



# ROOTaaS

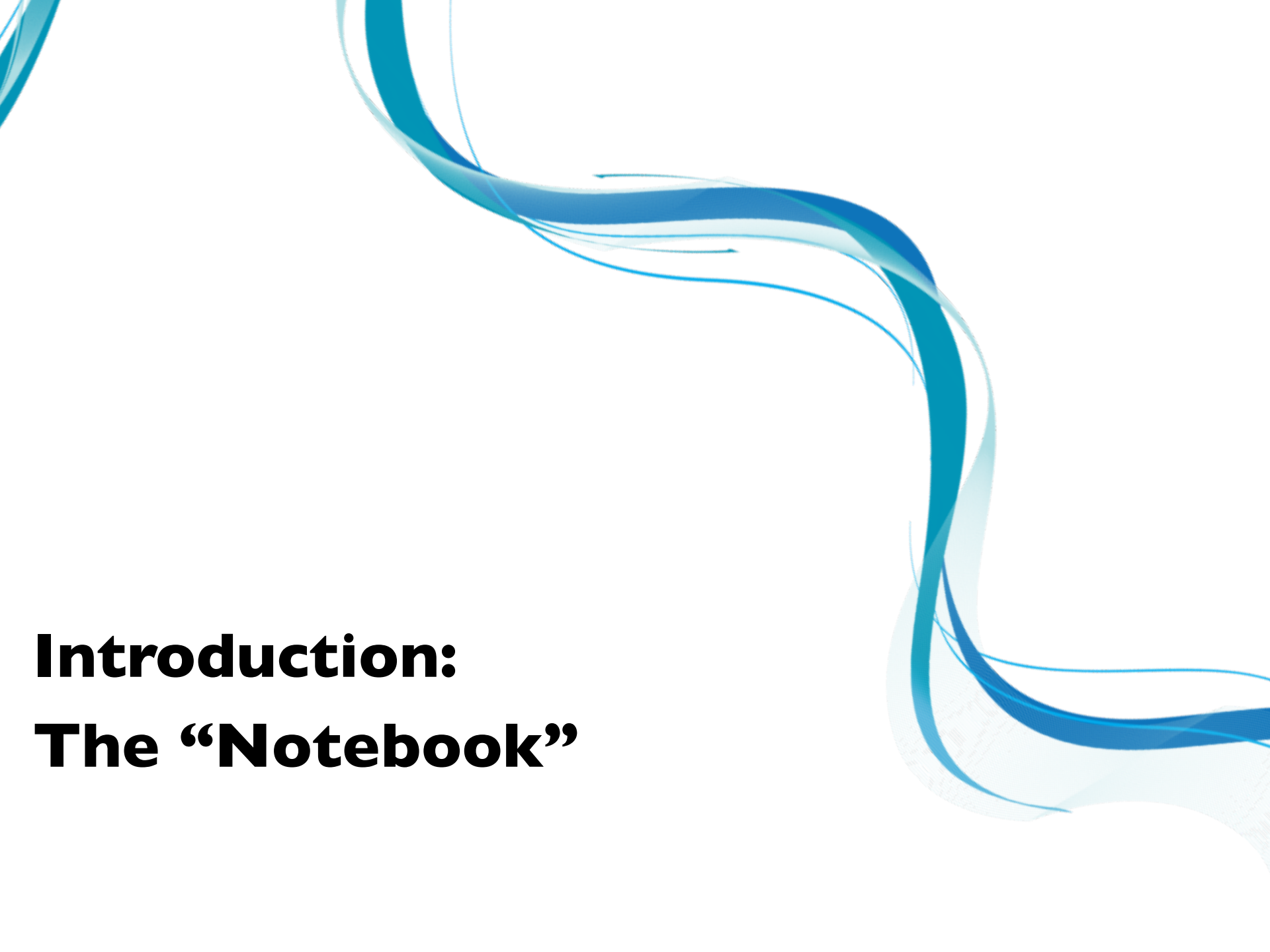
## ROOT as a Service

P. Mato, D. Piparo, E. Tejedor, for the ROOT Team

Alice Offline week


*24/11/2015*



The background features several flowing, wavy lines in various shades of blue, ranging from light sky blue to deep navy blue. These lines originate from the top left and curve downwards and to the right, creating a sense of movement and depth. The lines are layered, with some appearing more prominent than others, giving the design a dynamic and modern feel.

# **Introduction: The “Notebook”**

A web-based interactive computing interface and platform that combines code, equations, text and visualisations.



Many supported languages: Python, Haskell, Julia... One generally speaks about a “kernel” for a specific language

In a nutshell: an “interactive shell opened within the browser”

Also called:  
“Jupyter Notebook” or  
“IPython Notebook”

<http://www.jupyter.org>



# Start a Notebook in a Laptop

```
$ ipython notebook
```

This command:

1. Starts a local notebook server
2. Connects to it via the browser

ROOT Notebook Functionalities

Control Panel Logout

File Edit View Insert Cell Kernel Help Python 2

Code Cell Toolbar: None

# Welcome to the Notebook Technology

This is a markdown cell. You can add LaTeX code:  $\sum_{n=-\infty}^{\infty} |x(n)|^2$

The image shows a web browser window displaying a Jupyter Notebook. The browser's address bar shows 'localhost:8888/tree'. The notebook's header includes a 'ROOT' logo, the title 'Notebook Functionalities', and buttons for 'Control Panel' and 'Logout'. Below the header is a menu bar with 'File', 'Edit', 'View', 'Insert', 'Cell', 'Kernel', and 'Help'. A toolbar contains icons for file operations and cell execution, along with a 'Code' dropdown menu and a 'Cell Toolbar: None' dropdown. The main content area features a large heading 'Welcome to the Notebook Technology' and a paragraph explaining that it is a markdown cell. It includes a LaTeX equation: 
$$\sum_{n=-\infty}^{\infty} |x(n)|^2$$
. Below this is a code cell with the following Python code: 

```
In [1]: def thisFunction():  
        return 42
```

Code

localhost:8888/tree

# ROOT Notebook Functionalities

Control Panel Logout

File Edit View Insert Cell Kernel Help Python 2

Code Cell Toolbar: None

## Welcome to the Notebook Technology

This is a markdown cell. You can add LaTeX code:  $\sum_{n=-\infty}^{\infty} |x(n)|^2$

```
In [1]: def thisFunction():  
        return 42
```

This is a notebook in Python

Code

Home \* +

localhost:8888/tree

ROOT Notebook Functionalities

Control Panel Logout

File Edit View Insert Cell Kernel Help Python 2

Code Cell Toolbar: None

# Welcome to the Notebook Technology

This is a markdown cell. You can add LaTeX code:  $\sum_{n=-\infty}^{\infty} |x(n)|^2$

```
In [1]: def thisFunction():
        return 42
```

```
In [2]: thisFunction()
```

```
Out[2]: 42
```

**Code**



```
In [1]: def thisFunction():  
        return 42
```

```
In [2]: thisFunction()
```

```
Out[2]: 42
```

```
In [3]: %%bash  
        curl rootaasdemo.web.cern.ch/rootaasdemo/SaaSfee.jpg \  
        > SF.jpg
```



We can invoke commands in the shell...

