



“Dropbox” For Science

Cloud Storage&More

Jakub T. Mościcki, CERN IT-ST

CGW 2016

domain logic

data science & education

workflows ✂ analysis environments ✂ publishing ✂ tagging

non-traditional working storage

synchronization ✂ sharing ✂ cloud ✂ mobile ✂ federation ✂ apps ✂ roaming

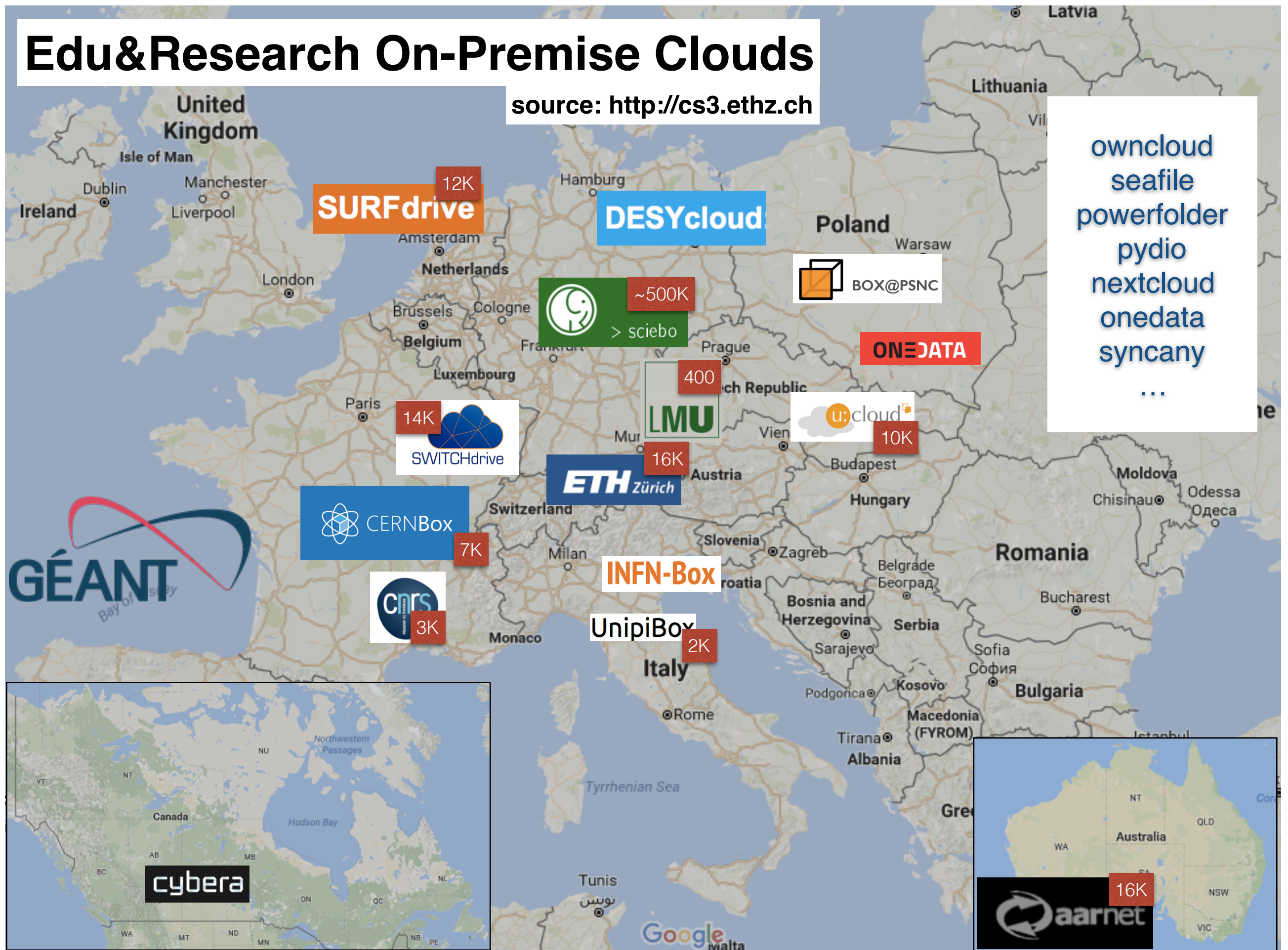
storage abstraction & protocols

files ✂ HTTP ✂ objects ✂ filesystems ✂ POSIX ✂ S3 ✂ SWIFT

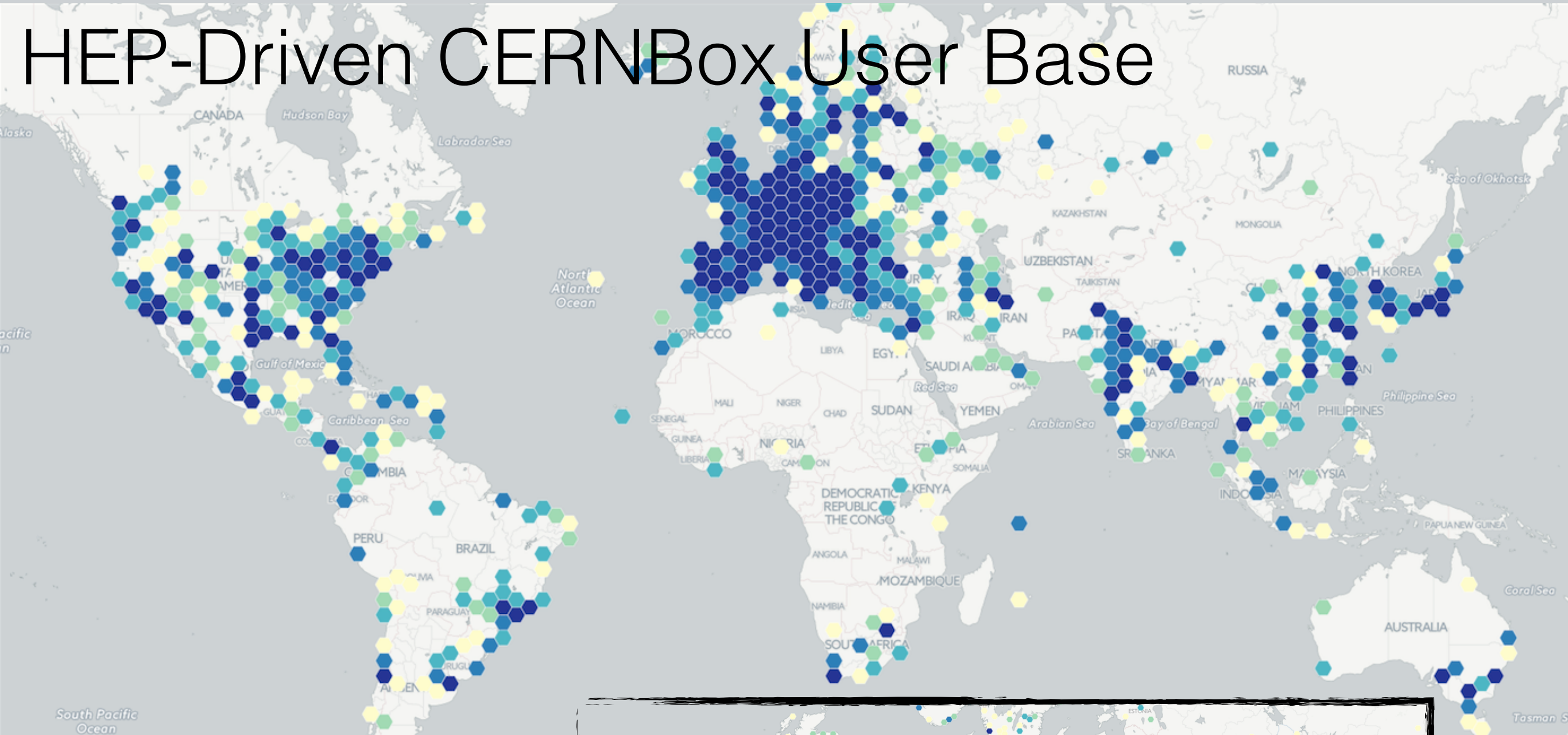
storage technology

Edu&Research On-Premise Clouds

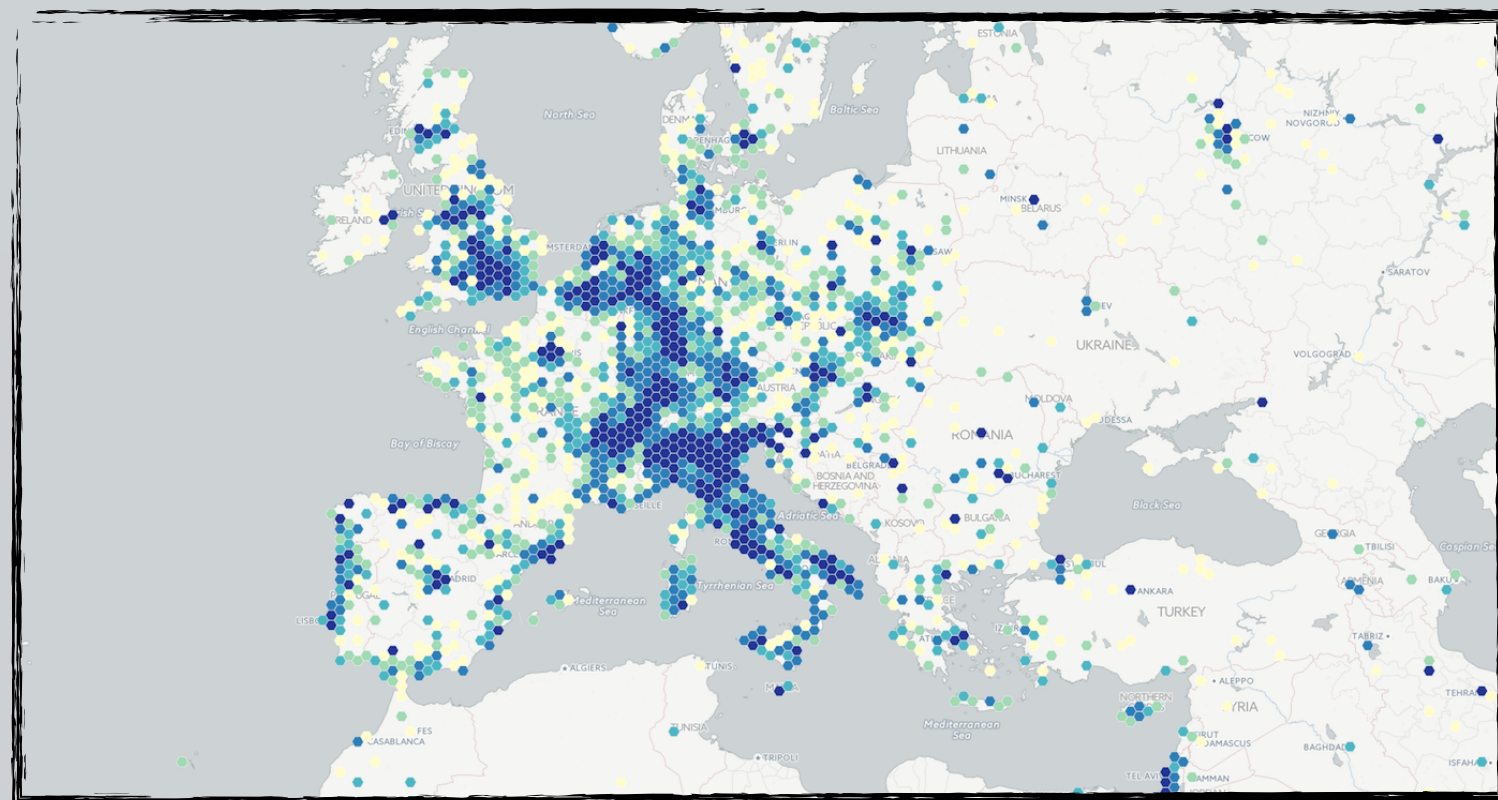
source: <http://cs3.ethz.ch>



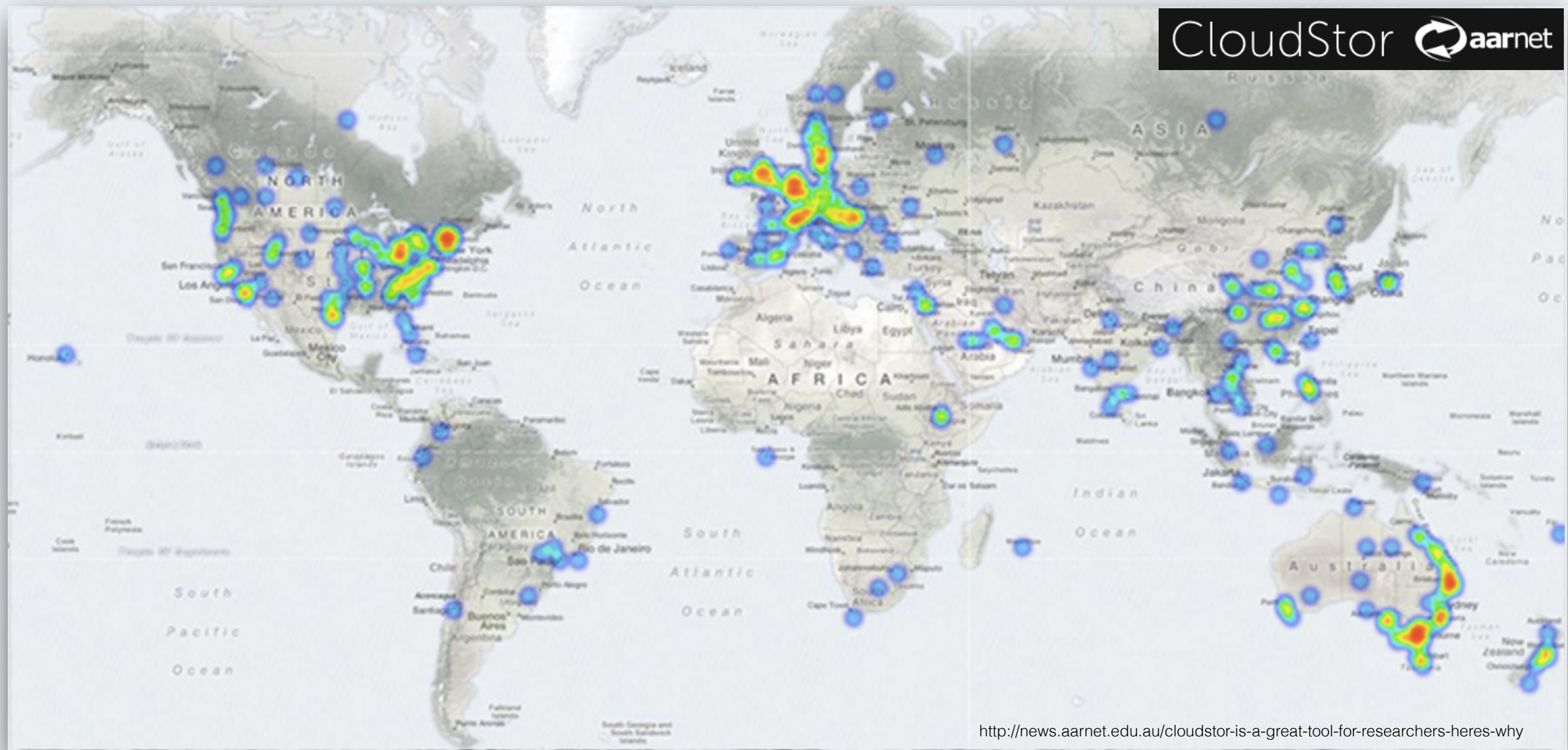
HEP-Driven CERNBox User Base

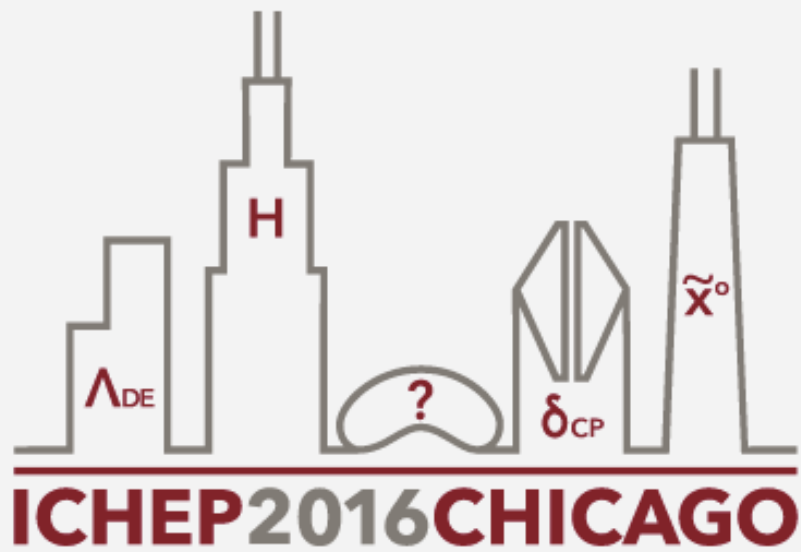


Q1-Q2 2016
~6000 users



