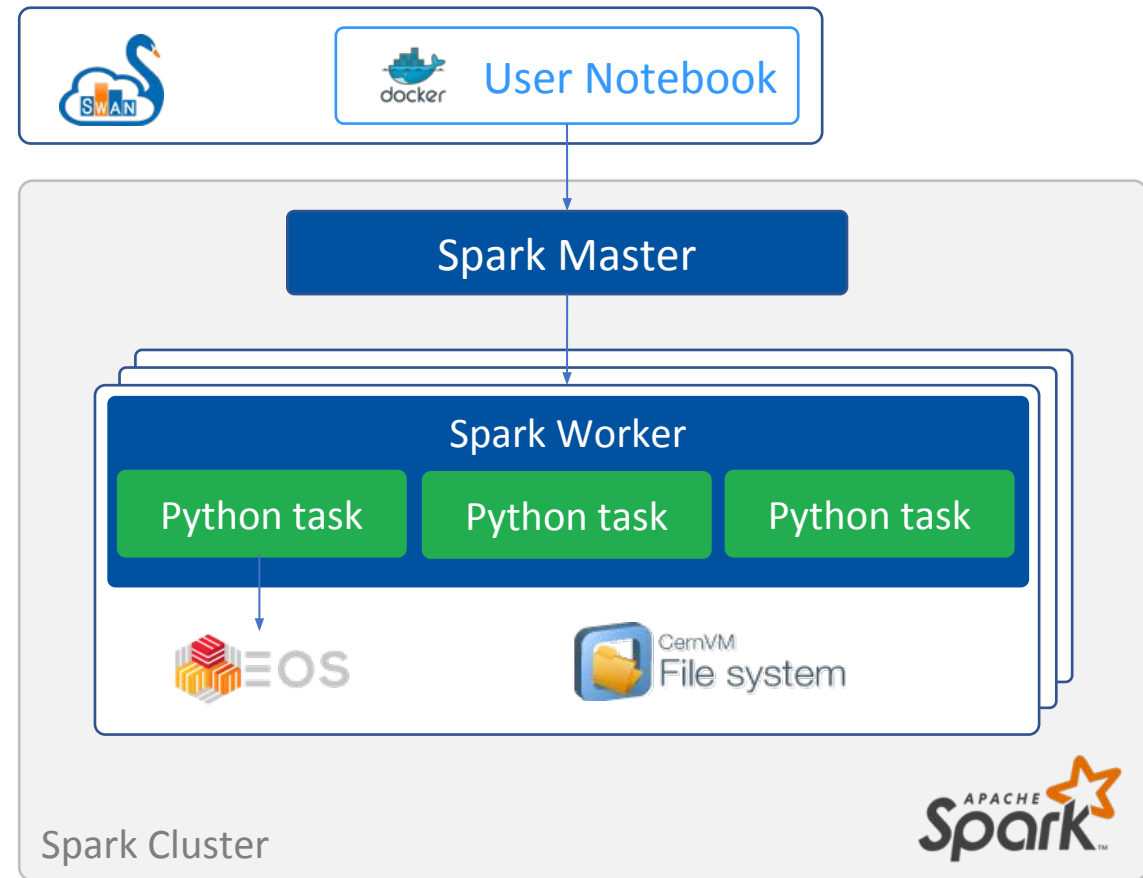
Motivation

- > Beams department at CERN built and operates the LHC
 - Data: monitoring LHC accelerator hardware devices
 - Analysis of logs
 - Different from physics data of the experiments
- > With increase in data, they adopted Spark
 - Spark: a modern cluster management system allowing distributed and interactive data analysis
 - But it was missing a unified platform for analytics visualization



Integration with Spark

- > Connection to CERN Spark Clusters
- > Same environment across platforms
 - User data - EOS
 - Software - CVMFS
- > Graphical Jupyter extensions developed
 - Spark Connector
 - Spark Monitor





Spark > Spark_Simple

Last Checkpoint: a minute ago (autosaved)

FILE EDIT VIEW INSERT CELL KERNEL HELP



Markdown

Simple example with Spark

This notebook illustrates the use of [Spark](#) in [SWAN](#).

The current setup allows to execute [PySpark](#) operations on a local small datasets.

In the future, SWAN users will be able to attach external Spark cluster. Moreover, a Scala Jupyter kernel will be added to use Spark from the notebook.

