

A photograph of the SOLEIL II building, a modern structure with a curved facade, large glass windows, and a prominent metal grid roof structure. The building is set against a clear blue sky and a green lawn.

SOLEIL UI experience, web technology, etc. and future plan

UI status around control system

UI Organization

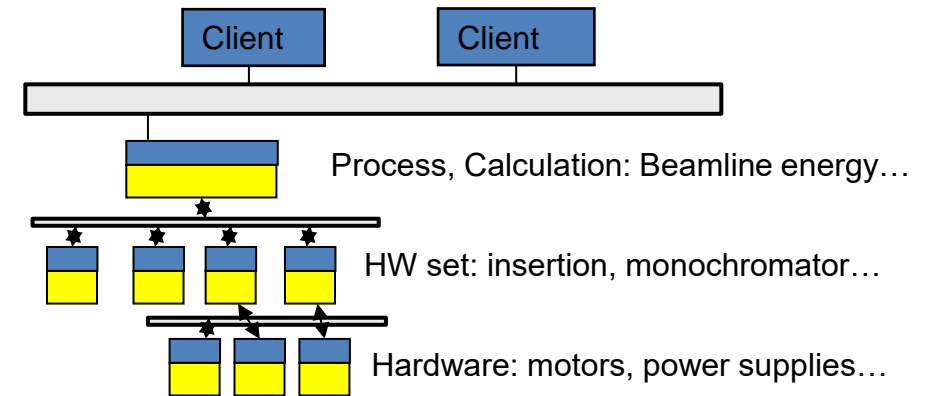
Future plans



UI status around control system

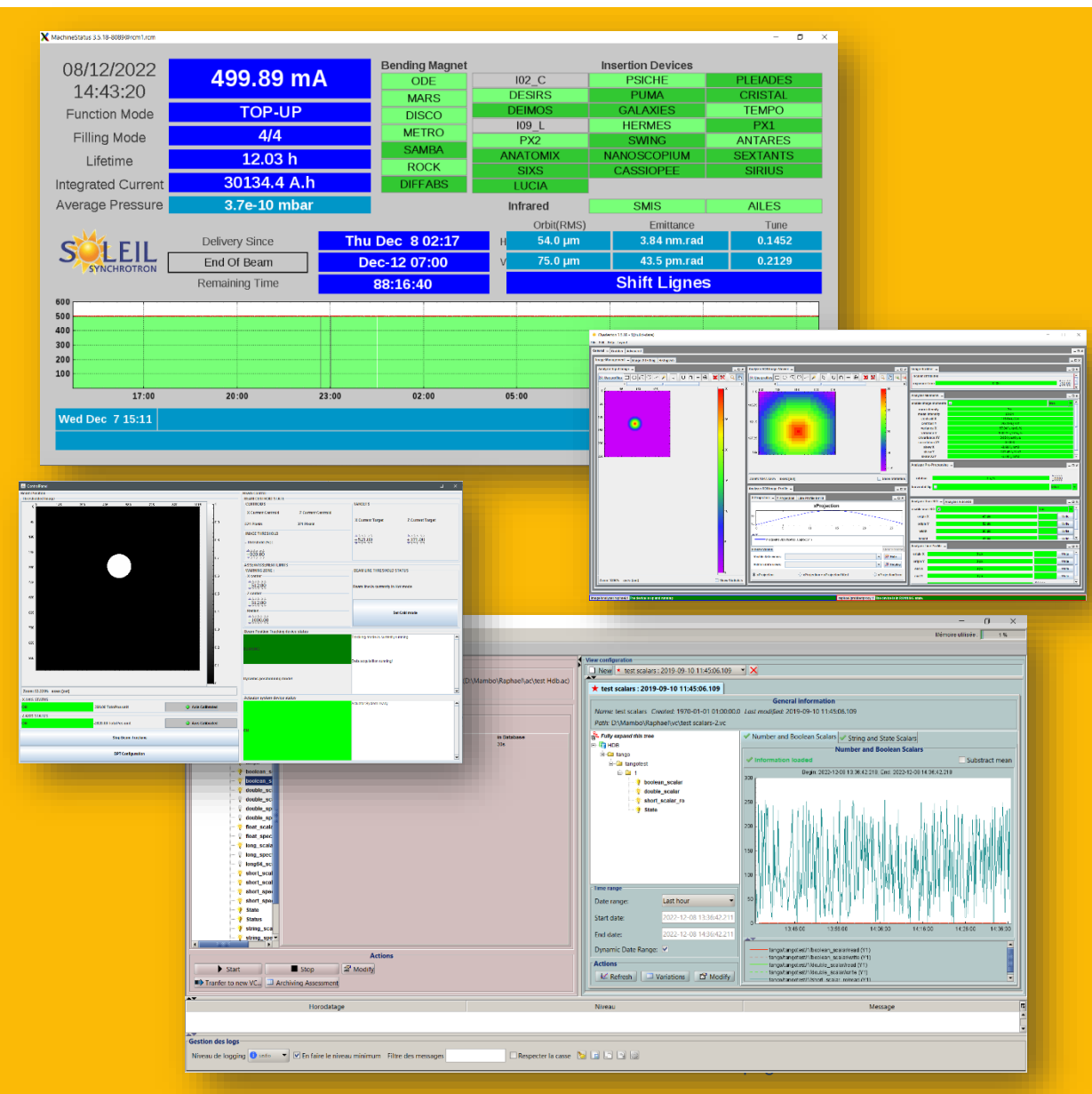


- Tango as the de facto standard at SOLEIL
 - Tango is used across all control systems (Accelerators, Beamlines and Labs)
- Our strategy is to integrate everything into Tango devices (a precursor of SOA/ μ Service architecture?):
 - Hardware (motors, vacuum systems, power supplies, etc.)
 - Hardware sets (insertions devices, monochromators, etc.)
 - External systems (Building Management in OPC, LINAC in LabVIEW etc.)
 - Calculations, orchestration, workflows (beamline energy, acquisitions processes, experimental data management, archiving, etc.)