**Shell Commands**

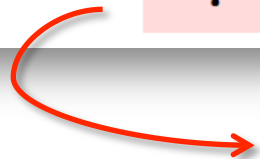
```
In [1]: def thisFunction():  
        return 42
```

```
In [2]: thisFunction()
```

```
Out[2]: 42
```

```
In [3]: %%bash  
        curl rootasdemo.web.cern.ch/rootasdemo/SaaSfee.jpg \  
> SF.jpg
```

```
% Total      % Received % Xferd  Average Speed   Time  
Time        Time       Current           Dload  Upload   Total  
Spent       Left      Speed  
100 128k  100 128k    0      0 2731k      0  --:--:--  
--:--:--  --:--:-- 2787k
```



... and get their output

**Shell Commands**

```
In [1]: def thisFunction():  
        return 42
```

```
In [2]: thisFunction()
```

```
Out[2]: 42
```

```
In [3]: %%bash  
        curl rootasdemo.web.cern.ch/rootasdemo/SaaSfee.jpg \  
> SF.jpg
```

```
% Total      % Received % Xferd  Average Speed   Time  
Time         Time    Current                Dload  Upload   Total  
Spent       Left  Speed  
100 128k  100 128k    0      0 2731k      0  --:--:--  
--:--:--  --:--:-- 2787k
```

```
In [4]: from IPython.display import Image  
        Image(filename="./SF.jpg",width=225)
```

```
In [1]: def thisFunction():  
        return 42
```

```
In [2]: thisFunction()
```

```
Out[2]: 42
```

```
In [3]: %%bash  
        curl rootaasdemo.web.cern.ch/rootaasdemo/SaaSFee.jpg \  
> SF.jpg
```

```
% Total      % Received % Xferd  Average Speed   Time  
Time        Time      Current                Dload  Upload   Total  
Spent       Left     Speed  
100 128k  100 128k    0      0 2731k      0  --:--:--  
--:--:-- --:--:-- 2787k
```

```
In [4]: from IPython.display import Image  
        Image(filename="./SF.jpg",width=225)
```

```
Out[4]:
```



Images

In [1]:

**In a browser**

In [2]: `thisFunction()`

**Text and Formulas**

Out[2]: 42

**Code**

In [3]: `%%bash  
curl rootaasdemo.web.cern.ch/rootaasdemo/SaaSfee.jpg \  
> SF.jpg`

**Shell Commands**

% Total Time	% Received Time	Current Speed	Dload	Upload	Time
Spent	Left	Speed			Total
100 128k	100 128k	0 0 2731k		0	--:--:--
--:--:--	--:--:--	2787k			

In [4]: `from IPython.display import Image  
Image(filename='./SF.jpg',width=225)`

Out[4]:



**Images**



- The ROOTaaS project
- Integration of ROOT with the Notebook technology
  - Programming model and usability for data analysis
- A new service: ROOT notebooks within the CERN IT services' portfolio
  - Spotlight on storage

## Data mining “as a service”

Interface: Notebooks

Goals:

- Analyse data via a web interface
  - Calculations, input and results “in the cloud”
- Use **ROOT** only with a web browser
  - Platform independent ROOT based data analysis
- **Allow easy sharing of scientific results:** plots, data, code
  - Storage is crucial
- **Simplify teaching** of data processing and programming
- **C++, Python** and other languages
  - Also interfaced to ROOT

pandas  
 $y_{it} = \beta x_{it} + \mu_i + \epsilon_{it}$





# Integration of ROOT with Notebooks

ROOT

iPyROOT  
(ROOT-Notebooks integration)





Terminal

Control Panel

Logout

Files Running Clusters

A Choice of Kernels

Select items to perform actions on them.

Upload New

Home icon

- PresentationNotebooks
- cernbox
- HowTo\_ROOT-Notebooks.ipynb
- HowTo\_ROOT-Notebooks\_Long.ipynb
- My First Notebook.ipynb
- Untitled.ipynb

- Text File
- Folder
- Terminal
- Notebooks
- Python 2
- Python 3
- ROOT Prompt



File Edit View Insert Cell Kernel Help Python 2

Code Cell Toolbar: None

```
In [1]: import ROOT # This triggers the integration layer
```

```
Welcome to ROOTaaS 6.05/01
```

File

Edit

View

Insert

Cell

Kernel

Help



Python 2



Code

Cell Toolbar: None

```
In [1]: import ROOT # This triggers the integration layer
```

```
Welcome to ROOTaaS 6.05/01
```

```
In [ ]: %%cpp
        auto myHisto = TH|
```

```
TH1
TH1C
TH1D
TH1F
TH1I
TH1K
TH1S
TH2
TH2C
TH2D
```

**C++ Cells in Python  
Notebooks**

**ROOT Tab  
Completion**

File Edit View Insert Cell Kernel Help Python 2

Code Cell Toolbar: None

```
In [1]: import ROOT # This triggers the integration layer
Welcome to ROOTaaS 6.05/01
```

```
In [2]: %%cpp
auto myHisto = TH1F("h", "MyData;X;Y", 64, -4, 4); // C++11
```

File Edit View Insert Cell Kernel Help Python 2

Code Cell Toolbar: None

```
In [1]: import ROOT # This triggers the integration layer
```