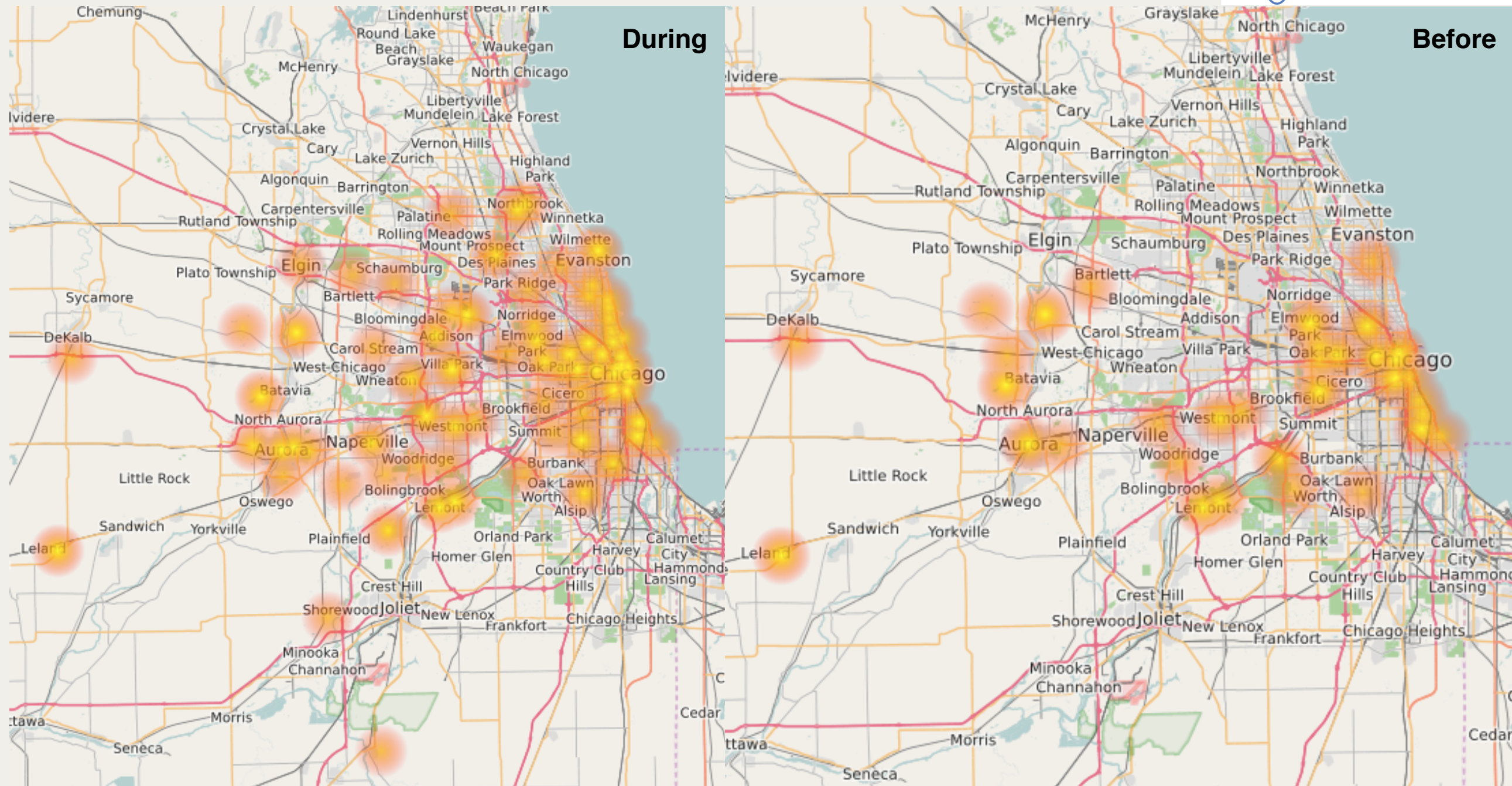
Non-HEP User Community Example





38th INTERNATIONAL CONFERENCE ON HIGH ENERGY PHYSICS

AUGUST 3 - 10, 2016
CHICAGO

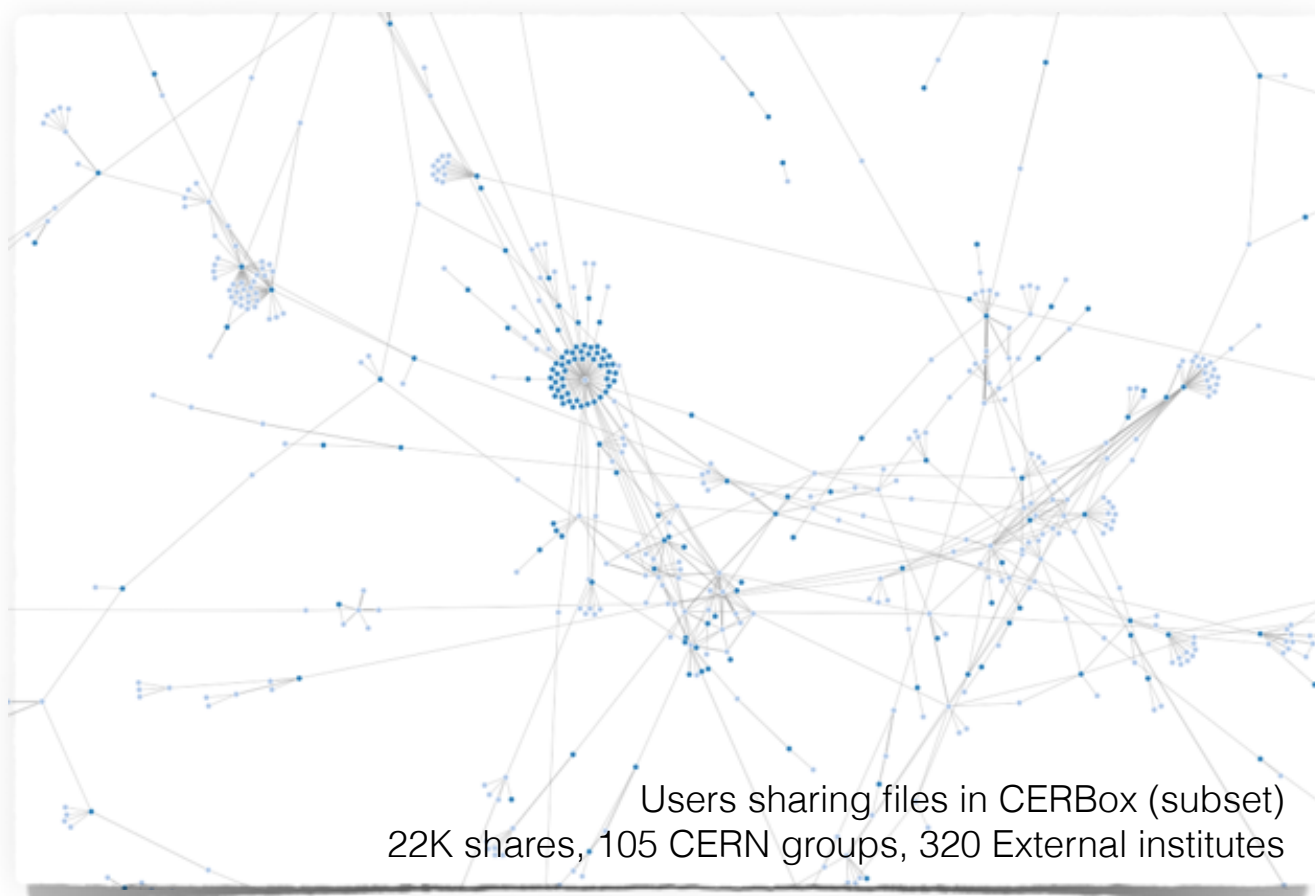


Sharing in on-premise storage clouds

- Sharing made *very* easy



- A successful model



Jupyter_Notebooks

★ 0 kB, seconds ago

Sharing CERBox Info

Share with users or groups ...

Jakub Moscicki

☐ write + delete

☒ Share link

<https://cernbox.cern.ch/index.php/s/nUS>

☐ Password protect

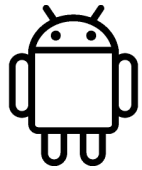
☐ Allow editing

☒ Set expiration date

09-10-2017

The public link will expire no later than 365 days after it is created

Sync&Share Data Repositories



iOS



CERNBox



/home

/eos/cms

/eos/atlas

/eos/project

