SPARK+AI
SUMMIT EUROPE

CERN´s Platform for Data Analysis with Spark

Enric Tejedor, Prasanth Kothuri, CERN

#SAISExp1

CERN

Outline

1. What we do at CERN
2. Interactive data analysis with Spark
3. Use cases
   1. Physics data
   2. Controls data

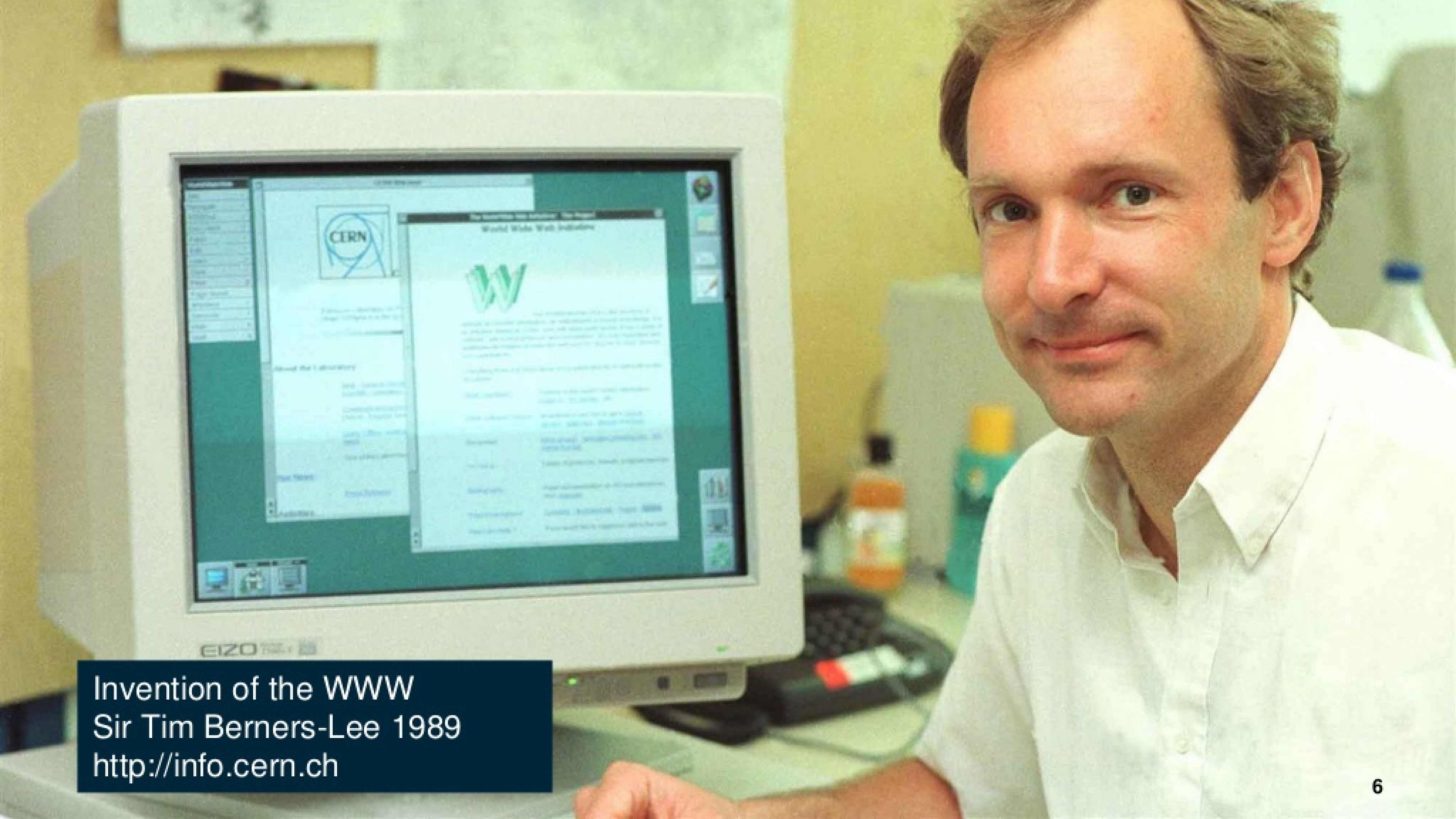
Founded in 1954
Mission: fundamental research in Physics

CERN

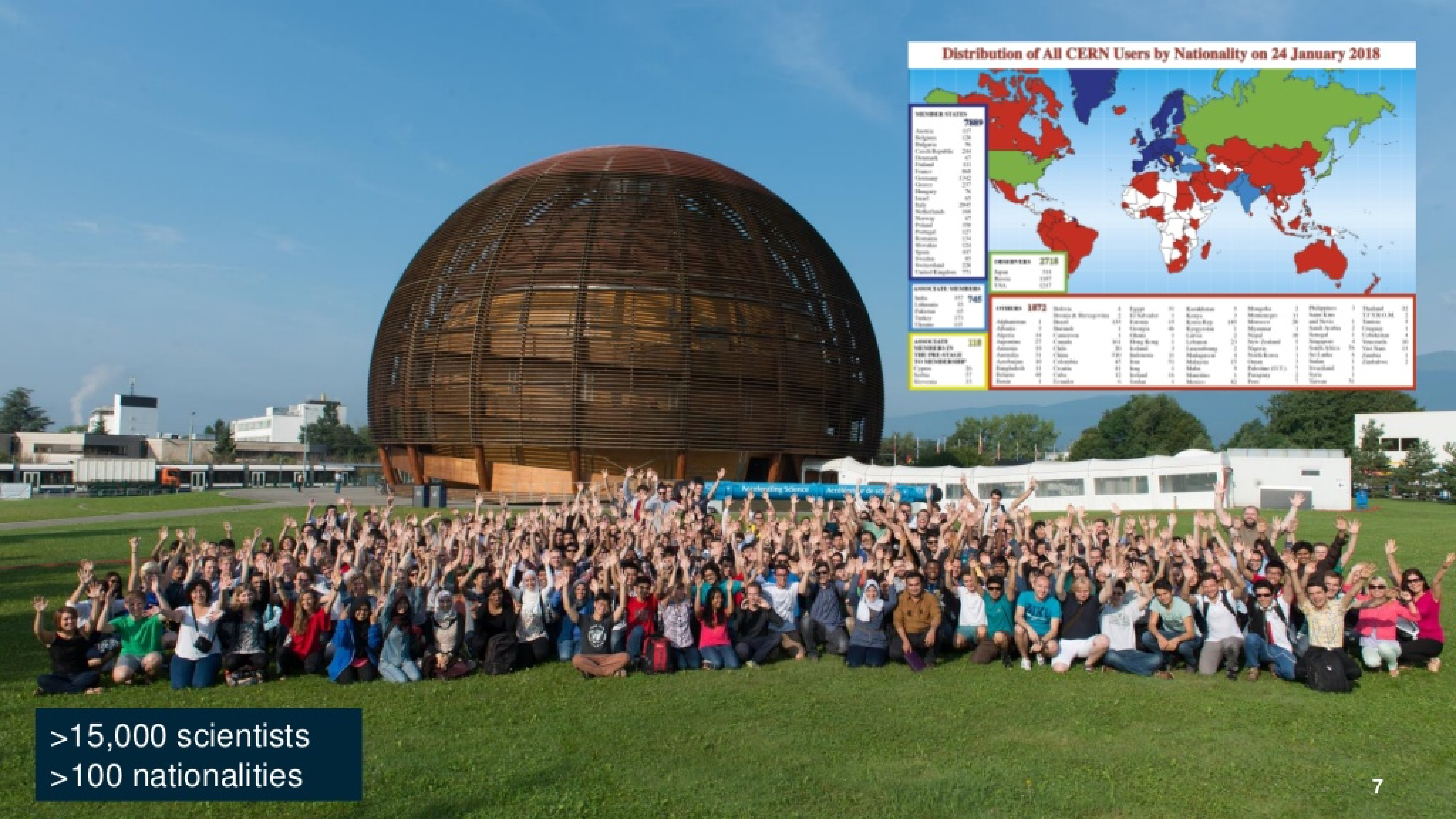
Accelerating Science

The **Large Hadron Collider** (LHC)
The largest scientific experiment to date
1 PB / second of raw physics data