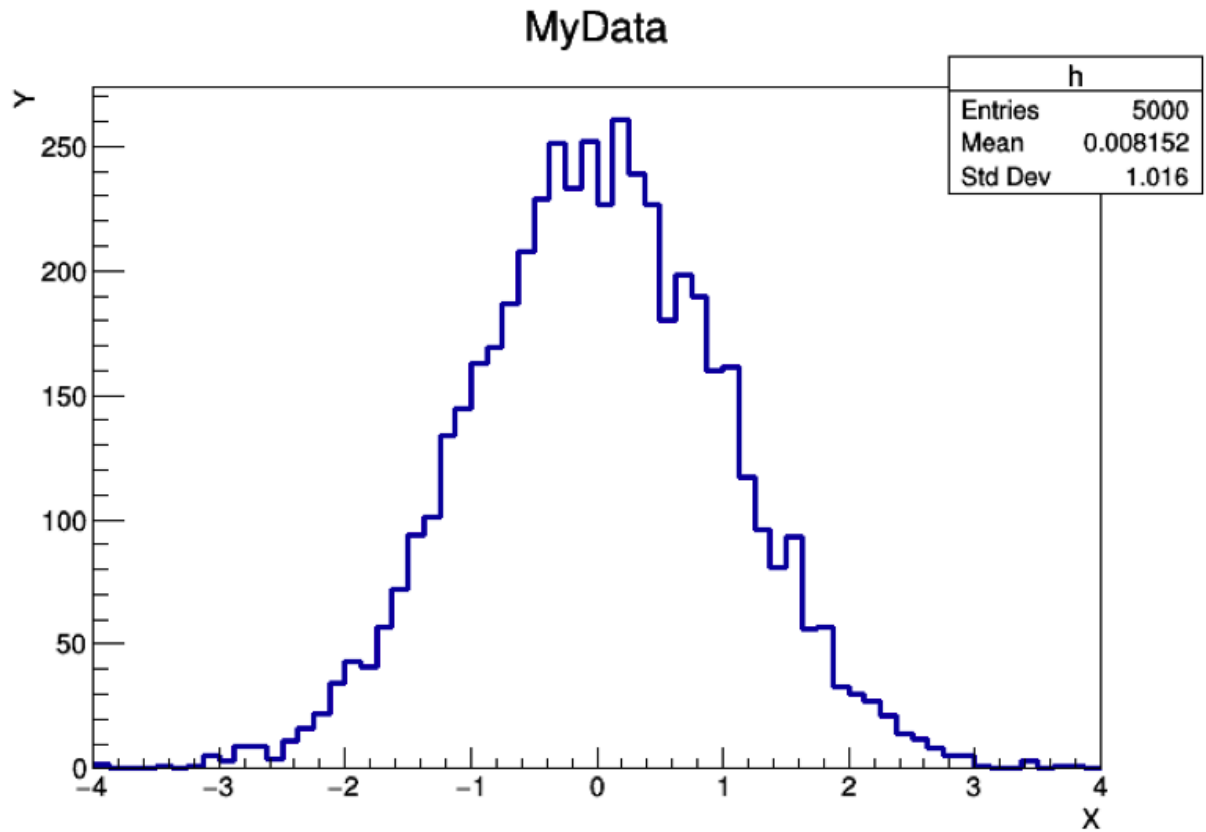
```
Welcome to ROOTaaS 6.05/01
```

```
In [2]: %%cpp
auto myHisto = TH1F("h", "MyData;X;Y", 64, -4, 4); // C++11
```

```
In [3]: h = ROOT.myHisto # Find the variable back in Python!
h.FillRandom("gaus")
c = ROOT.TCanvas()
h.Draw()
c.Draw()
```

**C++-Python  
Interoperability**

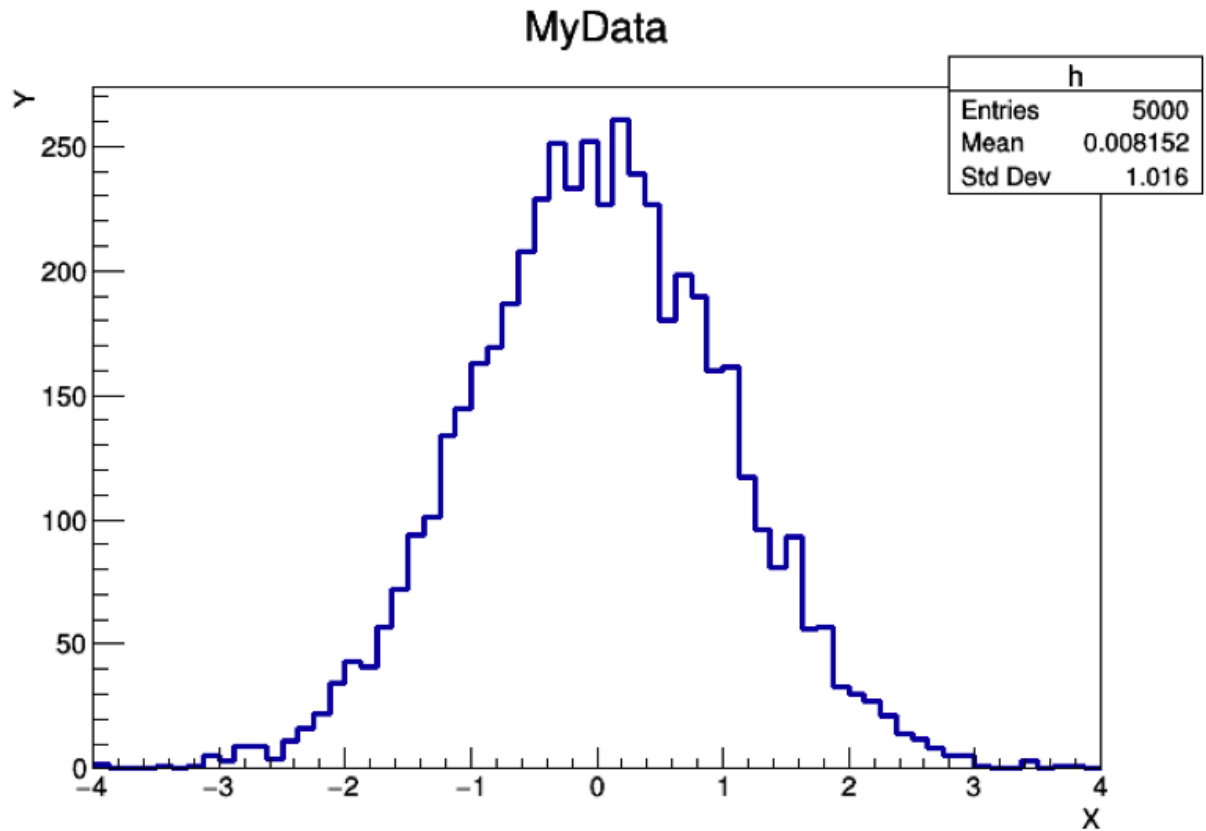
c.Draw()



**Seamless  
display of  
graphics**

c.Draw()

Syntax  
Highlighting



```
In [4]: %%cpp -d
double myG(double* x, double* par){
    auto res = (x[0]-par[1])/par[2];
    auto e = -.5 * res * res;
    return par[0] * exp(e); // declare function
}
```

```
In [4]: %%cpp -d
        double myG(double* x, double* par){
            auto res = (x[0]-par[1])/par[2];
            auto e = -.5 * res * res;
            return par[0] * exp(e); // declare function
        }
```

```
In [5]: f = ROOT.TF1("myGf",ROOT.myG,-5,5,3)
        f.SetParameters(200,0,1);f.SetParNames("N","mu","sigma")
        fr = ROOT.h.Fit(f,"S") # Capture printouts
```



```
In [4]: %%cpp -d
double myG(double* x, double* par){
    auto res = (x[0]-par[1])/par[2];
    auto e = -.5 * res * res;
    return par[0] * exp(e); // declare function
}
```

```
In [5]: f = ROOT.TF1("myGf",ROOT.myG,-5,5,3)
f.SetParameters(200,0,1);f.SetParNames("N","mu","sigma")
fr = ROOT.h.Fit(f,"S") # Capture printouts
```

```
FCN=47.4997 FROM MIGRAD      STATUS=CONVERGED      69 CALLS      70 TO
TAL
```

```
EDM=2.04372e-09      STRATEGY= 1      ERROR MATRIX ACC
```