~1PB

~100 PB

Cloud Data Analysis Model

In a browser

Scientific & Educational Notebooks

Code



Notebook Functionalities



Control Panel

Logout

File Edit View Insert Cell Kernel Help Python 2

Code Cell Toolbar: None

```
In [1]: cout << "From this point on...\n"
        << "... it's only C++ ..." << endl;
```

From this point on...
... it's only C++ ...

We now create a [ROOT histogram](#) and fill it with [random numbers](#) distributed according to a Gaussian.

```
In [1]: TH1F h("gauss", "Example histogram", 64, -4, 4);
        h.FillRandom("gaus");
```

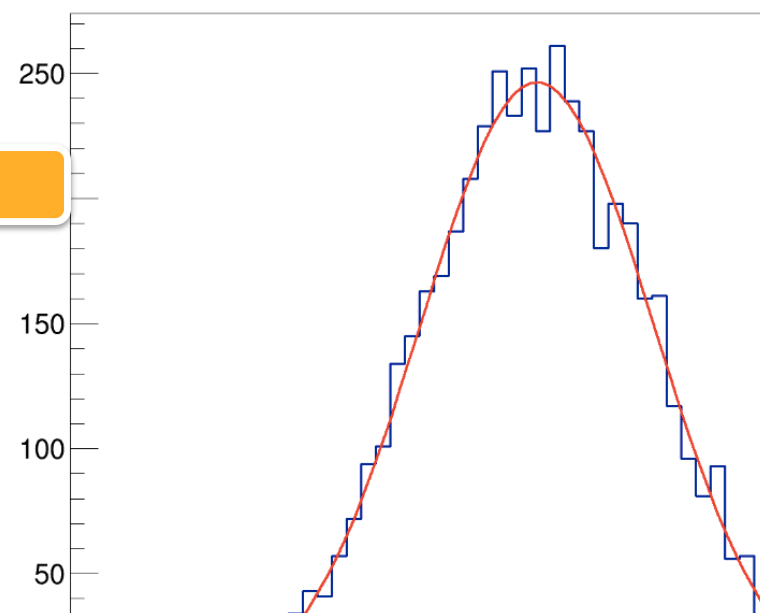
Now, we create a [canvas](#), the entity which holds graphics primitives.

```
In [4]: h.Fit("gaus", "S");
        c.Draw();
```