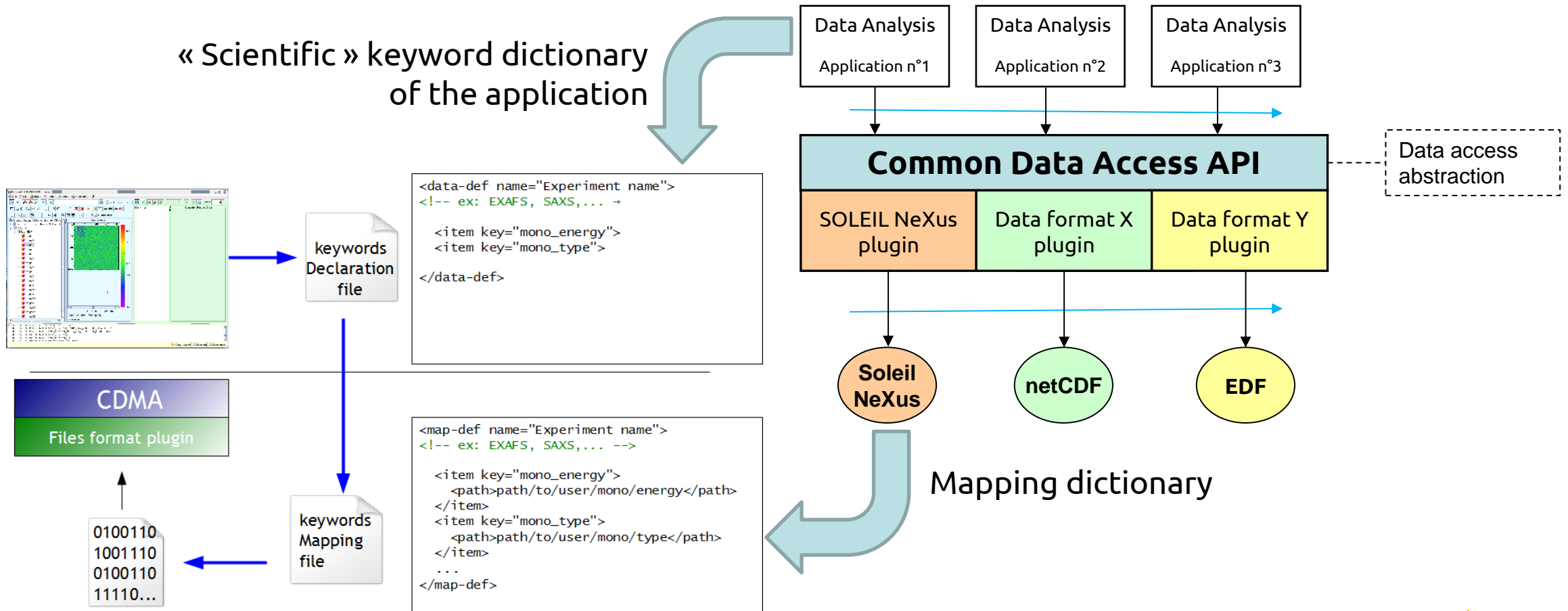


- Key outcomes:
 - Seamless integration with all Tango clients (GUIs), and in archiving systems
 - Built-in data correlation
 - Users gains autonomy through the Tango client API (Python, Matlab, LabVIEW, Igor Pro)

- Comete Framework at SOLEIL
 - SOLEIL uses **Java Swing** for building common GUIs.
 - The **Comete framework** was developed to simplify this process.
- Key features
 - **Widget library**: A collection of ready-to-use interface components.
 - **Data sources**: Easily connect to various data inputs.
 - **Connection management**: Intermediaries handle communication between widgets and data sources.
- Benefits
 - **Fast GUI development**: Quickly create new interfaces.
 - **Simplified integration**: Streamlines the connection of data to user interfaces.



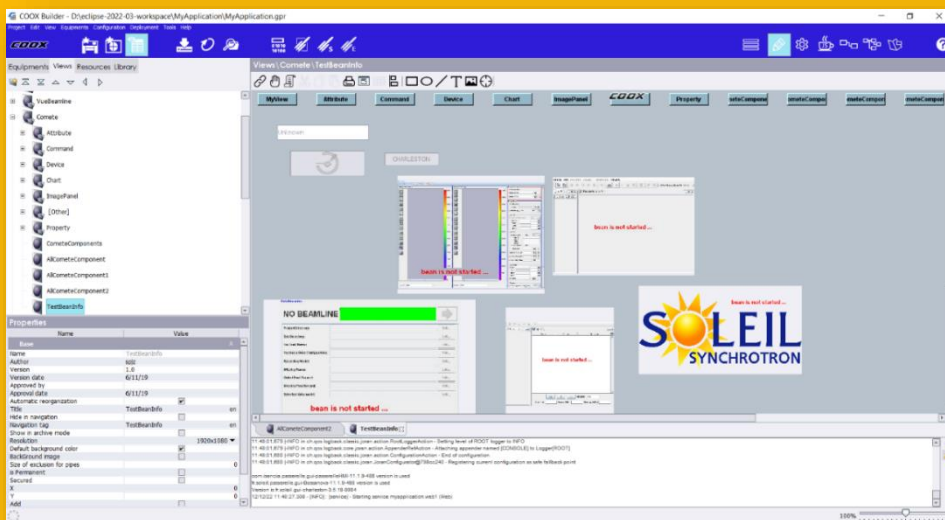
- Make applications independent of the storage structure of the accessed data.



- Widely used by beamlines, providing full autonomy in designing views and synoptics.
 - A **no-code tool** for creating GUIs using drag & drop.
 - Supplied and maintained by the French company **CODRA**.
 - Build in Java, allowing the use of our **Comete widgets**