Discovery of the Higgs boson
ATLAS & CMS 2012



Invention of the WWW
Sir Tim Berners-Lee 1989
http://info.cern.ch

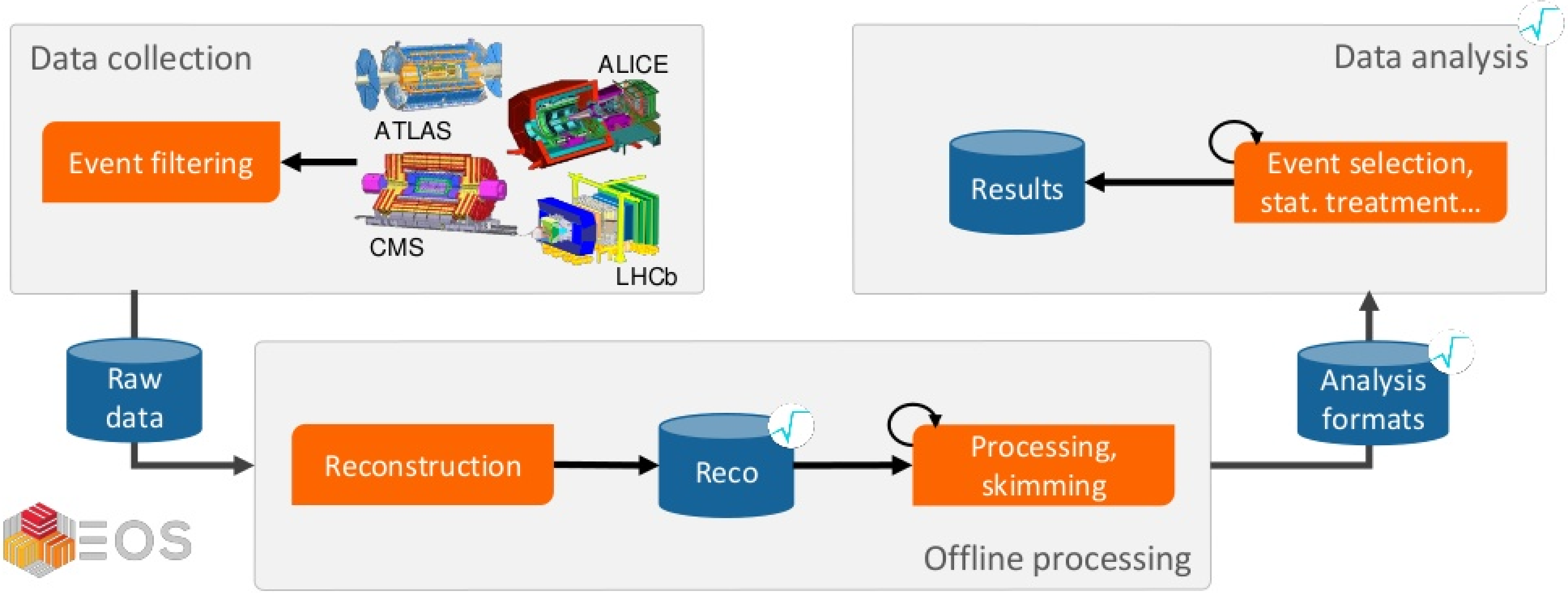


>15,000 scientists
>100 nationalities

Physics Data Processing and Analysis at CERN

#SAISExp1

LHC Data Pipeline at CERN

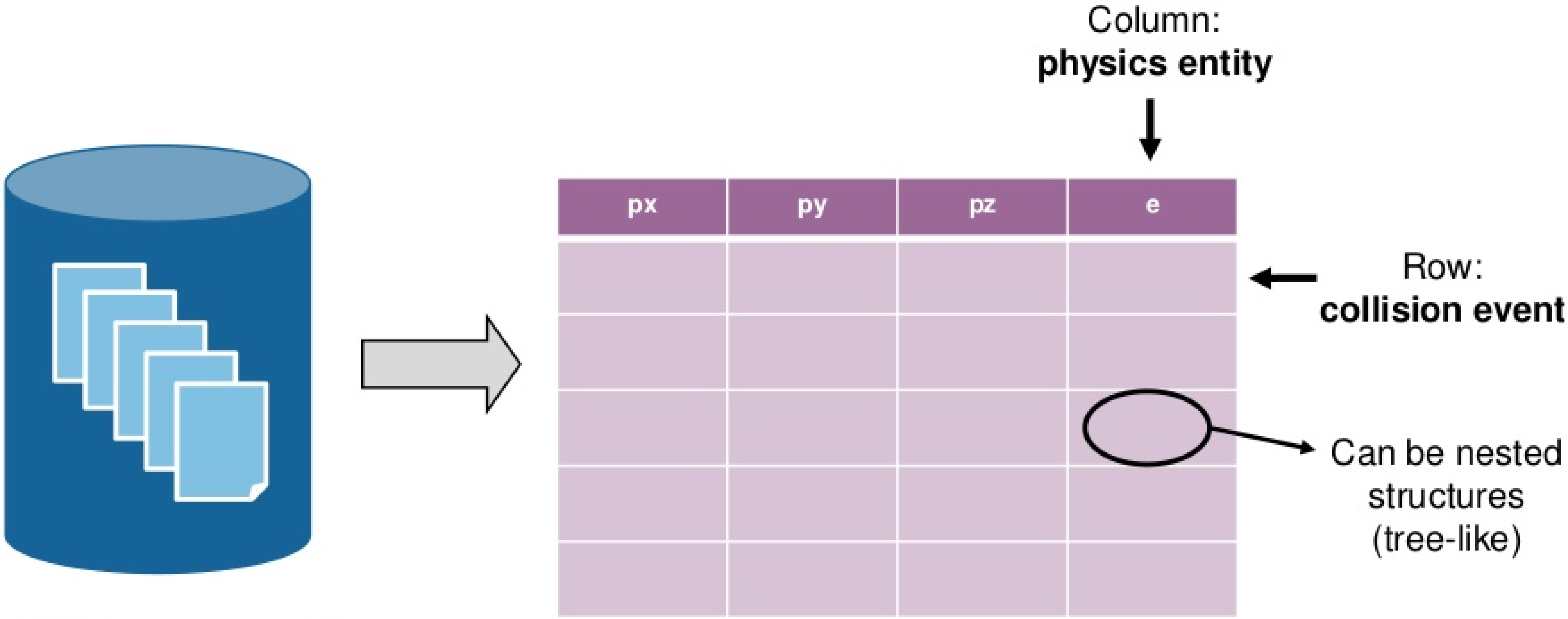


ROOT

- **The** data analysis framework for high-energy physics (HEP)
- Data processing, statistical analysis, visualisation, I/O and storage
- **~1 EB** stored in ROOT format
  - Binary, compressed
  - Columnar

https://root.cern

ROOT Dataset



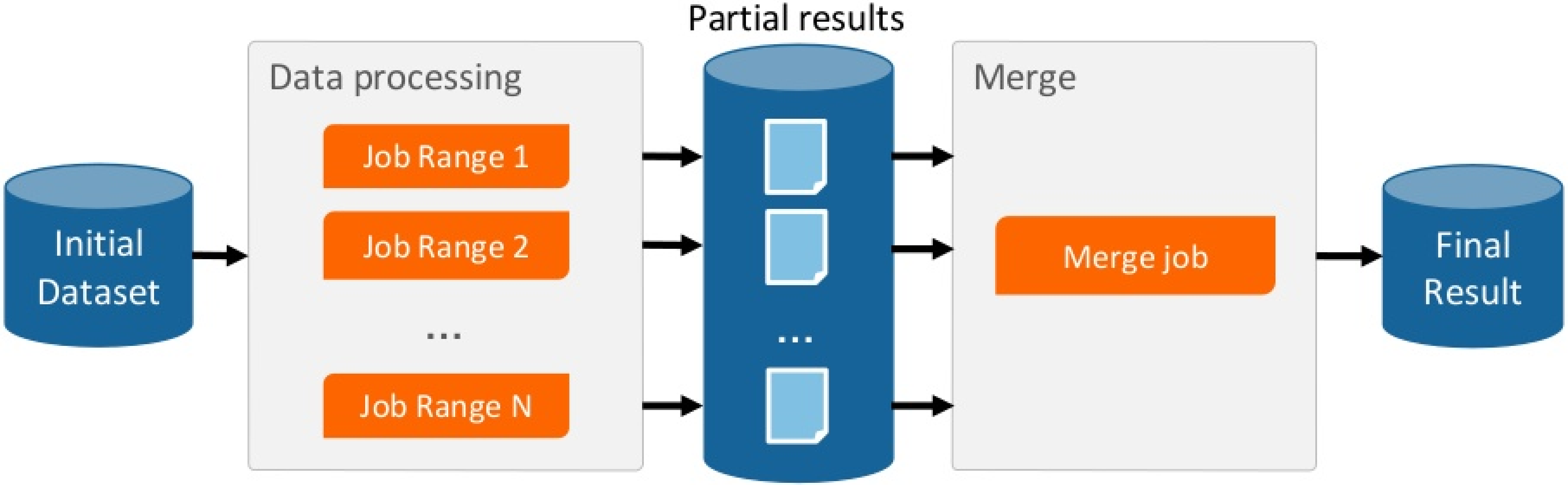
ROOT dataset stored in
one or multiple files

