

Data management on Control Systems & more

G.Abeillé – ISAC (Acquisition & Control Systems Engineering)

- Tango kernel
- Tango archiving
- Experimental data storage
- Interoperability challenges

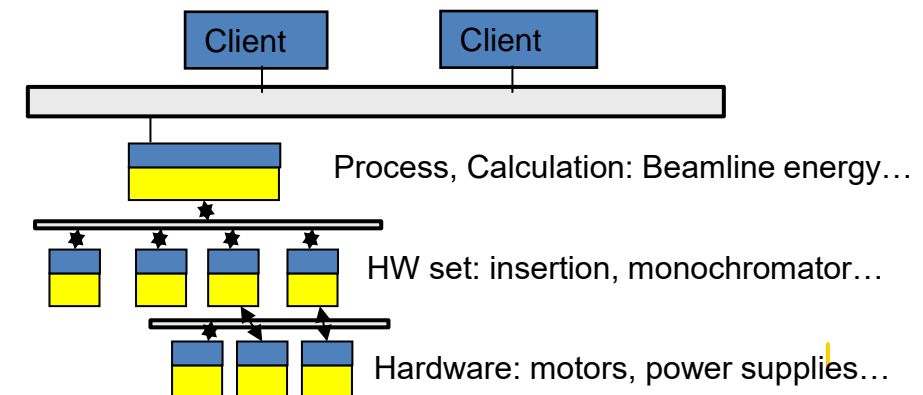




Tango kernel



- Tango has become a SOLEIL standard de-facto
 - Tango is used on all control systems (Accelerators, Beamlines and Labs)
- Strategy to integrate everything in Tango devices:
 - Hardware (motors, vacuum, power supplies, ...)
 - Hardware sets (insertions devices, monochromators,...)
 - External systems (Building Management in OPC, LINAC in LabVIEW....)
 - Calculations, orchestration, workflows (beamline energy, beamline acquisitions processes, experimental data management, archiving...)
- Outcomes:
 - A seamless integration in all Tango clients (GUIs), in archiving...
 - Built-in data correlation
 - Autonomy to our users with Tango client API (Python, Matlab, Labview, Igor Pro)



- **Tango cpp 9.2.5 / PyTango 9.2.2 / JTango latest**
 - Very few events, only pushed by code
- **ISAC control develops & deploys :**
 - Servers in mostly in C++/Java. Increasing number of Python classes.
 - Clients in Java (GUI in SWING) / Python (SOLEIL CLI, Spyc)
- **Accelerators teams:**
 - Devices in Python (mostly Diagnostics)
 - Physicists: Clients/GUI in Matlab (SOFB, Tune FB, Transversal FB...)
 - Operators: Clients/GUI in Labview/Python (Supervision, Top-up injection process...)
 - Igor for magnets measurements facility
- **Beamlines teams:**
 - Devices in Python
 - Clients in Python (scripting)
 - Bindings Labview/Igor/Matlab
 - Using most GUI from ISAC



- Improvements for SOLEIL II
 - Large use of Tango events
 - Improve administration of Tango databases
 - Manage changes along with software deliveries
 - track all changes
- Tango versions upgrades strategy?
 - One shot on all CS?
- RETEX Tango databases management
 - Process for deployment/ update configuration of devices?
 - Reporting?
- Events management?
 - Polling vs by code?
 - Underlying network architecture? 1 single DNS for all CS?
- How is interconnexions is done between CS?
- Open Telemetry
 - Status and plans at MAXIV?
 - Signoz usage? SOLEIL would need help installing it for validating Jtango
- Management of user autonomy
 - Guiding them? How? Dedicated work groups? On architecture? Development? Code review?



Tango archiving



Developed as a FOSS by SOLEIL since ~2006

HDB

Historical DataBase

Timeseries of Tango attributes

Long term storage

Min. period 10s

TDB

Temporary DataBase

Timeseries of Tango attributes
Short term storage (data deleted at each shutdown)

Min period 100ms

SNAP

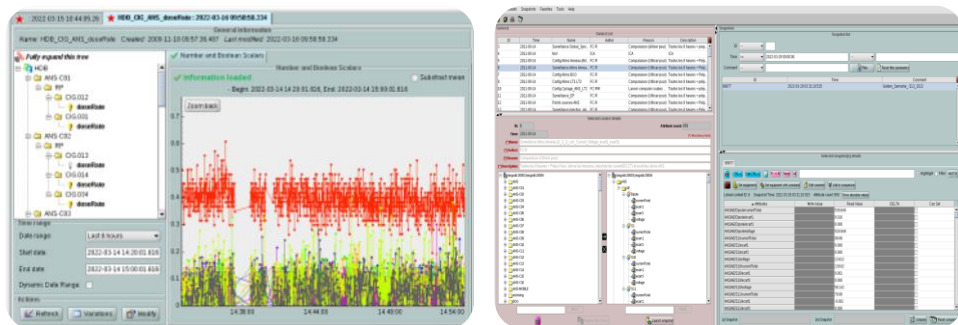
On-demand attributes storage

On-demand stored attributes writes

Comparison between snapshots

Insertion and data fetching
is done through
Tango devices & GUIs





Used by SOLEIL operational staff (operators, accelerators physicists, beamline scientists, engineers..) :
incident analysis, preventive maintenance, testing new operational modes...

HDB/TDB critical for accelerators, no beam injection if the archiving is not operational

Configuration managed by operational staff

Java SWING GUIs for configuration and data exploitation (Mambo/Bensikin)

Accelerators
HDB/TDB/SNAP



Beamlines
HDB / SNAP



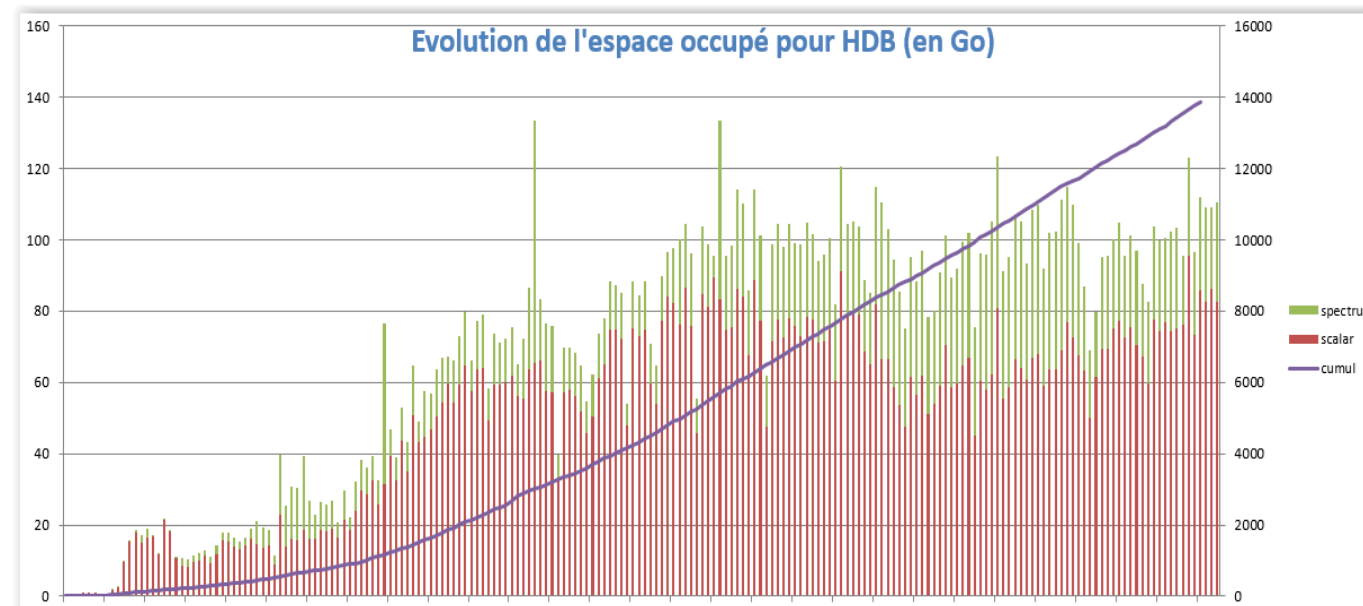
Labs

HDB or TDB

Insertions,
Vacuum, Diagnostics, Power Supplies...