Shell Commands

Images

Example histogram



gauss	
Entries	5000
Mean	0.008152
Std Dev	1.016

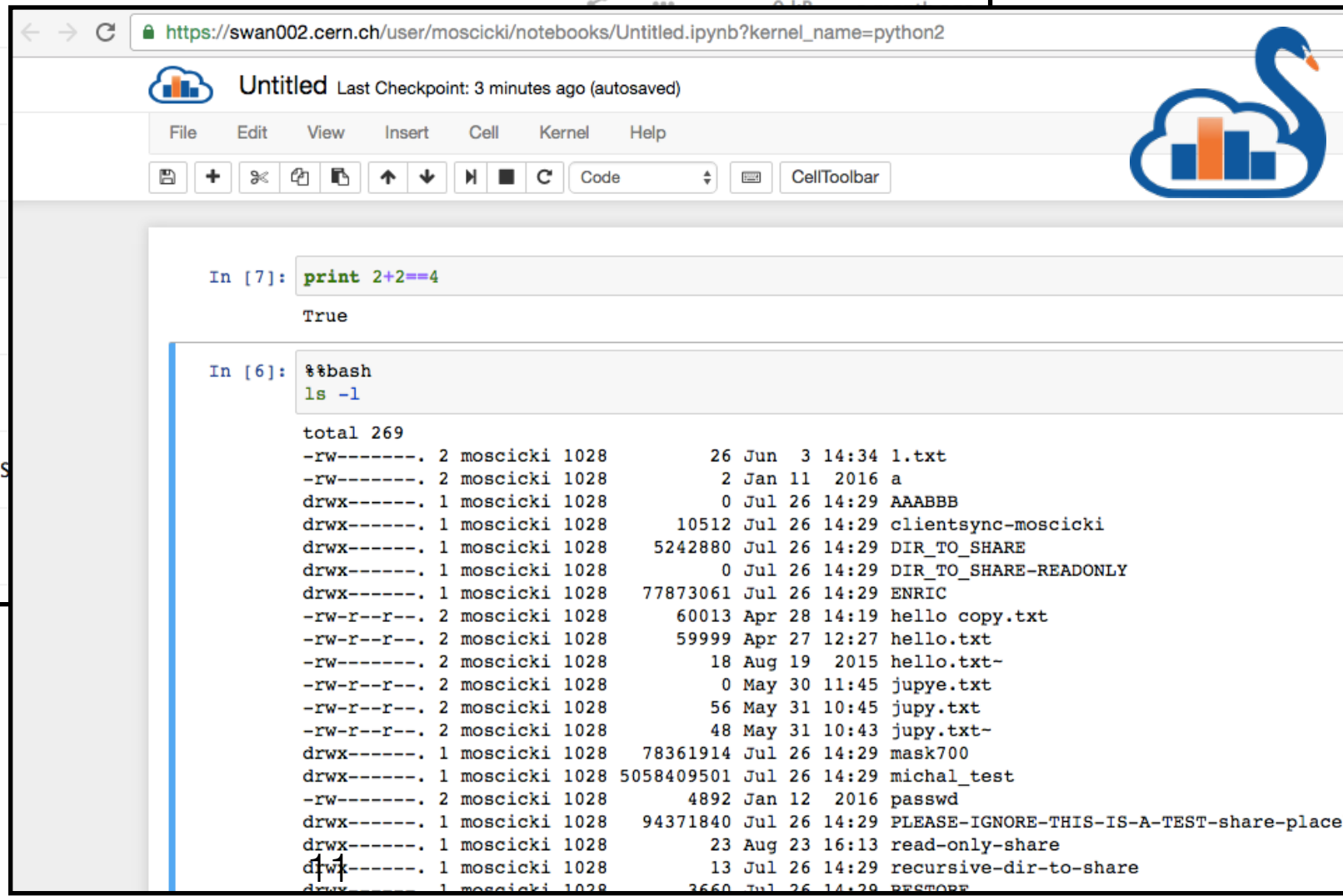
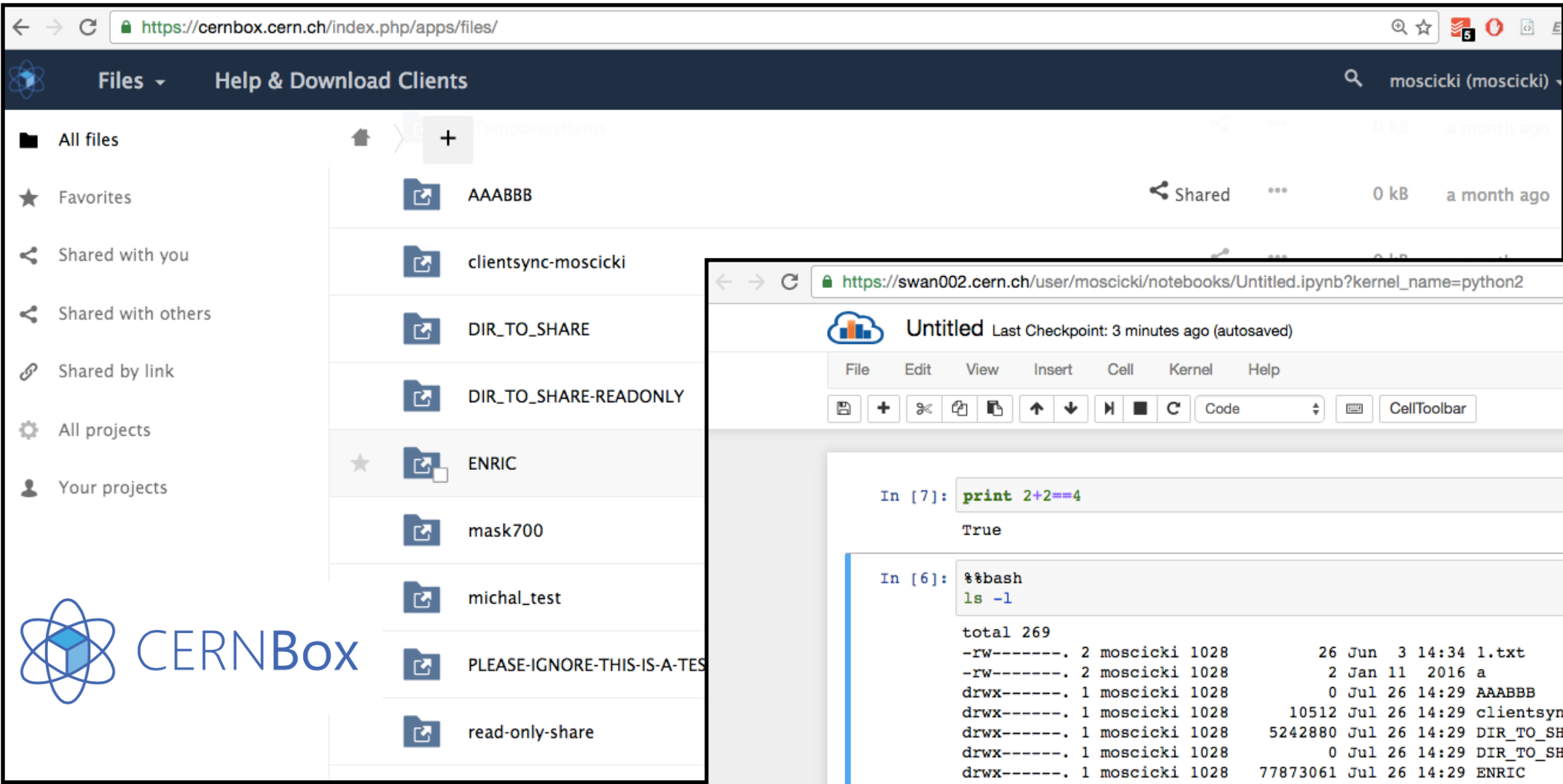
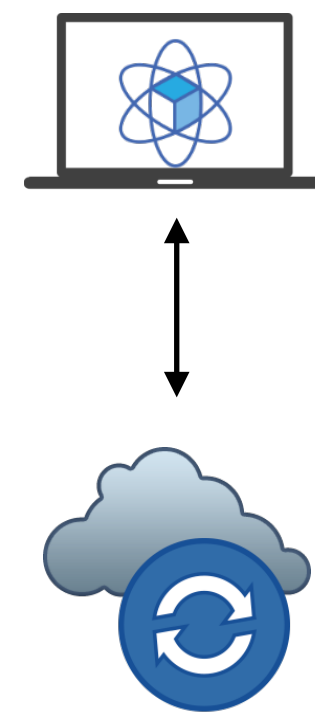


SWAN Service

Text and Formulas



