SWAN: service for web based analysis

P. Mato, E. Tejedor, D. Piparo, D. Castro
E. Bocchi, J. Moscicki, M. Lamanna

https://swan.web.cern.ch

Mar 27th, 2018
WLCG-HSF workshop

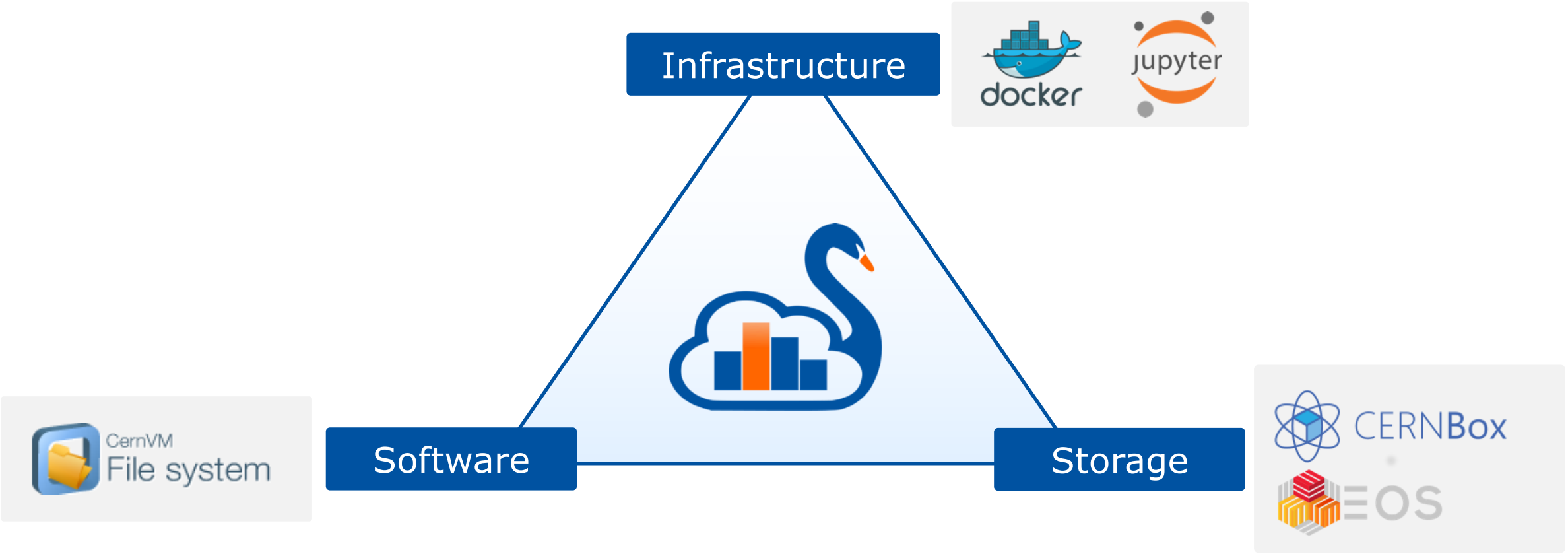
Introduction

Motivation

- Data Analysis only with a web browser
  - Available everywhere and at anytime
  - Integrated with other analysis ecosystems : ROOT, R, Python, Octave, ...
- Easy to use (but powerful)
  - No local installation and configuration needed
- Sharable scientific results
  - Plots, data, code
- Integration with CERN resources
  - Access software, user/experiments data, computing resources

SWAN

Integrating services



Jupyter - The Notebook as Interface

- A web-based interactive interface and platform that combines code, equations, text and visualisations
  - Ideal for sharing/collaboration
  - … In a nutshell: an “interactive shell opened within the browser”
- Many supported languages (kernels)
  - In SWAN: Python, ROOT C++, R and Octave
- Interactive, usually lightweight computations
- Very useful for some use cases
  - Final steps of an analysis, exploration, teaching, documentation and reproducibility

Text

Code

Graphics

Simple_ROOTbook_py (unsaved changes)

Control Panel | Logout

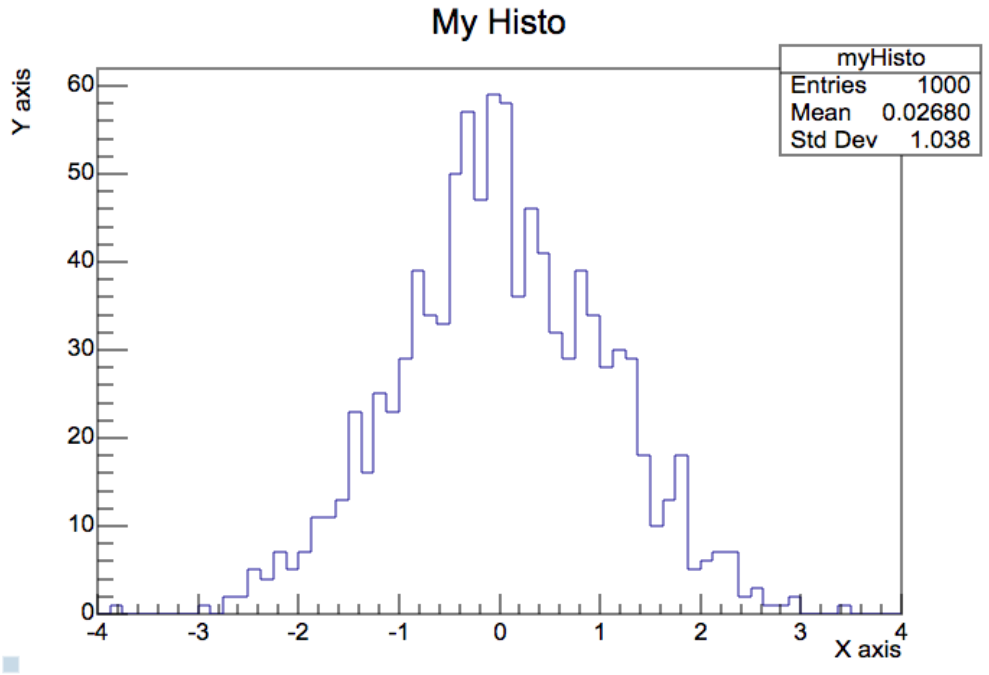
File | Edit | View | Insert | Cell | Kernel | Help | Trusted | Python 2