The LHC Computing Grid

Started in 2002
Provide processing power and data access to physicists
~170 centres in 42 countries
Running 24/7/365

Distributed Computing in HEP

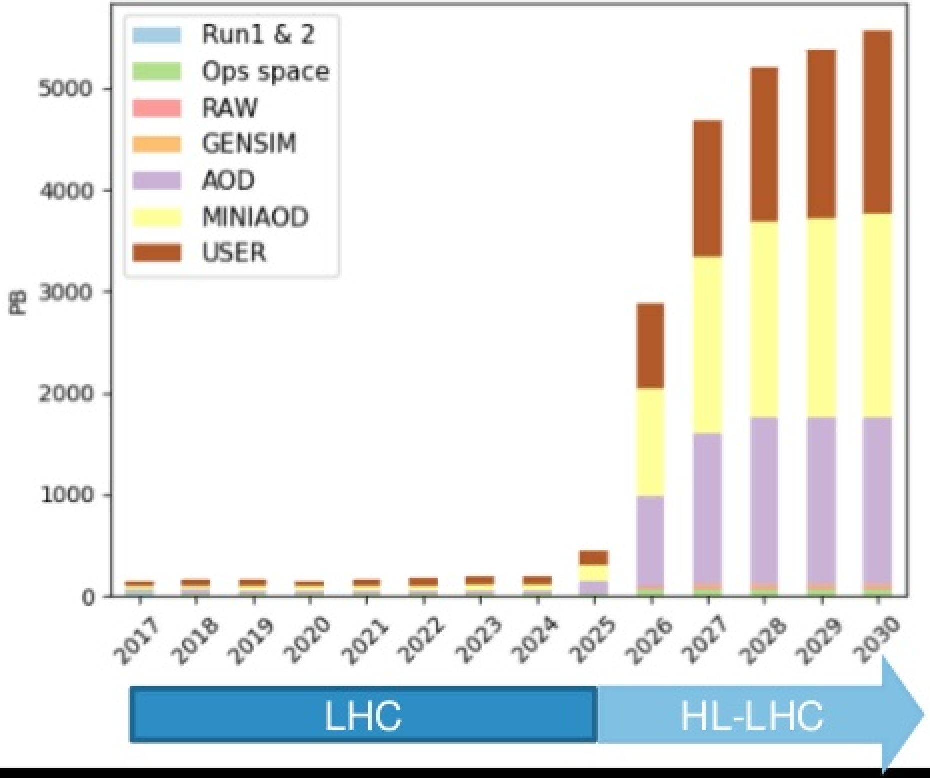
- Physicists use Grid and batch resources to process LHC data in parallel



HL-LHC: Even More Data!

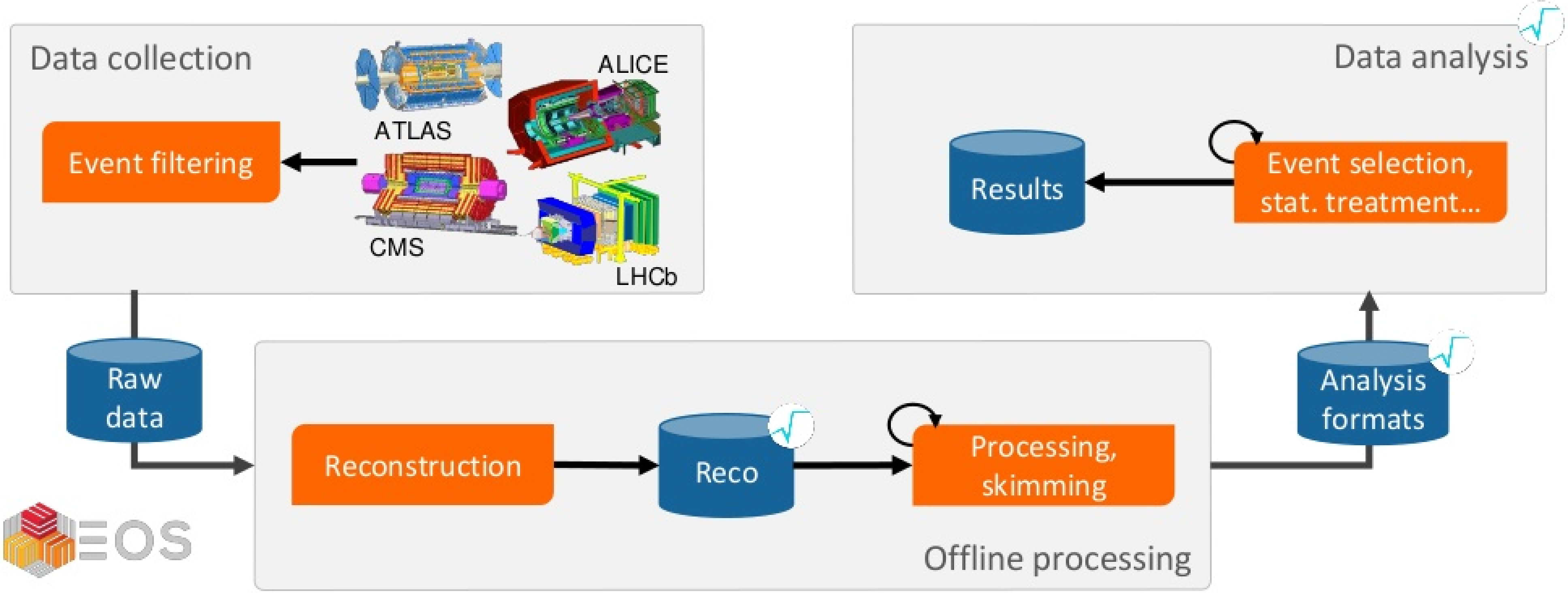
- Coming upgrade: High-Luminosity LHC
- **30x more data** collected
- Big challenge for software and computing

