



NXCALS in SWAN

Jakub Wozniak

BE/CO/DS



Agenda

- What is CALS & why to build NXCALS?
- Project status
- Why do we need SWAN?



CERN Accelerator Logging Service

Vital for Beam Operations!

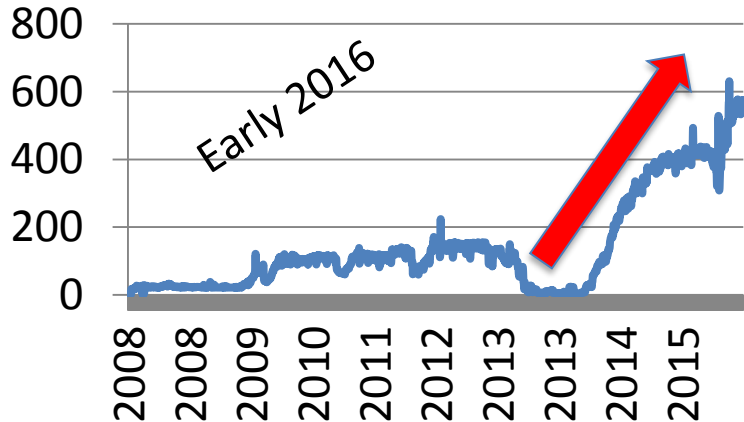
- Mandate
 - Stores data from **accelerator complex related devices**
 - Information for **acc performance improvement**
 - **Decision support** system for management
 - Avoids **duplicate logging** efforts
- CALS in numbers
 - Built for **1 TB / year** throughput
 - Currently **1.2 TB / day** for all DBs
 - 1,500,000 signals
 - 5,000,000,000 dp/day, 1.6E12 dp/year
 - 6,000,000 extraction requests per day
 - Soon reaching **Peta Bytes (BigData) stored (~0.5PB)**
 - **Over 1000 users**



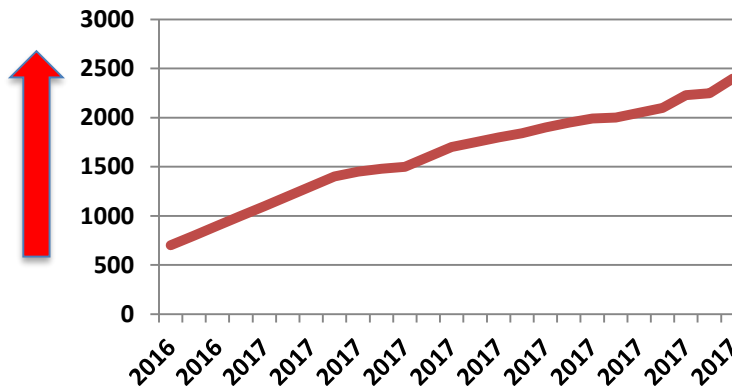
Why NXCALS?

- Aging and non-scalable technology
- Not designed for TBs / day data rates (large vectors, high frequency data)
- Main stream analytic techniques not available (streaming, cluster exec.)

Size in GB / day



1.5 years later...





NXCALS & SWAN

- Spark Web Notebooks (like Jupyter):
 - **Web interface** with built-in Spark integration
 - Data visualisation (**tables, charts**, etc.)
 - Dynamic input forms and **data widgets**
 - Support **work in collaboration** and publishing results online
 - **Natural and very user-friendly** for **Data Scientists**



SWAN (Service for Web based ANalysis) is a platform to perform interactive data analysis in the cloud.

To avoid work duplication
NXCALS bet on SWAN!

Very productive collaboration. Big THANK YOU to our EP-SFT and IT-DB colleagues !