Code

Displaying graphics

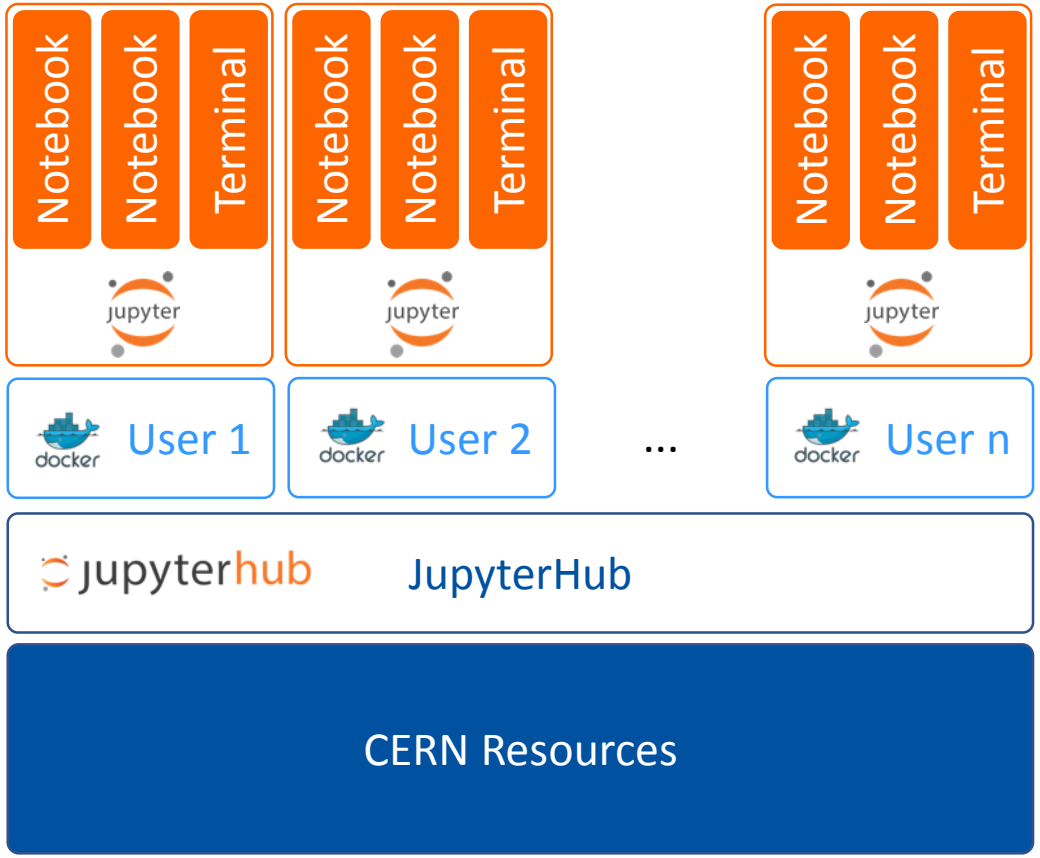
We can now draw the histogram. We will at first create a canvas, the entity which in ROOT holds graphics primitives. Note that thanks to JSROOT, this is not a static plot but an interactive visualisation. Try to play with it and save it as image when you are satisfied!

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



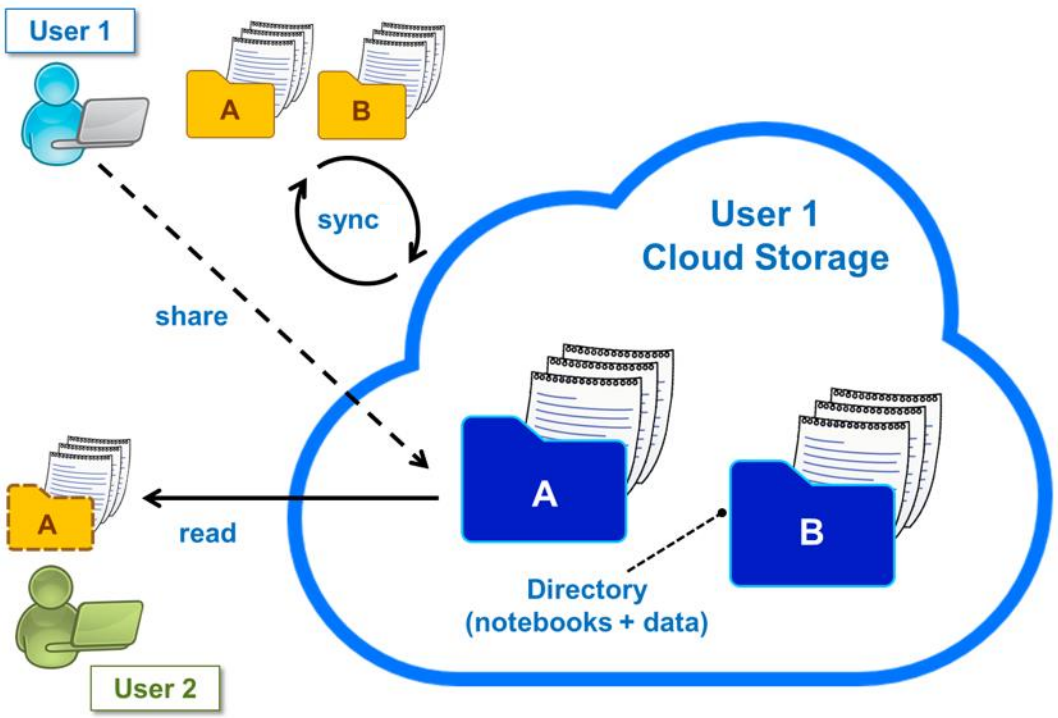
Integrating Jupyter

- Jupyterhub to allow multiple Jupyter instances
  - Single instance of Jupyter per user
- User sessions spawned as Docker containers
  - Enforces resource limits per user
  - To isolate users work
- Configurable environments through user defined scripts



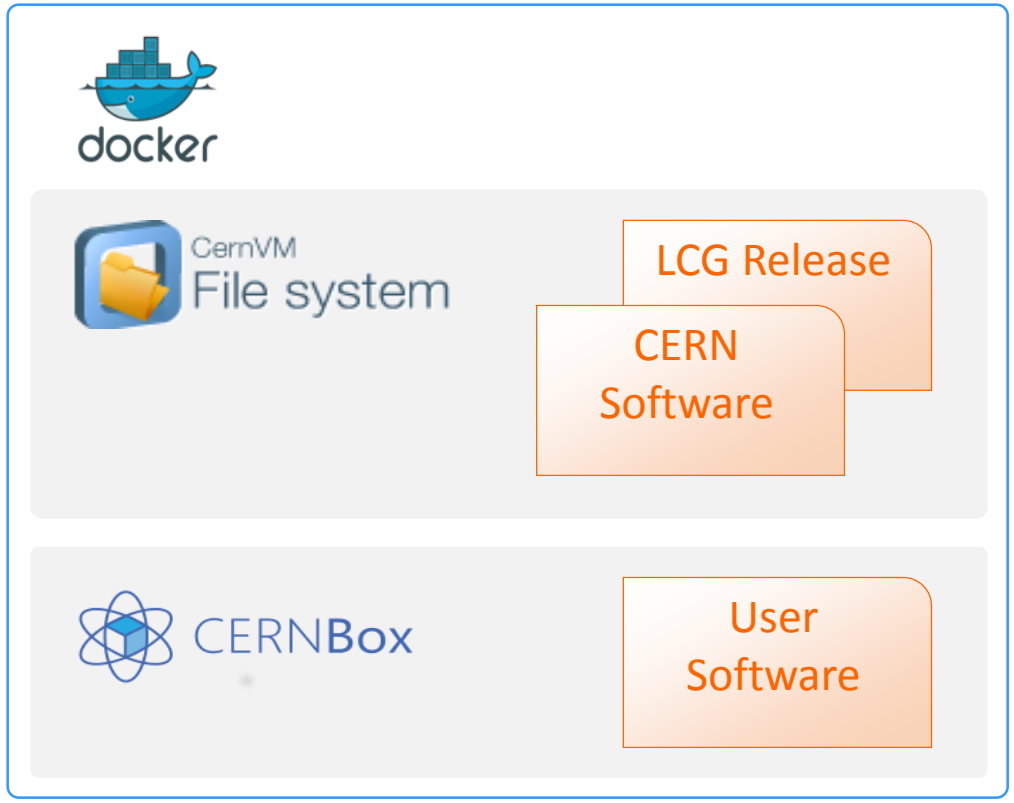
Storage

- User personal user space
  - Synchronized through CERNBox
  - All files synced across devices, the cloud and other users
- Uses EOS mass storage system
  - All experiment data potentially available