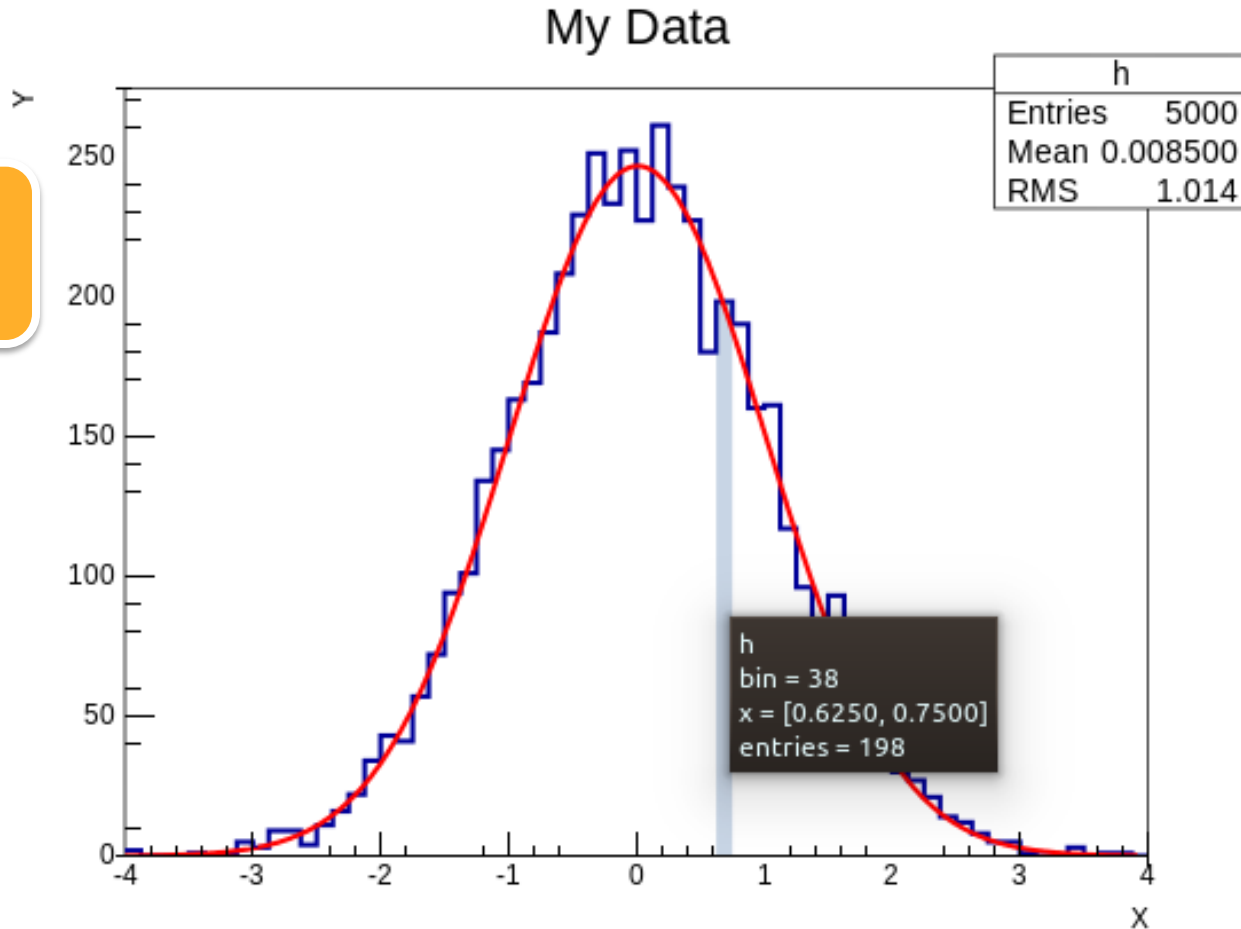
```
URATE
```

EXT	PARAMETER	STEP	FIRST		
NO.	NAME	VALUE	ERROR	SIZE	DERIVATIVE
1	N	2.46469e+02	4.31493e+00	1.19092e-02	-5.38026e-06
2	mu	1.04793e-02	1.43576e-02	4.87640e-05	4.15093e-03
3	sigma	1.00316e+00	1.03818e-02	2.86307e-05	-2.55310e-04

```
1 N 2.46469e+02 4.31493e+00 1.19092e-02 -5.38026e-06
2 mu 1.04793e-02 1.43576e-02 4.87640e-05 4.15093e-03
3 sigma 1.00316e+00 1.03818e-02 2.86307e-05 -2.55310e-04
```

```
In [6]: ROOT.enableJSVis() # Not active by default yet!
c.Draw()
ROOT.disableJSVis()
```

**JSROOT  
Visualisation**



```
In [10]: %%cpp -a
// Create dictionaries, a library and load it
#include <string>
class myClass{
public:
    myClass(){};
    myClass(const char* name):fName(name){};
    const char* getName() const{return fName.c_str();}
private:
    std::string fName = "";
};
```

```
Info in <TUnixSystem::ACLiC>: creating shared library
/home/rwl5u099/PresentationNotebooks/e9c1711f_C.so
```

```
In [12]: myObj = ROOT.myClass("theName")
ofile = ROOT.TFile("ofile.root", "recreate")
h.Write()
ofile.WriteObjectAny(myObj, "myClass", myObj.getName())
ofile.Close()
```

```
In [13]: %%bash
rootls -l ofile.root
```

```
TH1F      Sep 11 15:29  h          "MyData"
myClass   Sep 11 15:29  theName   "object title"
```