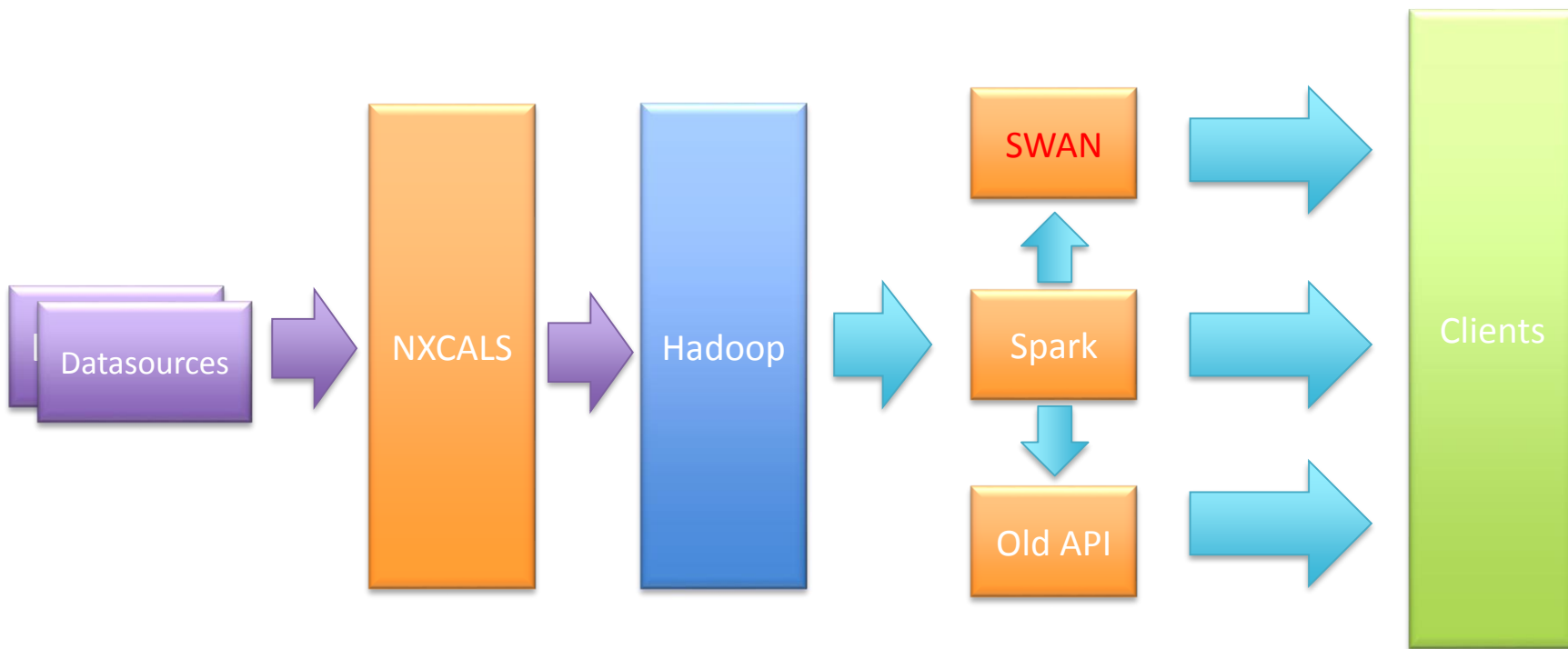


Benefits

- Open door for **BigData Queries, Tools & Techniques**
 - Bring the **computation** to the **data** (reverse of paradigm)
- More **generic & accommodating NXCALS system...**
 - ... that just **stores rows {f1,f2...}** of arbitrary structure
 - **Avoids** many **scattered “logging” systems**
 - **Keeps together & shares all data**
 - Acc. Devices, Industrial Controls, Tracing, Post-Mortem, MDs, Analysis Results, Alarms, etc
 - **Allows client-oriented data governance** (policy per client)
- **Spark analytics**
 - **Python & Notebooks/Dashboards**, Machine Learning, R, ...
- **Horizontal scaling** to cover ever growing needs
- **Cost & maintenance reduction**
- Strengthen **collaborations** between **groups**
- “Attractive” technology -> easier to recruit for



NXCALS Architecture





Overall Status

- Core of NXCALS ready
- Taking data from CMW devices
- Ready to accept clients (early adopters)
 - PM clients (ongoing)
 - ABP (for MDs, LHC Luminosity, etc) – **waiting for SWAN to have production grade SLA**



Next Activities (until LS2)

- **Client use-cases (i.e. for analytics) (WIP)**

– **main driving factor!**

- **Data migration from CALS (WIP)**
- **Low latency Streaming API from Kafka (WIP)**
- **Simplified Extraction API & backport of old CALS API**
- **WinCCOA** integration & datasources
- Installation on **new hardware** (CO and Hadoop in IT)
- Storing analysis results back to NXCALS as entities
- **Timber** rewrite to web, incorporate Web Statistics App?
- **Reporting System?**
- **NXCALS Configuration** Application (needed for Variables)
- **Gobblin** replacement?
- ...



SWAN Summary

- **NXCALS uses SWAN to target Python/Jupyter users**
 - **~10 users for starters**, will grow to **around 30-40** or more depending on the success rate
- **Requirements for SWAN:**
 - **pySpark (demonstrated)** & other already available Python packages (numPy, pandas, etc)
 - Security with Kerberos
 - At least 4GB memory for the driver/kernel (TBD)
 - Best effort **24/7 support** required (like for NXCALS)
 - **Stability == Good publicity == Success**