- TDB for accelerators:
 - ~ 25 000 Tango attributes
 - ~ 4 500 inserts/s
 - Data deleted at each shutdown
- HDB for accelerators :
 - ~ 32 000 Tango attributes
 - ~ 850 inserts/s
 - Increase of ~150GB / Month.
 - Total since 2006: 17 TB
- HDB for 29 Beamlines
 - ~8830 attributes
 - Total of 3.5 TB

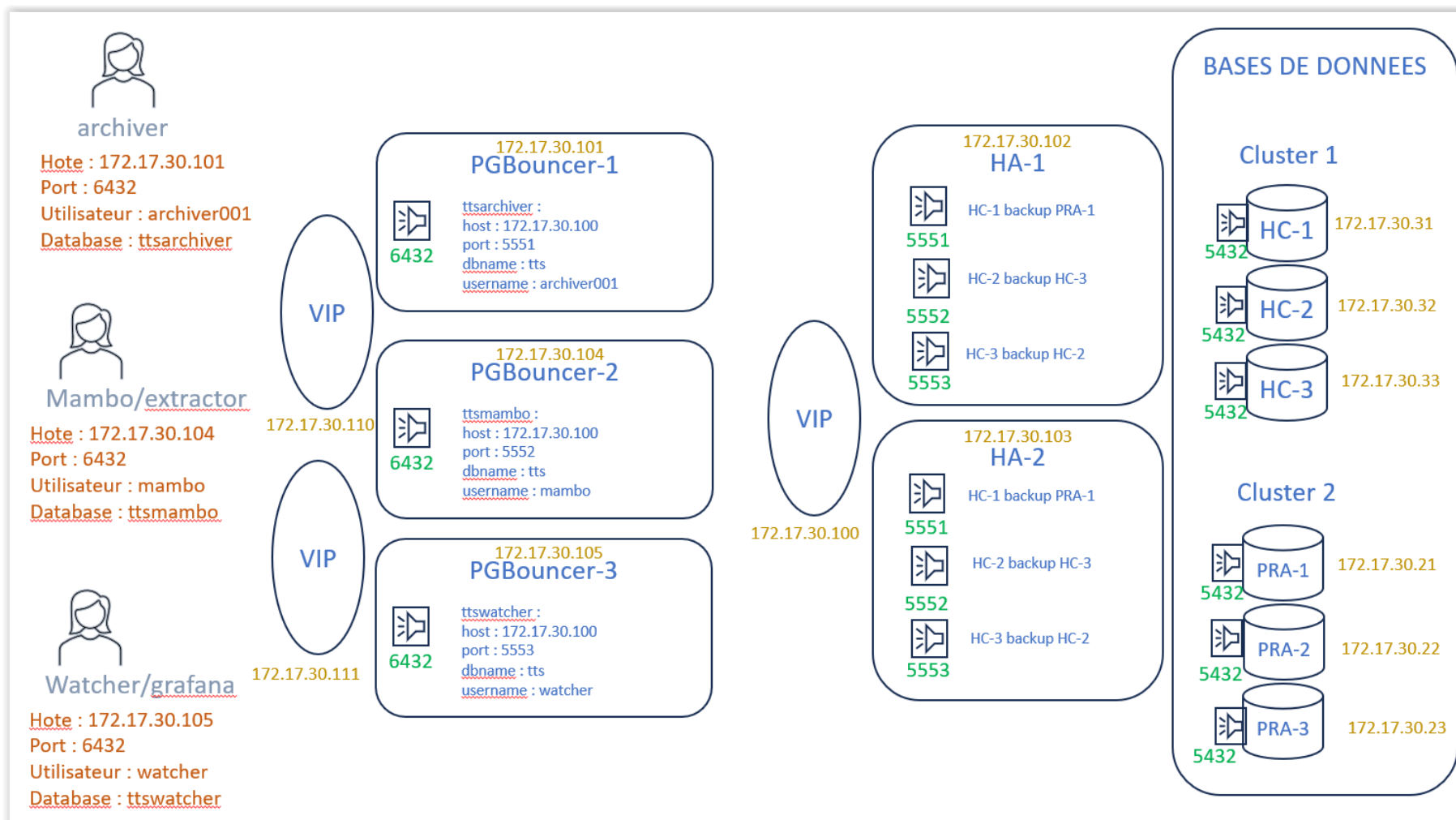


- Oracle is too expensive, and FOSS efficient alternatives are now well established
- Replace all our Oracle DB to FOSS alternatives for SOLEIL II
- On-going replacing of HDB/TDB by one Timescale DB:
 - Using HDB++ schema with some few additions (one more table for polling configuration)
 - Keep all existing software stack and configurations for smooth user migration
 - Switched TDB to Timescale in operation beginning of 2024
 - In progress:
 - Long term data retention strategy
 - Reliance strategy for maximum availability
 - Performance optimizations for data extraction
 - Switch users of accelerators from HDB to Timescale beginning of 2025
 - Oracle data migration strategy not yet established
- Snap DB alternative study will be done afterwards



<https://gitlab.synchrotron-soleil.fr/software-control-system/tango-controls-archiving>

- Usage HAProxy / PgBouncer under tests



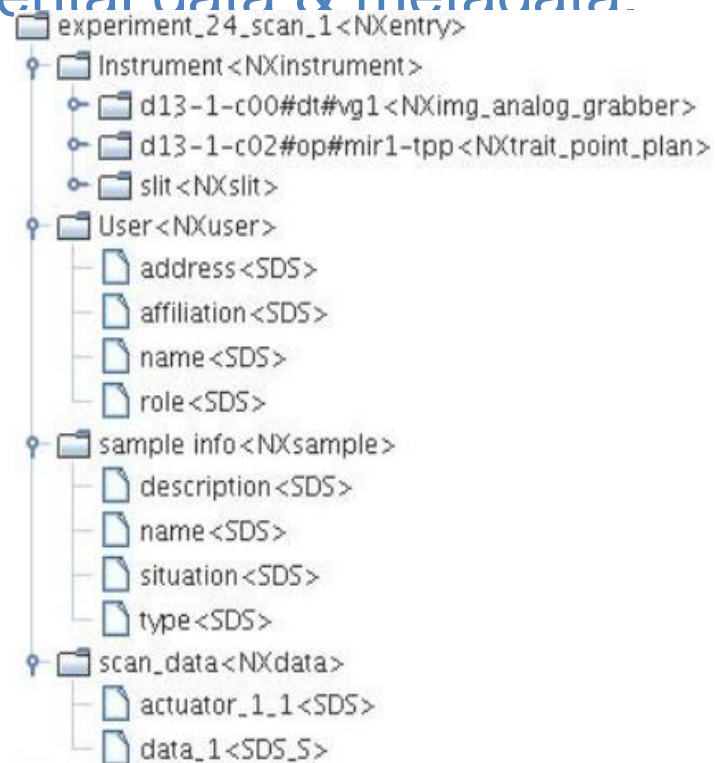
- Feedback on Timescale over K8S?
- Monitoring? on user data extractions?
- How do you manage long term storage?
- Data decimation? Aggregation?
- Use cases with IA on archiving databases?