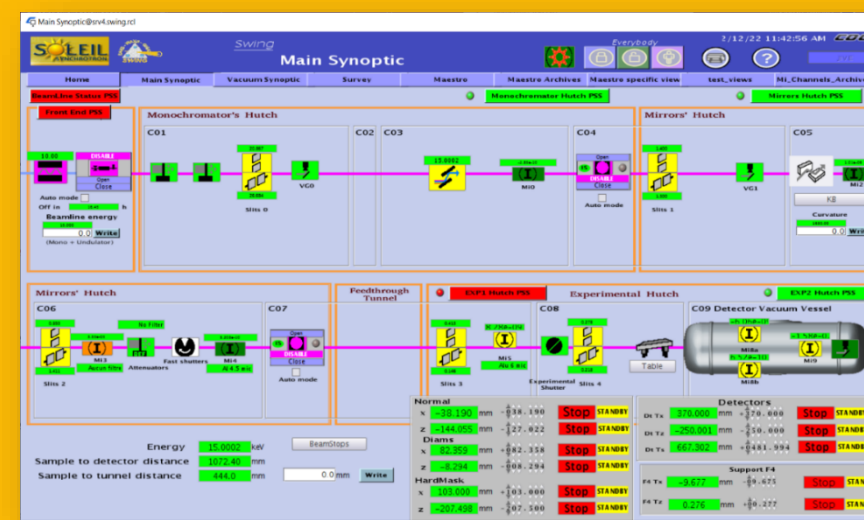
Builder

Creation of views by drag'n'drop
(uses SOLEIL's components with Comete)



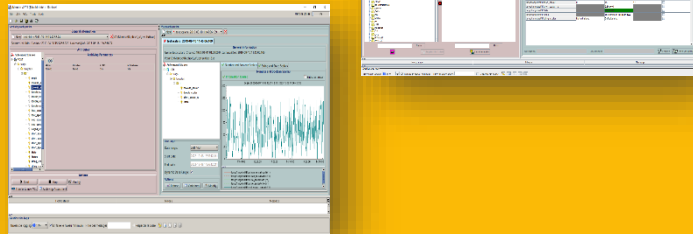
Viewer

Runs the views previously created with
Builder

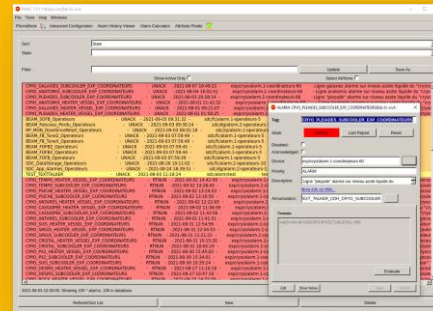


Tango Archiving

Configure & plot
HDB & TDB & Snap



Alarms - PANIC

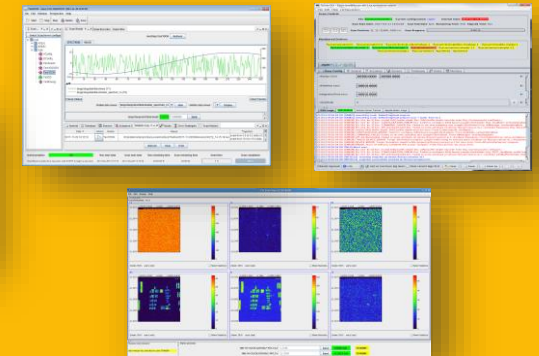


Acquisition

Spyc: SOLEIL Python CLI

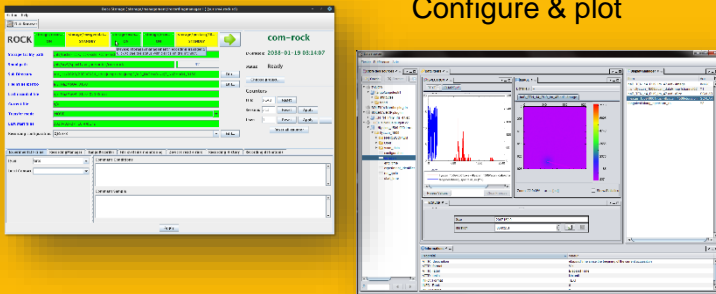


Fronts for scan devices
(FlyScan & ScanServer)