Common and synchronised file space



Access&Sharing

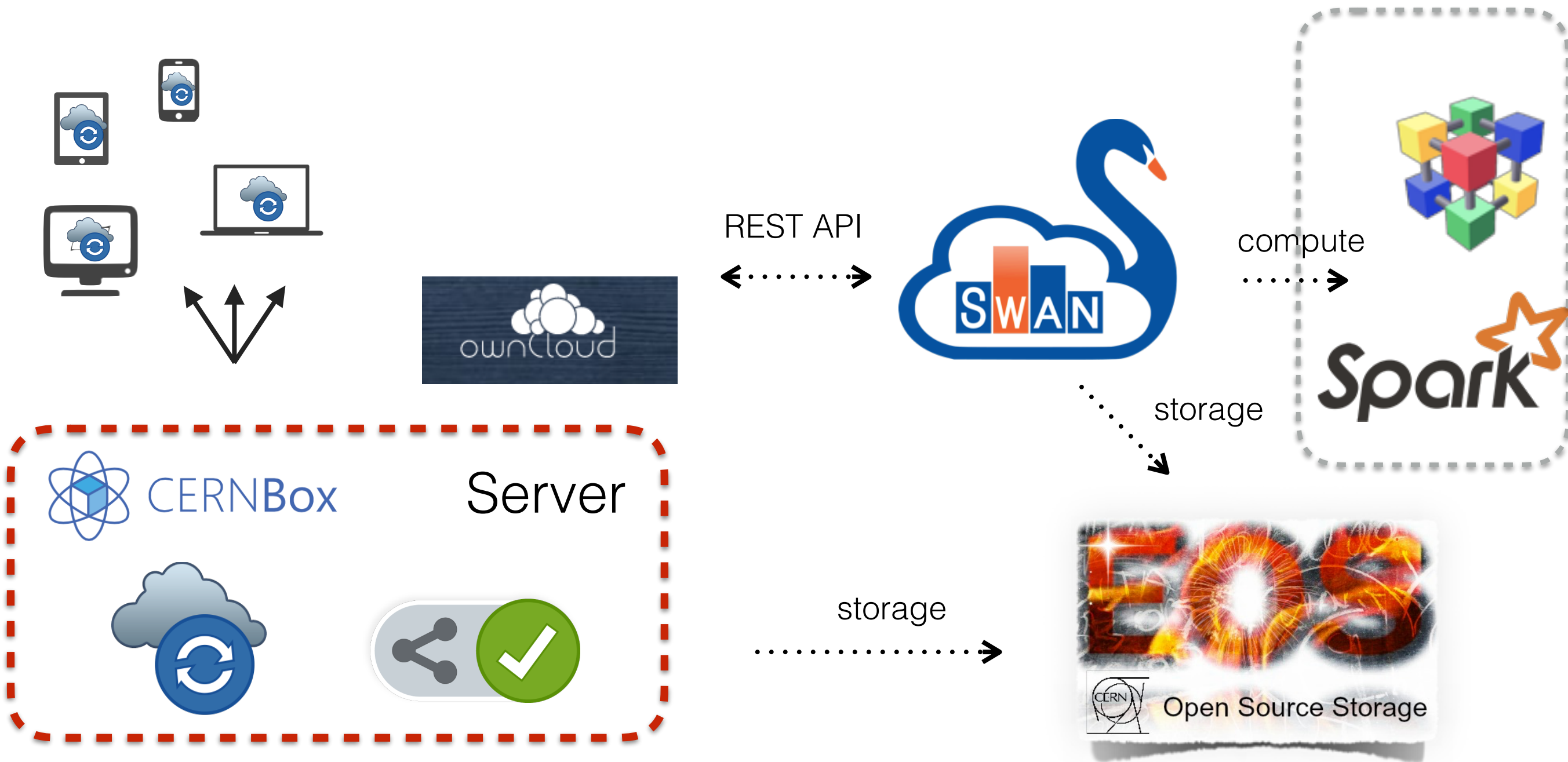


Open Notebooks directly from CERNBox website

		2 kB	4 months ago
		2 kB	4 months ago
		2 kB	4 months ago
			4 kB 3 months ago
			4 kB 4 months ago
156 kB			