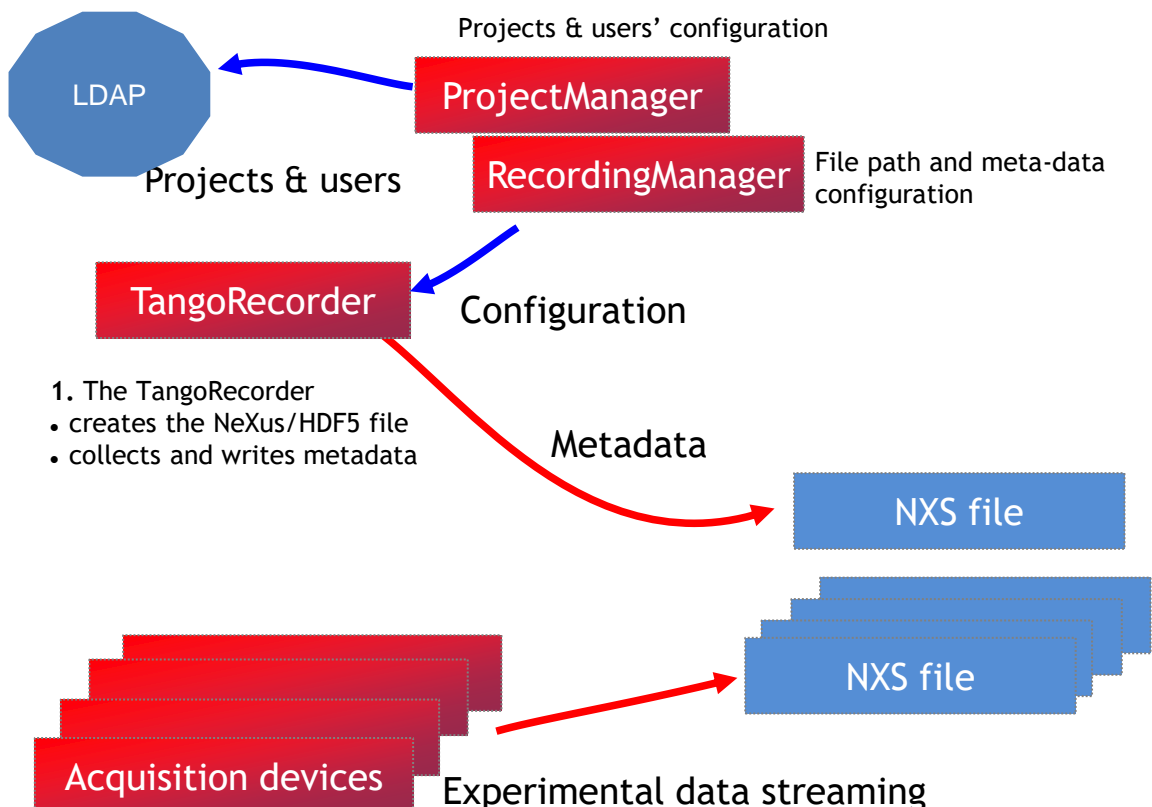
Experimental data recording



- We provide a set of Tango tools for recording experimental data & metadata.
- Used on 24 out of 29 beamlines
- Key features :
 - NeXus/HDF5 file format
 - Flexible configuration for beamlines:
 - Files' path, metadata
 - Integrated w/ our DAQ systems (see DAQ pres.)
 - Using lib HDF 1.8

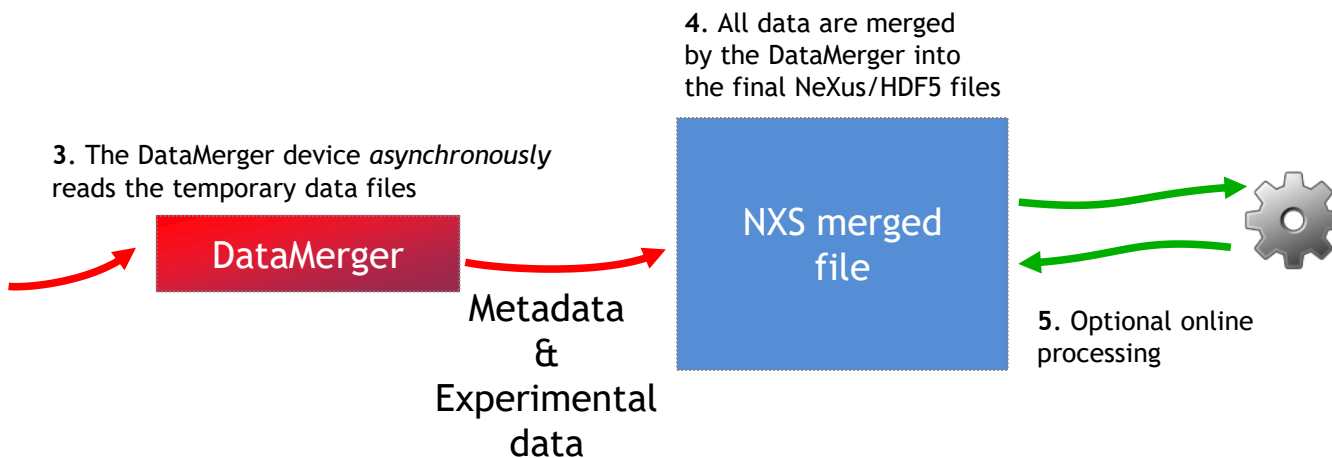


- The whole acquisition process is orchestrated by Tango devices
- The actual architecture has been refactored to optimize performance & configuration flexibility



1. The TangoRecorder
 • creates the NeXus/HDF5 file
 • collects and writes metadata

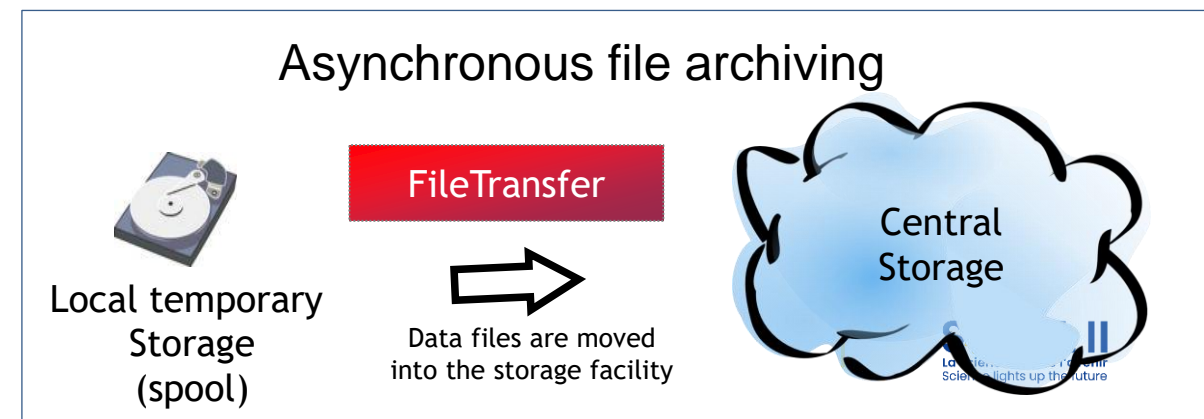
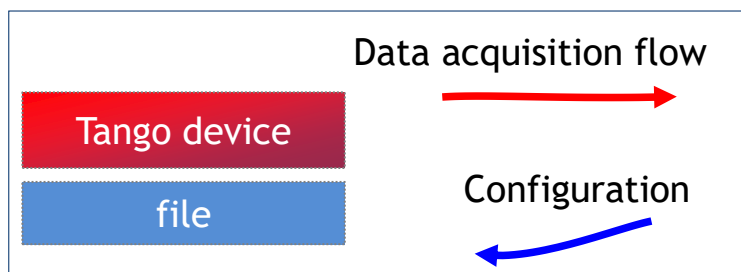
2. NeXus/HDF5 files containing the experimental data



3. The DataMerger device *asynchronously* reads the temporary data files

4. All data are merged by the DataMerger into the final NeXus/HDF5 files

5. Optional online processing



Configuration & Monitoring

The screenshot shows the 'Data Storage' configuration window for 'com-rock'. It features a top navigation bar with 'ROCK' and several status indicators (ON, STANDBY). The main area is divided into configuration fields on the left and a status/monitoring section on the right. The status section shows 'Expiration: 2038-01-19 03:14:07' and 'Status: Ready'. Below this are 'Counters' for File (5148) and Session (713), each with 'Reset' and 'Apply' buttons. The bottom section contains tabs for 'Experimental Frame', 'RecordingManager', 'TangoRecorder', etc., and a 'Comment Conditions' field.