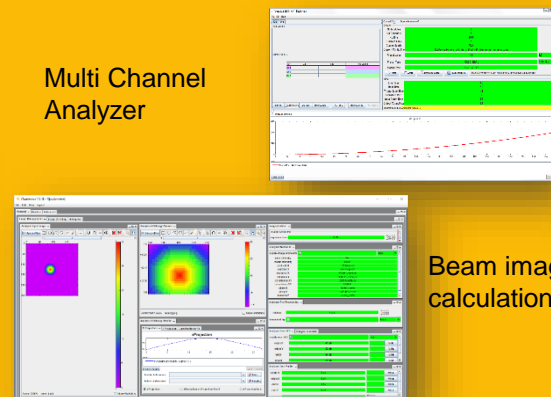
Experimental data recording

Configure & plot



Detectors

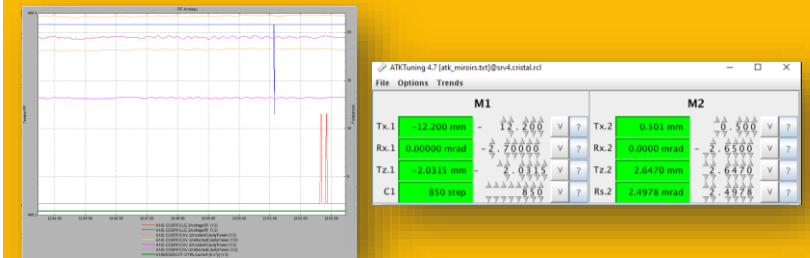
Multi Channel Analyzer



Beam image calculations

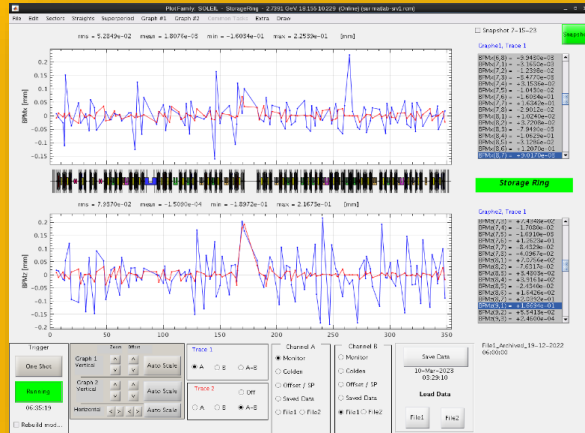
Tango tools

Large usage of ATKPanel, ATKtuning, Jive, Astor, ATKtrend

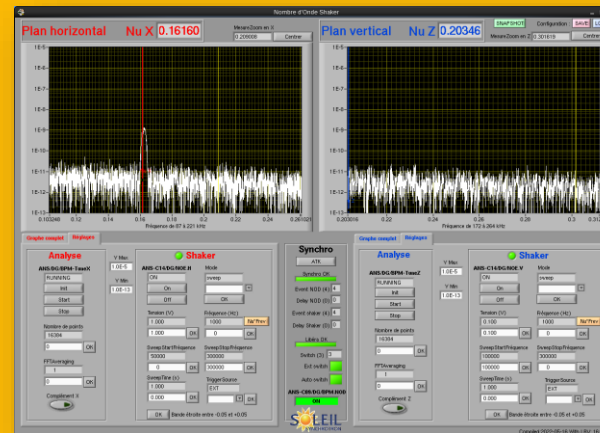


All the Java GUI based on Comete are also beans that can be integrated into COOX screens