Import the necessary modules

The pyspark module is available to perform the necessary imports.

```
In [ ]: from pyspark import SparkContext
```

Spark clusters connection



You are now connected

The following variables were instantiated:

- > `sc` = `SparkContext`
- > `spark` = `SparkSession`

[Show/Hide connection logs](#)

Go to the notebook



Spark Monitor

- > Allows to monitor Spark jobs from within the notebook
 - Interactive display appears **automatically** in the output of the cell
- > Multiple views
 - Progress bars for jobs and stages
 - Task timeline
 - Resource utilization



```
In [ ]: def f(x):  
        global a  
        a+=x  
        RDD9.foreach(f)  
        RDD9.foreach(f)  
        print(a.value)  
        #Display should appear automatically
```



Access to Computing Resources

Batch jobs



Connecting More Resources

> Ongoing effort: submit **batch jobs** from the notebook

- Monitoring display
- Jobs tab



Projects Share **Jobs** CERNBox

Select jobs to perform action on them.

Job ID Jobs per page

☐ 0	Job ID	Job Name	Backend	Application	File Name	Status	Submission Time	Runtime
☐	3		Condor	Executable		SUBMITTING	Jul 24th, 3:26 pm	-
☐	2		Condor	Executable		NEW	-	-
☐	1		Localhost	Executable		NEW	-	-
☐	0		Localhost	Executable		COMPLETED	Jul 24th, 3:21 pm	00 seconds

In [6]: %%ganga
j=ganga.Job()
j.submit()

Backend: LOCALHOST Application: EXECUTABLE Splitter: None 0 SUBJOBS

Job ID	Job Name	Status	Subjobs	Submission Time	Runtime
0		COMPLETED	No Subjobs	Jul 24th, 3:21 pm	00 seconds



Education, Outreach



Notebooks for Teaching

- > SWAN and notebooks are very useful for teaching
 - Teachers prepare exercises in the form of notebooks
 - Students complete and run the exercises
- > Used frequently as support for:
 - CERN Summer student courses: ~150 students, data analysis with ROOT
 - CERN School of Computing exercises: ~70 students, parallelism
 - CERN ATLAS PhD student courses: ~50 students, declarative data analysis
 - Machine learning tutorials at CERN





Science Box: SWAN on Premises

- > UP2University European Project
 - Bridge the gap between secondary schools, higher education and the research domain
 - Partner universities (OU, UROMA, NTUA, ...), pilot schools
 - <http://up2university.eu>
- > SWAN used by students to learn physics and other sciences
 - Let them use the very same tools & services used by scientists at CERN
 - Integrating Jupyter in teaching environment
 - Complete SWAN package, with CERNBox, EOS and CVMFS