**All the power of ROOT: Dictionaries, I/O, runtime loading of libraries**



“import ROOT” turns on all notebook goodies

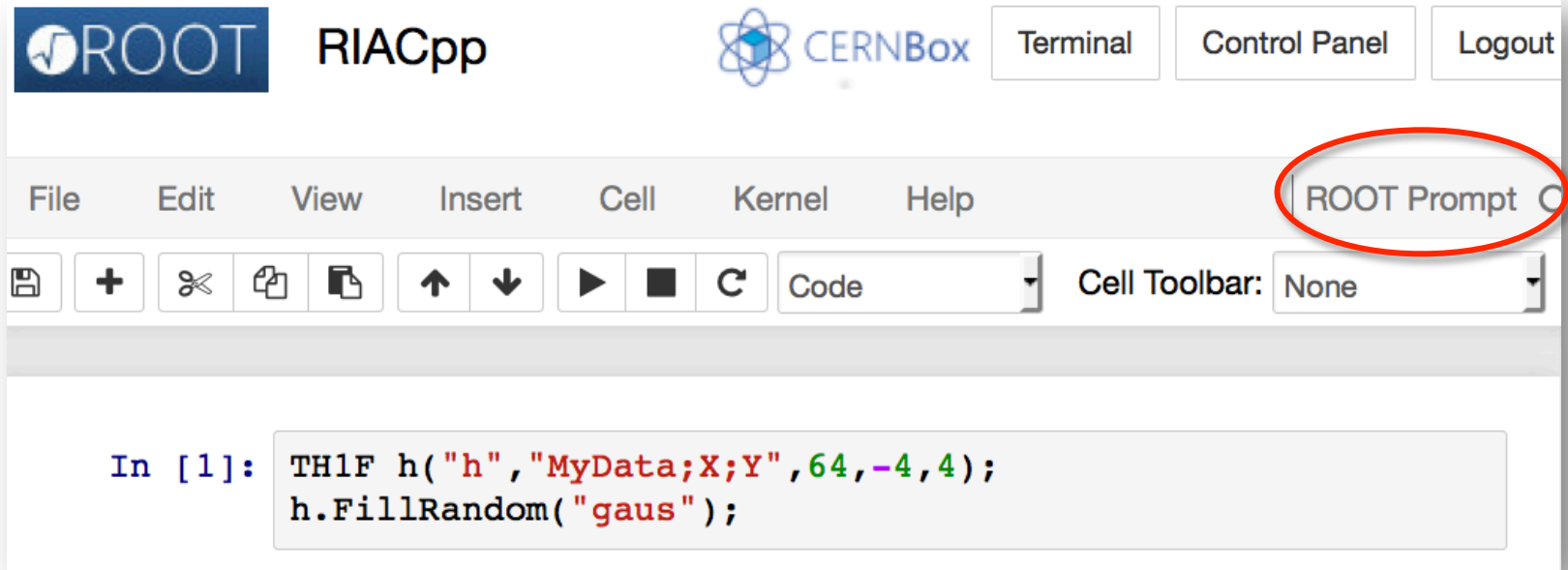
- Tab-completion
- C++ cells, ACLiC
- Display of graphics
- Syntax highlighting

All the power of ROOT and the ROOT Python bindings, PyROOT, are there

**Like before, but better**

Select items to perform actions on them.

[Upload](#)[New ▾](#)[PresentationNotebooks](#)[cernbox](#)[HowTo\\_ROOT-Notebooks.ipynb](#)[HowTo\\_ROOT-Notebooks\\_Long.ipynb](#)[My First Notebook.ipynb](#)[Untitled.ipynb](#)[Text File](#)[Folder](#)[Terminal](#)[Notebooks](#)[Python 2](#)[Python 3](#)[ROOT Prompt](#)



ROOT RIACpp CERNBox Terminal Control Panel Logout

File Edit View Insert Cell Kernel Help ROOT Prompt

Code Cell Toolbar: None

```
In [1]: TH1F h("h", "MyData;X;Y", 64, -4, 4);  
        h.FillRandom("gaus");
```



RIACpp



Terminal

Control Panel

Logout

File

Edit

View

Insert

Cell

Kernel

Help

ROOT Prompt



Code

Cell Toolbar:

None

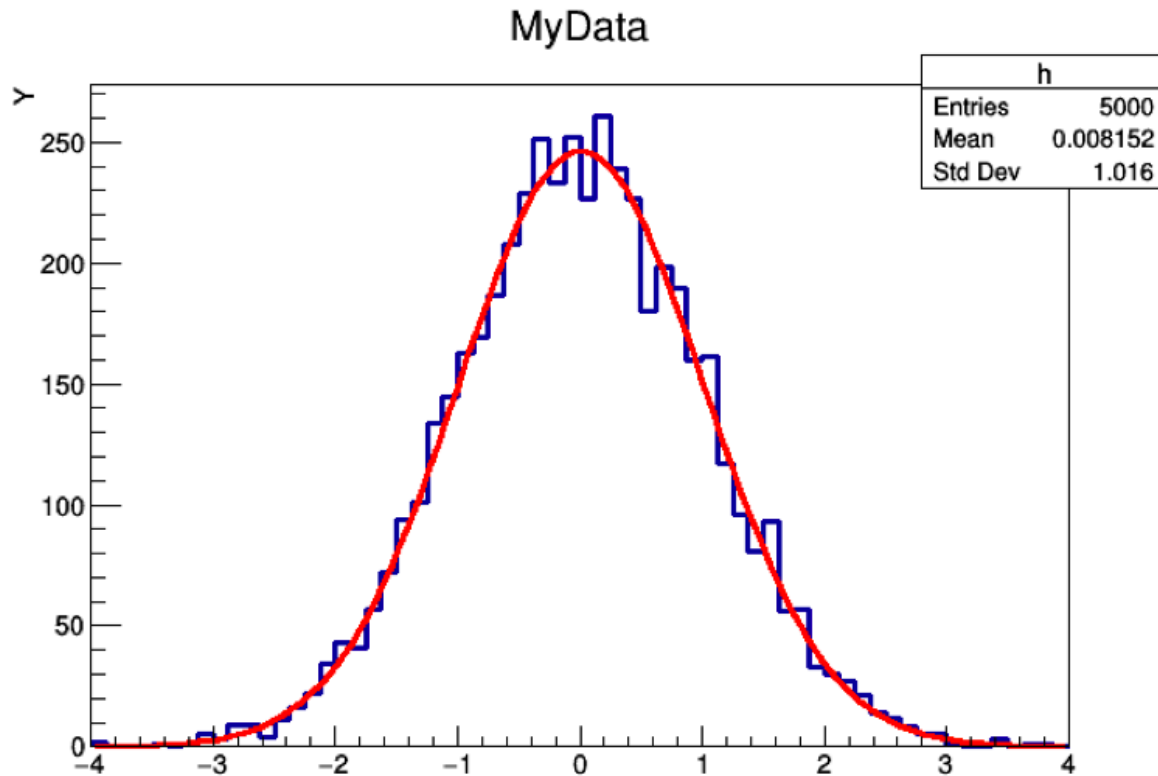
```
In [1]: TH1F h("h", "MyData;X;Y", 64, -4, 4);  
        h.FillRandom("gaus");
```

```
In [2]: .cpp -d  
        double myG(double* x, double* par){  
            auto res = (x[0]-par[1])/par[2];  
            auto e = -.5 * res * res;  
            return par[0] * exp(e); // declare function  
        }
```

```
In [4]: TF1 f("myGf", myG, -5, 5, 3);  
        f.SetParameters(200, 0, 1);  
        f.SetParNames("N", "mu", "sigma");  
        auto fr = h.Fit(&f, "S");  
        TCanvas c;  
        h.Draw(); c.Draw();
```