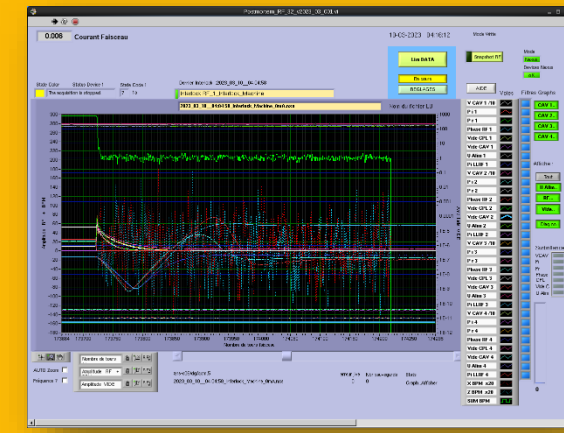
Beam positions - Matlab



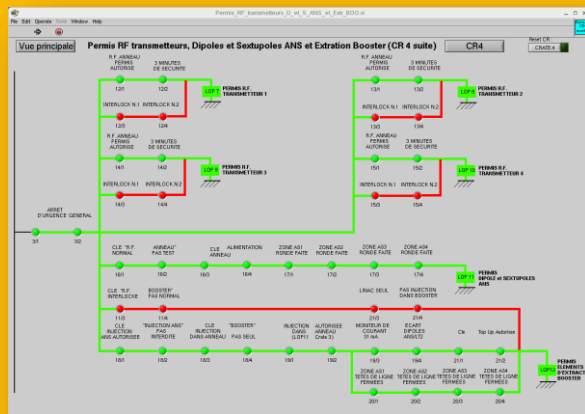
Beam Nu/WaveNumber measure - Labview



Postmortem analysis - Labview



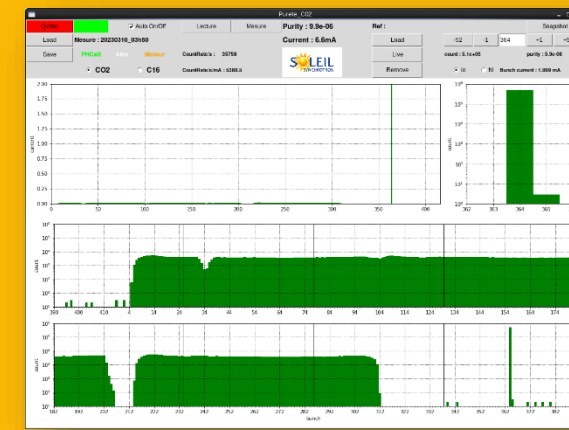
PSS permits (Personal Safety System) - Labview



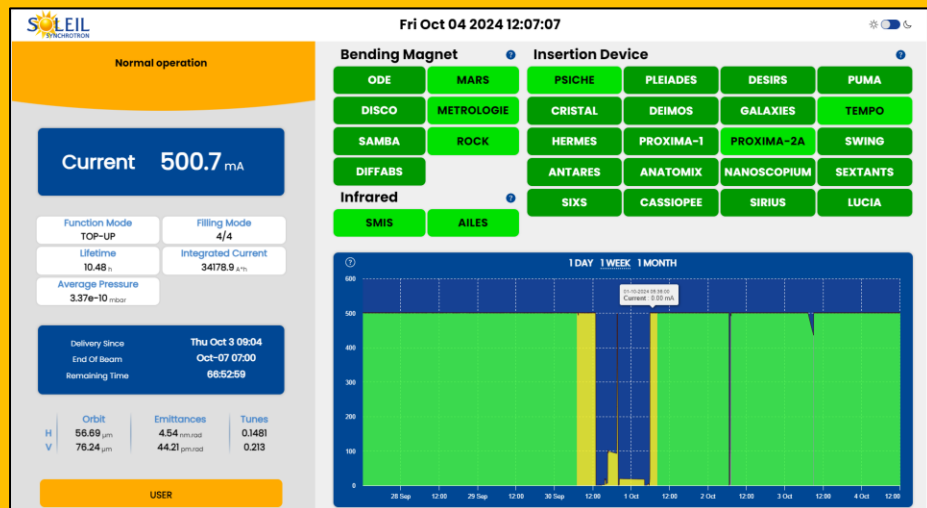
Utilities monitoring (AC & Fluids) - Labview