CMS estimated disk space required by tier

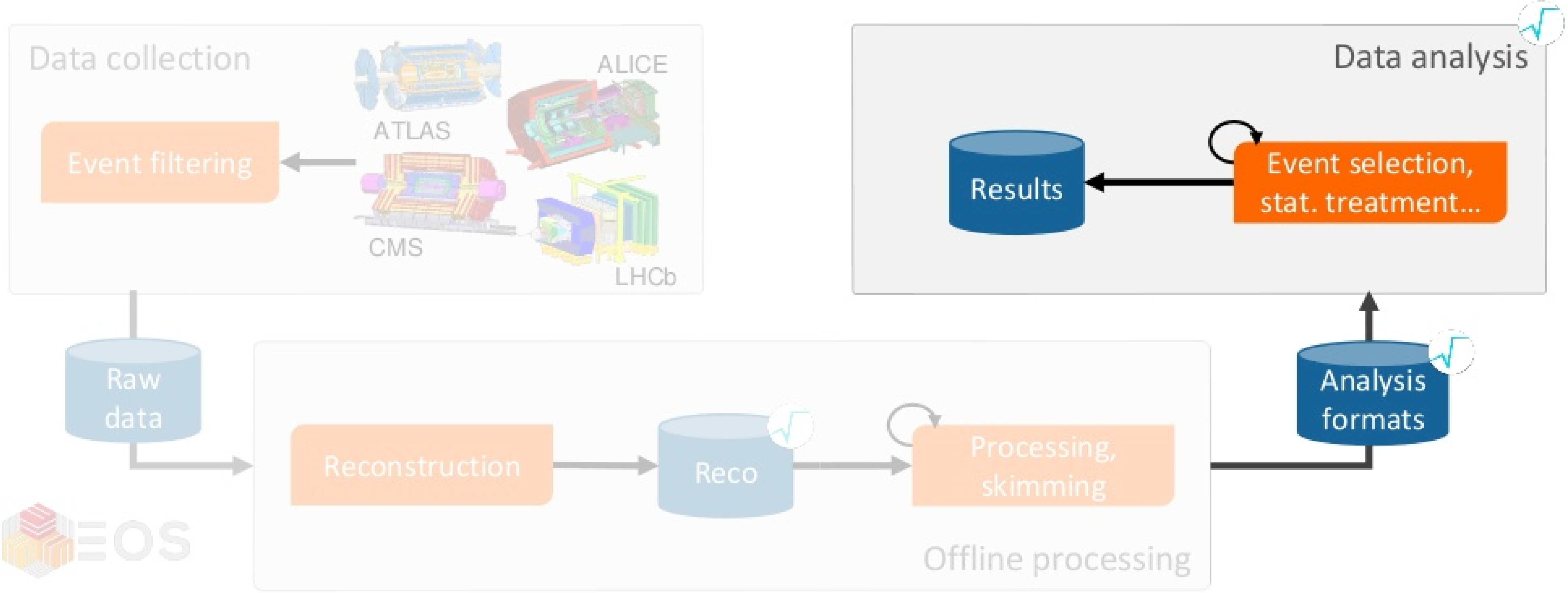


Source: HSF community white paper

LHC Data Pipeline at CERN



Final Steps of Data Analysis



Interactive Data Analysis

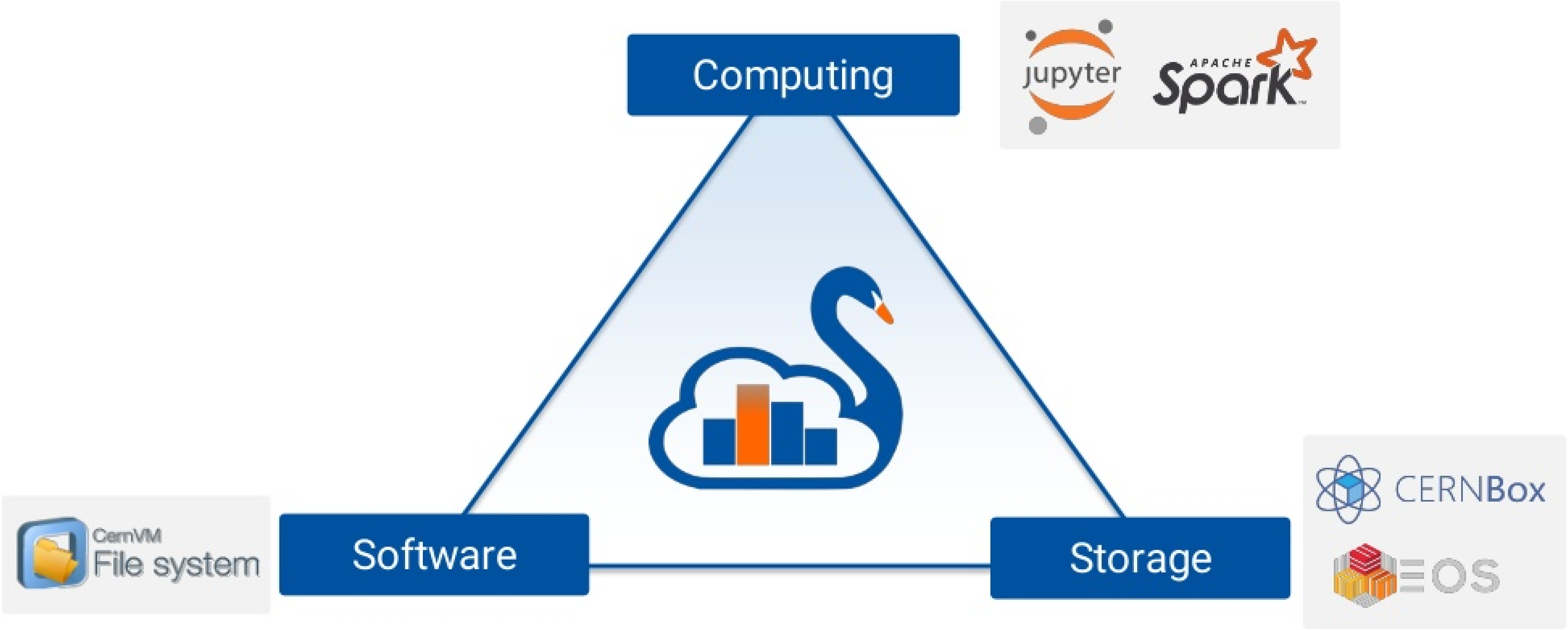
##SAISExp1

The SWAN Service

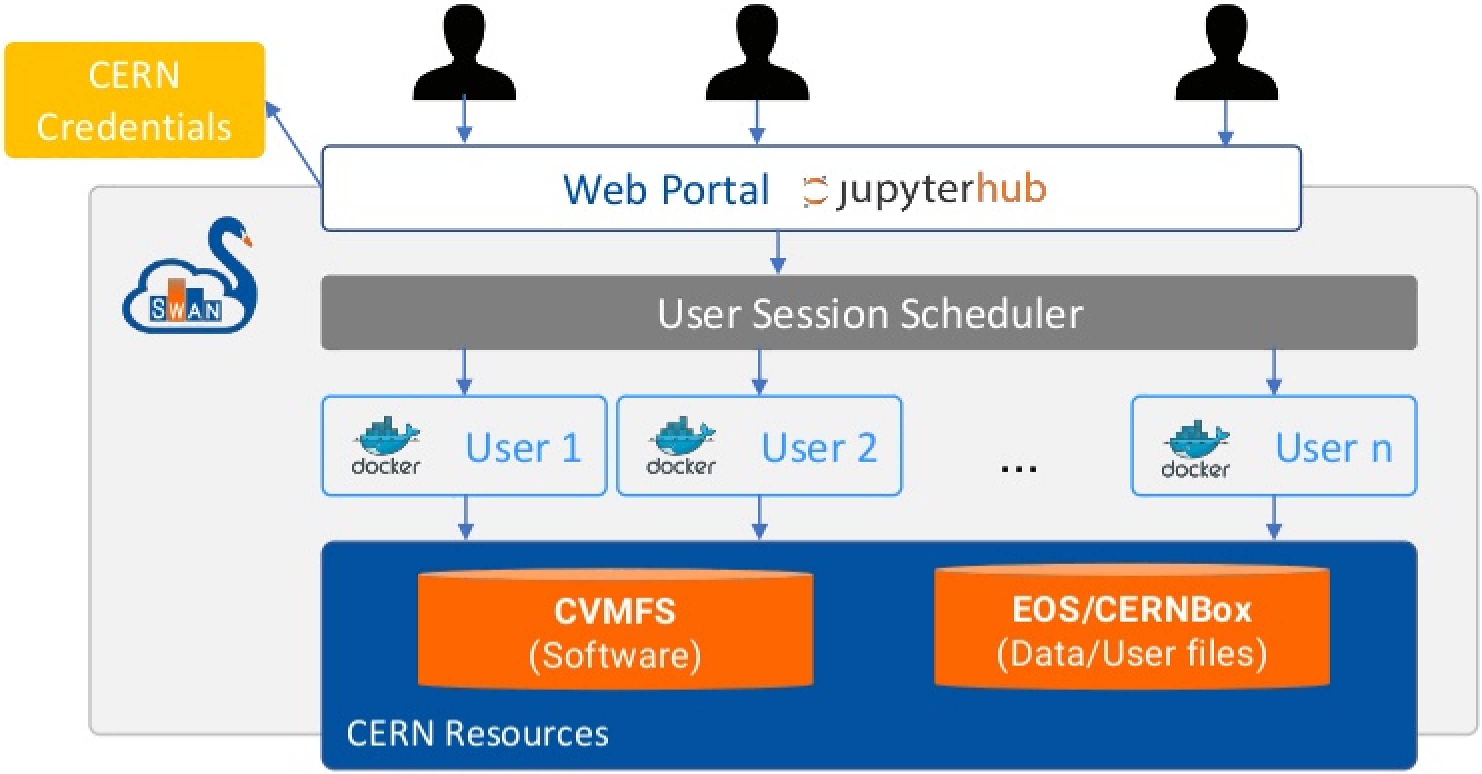
- **SWAN**: Service for Web-based Analysis
- **Interactive computing** platform for scientists
  - Based on Jupyter notebooks
- Analysis with only a web browser
- Easy **sharing** of results
- Integrated with CERN resources
  - Storage, software and computing