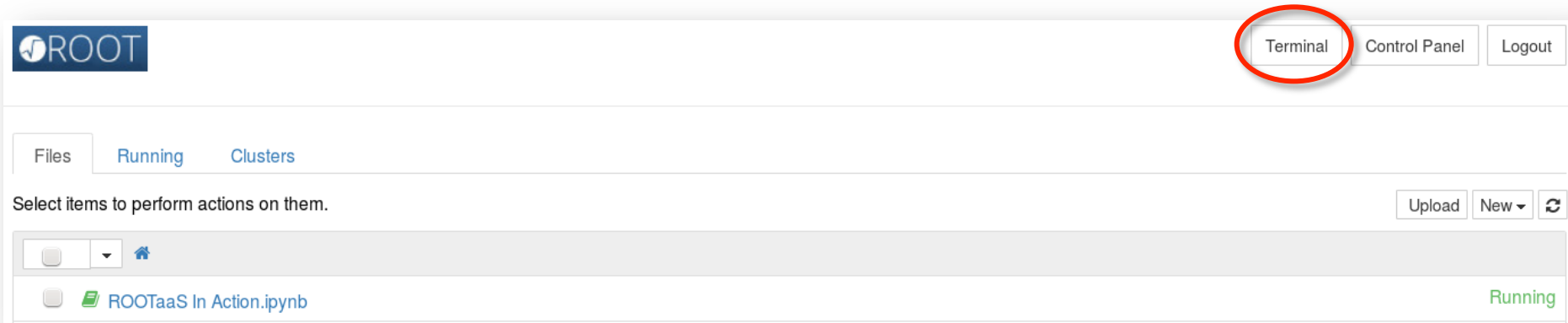
NO.	NAME	VALUE	ERROR	SIZE	DERIVATIVE
1	N	2.46469e+02	4.31493e+00	1.19092e-02	-5.38026e-06
2	mu	1.04793e-02	1.43576e-02	4.87640e-05	4.15093e-03
3	sigma	1.00316e+00	1.03818e-02	2.86307e-05	-2.55310e-04

Info in <TCanvas::MakeDefCanvas>: created default TCanvas with name c1

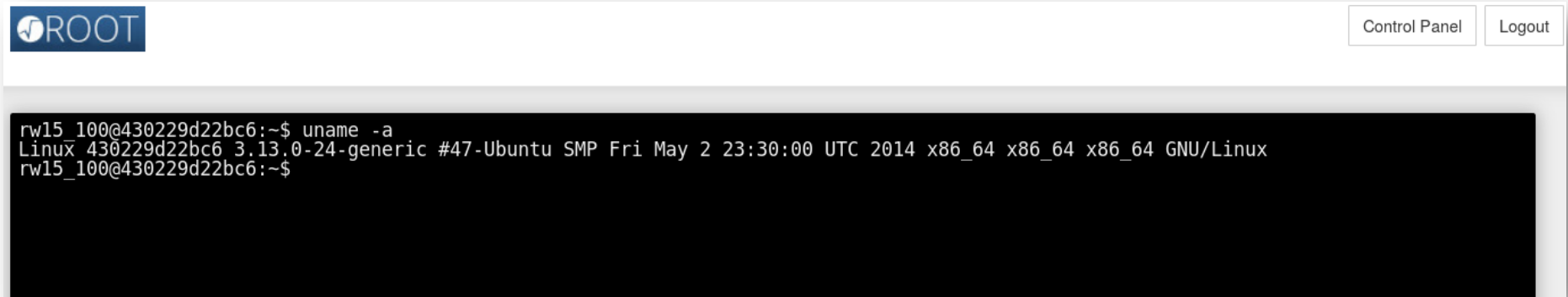




Make terminal available with one click!



The screenshot shows the ROOT web interface. At the top left is the ROOT logo. At the top right are three buttons: 'Terminal', 'Control Panel', and 'Logout'. The 'Terminal' button is circled in red. Below the header, there are three tabs: 'Files', 'Running', and 'Clusters'. Under 'Running', there is a text prompt 'Select items to perform actions on them.' and three buttons: 'Upload', 'New', and a refresh icon. Below this is a list of running jobs. The first job is 'ROOTaaS In Action.ipynb', which is in a 'Running' state, indicated by a green status label on the right.



The screenshot shows the ROOT terminal window. At the top left is the ROOT logo. At the top right are two buttons: 'Control Panel' and 'Logout'. The terminal window is black with white text. The text shows the output of the 'uname -a' command:

```
rw15_100@430229d22bc6:~$ uname -a
Linux 430229d22bc6 3.13.0-24-generic #47-Ubuntu SMP Fri May 2 23:30:00 UTC 2014 x86_64 x86_64 x86_64 GNU/Linux
rw15_100@430229d22bc6:~$
```



- **Examples** (15 already) from the *new ROOT Tutorials* can be found at:

<https://root.cern.ch/code-examples#notebooks>

both in Python and C++ (and mixed!)

- **“Howto”s**

“How To use ROOT in a Notebook” instructions and

“How To activate a *ROOT Prompt* kernel in Your IPython Notebook”

<https://root.cern.ch/howtos#Language%20Bindings>

- Notebook technology also adopted for writing the most “pragmatic” HowTos: <https://root.cern.ch/howtos>

Install ROOT, install IPython notebooks and...

```
$ root --notebook
```

This command:

1. Starts a local notebook server
2. Connects to it via the browser

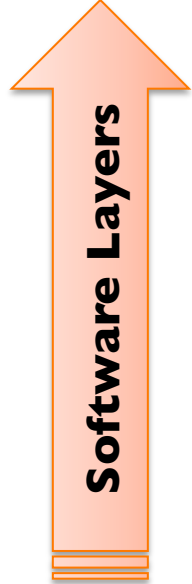
**Provides a C++ notebook  
and the rest of ROOT  
notebook goodies**



Delivered in ROOT 6.05, improved for 6.06 (about to be released)

Exciting future developments ahead! E. g.

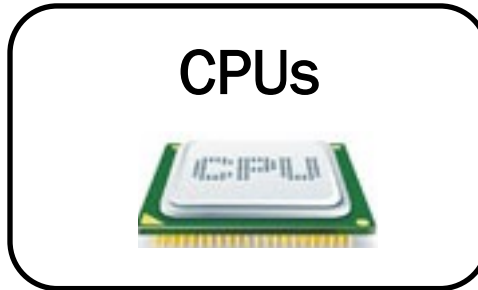
- Better integration of ROOT C++ notebooks
- Improved R support



ROOT

iPyROOT  
(ROOT-Notebooks integration)

JupyterHub +  
CERN Add-ons




# ROOTaaS and CERN services' Portfolio





# Integration With CERN's Ecosystem

- Centrally provide ROOT as a Service
- Authentication with **CERN credentials** (*IT Sec*)
- Infrastructure: **virtual machines** in OpenStack Cloud (*IT-OIS*)
- **Software distribution** (*PH-SFT, IT-PES*): CVMFS 
- **Storage access** (*IT-DSS*): CERNBox, EOS
  - All data potentially available!
- Synergy with **document sharing** (*IT-CIS*)
  - Notebook visualiser available in the next Indico release



Fruitful collaboration with IT-DSS group to make this happen!