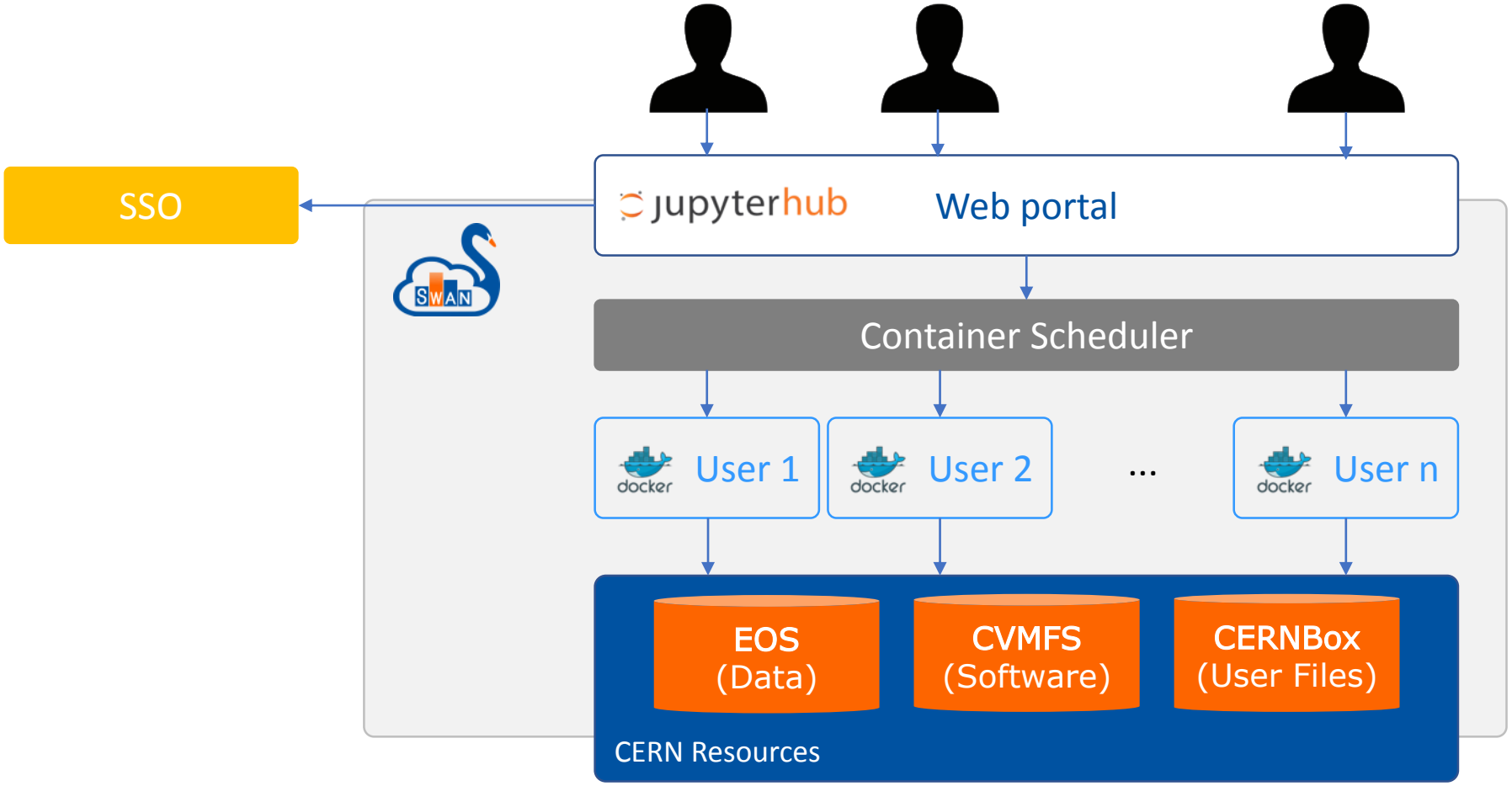


Software

- Software distributed through CVMFS
  - ”LCG Releases” - packing a large software stack with most common software
  - Experiment software used by researchers is available
- Possibility to install other libraries in user storage (CERNBox)



Architecture



Boxed SWAN

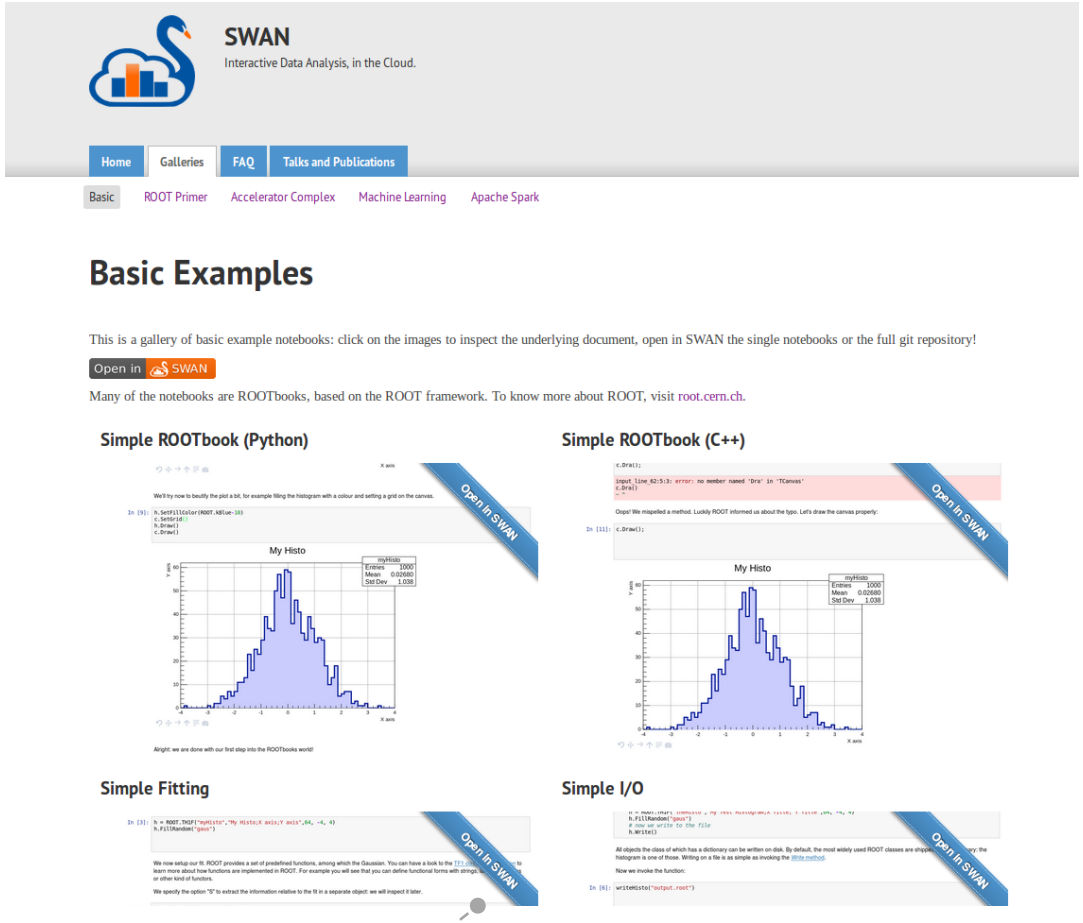
- Containerized version with all the needed infrastructure
  - Includes EOS, CERNBox, CVMFS and all Swan services (Jupyter Docker image, JupyterHub)
  - Available in https://github.com/cernbox/uboxed
- Easily deployable on premises
  - Installable in Linux systems
  - Based on Docker Compose