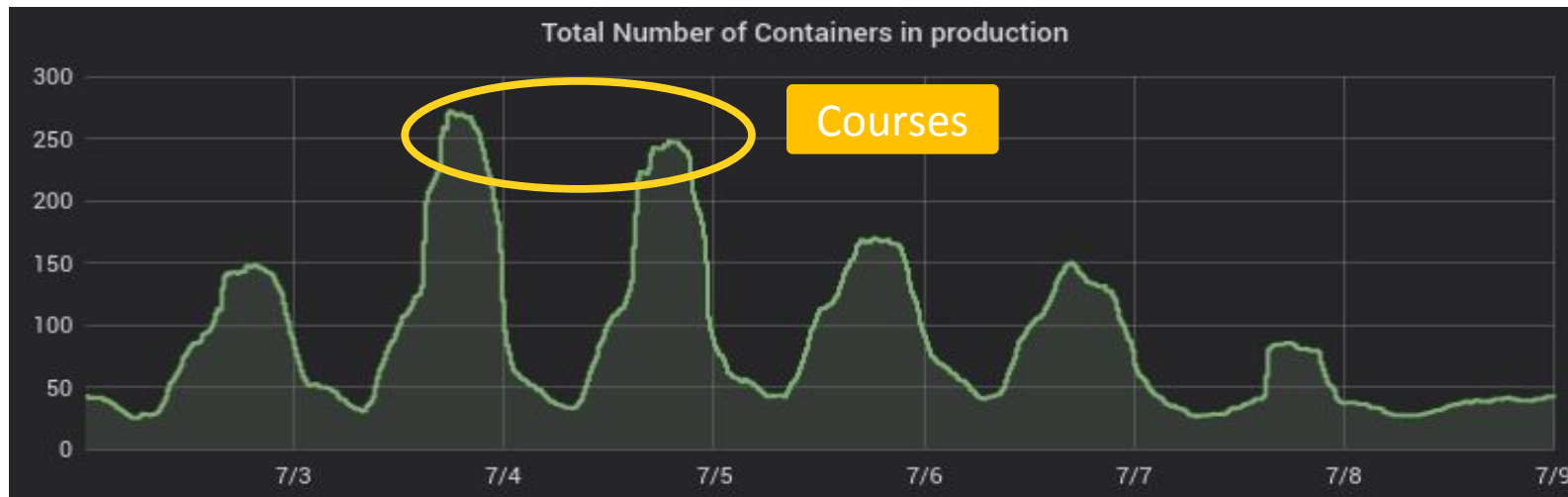


SWAN Users



Stats

- > **~200 user sessions** a day on average
 - Users doubled this year with new SWAN interface
- > Spark cluster connection: 15 – 20 % of users
 - SWAN as entry point for accessing computational resources





User community

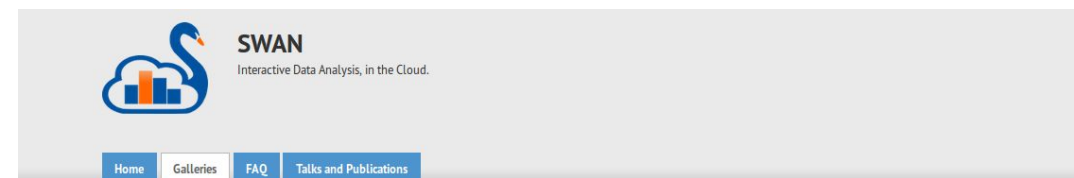
> SWAN development is guided by our user community

- New features (libs, kernels, ...) are requested by users from their real usage needs

> Gallery of examples

- Made in collaboration with our users
- Almost 50 notebooks in 7 categories

Example notebooks at cern.ch/swan



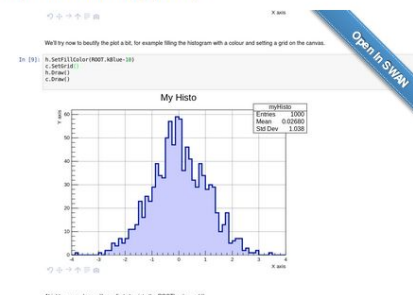
Basic Examples

This is a gallery of basic example notebooks: click on the images to inspect the underlying document, open in SWAN the single notebooks or the full git repository!

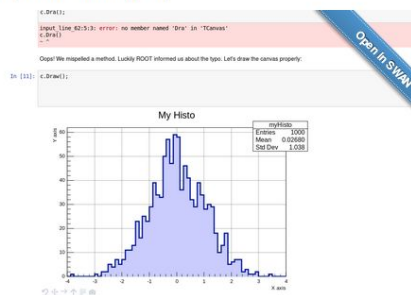
Open in  SWAN

Many of the notebooks are ROOTbooks, based on the ROOT framework. To know more about ROOT, visit root.cern.ch.

Simple ROOTbook (Python)



Simple ROOTbook (C++)



Simple Fitting



Simple I/O

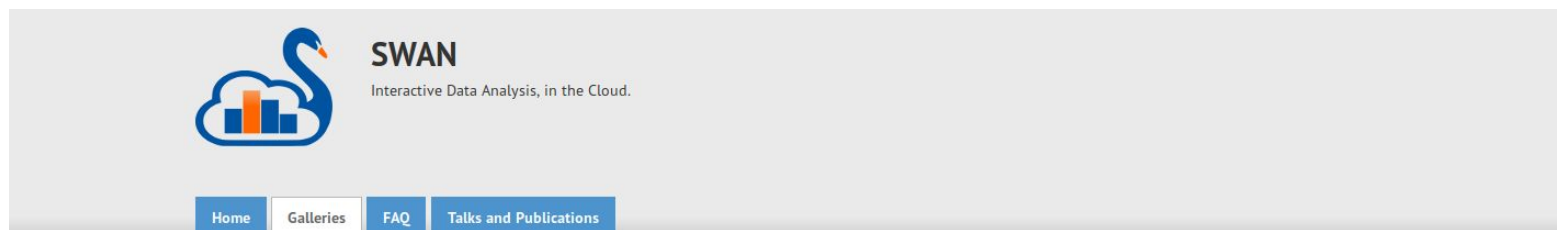


Access with only a click





Education, Too



Basic ROOT Primer Accelerator Complex FCC Beam Dynamics Machine Learning Apache Spark Outreach

Outreach

Reach out with SWAN! This section collect a series of outreach efforts involving SWAN.