https://swan.web.cern.ch

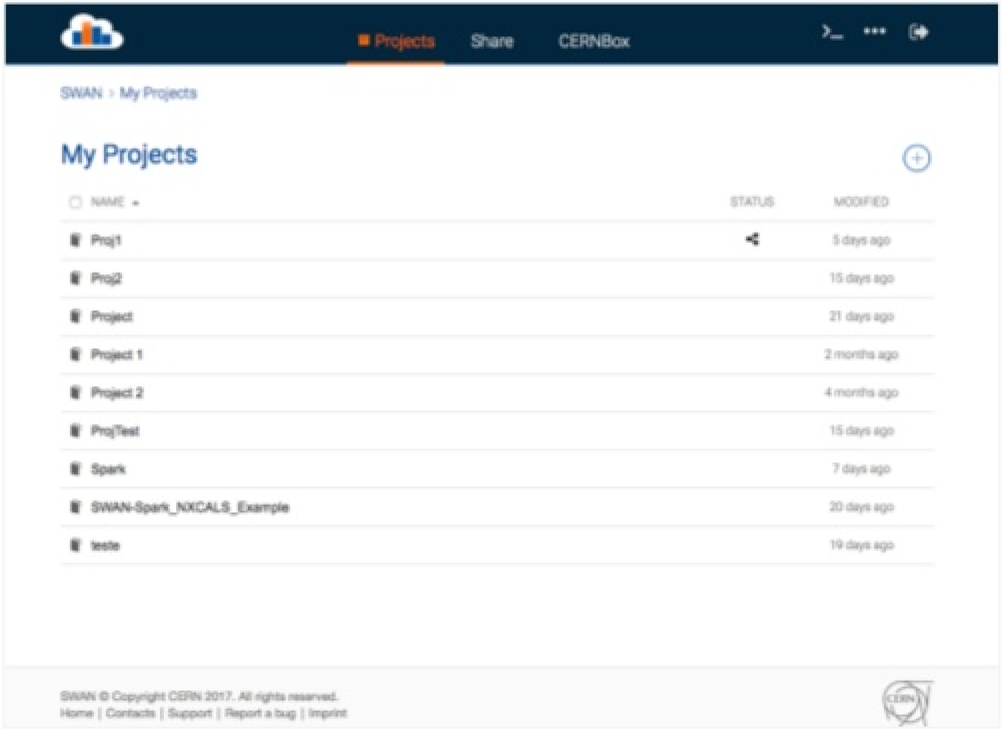
SWAN Pillars



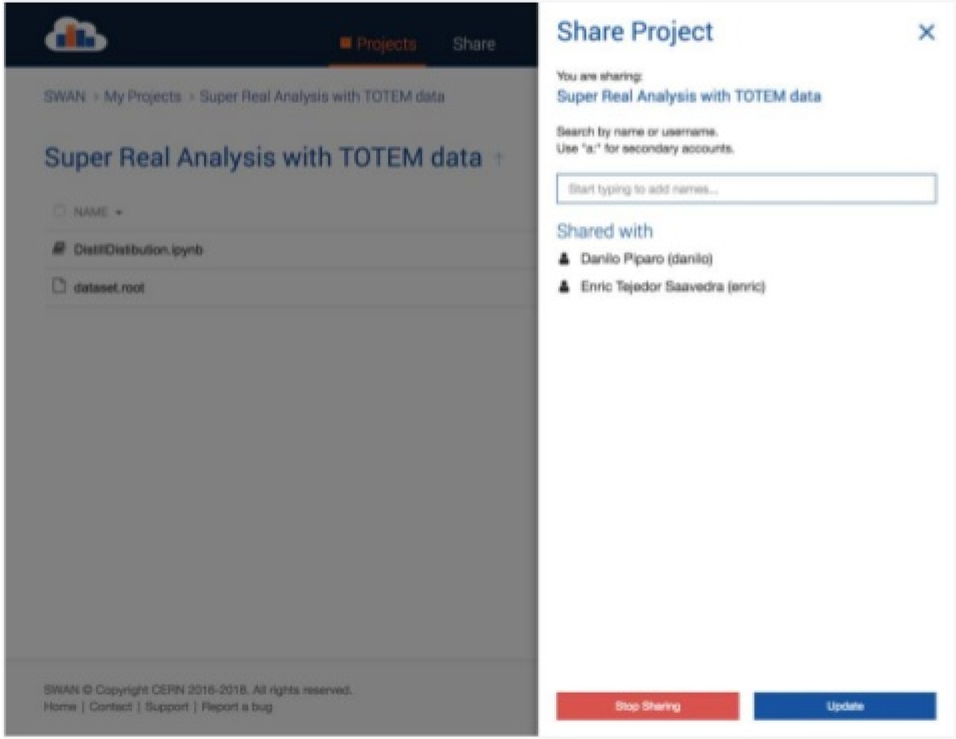
SWAN Architecture

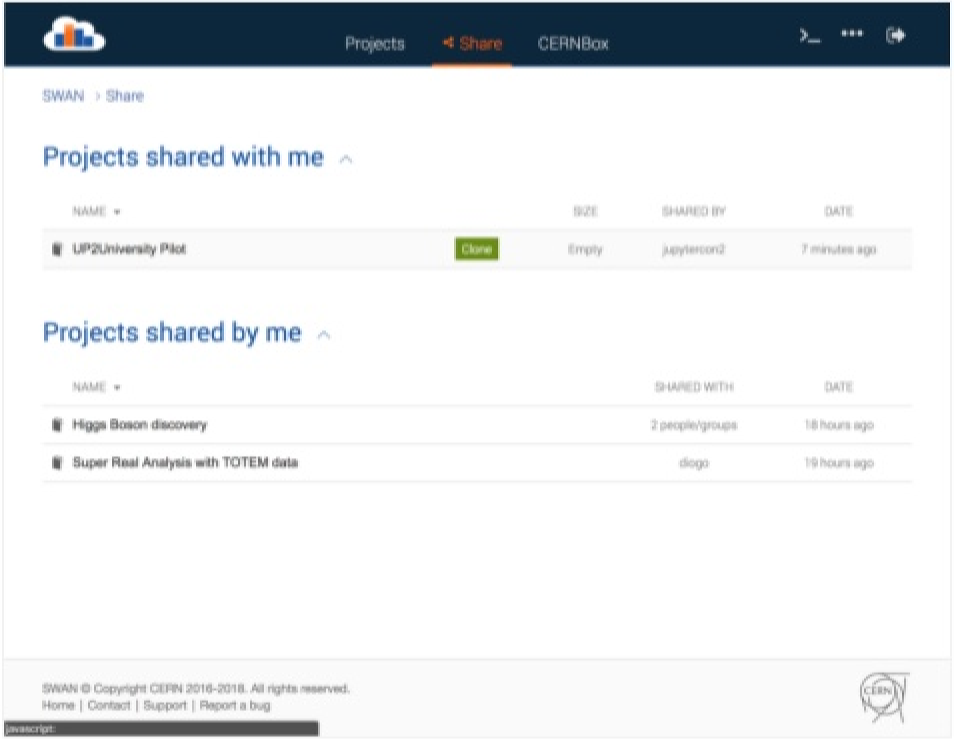


SWAN Interface: Notebooks

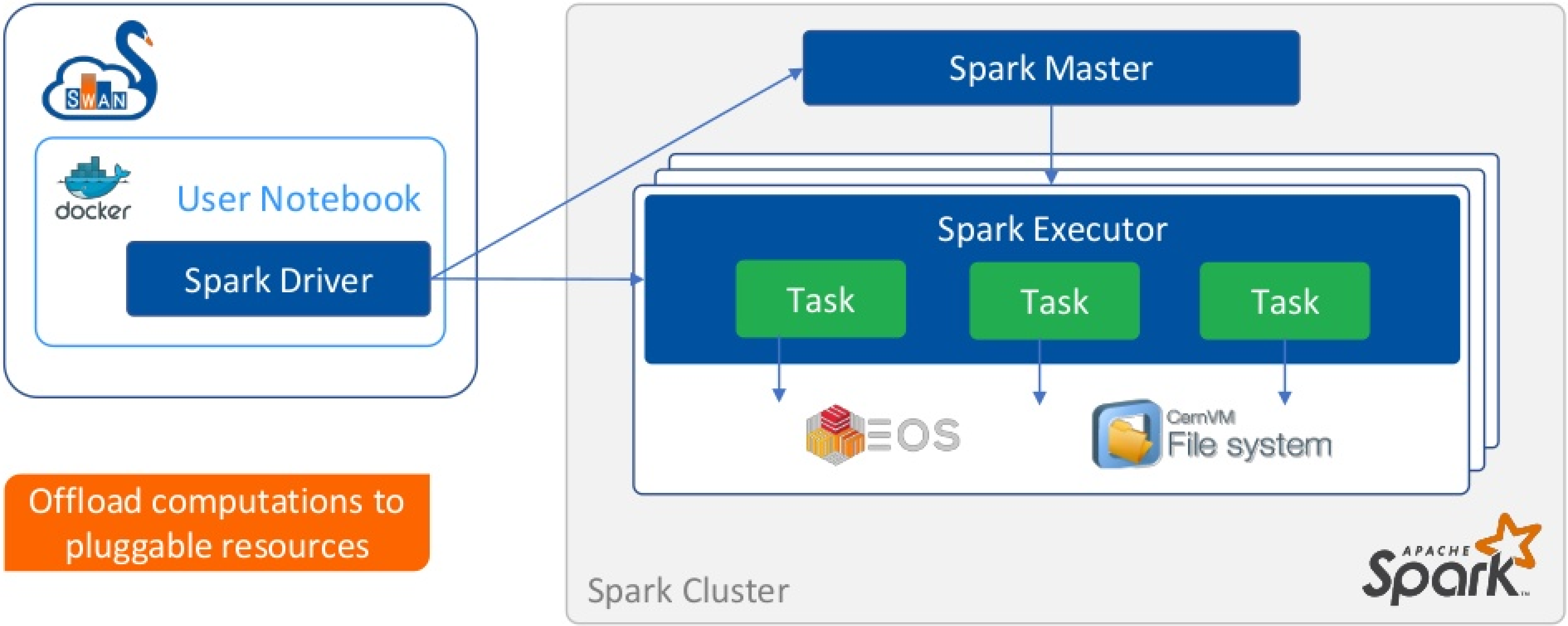


Sharing in SWAN



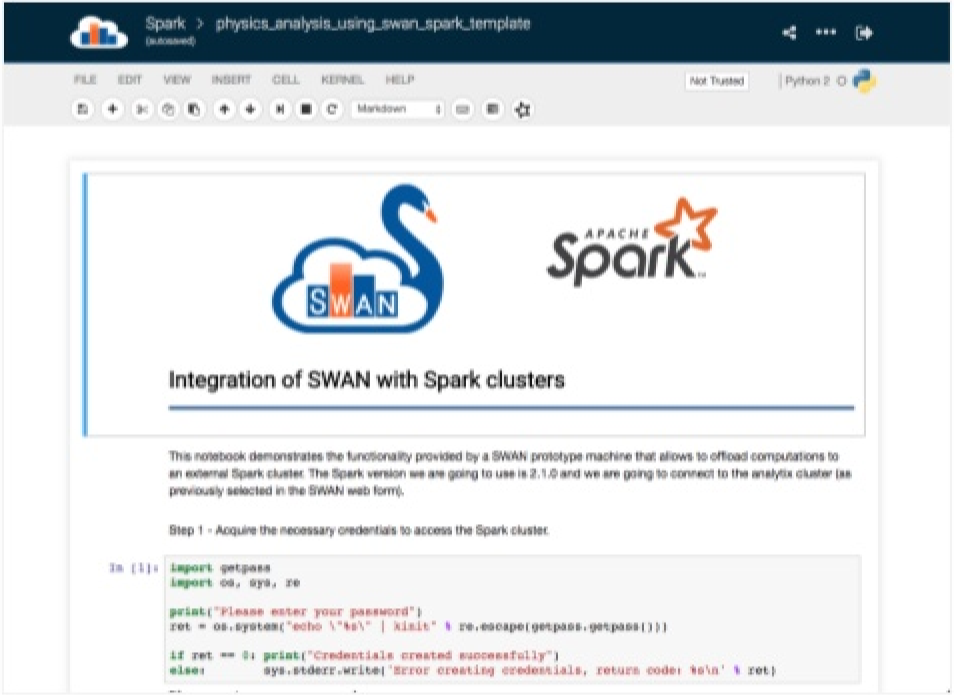


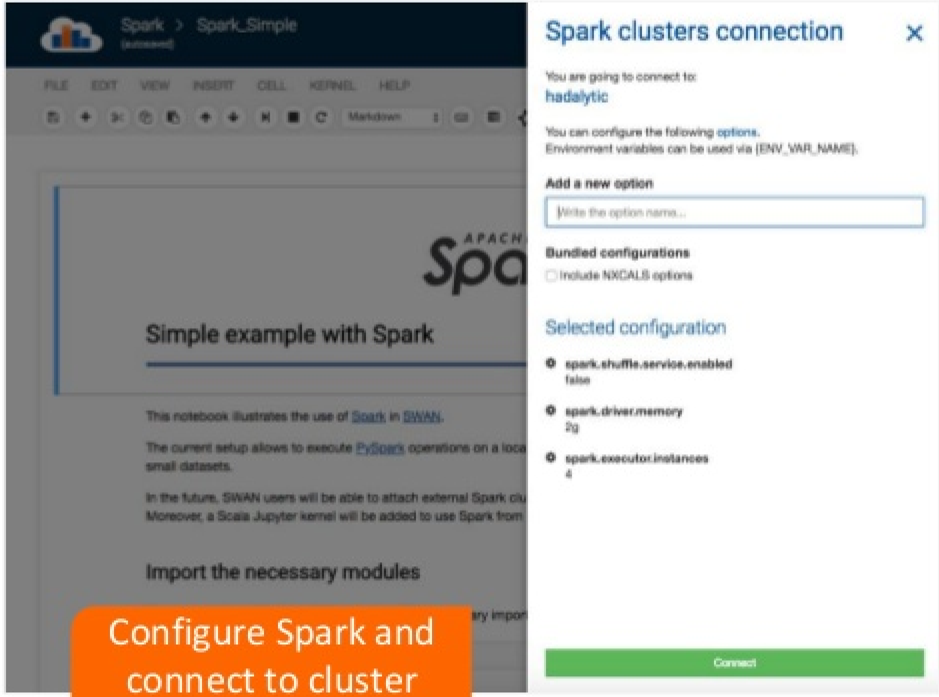
Integration with Spark



Offload computations to pluggable resources

Spark Connector





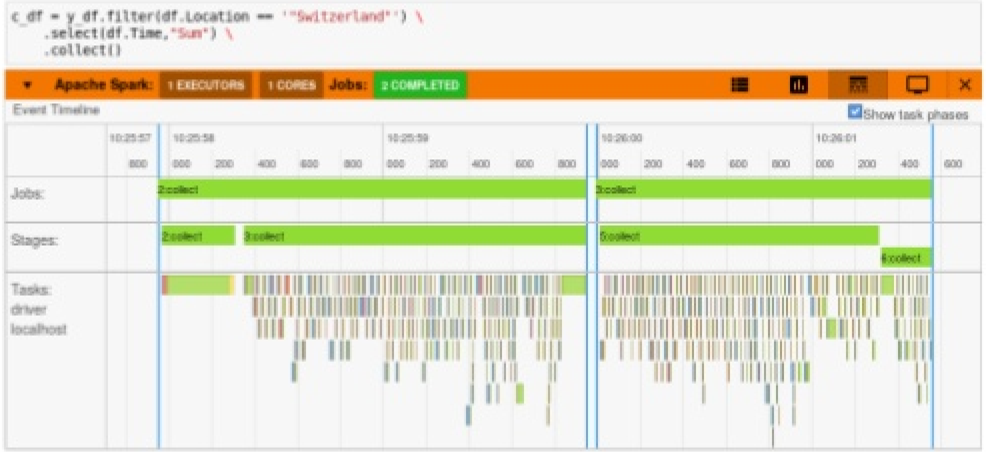
Configure Spark and connect to cluster with a click

Spark Monitor

[Code here!](#)