... and share Notebooks via CERNBox (REST API)
directly from SWAN website

Integration Architecture

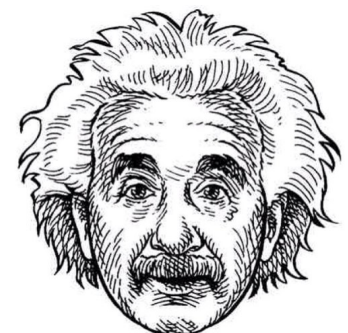


Federation: Open Cloud Mesh

- **interconnected, secure private clouds**
- **universal file access** with privacy, control and security an on-premises cloud provides.



Users' examples

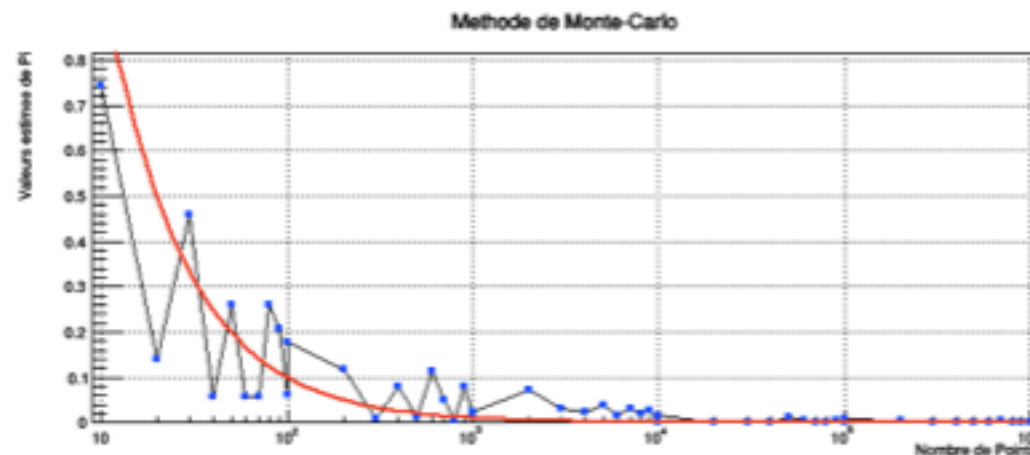


K-12 Education



```
cl.cd().SetLogx()  
  
gr = TGraph( n, x, y )  
gr.SetLineColor( 1 )  
gr.SetLineWidth( 1 )  
gr.SetMarkerColor( 4 )  
gr.SetMarkerStyle( 21 )  
gr.SetTitle( 'Methode de Monte-Carlo' )  
gr.GetXaxis().SetTitle( 'Nombre de Points' )  
gr.GetYaxis().SetTitle( 'Valeurs estimee de Pi' )  
  
gr.Draw( '' )  
  
f1 = TF1( 'f1', '10/x', 0, 1000000 )  
f1.SetLineWidth( 2 )  
f1.Draw( "Same" )  
  
cl.Draw()
```

TCanvas::Constructor:0: RuntimeWarning: Deleting canvas with same name: cl



Mano, 14 years old college student

- discovering mathematics
- interested in programming&sciences
- sharing notebook with his professor and classmates

Comme vous pouvez le voir ci dessus la marge d'erreur est tres grande quand il y a peu de points. Nous avons traces une courbe faisant une moyenne mais comme nous pouvons le voir elle ne va pas, a la fin elle est a 0 alors qu'elle devrait etre un peu au dessus ce qui compte beaucoup car la marge d'erreur est faible et ce qui est beaucoup moins important au debut ou la marge d'erreur est grande

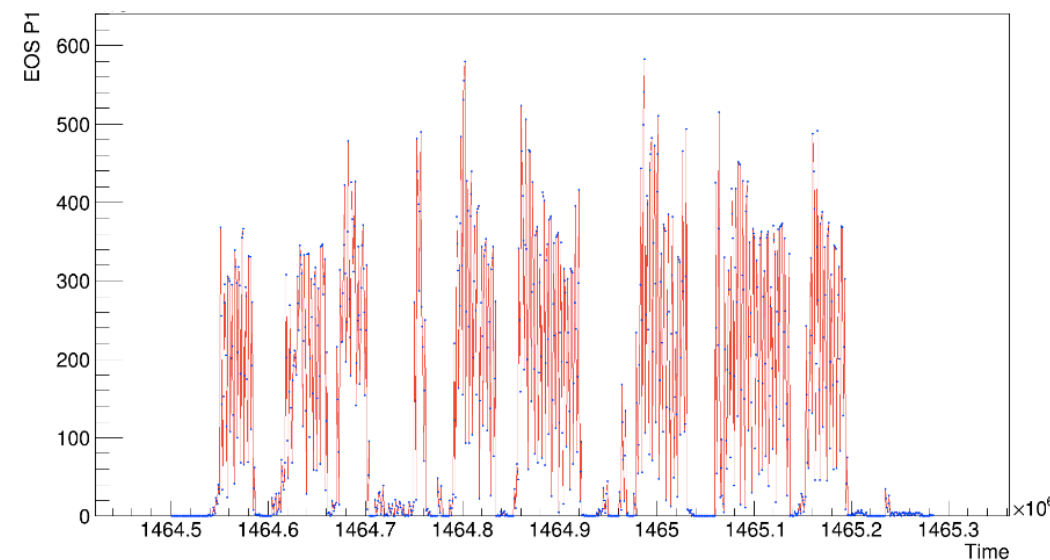
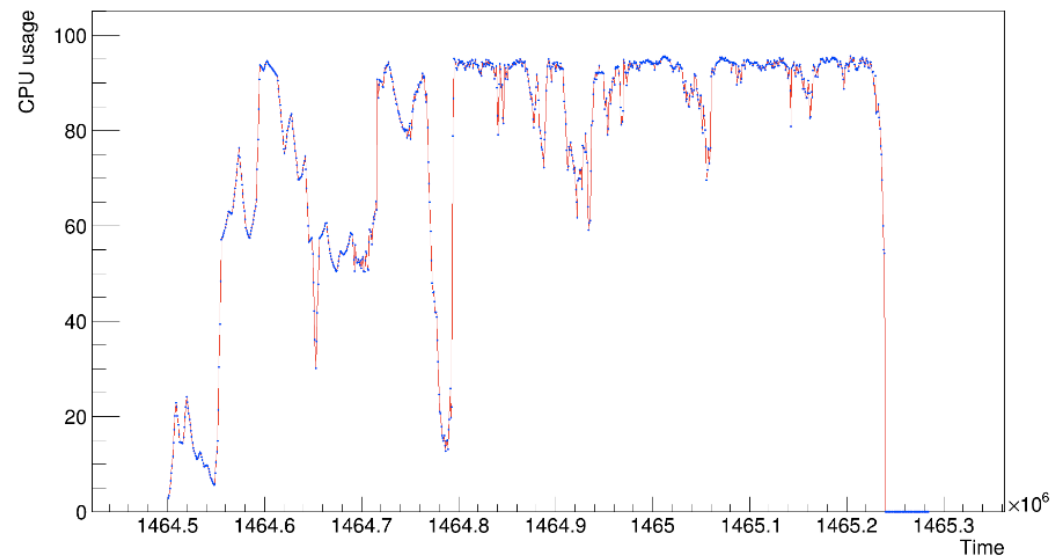
IT Systems Engineering



```
In [10]: #plot1 = scatterPlot(timeVal, inVal, "NET in | Time", "Time", "NET in", SAVE_CHART, "NETin_Time.png", "ALP")
plot2 = scatterPlot(timeVal, cpuVal, "CPU usage | Time", "Time", "CPU usage", SAVE_CHART, "CPU_Time.png", "ALP")
plot3 = scatterPlot(timeVal, EOSP1Val, "EOS P1 | Time", "Time", "EOS P1", SAVE_CHART, "EOSP1_Time.png", "ALP")
plot4 = scatterPlot(cpuVal, EOSP1Val, "EOS P1 | CPU", "CPU", "EOS P1", SAVE_CHART, "EOSP1_CPU.png", "AP")

Info in <TCanvas::Print>: png file CPU_Time.png has been created
Info in <TCanvas::Print>: png file EOSP1_Time.png has been created
Info in <TCanvas::Print>: png file EOSP1_CPU.png has been created
```