Beam purity measure - Python



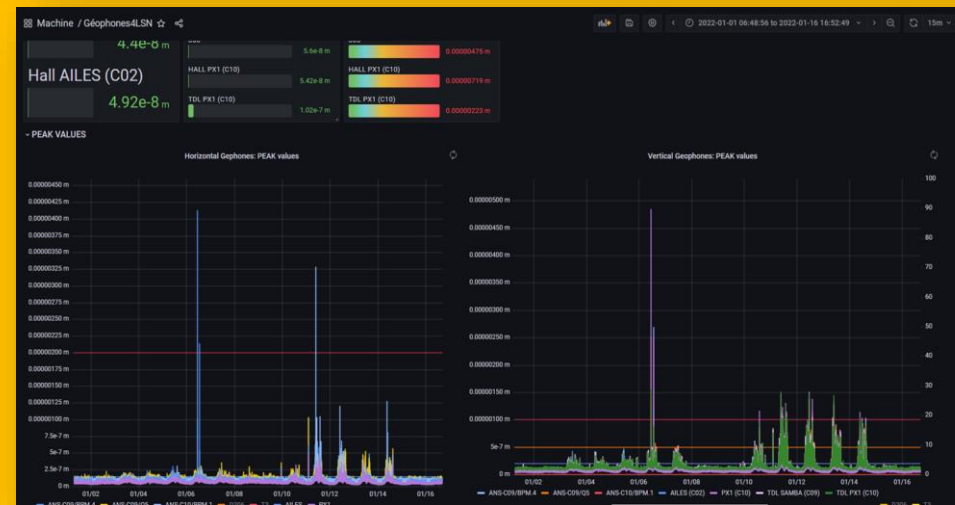
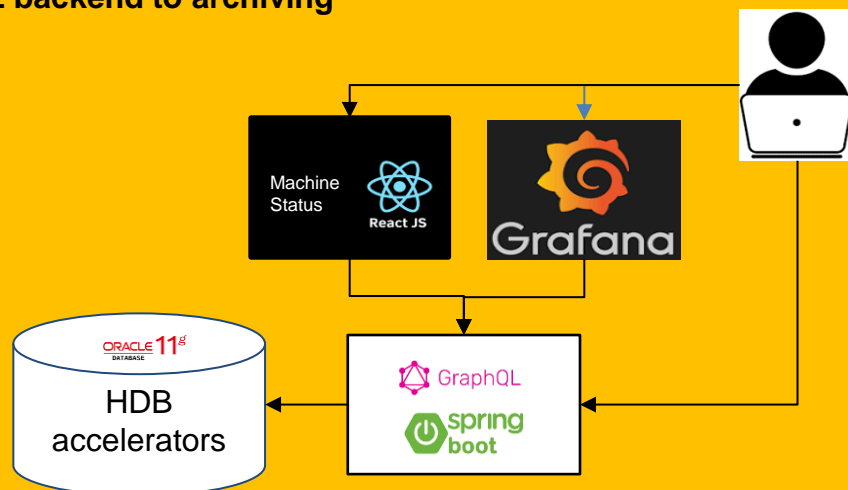
Machine status



Grafana



GraphQL backend to archiving



Taurus vs. Comete

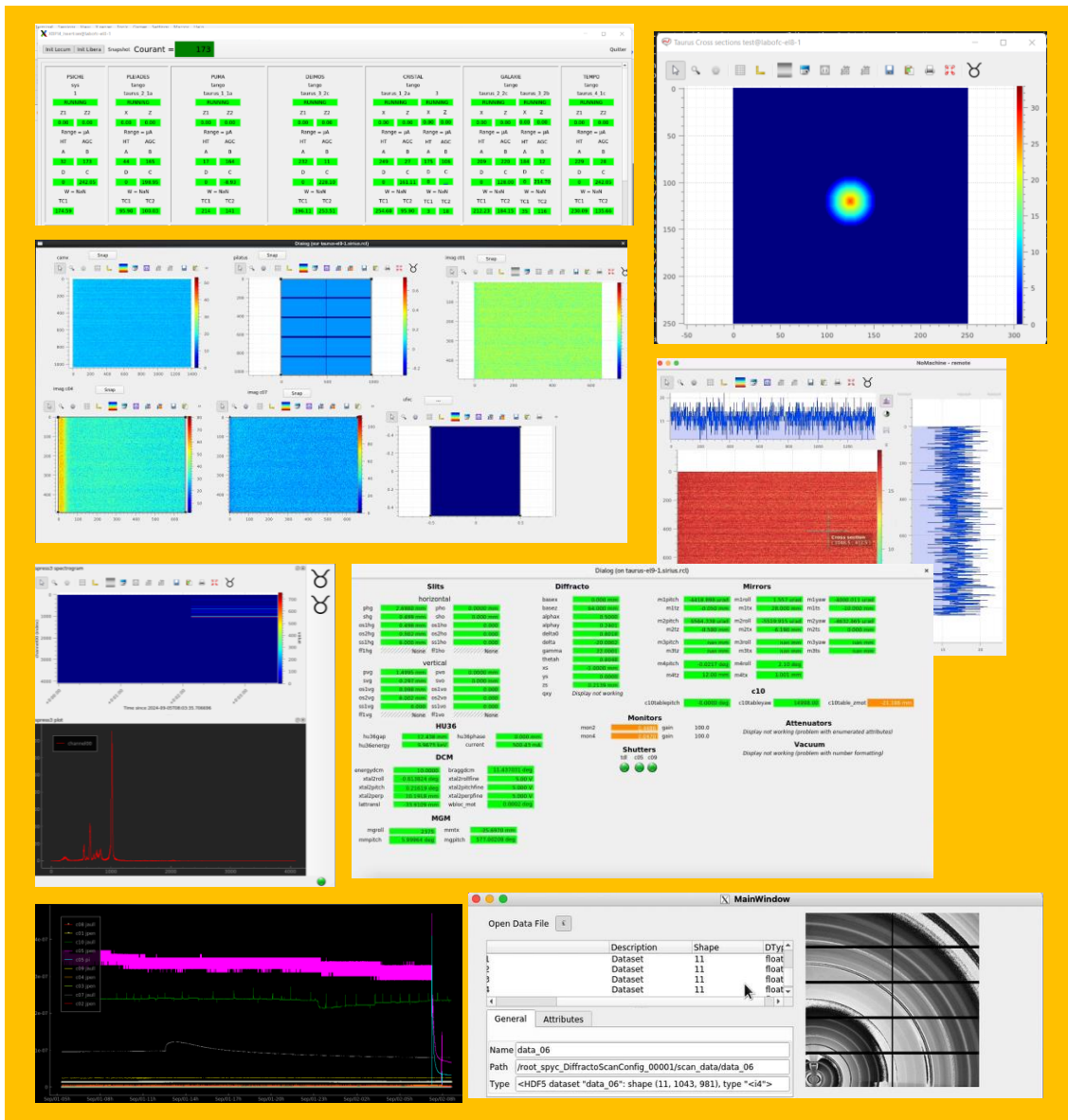
- Can Taurus widgets replace the SOLEIL Comete for GUI ?
 - TaurusImageDialog vs. Comete ImageViewer
 - TaurusPlot vs. Comete Chart
 - Basic chart and image widgets need further development to match Comete's features.
- Performance is similar between Comete and Taurus