SWAN and the community

HEP 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 **swan.web.cern.ch**



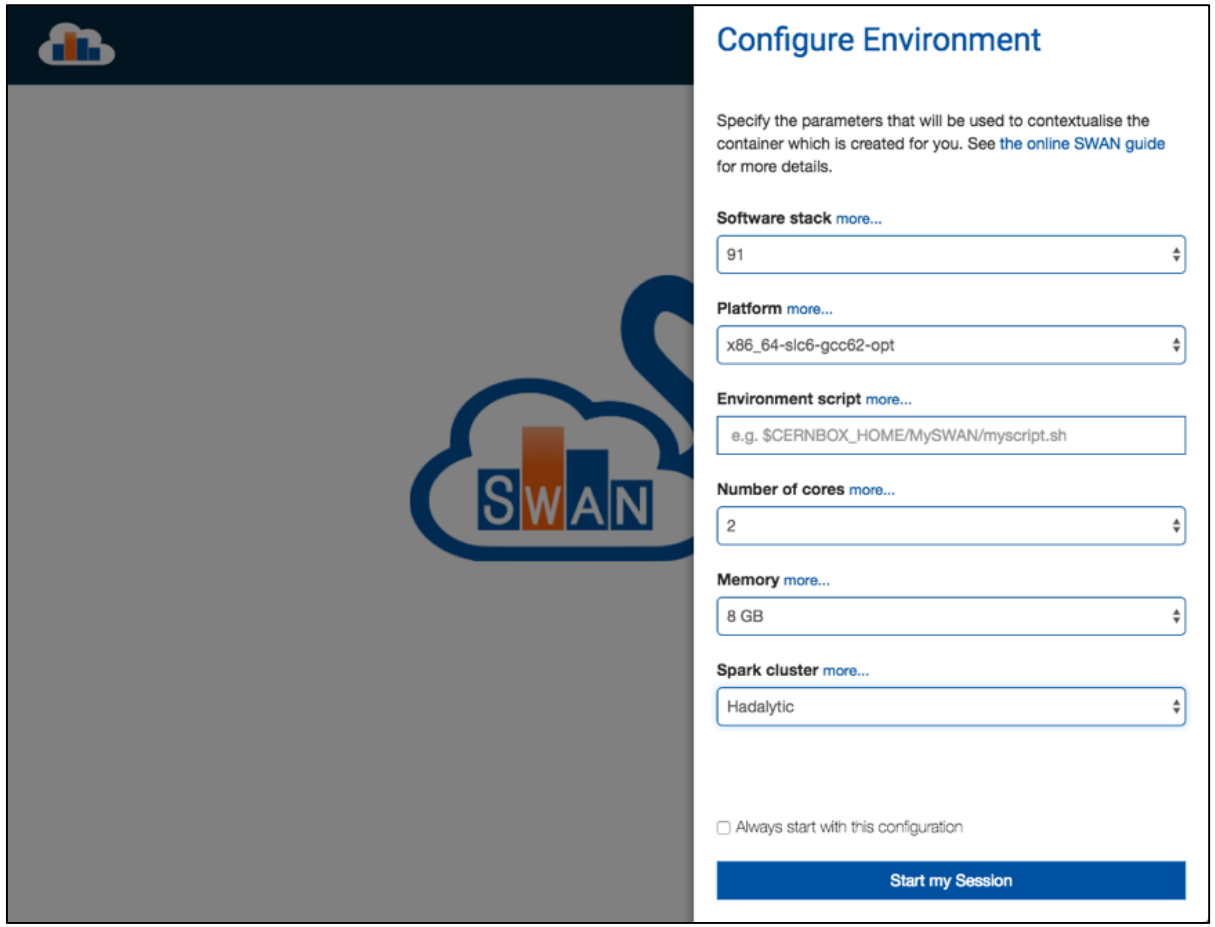
Access with only a click

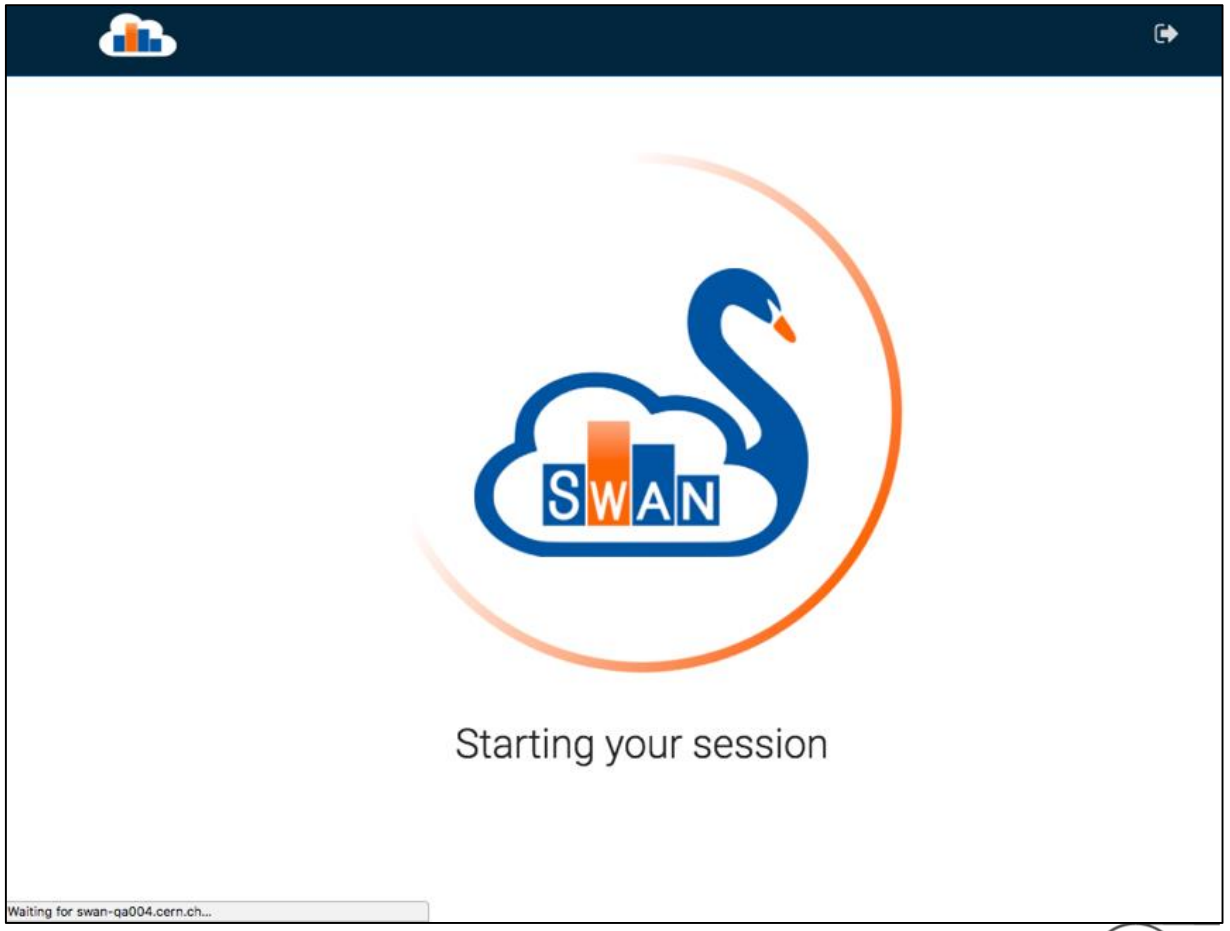
Other Comunities

- Building block in UP2University European Project
  - Bridge the gap between secondary schools, higher education, and the research domain
  - SWAN used by students to learn Physics and other Natural Sciences
  - Let the kids use the very same tools & services used by real researchers doing Big Science at CERN
  - Using Boxed distribution