- Jupyter notebooks can be uploaded to Indico: automatic rendering
- See this very contribution in indico for more details!

10:45 **ROOT as a service 25'**

Speakers: Danilo Piparo (CERN), Enric Tejedor Saavedra (CERN)



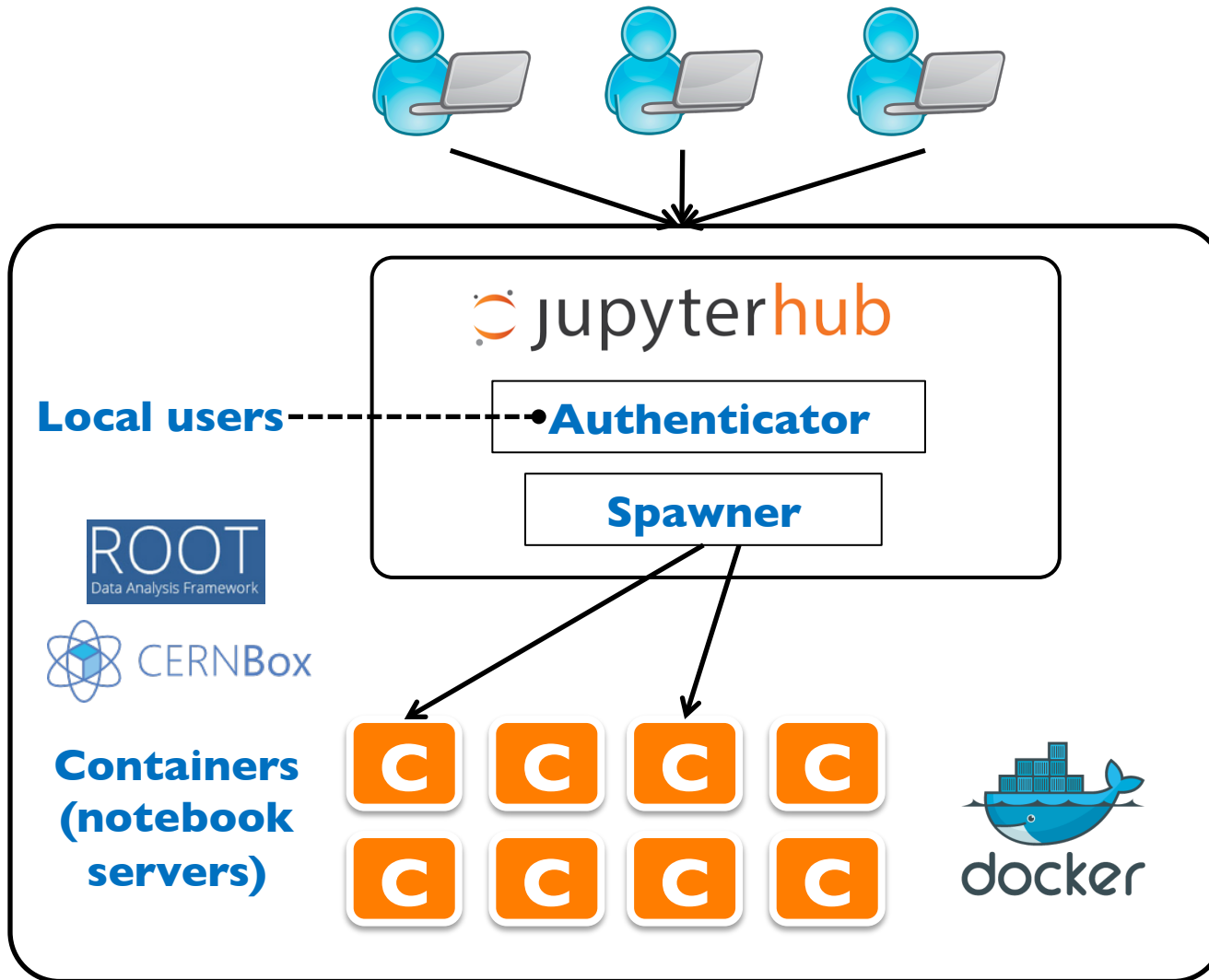
Double32\_pycpp.ip...



th2polyEurope\_cpp...