Google Summer of Code

- Bridge the gap between interactive computing and distributed data processing
- Automatically appears when a Spark job is submitted from a cell
- Progress bars, task timeline, resource utilisation



Physics Data Use Case

#SAISExp1

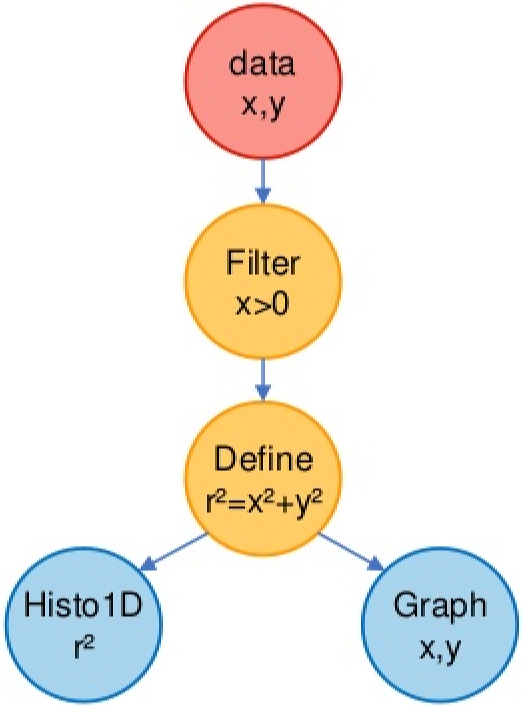
The HEP DataFrame

- **RDataFrame**
  - Implemented in C++, interfaced also to Python
  - Tailored for ROOT and HEP

```python
df = RDataFrame(dataset)

df2 = df.Filter('x > 0')
        .Define('r2', 'x*x + y*y')

h = df2.Histo1D('r2')
g = df2.Graph('x', 'y')
```