Particle open data teaching (Hiukkasfysiikan avoin data opetuksessa)

Esim-pseudorapideetti-mittatarkkuus (autosaved) Control Panel Logout

File Edit View Insert Cell Kernel Help Python 2

Markdown CellToolbar

Lähdetäänpä tutkimaan!

Lähdetään seuraavaksi tarkastelemaan, miten pseudorapideetin vaikutus mittatarkkuuteen voidaan havaita CMS-ilmaisimen keräämän oikean datan avulla. Käytetään CMS:n vuodelta 2011 kerättyä dataa [1], josta on valittu 10851 törmäystapahtumaa (events) tiedostoon "Zmumu_Run2011A_massoilla.csv". (Karsinta on suoritettu koodilla, joka on avoimesti saatavilla osoitteessa <https://github.com/tpmccauley/dimuon-filter>.)

Tiedostoon on valittu niitä törmäystapahtumia, joissa syntynyt Z-bosoni on hajonnut myoniksi μ^- ja antimyoniksi μ^+ . Ilmaisim on havainnut nämä myonit ja mitannut niiden liikemäärät.

$Z^0 \rightarrow \mu^- \mu^+$

SWAN, the CERN Service for Web based ANALysis, is not only made for analysis of scientific data but also the ideal platform for outreach. Paavo Rikkilä (CMS) put together an introductory course about experimental HEP for future high school teachers. The result is great: check it out in SWAN!

Open in SWAN

Github repository: <https://github.com/cmsopendata-finland/kurssimateriaali>

Example notebooks at cern.ch/swan

- Finnish High School program





Integrate Software Packages



ROOT 6.15/01
Reference Guide

[ROOT Home](#) [Main Page](#) [Tutorials](#) [Functional Parts](#) [Namespaces ▾](#) [All Classes ▾](#) [Files ▾](#) [Release Notes](#)

Data Frame tutorials

Tutorials

Files

These examples show the functionalities of the RDataFrame class.

Files

file [df001_introduction.C](#)

[View](#) [Notebook](#) [Open in SWAN](#) This tutorial illustrates the basic features of the RDataFrame class, a utility which allows to interact with data stored in TTrees following a functional-chain like approach.

file [df001_introduction.py](#)

[View](#) [Notebook](#) [Open in SWAN](#) This tutorial illustrates the basic features of the RDataFrame class, a utility which allows to interact with data stored in TTrees following a functional-chain like approach.

file [df002_dataModel.C](#)

[View](#) [Notebook](#) [Open in SWAN](#) This tutorial shows the possibility to use data models which are more complex than flat ntuples with RDataFrame

file [df002_dataModel.py](#)

[View](#) [Notebook](#) [Open in SWAN](#) This tutorial shows the possibility to use data models which are more complex than flat ntuples with RDataFrame

file [df003_profiles.C](#)

[View](#) [Notebook](#) [Open in SWAN](#) This tutorial illustrates how to use TProfiles in combination with the RDataFrame.

file [df003_profiles.py](#)

[View](#) [Notebook](#) [Open in SWAN](#) This tutorial illustrates how to use TProfiles in combination with the RDataFrame.

Example notebooks at **root.cern**

Looking ahead



Future work/challenges

- > Move to Jupyterlab
 - Porting the current extensions
 - ROOT is already migrating its graphics to JavaScript only
 - **Ultimate blur between local/cloud resources and apps**
- > New architecture made to scale
 - Based on Kubernetes, leveraging CERN Container Service
- > Exploitation of GPUs
 - ML found its place in HEP
 - Speed up computation of GPU-ready libraries (e.g. TensorFlow)
- > Moving beyond LHC
 - New challenges from HL-LHC
 - Even small dataset will not fit inside users laptops: need R&D - Analysis Facilities?

Where to find us



Where to find us

> Contacts

- swan-talk@cern.ch
- <http://cern.ch/swan>

> Repository

- <https://github.com/swan-cern/>

> Science Box: SWAN distribution **deployable on premises**

- <https://cern.ch/sciencebox>
- Includes all SWAN components: CERNBox/EOS, CVMFS, JupyterHub
- Deployable through Kubernetes or docker-compose

Conclusions



Conclusion

- > We successfully integrated CERN computing services with Jupyter
 - Increase value of existing services' portfolio (storage, batch, auth)
 - Successfully boosting productivity of scientists
 - Jupyter became a new entry-point to our computing resources and data
- > Successful model for teaching activities
 - Tutorials for software, data analysis, ML, Spark...
 - European project focused on higher education
- > Blending desktop and web application can help reach more users
 - Can Jupyterlab help?