CPU usage | Time



M. Lizzit, M. Lamanna IT-ST

Network&Storage performance analysis

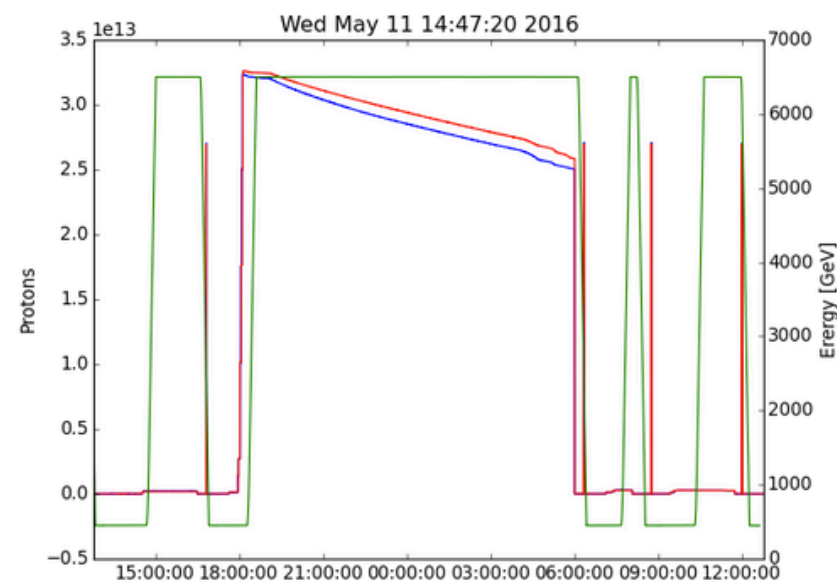
- correlation of system activity
- data transferred from reconstruction farm to EOS storage cluster
- CPU usage of reconstruction farm
- EOS storage cluster IO activity

LHC Engineering



```
In [3]: now=time.time()
ib1="LHC.BCTDC.A6R4.B1:BEAM_INTENSITY"
ib2="LHC.BCTDC.A6R4.B2:BEAM_INTENSITY"
nrg="LHC.BOFSU:OFSU_ENERGY"
data=db.get([ib1,ib2,nrg],now-3600*24,now)
```

```
In [4]: plt.clf()
tt,vv=data[ib1]
plt.plot_date(epoch2num(tt),vv,'-b',label='Beam1')
tt,vv=data[ib2]
plt.plot_date(epoch2num(tt),vv,'-r',label='Beam2')
plt.ylabel('Protons')
plt.twinx()
tt,vv=data[nrg]
plt.plot_date(epoch2num(tt),vv,'-g',label='Energy')
plt.ylabel('Energy [GeV]')
plt.title(time.asctime(time.localtime(now)))
```



R. De Maria, LHC BEAM

- Read measurements coming from pick-ups in a database
- Plot time series
- Needs also SciPy and to share the notebooks with his colleagues

LHC Research



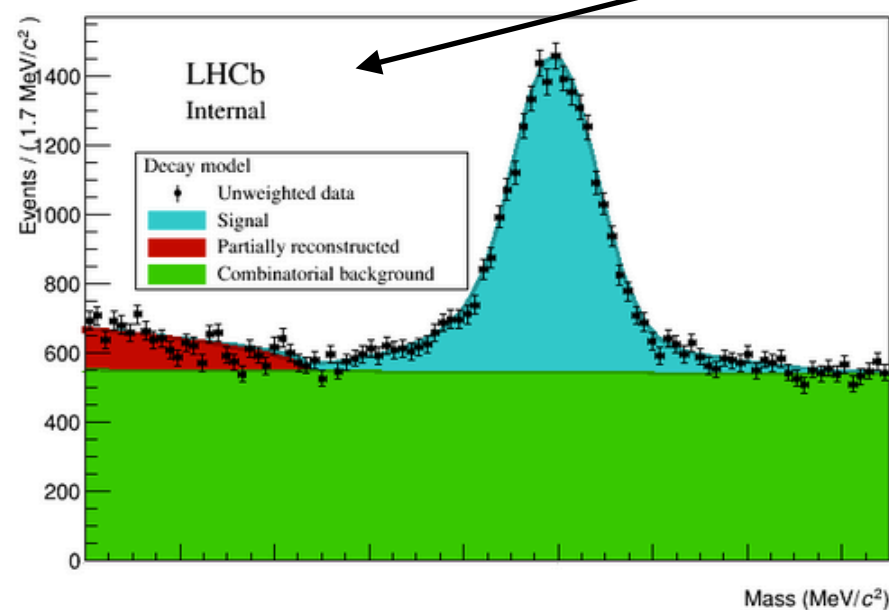
```
title = { "model": "Signal" , "pdfBkg" : "Partially reconstructed" , "cmbBkg": "Combinatorial background"}

for (component, color) in [ ("model",kCyan), ("pdfBkg",kRed), ("cmbBkg",kGreen)]:
    model.plotOn (frame, LineColor(color+2) , DrawOption('L'), Components(component), LineWidth(5))
    model.plotOn (frame, FillColor(color+1) , DrawOption('F'), Components(component), LineWidth(0), Name("P"+component)
    )
    leg.AddEntry ( frame.findObject ("P"+component), title[component] , "F" )

data.plotOn ( frame, MarkerColor ( ROOT.kBlack ) )
frame.Draw()
Graphics().lhcbMarker(0.2,0.8, "Internal")

leg.Draw()
ROOT.gPad.Draw()
```

Results coming
from real data!
(published now)




L. Anderlini

Rare B meson decay in LHCb

- Read data from EOS
- Setup complex fit
- Document and inspect results

Community

CS3 Workshops

 **Workshop on Cloud Services
for File Synchronisation and Sharing**

**CERN 17-18
November 2014**

<https://indico.cern.ch/event/336753/>
Abstract submission: 30 September
Registration: 31 October

ORGANISING COMMITTEE
Miguel Branco
Massimo Lamanna
Jakub T. Moscicki

Cloud Services for Synchronisation and Sharing (CS3)
Cloud Storage Services for Novel Applications and Workflows

ETH Zürich - Switzerland
18-19 January 2016

<https://cs3.ethz.ch>
Abstract submission: 1 September – 15 November 2015

PROGRAMME COMMITTEE
Massimo Lamanna (CERN), Luca Mascetti (CERN),
Jakub Mościcki (CERN) and Tilo Steiger (ETH)

Cloud Services for Synchronisation and Sharing (CS3)
Novel applications, cloud storage technology, collaborations

SURFsara Amsterdam
30 Jan - 1 Feb 2017

<https://cs3.surfsara.nl>

PROGRAMME COMMITTEE
Guido Aben (AARNet), Massimo Lamanna (CERN)
Luca Mascetti (CERN), Jakub Mościcki (CERN)
Tilo Steiger (ETH), Ron Trompert (SURFsara)