Data Visualization

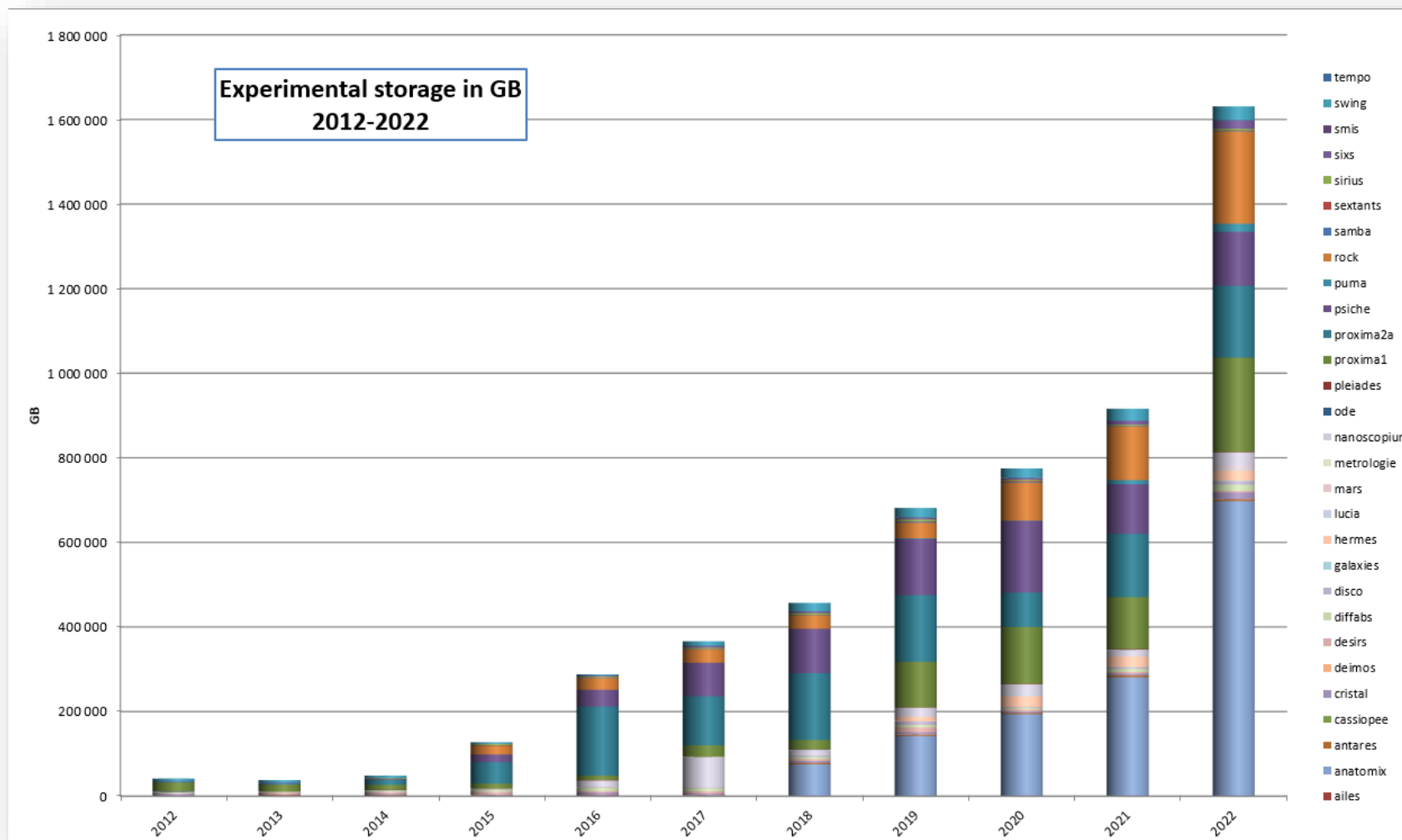
The screenshot displays the 'DataBrowser' interface with four main panes:

- 1 - Datasource tree:** A hierarchical tree view showing the data structure, including 'NEXUS' and 'TANGO' folders.
- 2 - Data items viewer:** A chart showing two data series: 'KatyScanImage.nxs - /a/scan_data/trajectory_1_1 (Y1)' (blue line with dots) and 'tango/tangotest/1/float_spectrum (Y1)' (red line with dots). The x-axis ranges from 0 to 15, and the y-axis from 0 to 30.
- 3 - Display manager:** A table listing the data items and their axes.
- 4 - Item informations:** A table providing detailed metadata for the selected item.

Item	Axis
KatyScanImage.nxs - /a/scan_d_ IMAGE	
KatyScanImage.nxs - /a/scan_d_ Y1	
tango/tangotest/1/float_spectrum Y1	

Property	Value
Description	A float spectrum attribute
Format	NUMERICAL
Label	float_spectrum
Max alarm	Not specified
Min alarm	Not specified
NumericalFormat	%6.2f
Rank	1
Shape	[4096]
Type	SPECTRUM
Unit	No unit

- 2022 central storage files: ~1.6 PB for all beamlines



- No plan yet to change the existing architecture
 - Currently upgrading to libhdf 1.14.3
 - Next steps:
 - Study usage of AD (Application Definition from NeXus)
 - Simplify metadata configuration for end-users
- File compression in test on 1 beamline.
 - Will be activated on demand of beamlines
- Reduce latencies w/ data streaming from detectors on-going study for:
 - Cf DAQ pres.
 - High throughput detectors
 - online analysis

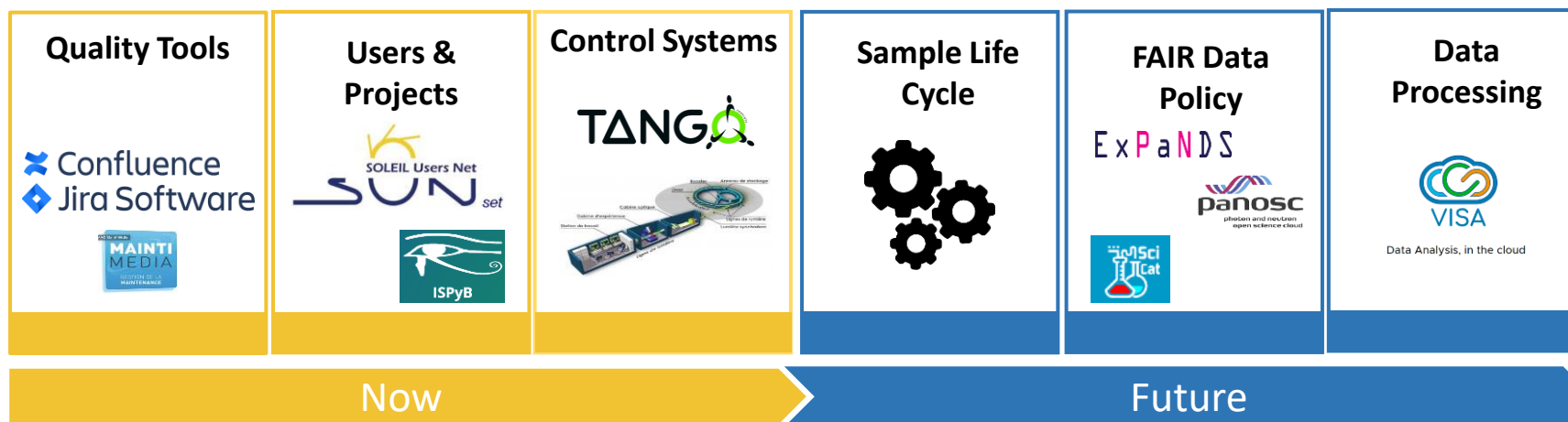


- Files rights management? Architecture for interconnections between DUO, IAM? And CS?
- How do you manage metadata configuration?
 - Who is doing the configuration?
 - Do you manage a standardization / quality of metadata?
 - Save all in Nexus files and SciCat?
- Do you support Nexus? AD? VDS?