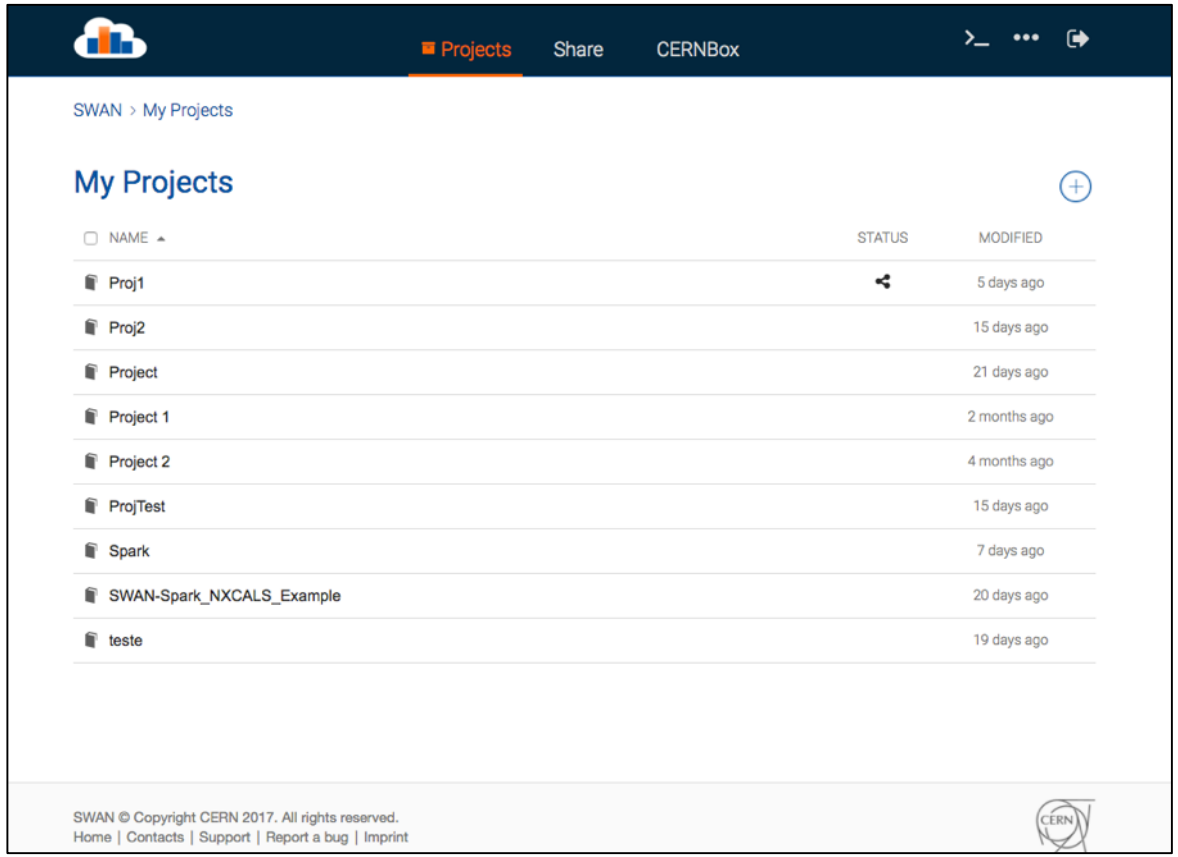
Recent developments

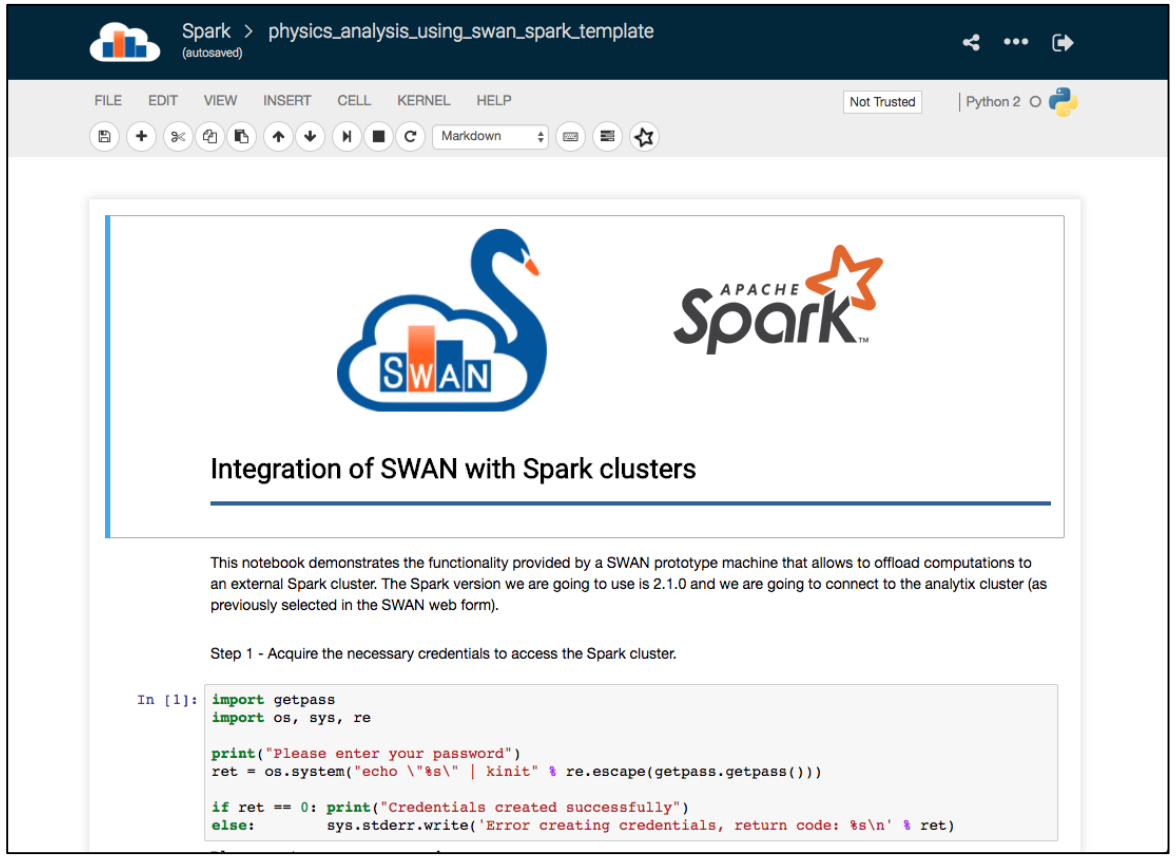
New User Interface





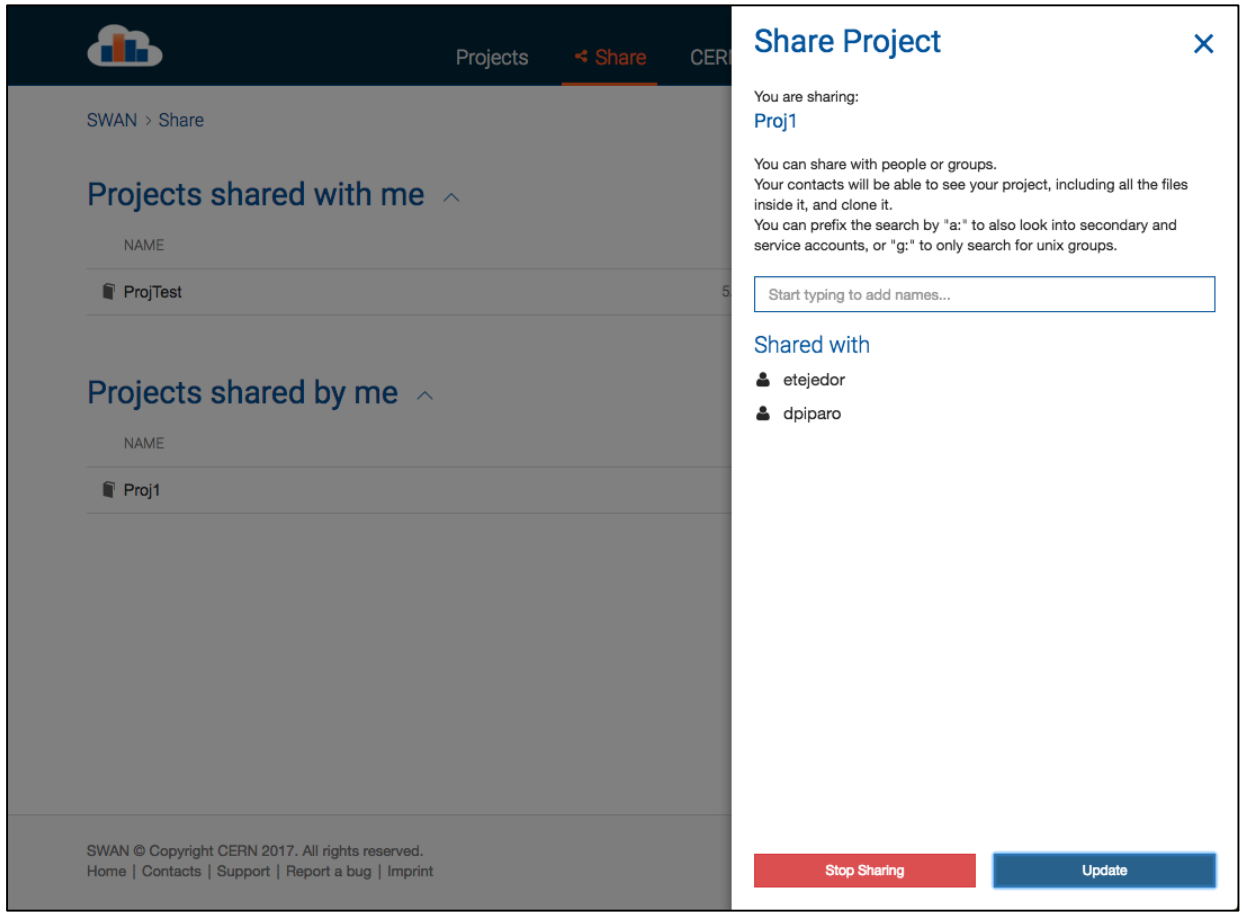
New User Interface





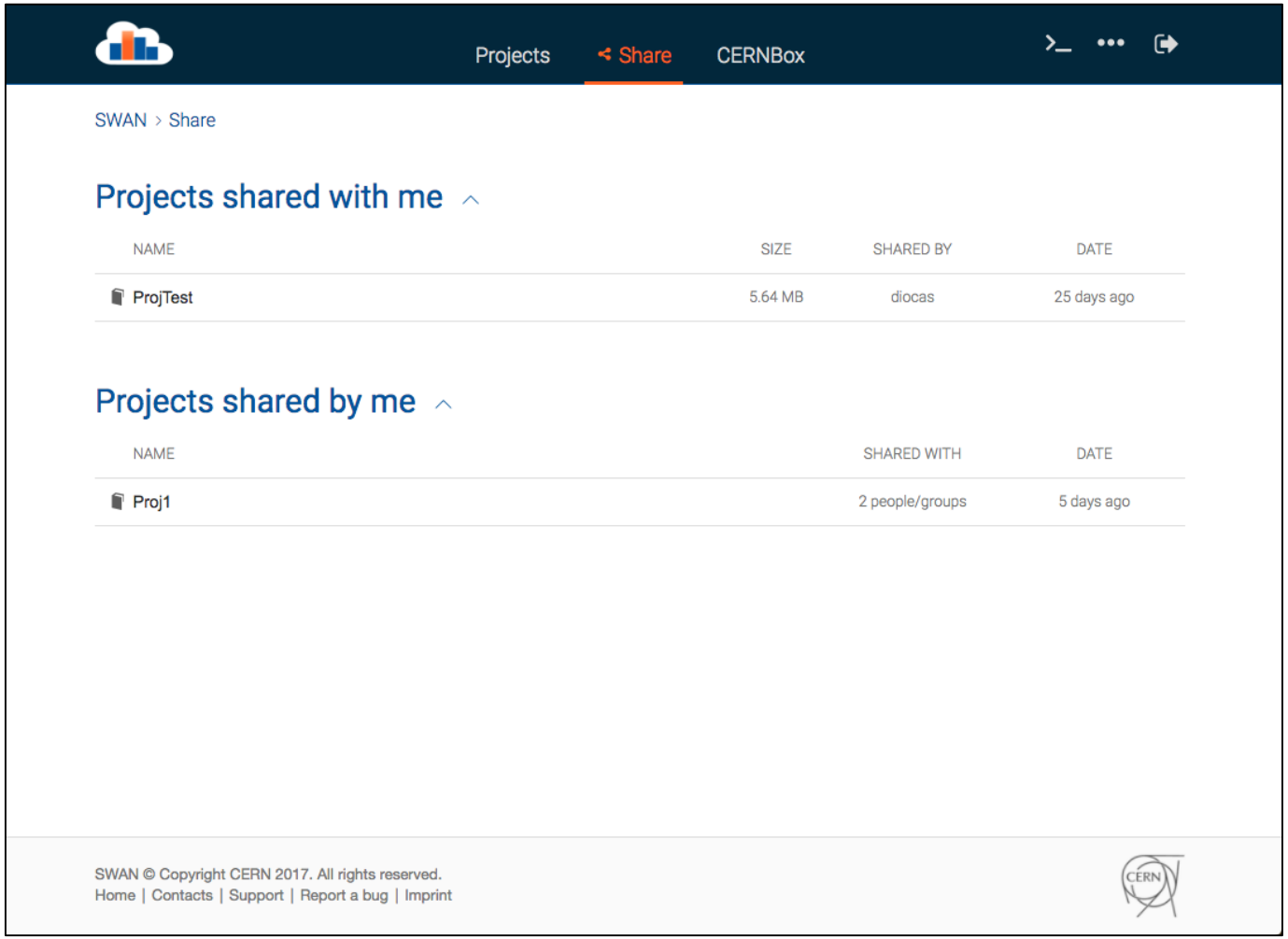