2014



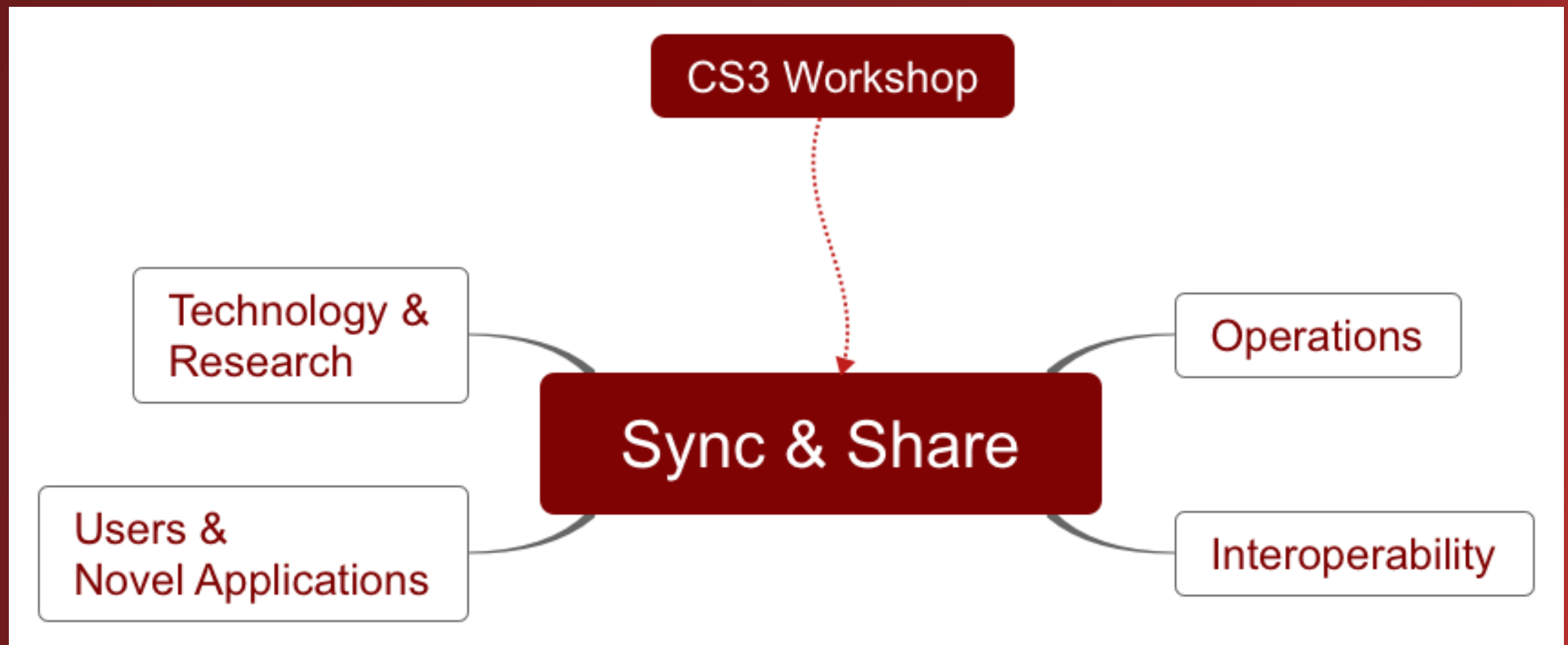
2016



2017



Building community knowledge



Tech & Vendors



Syncany

ONE DATA

IBM

Cloud Services for Synchronisation and Sharing (CS3)

Novel applications, cloud storage technology, collaborations

<https://cs3.surfsara.nl>

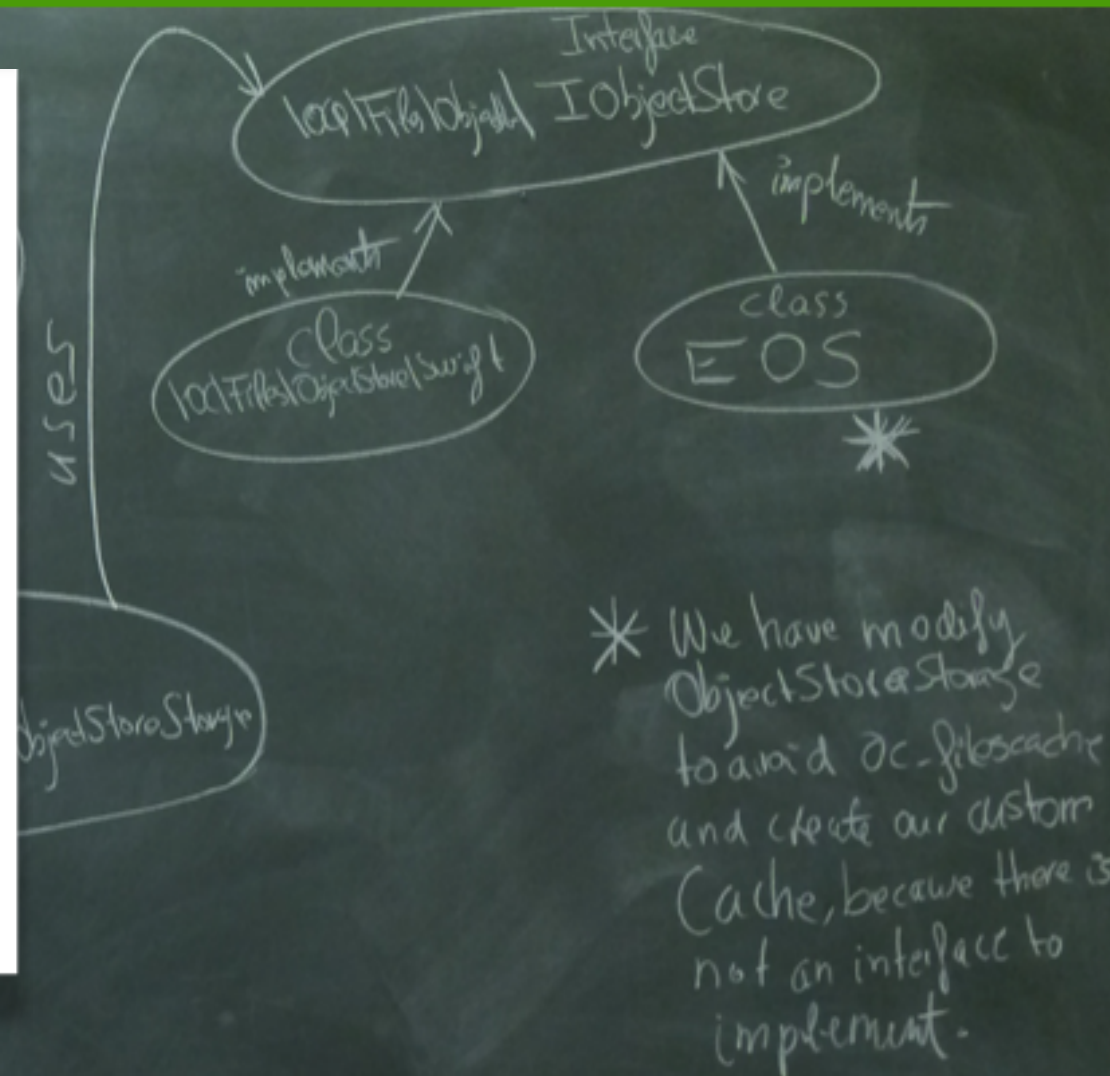
- Home
- What is the workshop about?
- Programme
- Important dates
- Abstract submission
- Accommodation
- Registration
- Participants
- Practical information
- Previous workshops

Search ...

Workshop on Cloud Services for Synchronisation and Sharing (CS3)

January 30st – February 1st 2017
Amsterdam, The Netherlands





SURFsara Amsterdam
30 Jan - 1 Feb 2017

<https://cs3.surfsara.nl>

PROGRAMME COMMITTEE

Guido Aben (AARNet), Massimo Lamanna (CERN)
Luca Mascetti (CERN), Jakub Mościcki (CERN)
Tilo Steiger (ETH), Ron Trompert (SURFsara)