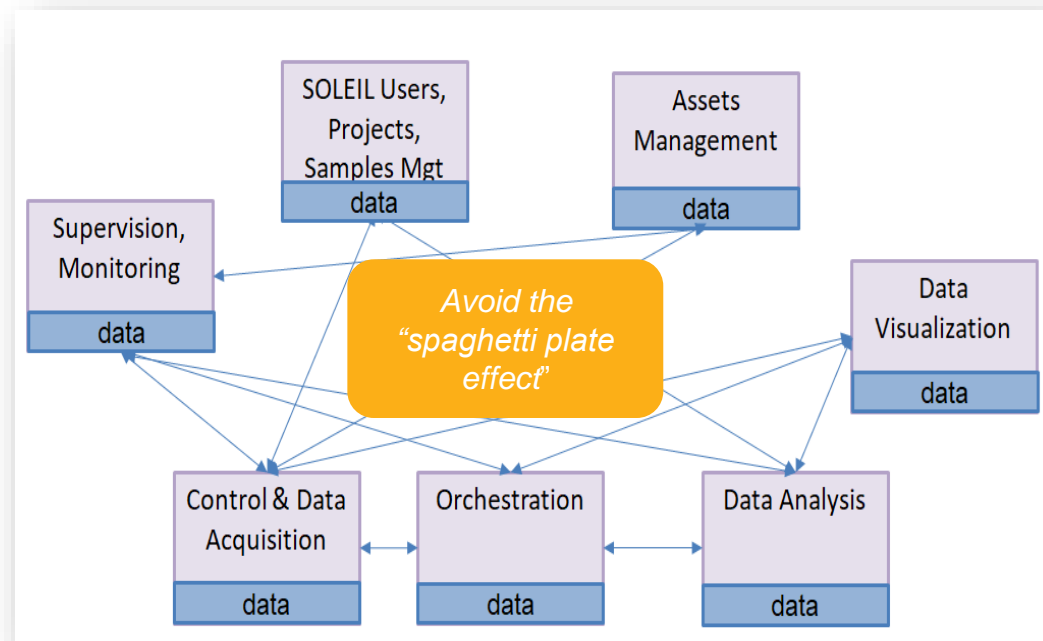
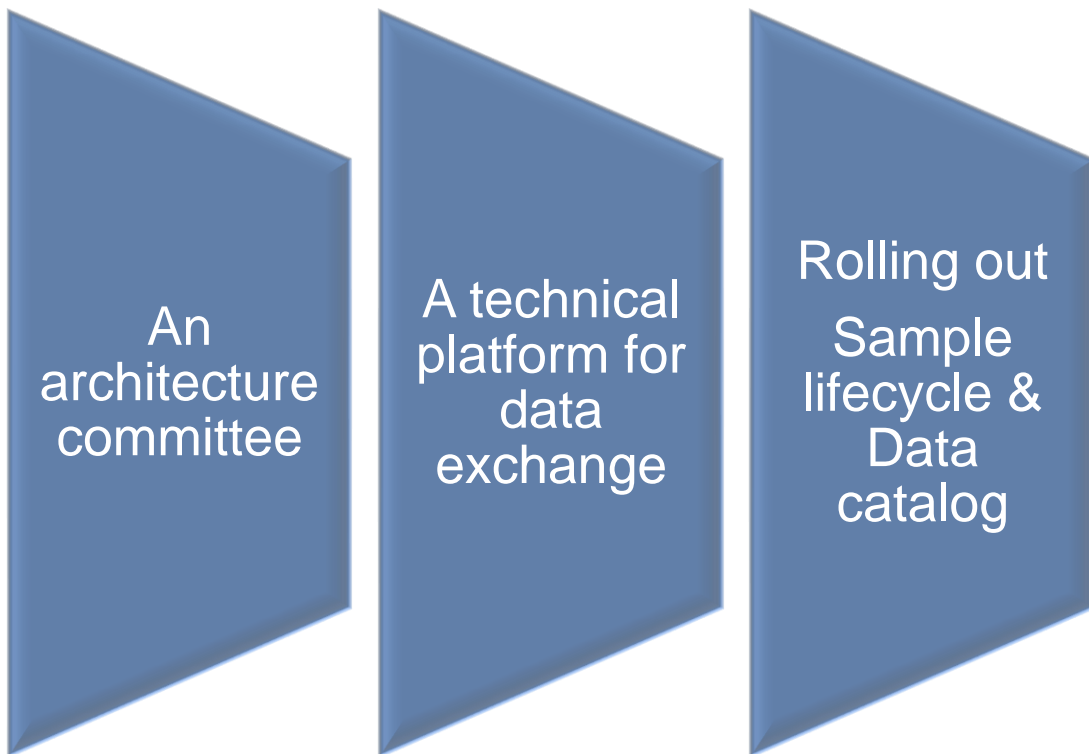


Interoperability challenges





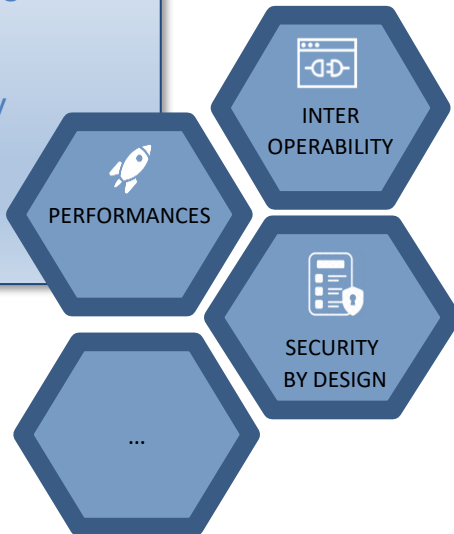
3 workstreams:



- Started one year ago
- Led by SOLEIL IT strategy director
- Includes 1 person per IT team

Set of architectural principles

Common consensus
 Design guide
 Applied proactively
 Evolving

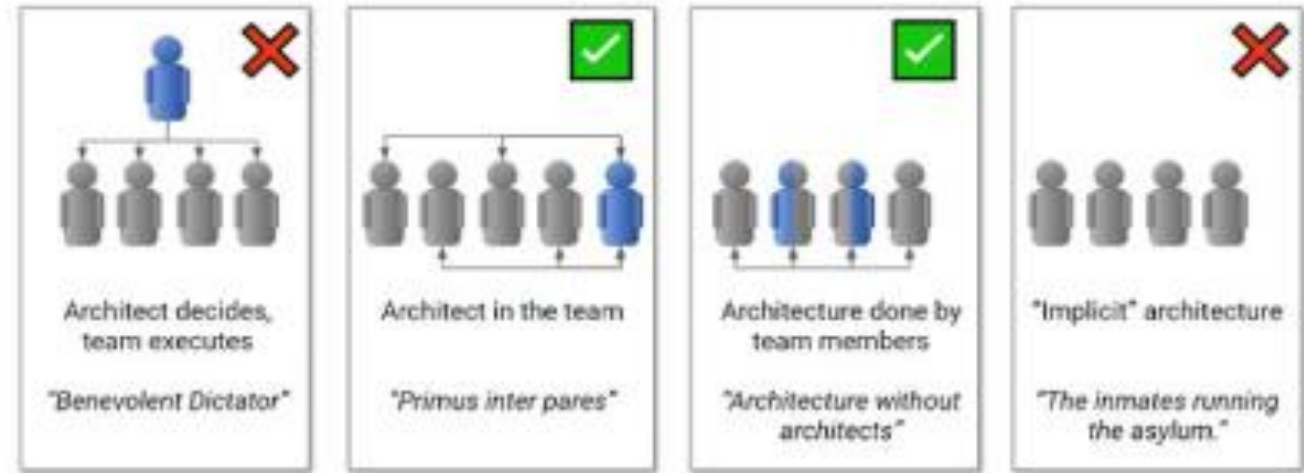


Foster and review projects architecture

*For new projects
 or major refactoring*

Dialog between architects, project managers, experts...