Sharing made easy

- 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



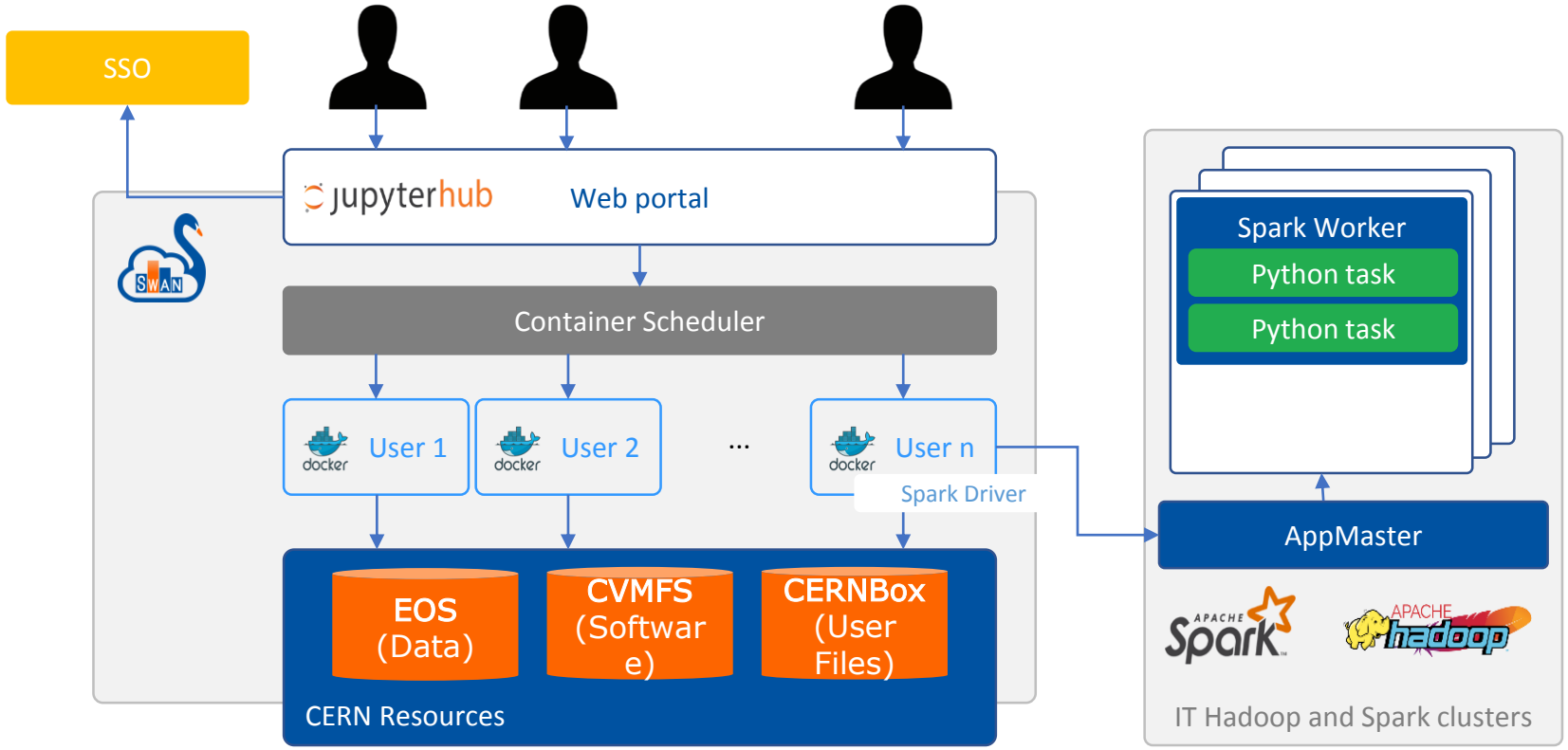
Sharing made easy

- Users can clone a shared Project directly from the interface
  - Jupyter doesn't allow concurrent editing