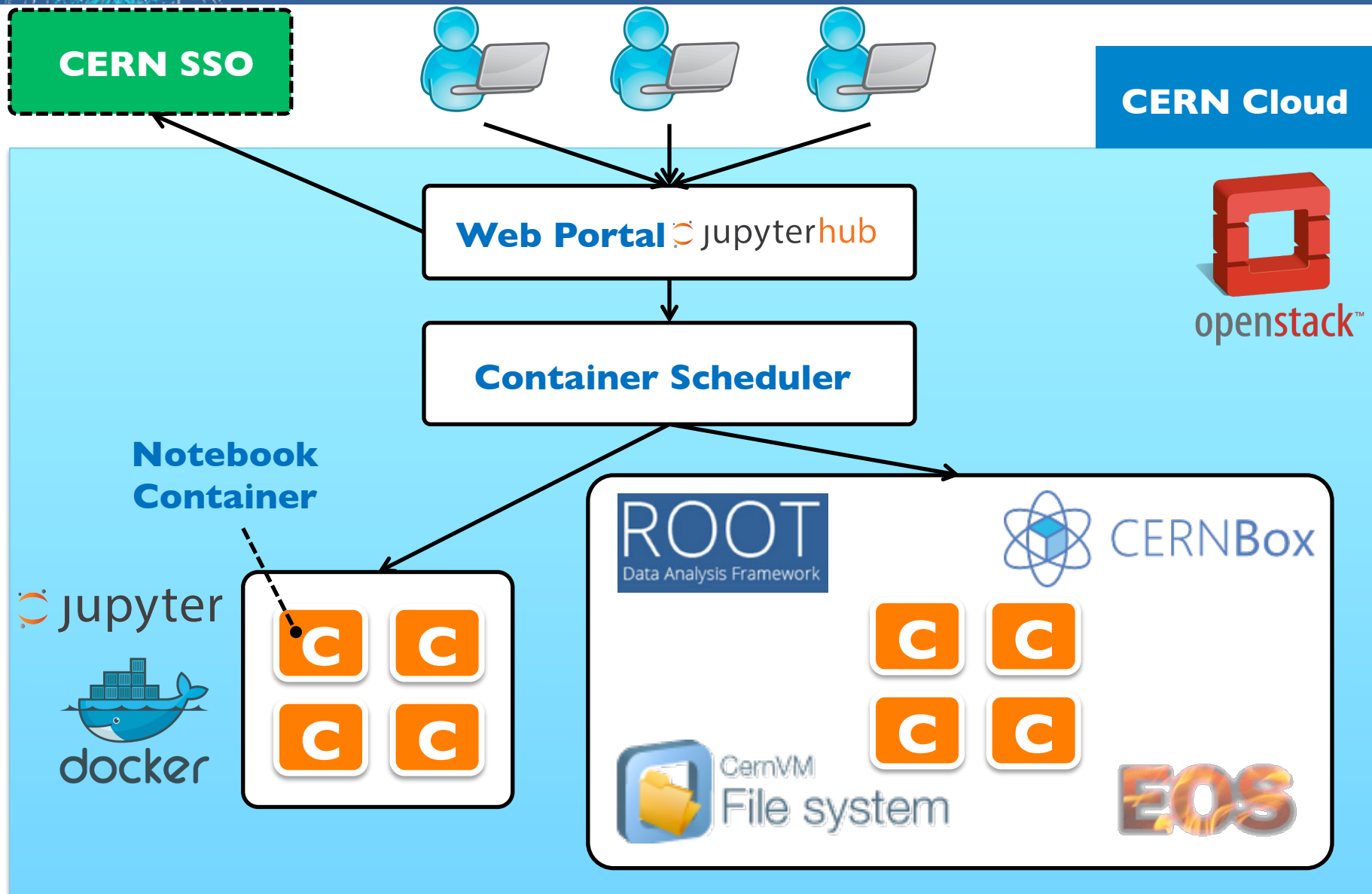
# What we have now: Demo Server



**One single powerful machine**



# What we are working on



**Distributed system, complementary to Ixplus**



# Potential “Daily Use-Case”

- Launch jobs on the batch farm
- Access notebook on a VM in OpenStack Cloud
- Inspect produced data via CERNBox/EOS from the notebook
- Create plots and output data with software provided via CVMFS
- Share, access plots (and output data!) on the web with CERNBox web interface
- Security and confidentiality guaranteed by the usual CERN standards



Summer Student Program, ROOT tutorial, last session: Interactive notebooks offered

- Single 24 cores box, Beta version of the software layer
- **50 participants, perfect scaling, a success!**
  - <https://indico.cern.ch/event/407519>

Data Science @ LHC Workshop, C++ notebook TMVA tutorial:  
<http://indico.cern.ch/event/395374/>

- ROOT is now integrated with notebooks
  - Python and C++ interactive shells
  - Tab completion, C++/Python integration, syntax highlighting, graphics inlining, shell commands
  - Available now!
- Integration with the CERN services portfolio
  - Collaborating with IT department
    - In particular with IT-DSS
  - Work in progress, usable demo server available
  - Pilot service proposal submitted to PH, IT and WLCG heads
- Useful for training and education, and potentially for interactive analysis

**Backup slides**



# Integration: Main Requirements

- Code in macros/programs usable in notebooks (and vice versa)
- Provide a novel ROOT Prompt (C++) kernel
  - A notebook which is a web based ROOT prompt
- Easy access to well known ROOT and notebooks features
- Provide clear, useful examples and documentation

**Requirements satisfied**  
**Delivered in release 6.05/02**

Now it's time to take a tour of the new provided functionalities!



**Jupyterhub**: manages login of users and redirection to notebook

- Existing solution: <https://github.com/jupyter/jupyterhub>
- Allows encapsulation: spawn Linux container at logon
  - User isolated from the host, modulo volumes explicitly mounted (CVMFS, CERNBox)
- Needs to be customised, e.g.:
  - CERN sign-on procedure
  - Docker image for the container



## EOS

Disk-based low latency storage infrastructure for physics users. Main target: [physics data analysis](#).

## CERNBox

Functionality analogous to Dropbox™. Synchronisation capabilities between user machines and central repository. [Data stored on EOS](#).

## Indico

[Manage complex conferences](#), workshops and meetings.

## CVMFS

[HTTP based network FS](#), optimized to deliver experiment software  
Files aggressively cached and downloaded on demand.



# The Demo





Time to go back see this workflow in action!

We will:

- Create a simple plot and a ROOT file with ROOTaaS
- Share it with CERNBox

 jupyter

Sign in

**Username:**

rw15u098

**Password:**

.....

Sign In



Terminal

Control Panel

Logout

Files

Running

Clusters

Select items to perform actions on them.

Upload

New ▾



cernbox



Terminal

Control Panel

Logout

Select items to perform actions on them.

Upload

New ▾



/ cernbox



..



tutorials



HowTo\_ROOT-Notebooks.ipynb



My First Notebook.ipynb



## My First ROOT Notebook

This is an example that aims to show the capabilities of ROOT once integrated in a notebook.

```
In [1]: import ROOT
```

```
Welcome to ROOTaaS 6.05/01
```

```
In [2]: h = ROOT.TH1F("myHisto", "My Title!;My X Axis;My Y Axis", 64, -4, 4)  
h.FillRandom("gaus")
```

```
In [3]: c = ROOT.TCanvas("myCanvas", "myCanvasTitle", 1024, 768)  
h.Draw()  
c.Draw()
```



This is an achievement. Let's save this plot and the histogram itself in a [ROOT file](#).

```
In [4]: c.Print("myPlot.pdf")
```

```
Info in <TCanvas::Print>: pdf file myPlot.pdf has been created
```

```
In [5]: ofile = ROOT.TFile.Open("myOutputFile.root", "recreate")
        h.Write()
        ofile.Close()
```

```
In [6]: %%bash
        ls
```

```
HowTo_ROOT-Notebooks.ipynb
My First Notebook.ipynb
myOutputFile.root
myPlot.pdf
tutorials
```

Now go and check on the **CERNBOX** web interface **your data!**












- All files
- Favorites
- Shared with you
- Shared with others
- Shared by link
- Deleted files
- 

<input type="checkbox"/>	Name ▲	Size	Modified
	.ipynb_checkpoints	0 kB	2 minutes ago
	My First Notebook.ipynbye0opi_m	0 kB	a few seconds ago
	tutorials	0 kB	7 days ago
	HowTo_ROOT-Notebooks.ipynb	214 kB	7 days ago
	My First Notebook.ipynb	24 kB	2 minutes ago
	myOutputFile.root	6 kB	3 minutes ago
	myPlot.pdf	14 kB	3 minutes ago





	HowTo_ROOT-Notebooks.ipynb	214 kB	7 days ago
	My First Notebook.ipynb	25 kB	25 minutes ago
★ 	my... .root   Download  Versions  Share	6 kB	a few seconds ... 
	myPlot.pdf	14 kB	27 minutes ago