Validate the application of the principles or agree on temporary derogations



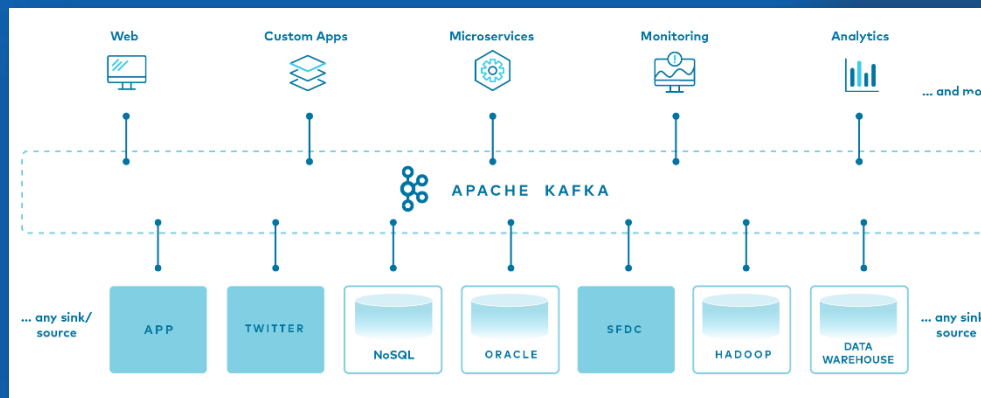
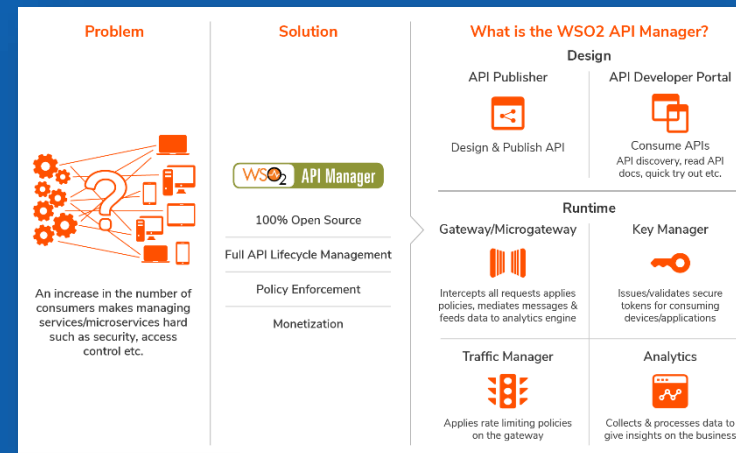
<https://twitter.com/ghohpe/status/1171379436739944449>

Provide a technical platform called PLUSS to exchange data between applications based 2 complementary solutions:

PLUS

API Manager with WSO2

- Provide a catalog of Web APIs (Open API/ REST, SOAP, GraphQL, ...)
- Control API usages (security, quotas...)



Event bus with Apache Kafka

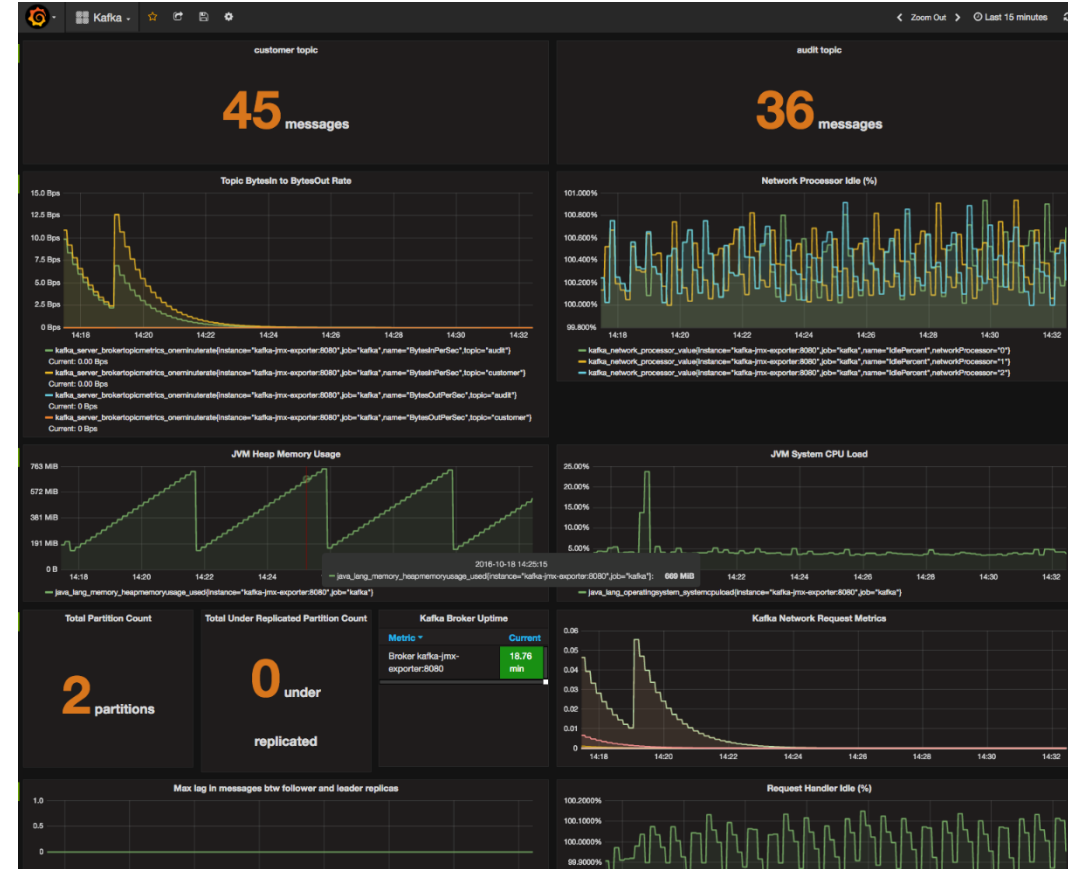
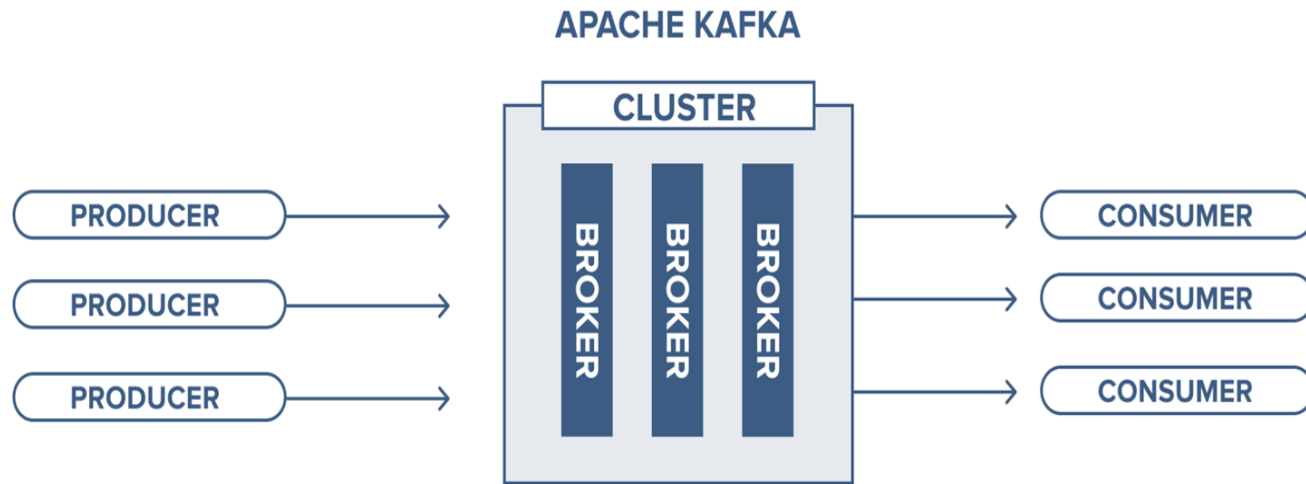
- HA, scalable, high throughput
- Decoupling between producers and consumers
- Schema registry
- Data transformation
- Large nr of existing connectors

PLUS:

Information System's Architecture Platform for SOLEIL.