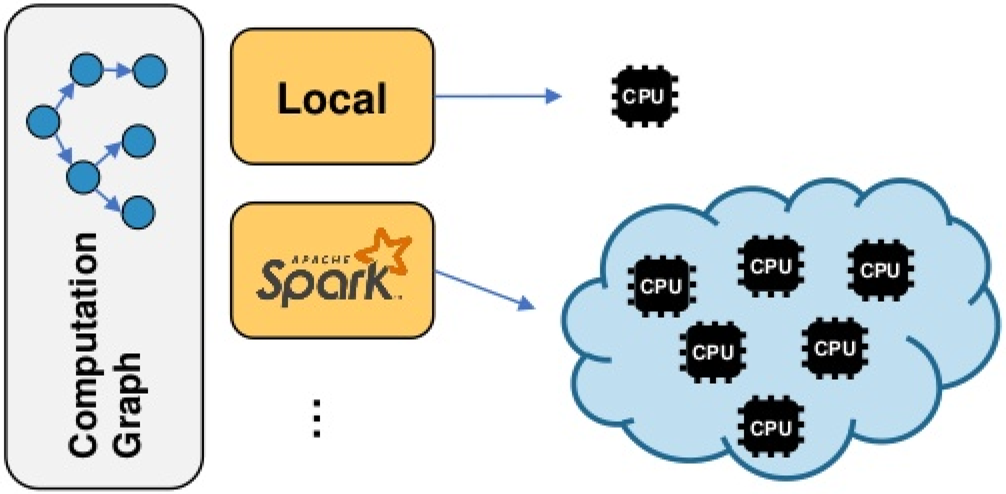


Distributed RDataFrame

- Exploratory work to parallelise RDataFrame computations with multiple backends

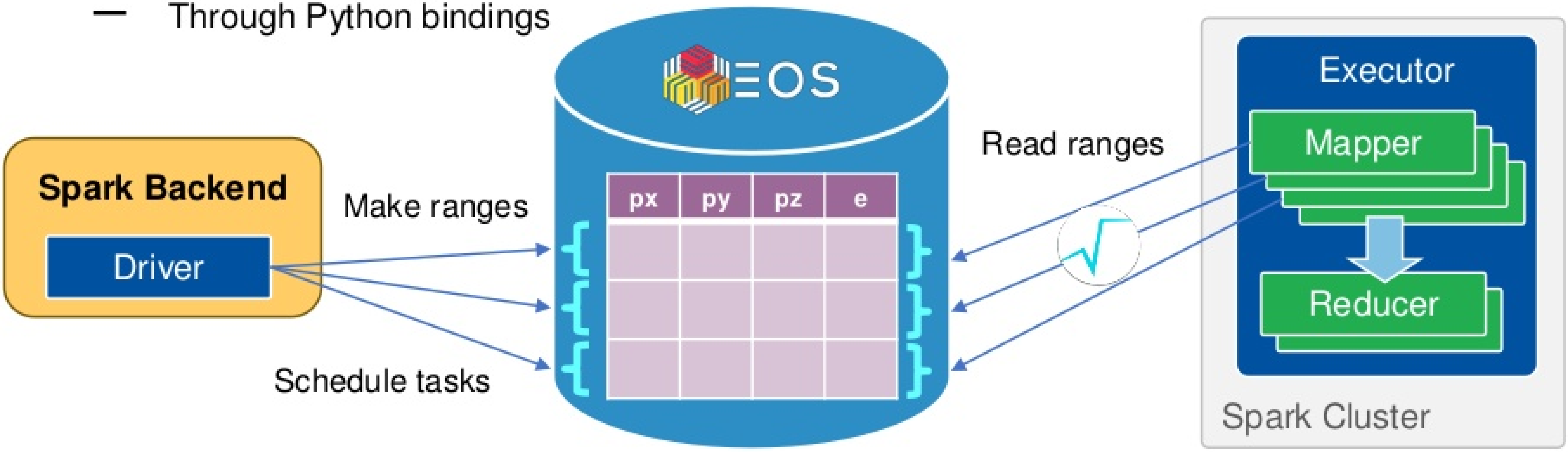
```
d = RDataFrame(dataset)
f = d.Define(...)
     .Define(...)
     .Filter(...)

h1 = f.Histo1D(...)
h2 = f.Histo2D(...)
g   = f.Graph(...)
```



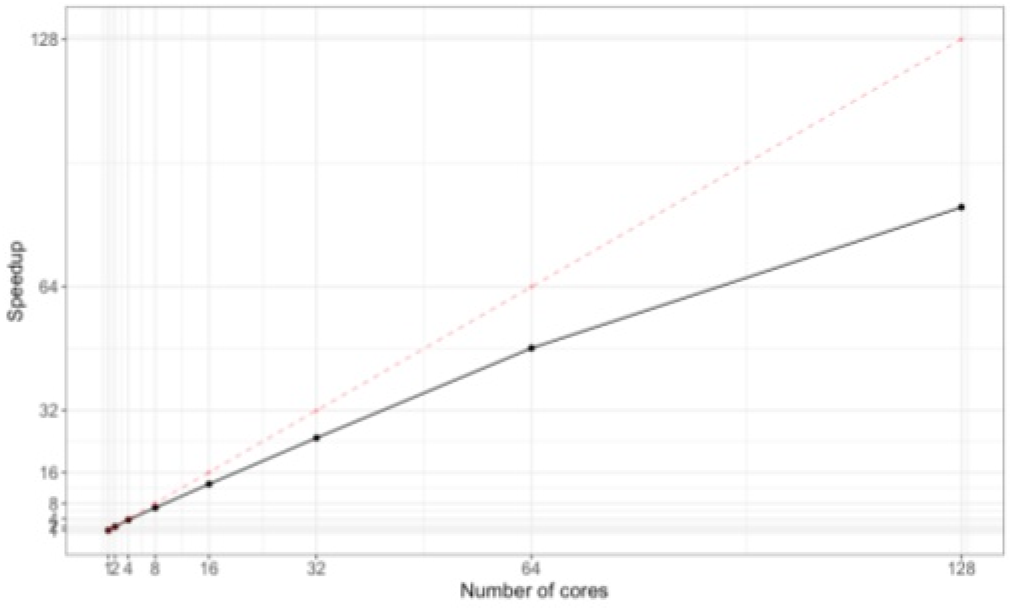
Spark Backend for RDataFrame

- **Map-reduce** workflow where every mapper runs the RDataFrame computation graph on a **range** of collision events
- Run **analysis in C++ with Spark**
  - Through Python bindings



Real Example: TOTEM Analysis

- TOTEM experiment analysis coded with RDataFrame
- **Spark** backend
- **4.7 TB** dataset on EOS
- Launched from **SWAN** to a dedicated Spark cluster
- **Get to physics results faster!**