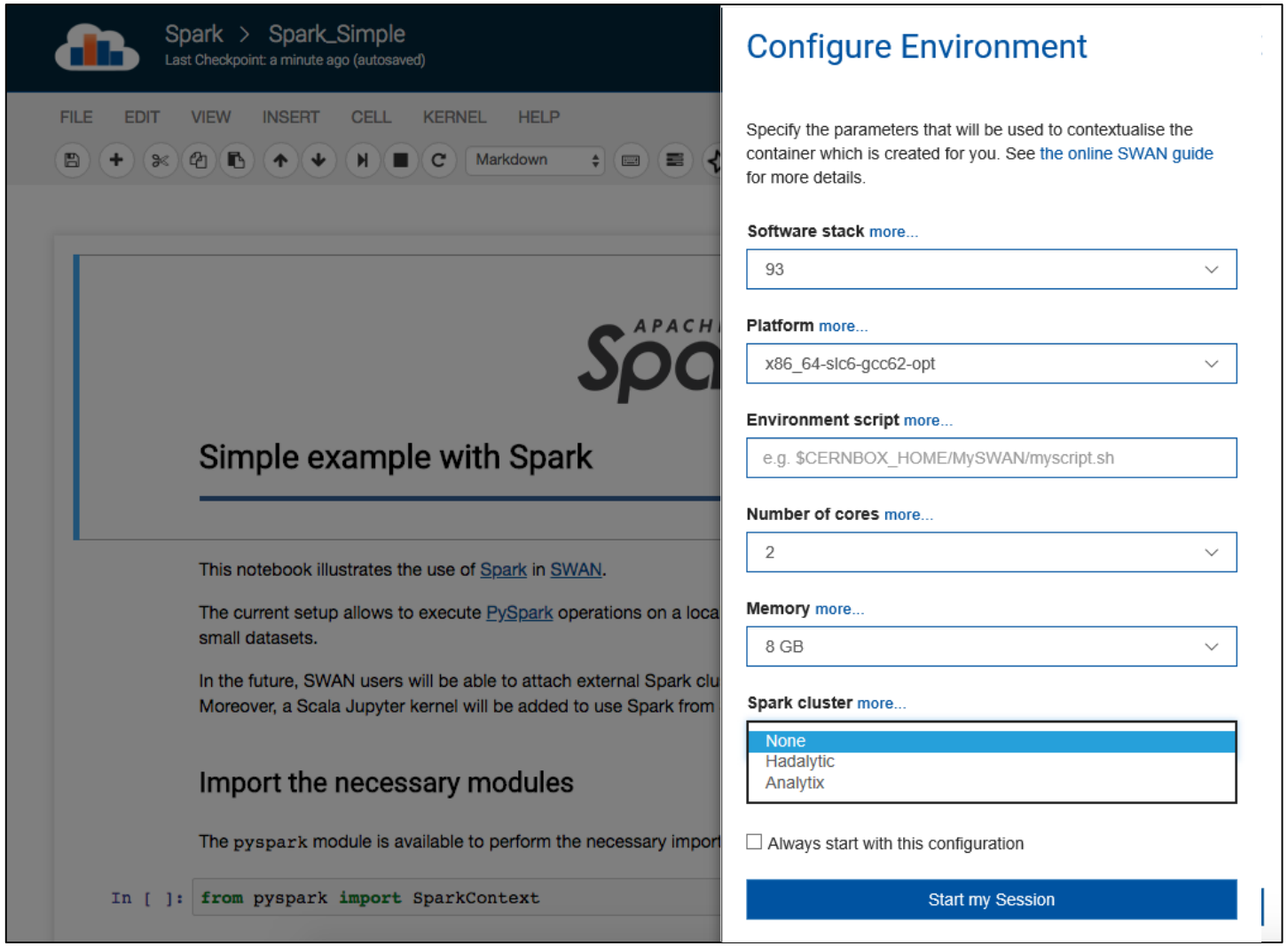


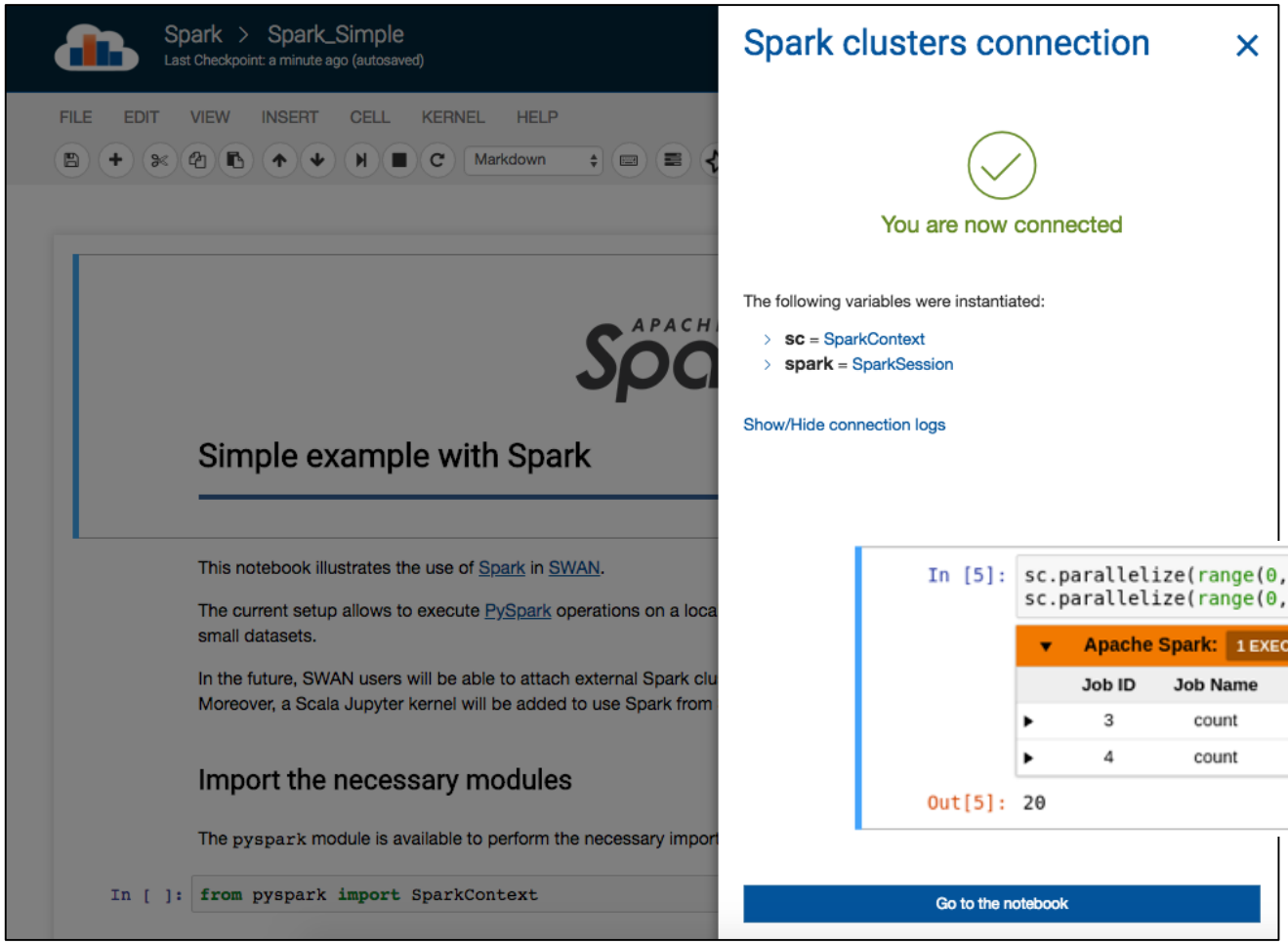
Integration with Spark

- Connection to CERN Spark Clusters
- User data accessed through EOS
- Graphical Jupyter extensions developed
  - Spark Connector
  - Spark Monitor

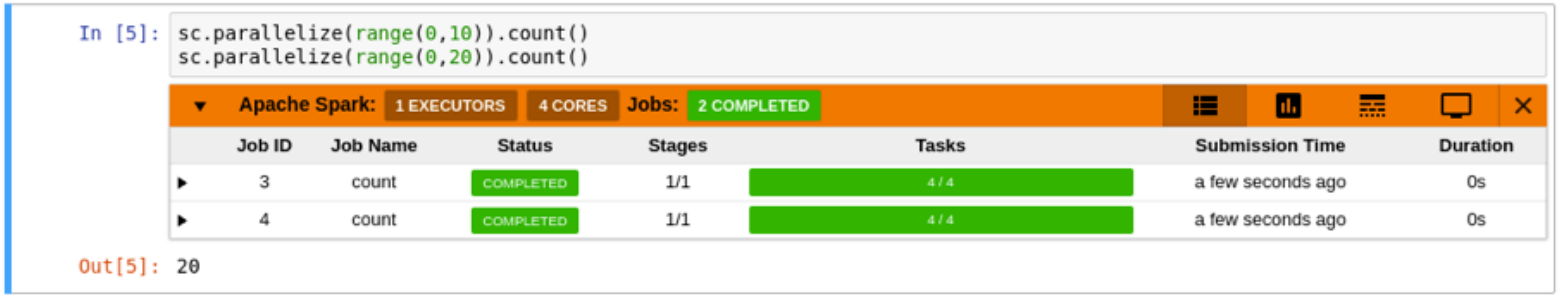


Integration with Spark





- Spark Monitor – jupyter notebook extension
  - For live monitoring of spark jobs spawned from the notebook
  - Access to Spark WEB UI from the notebook
  - Several other features to debug Spark application



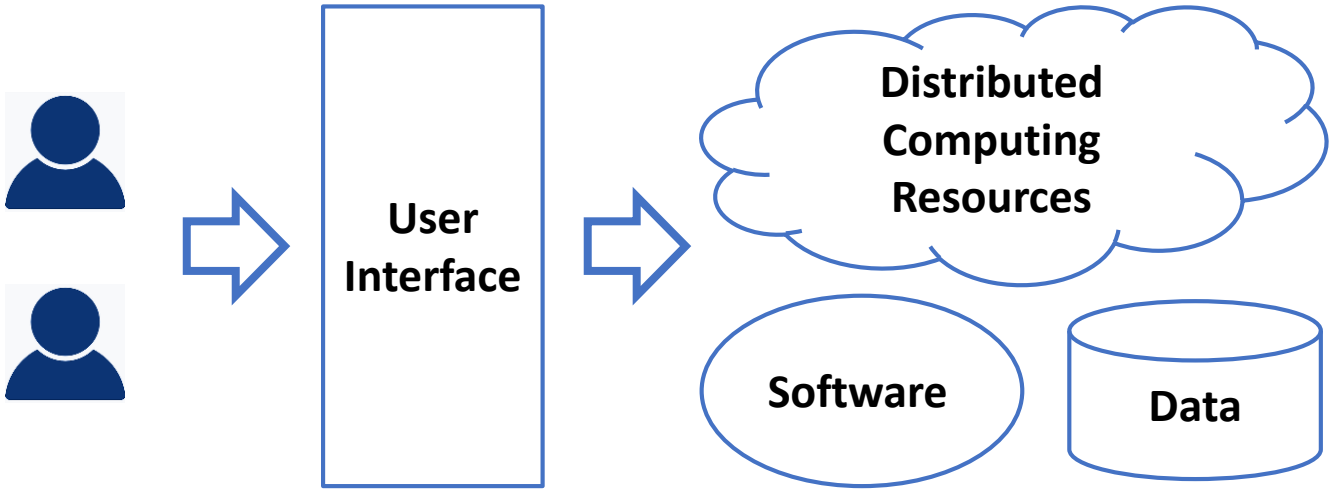
Further Research and Development

Scaling what you can do with your laptop

- **Batch/Grid jobs** has been a very successful model for distributed analysis in HEP, but:
  - Wrapper code / scripts necessary to perform submission
  - Not interactive: cannot view intermediate application results
  - Hard to monitor progress / estimate time to completion
  - Hard to debug and analyze performance
- As soon as they can, physicists work on **ntuples that fit in their computer**
  - Simpler, more user-friendly, interactive, full control of the resources
  - Will this still be possible for HL-LHC?

Interactive Distributed Analysis Environment

- **Easy entry** point to distributed execution
- **High-level** programming model
  - Declarative
  - No changes to go distributed
  - Interactive, easily modifiable
- Pluggable computational **HPC/cloud** resources
- Leverage on **big data** technologies from industry



Conclusion

Conclusion

- SWAN is a CERN service that provides Jupyter Notebooks on demand
- SWAN promotes a cloud based analysis model where users can do analysis only with their browser
- SWAN federates CERN services for software, storage and infrastructure so that users can find what they need in the service
- SWAN fosters collaboration and results sharing between scientists
- SWAN is an interface for large computing resources (Spark)
- SWAN is the first step towards a **truly scalable and interactive** distributed data analysis environment

SWAN: service for web based analysis

Thank-you