(Plateforme d'Urbanisation du Synchronon SOLEIL)

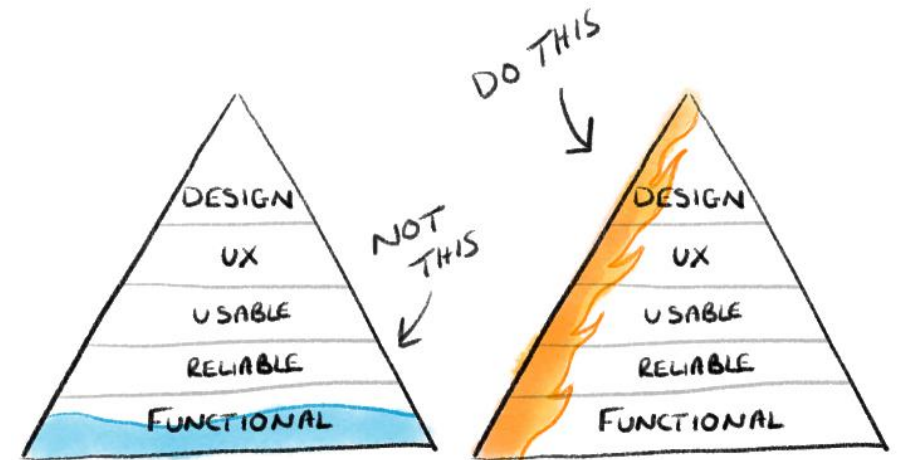
- WSO2 API Manager used in operation for one use case
- Apache Kafka used in dev environment
- Apache Kafka soon deployed in production env:
 - 3-nodes cluster on Docker
 - Monitoring w/ Prometheus/Grafana



- Build a new solution from scratch:
 - No generic solution at SOLEIL (ISpyB for only MX/AutobioSAXS beamlines)
- SLM application demands many interconnections
 - We started designing an architecture based on PLUS
- Develop an MVP of a Web application: minimal functionalities but fully operational

“The **Sample Lifecycle Management (SLM)** application centralise, organise sample information and make them available for the rest of the information system of SOLEIL (control systems, data catalog, etc.). It includes:

- Sample **unique identification** and description (UUID, DUO proposals and beamtime sessions description, sample’s components...)
- Tracking of **geographical locations**
- All **transformations** and **measures** performed on the sample
- All pertinent sample information stored in the **eelogbook** ”



Minimum Viable Product

Wireframe mockup and many exchanges/demos for user feedbacks before developing the real front end:

The image displays a wireframe overlay on a real screenshot of the SOLEIL Sample Management (SLM) web application. The wireframe is a dark blue vertical sidebar on the left side of the page, containing navigation links for 'Samples', 'Racks', 'Proposals', 'Locations', and 'Roles'. The real application interface is visible in the background, showing a 'Create Sample' form with various input fields and buttons.

SOLEIL Sample Management

Home · Samples · Create Sample

PUMA Beamline [Change your location](#) Jean Dupont

Record Several Samples Sample number 1

Proposal/Team* 20100850-2001 Type BAG Sample Identification

Proposal/Team 20100850-2001 Type BAG Sample Denomination* Moustique dans résine de pin 312

MP or PI Leia Organa Sample Short Name* MOUSTIQUE_312

Components*

Session

Metadata

Sample / Create Sample

Create Sample

Proposal/Team*

Proposal / Team 4

Title Analyse de Kryptonite

Type non-BAG

MP/PI Jeanne Dupuits

Session

Session PROXIMA-1 from 2015/10/...

Beamline PROXIMA-1 Station STATION-1

From Wed Oct 14 2015 To Mon Feb 14 2022

Proposal Components

<input type="checkbox"/>	Name	Acronym	Safety Level
<input type="checkbox"/>	Soufre	S	R

1-1 of 1 < >

Sample Identification

Sample Name

Short Name

External ID

Comment

Current Location

Location

Since

Current Rack

Rack Name

Rack ID

Rack Type

Sample Position M 1 N 1

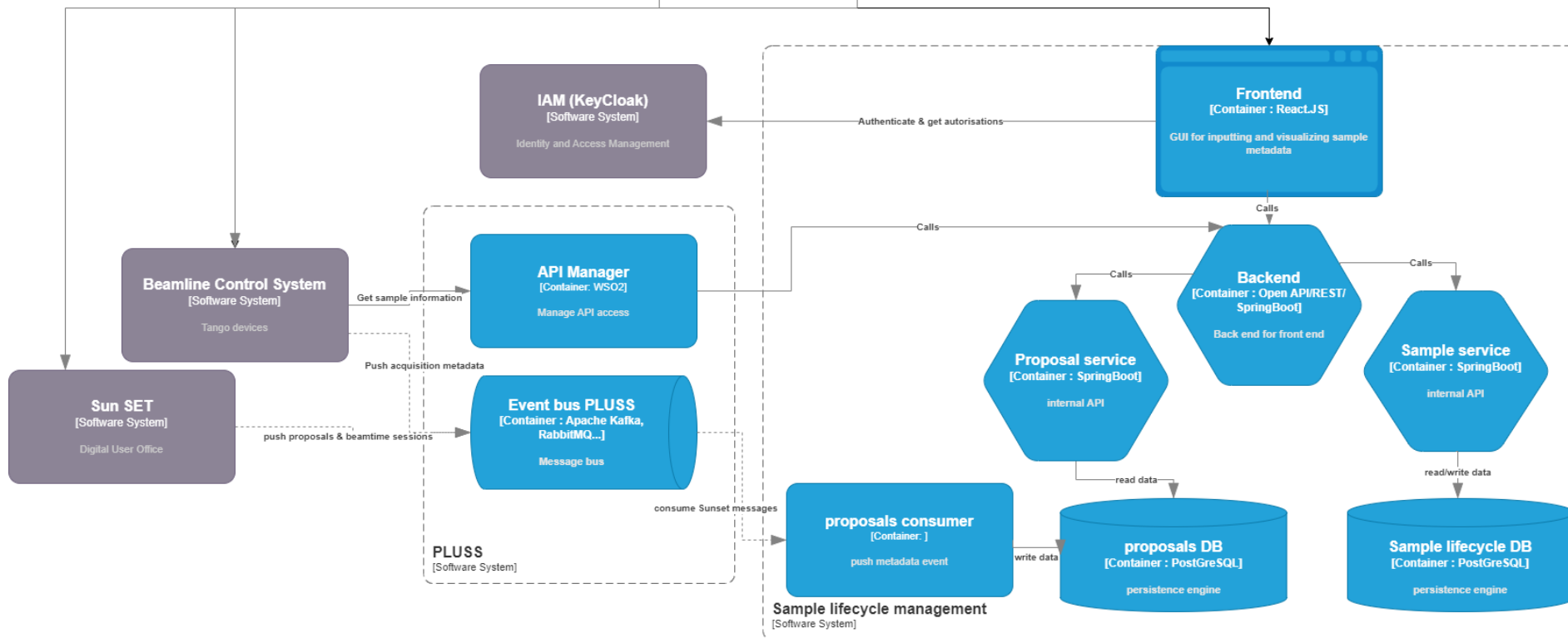
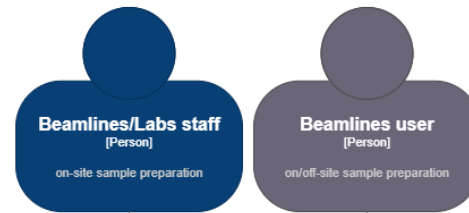
Since

Sample Short Name

File links

Legend

Person
Software System
Container
External Person



MVP development finished:

- PostgreSQL databases
- REST APIs /Java Spring Boot
- Frontend in React.JS/Refine
- KeyCloak connector for authentication and authorisations

• Interconnections with PLUS:

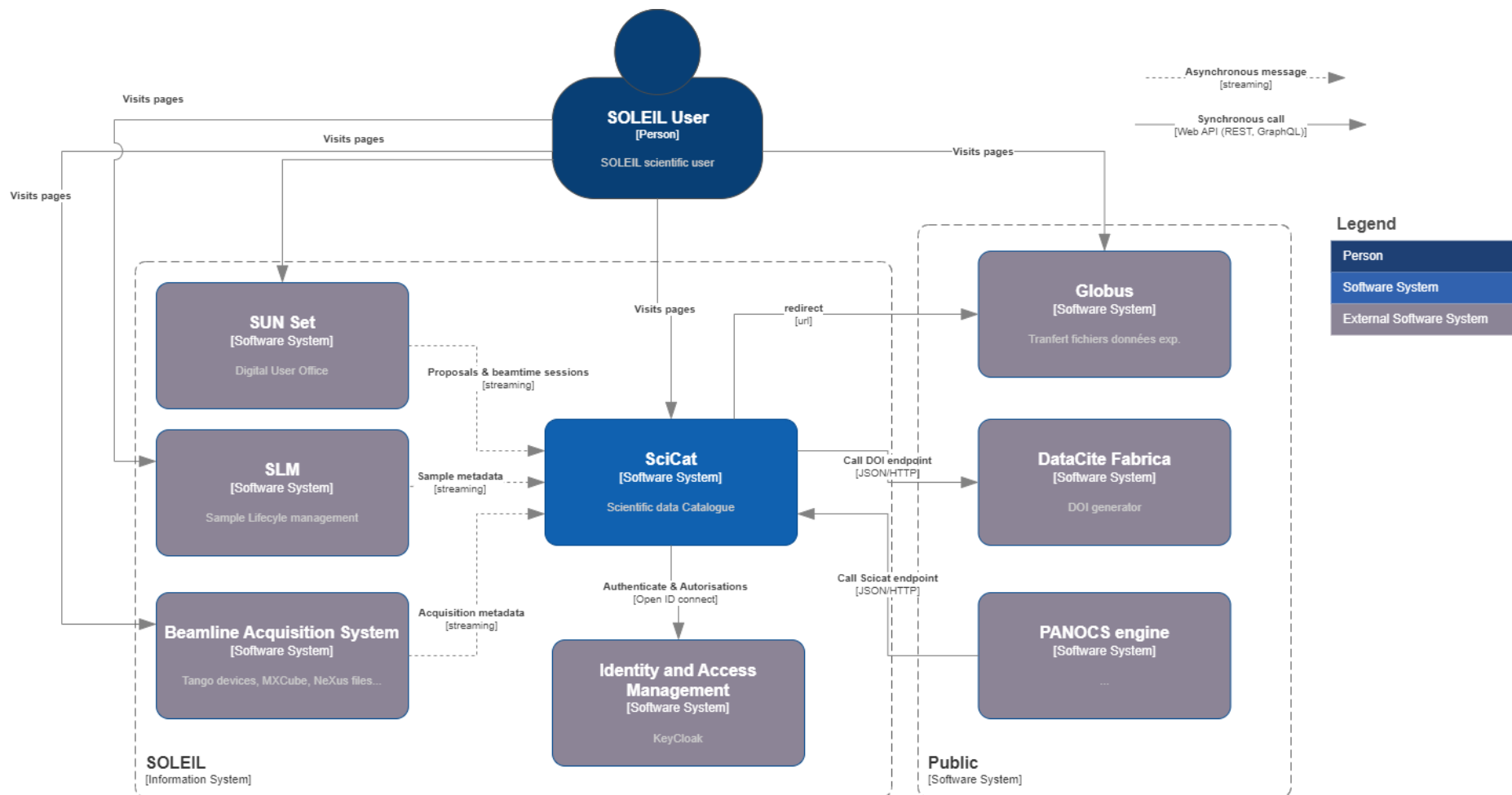
- Kafka for our DUO (SUN Set) proposals
- API Manager to expose sample data on control systems

- SOLEIL has to provide a public access to experimental -> provide a data catalog
- A SOLEIL project team is currently working on deploying SciCat (Collaboration PSI, ESS, MAX IV, DESY, MLZ, ALS, RFI, ...)
 - Starting with a few beamlines with basics fonctionnalités
- Rely on PLUSS to exchange data with the rest of the information system
 - Similar needs as the SLM application

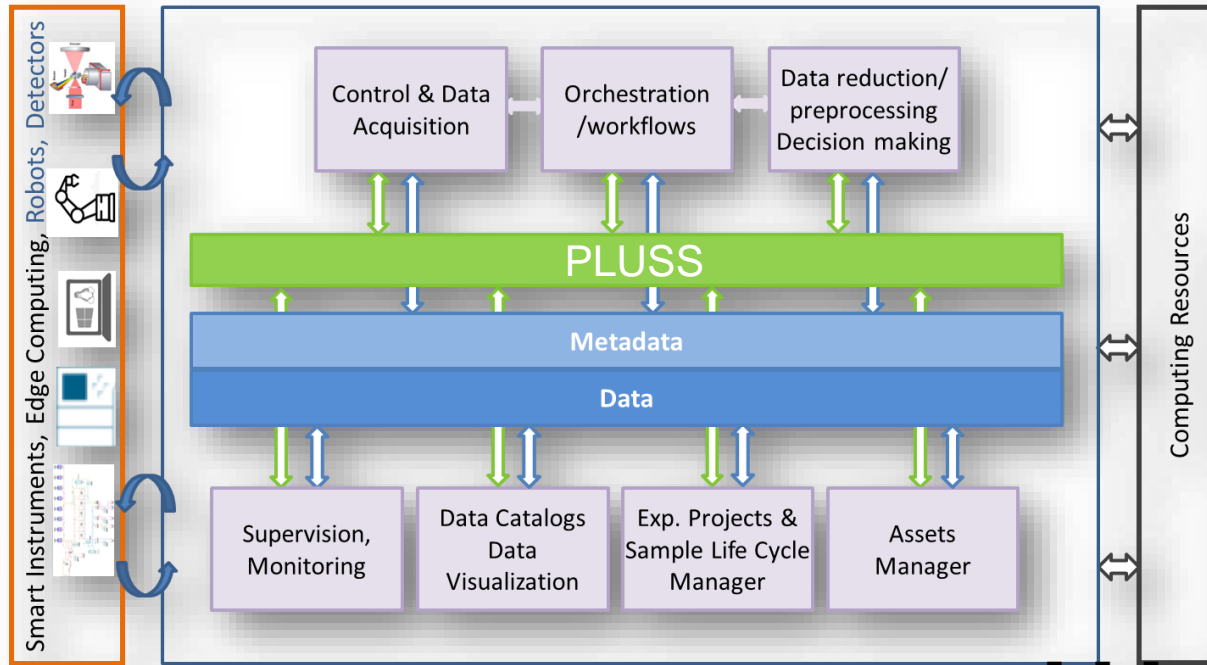


Kafka for getting :

- proposals from our Digital User Office
- ingestion of acquisition metadata from beamlines' control systems



- **Top priorities:**
 - Sample lifecycle management
 - Implement interactions of SciCat
 - Integrate Archiving GraphQL in API Manager (see GUI pres.)
- **Next ones:**
 - Future elogbook (evaluation of different solutions in progress)
 - New dedicated Digital User Office for industrial users
 - Data exchanges for SOLEIL II accelerators testing benches (Vacuum, Diagnostics, Magnets, Alignment)
- **Many other potential topics:**
 - Tango alarms (PANIC) history storage in an Elastic DB
 - Study for interactions between control systems and data analysis
 - Secured interconnections between control systems (Accelerators & Beamlines)
 - Maybe use of Kafka for high throughput asynchronous data exchanges on CS:
 - *Kafka is low-latency but not critical real-time, with limited message size*



- PLUSS covers transversal processes challenges:
 - Seamless data exchanges between information system components (Tango control systems, digital use office, data catalog, databases, files...)
 - It is an opportunity to unify our architecture
- PLUSS addresses **concrete uses cases** in an **iterative process**:
 - To **harness** these new technologies and their operations
 - To **foster** architecture activities inside projects
- A combined organizational and technical approach is mandatory for information system consistency.

- **Feedbacks on Kafka broker?**
 - Deployed over K8S? Nr of nodes?
 - Nr of topics? Throughputs?
 - Usage of schema registry? How is managing the governance of the schema?
 - Organisation for operation? Monitoring/Alerting?
- **Feedbacks of API Management?**
 - Do you have a wide number of API share around different development teams?
 - Do you manage the governance:
 - Open API mandatory?
 - API lifecycles changes?
 - Security?
- **Configuration management**
 - Interconnections between tests benches, cable DB, nomenclature DB, PLM...?
- **Observability / Monitoring**
 - Global architecture for applications monitoring, embedded system, IT infra monitoring?
 - Central data logging solutions like ELK?
- **Feedbacks on architecture activities**
 - Do you have a dedicated team or other organization to manage the global IT architecture?
 - How have you defined the overall architecture for the experiments data & metadata flows (from the proposals validation to data access for Open Data)? Dedicated persons/team? Coordination of different teams?