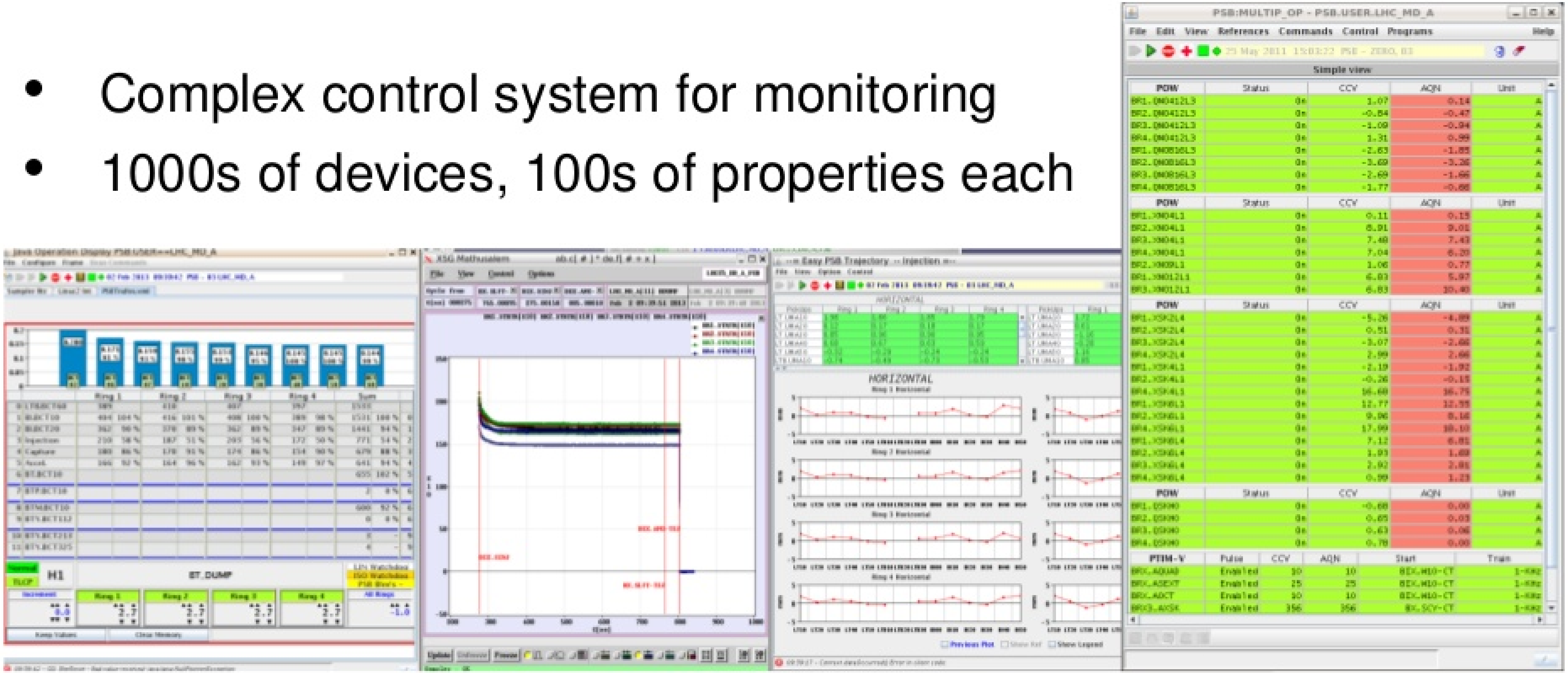
Controls Data Use Case

#SAISExp1

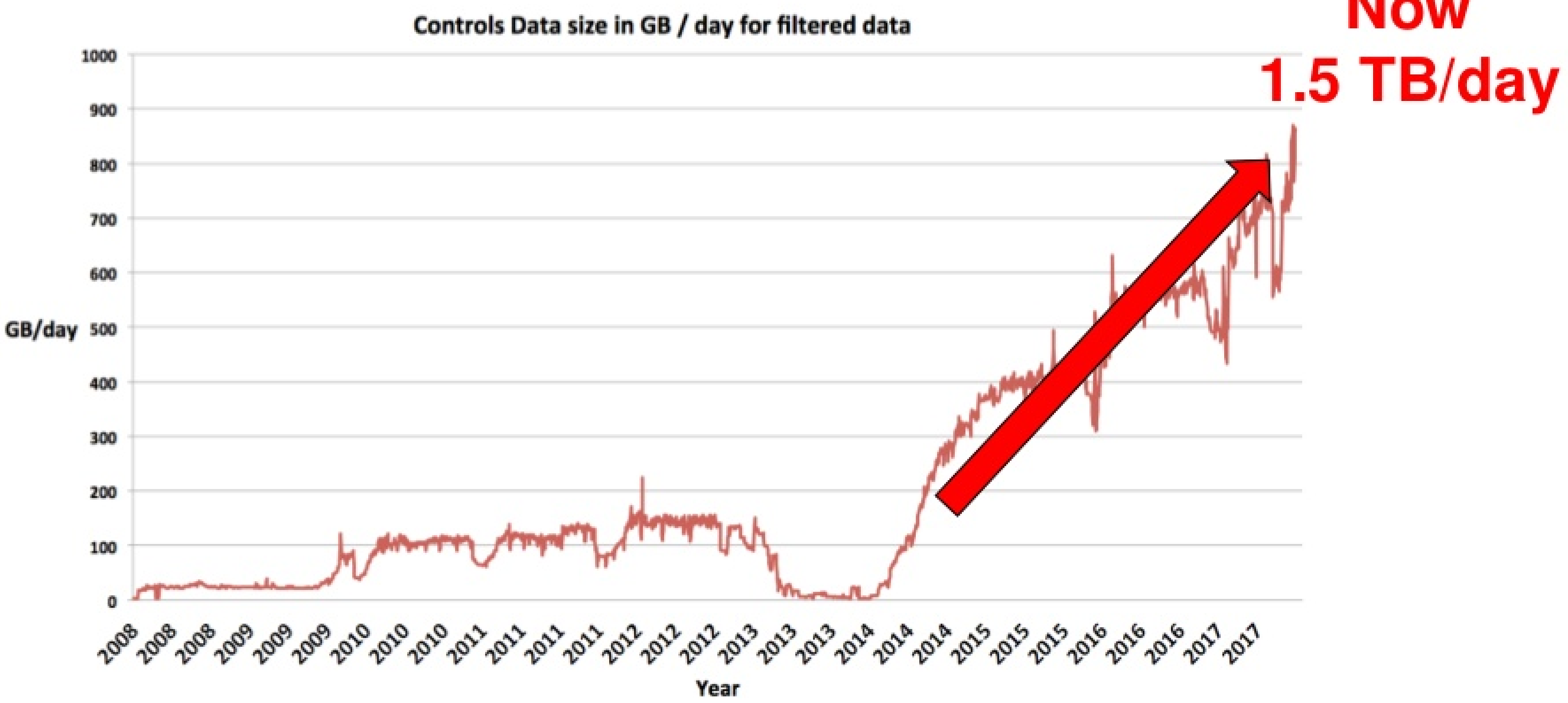
LHC: Huge machine, highly sophisticated
Control and monitoring are crucial

Hardware Controls

- Complex control system for monitoring
- 1000s of devices, 100s of properties each



Controls Log Data Growth



CERN Accelerator Logging Service

- Old system based on SQL databases
  - Hard to scale horizontally
  - Slow data extraction
- New system (NXCALS)
  - Data pumped into HBase and HDFS (Parquet)
  - **Spark** to extract and process data
  - **SWAN** to visualise + analyse

NXCALS Data Analysis in SWAN

- NXCALS rely on SWAN as their data analysis platform
- Connection to Spark clusters
- Access to software (data science Python ecosystem)

Inspect data

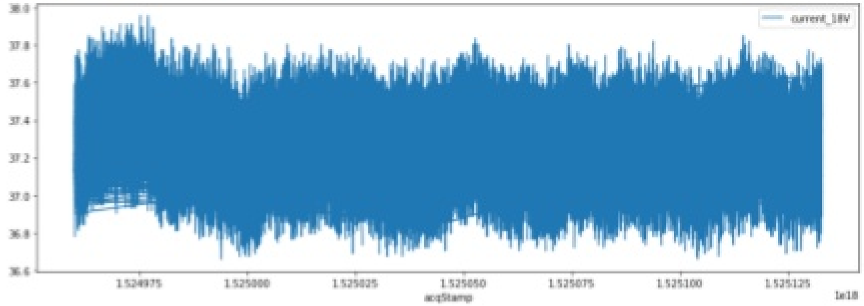
```
In [2]: dfl.select('acqStamp','voltage_18V','current_18V','device','pt100Value').toPandas()[:5]
Out[2]:
```

	acqStamp	voltage_18V	current_18V	device	pt100Value
0	1524960103132865000	NaN	37.301794	RADMON.PS-10	106.578911
1	1524960284134564000	NaN	NaN	RADMON.PS-10	107.246742
2	1524960322134942000	NaN	37.560940	RADMON.PS-10	106.504707
3	1524960353135244000	20.099066	NaN	RADMON.PS-10	107.068654
4	1524960911140548000	20.111261	37.698135	RADMON.PS-10	106.578911

Draw a plot with matplotlib

```
In [3]: import matplotlib
        import pandas as pd
        %matplotlib inline
```

```
In [4]: p_df = dfl.select('acqStamp','current_18V').toPandas()
        p_df.plot('acqStamp','current_18V',figsize=(15,5))
        # p_df.sort_values(by='acqStamp').plot(pd.to_datetime(p_df['acqStamp'],unit='ns'),'current_18V',figsize=(15,5))
Out[4]: <matplotlib.axes._subplots.AxesSubplot at 0x7fd8fa2bcc50>
```



Summary

#SAISExp1

Challenges

- The increase in physics and controls data volumes and complexity is **pushing software** at CERN
  - Adoption of Spark and other big data technologies still in its early stages
- **Large codebase** developed over decades
  - Cannot change overnight
- Changing the **mindset** of programmers takes time
  - Declarative analysis
  - Pushing computations to data

Future Directions

- **Bridge the gap** between data processing needs and technology evolution
  - Complement traditional ways with new strategies
- Combine **interactive analysis** with **easy access** to more processing power
  - Higher-level programming models
  - Pluggable computing resources

More on CERN and Spark: stay tuned for Luca's and Prasanth's presentations

Backup Slides

#SAISExp1

JupyterLab

- Jupyter is evolving towards a desktop-like environment
  - Notebook, terminal, file browser, editors, …
  - Highly customisable