Feedback

- Evaluation in progress** by the Accelerators teams and one beamline
- First feedback from these developer users on Taurus is positive

Our recent challenges

- Support of old Tango IDL
- Handling Tango host aliases
- Managing Tango / JTango events



UI Organization



Accelerators as owner of

Labview apps
Python apps
Matlab apps

Accelerators as user of

Tango GUI (Jive, ATK...)
PANIC
Databrowser
Mambo / Bensikin
MachineStatus
Grafana

152 LabVIEW apps
56 Python apps
114 Matlab apps
60 Grafana dashboards

Accelerators staff develop their own scripts/devices/GUI

- LabView, maintained by operators, acc. groups
- Matlab, maintained by physicists
- Python (TkInter, scripts, Tango devices), maintained by operators, physicists and acc. groups

GUI apps that embed the process are critical

- Top-up Injection process (Labview)
- Some feedbacks (Slow orbit feedback, ...) (Matlab) ...

Lots of accelerators' GUI are only for experts, and not for operations

Beamlines as owner of

Python apps
Igor Pro apps
Labview apps

Beamlines as user of

Tango GUI (Jive, ATK...)
PANIC
COOX
Salsa / FlyScan GUIs
Datastorage/Databrowser
Mambo/Bensikin
MachineStatus

Beamline staff develops their own scripts/devices/GUI

- Python (TkInter, scripts, Tango devices)
- On a few beamlines and laboratories
 - Igor Pro,
 - LabVIEW
 - Matlab

Beamline staff creates and maintains views of COOX synoptics

IT teams as owner of

Tango generic GUI (Jive...)
 Comete framework
 GUI based on Comete
 PANIC
 COOX
 Spyc
 Tango Bindings
 Grafana

*Support on deployment of products
 (Matlab, Labview, Python...)*

Tango

Nexus

IT teams provide common GUI solutions (Comete, COOX, Tango binding)

- Development, delivery, deployment, training and support

IT teams provide a very limited support for Matlab, Labview, Python and Igor Pro

- Provides help for deployment of these products, their licenses and their tango bindings

Resources for GUI (not fulltime)

- ISAC :
 - 1 expert for Java Swing GUI development
 - 1 expert for back-end in Java
 - 1 expert for front in React.JS
- ISI:
 - 1 sys. admin for products installation, OS upgrades

Accelerators

Accelerators as owner of

Labview apps
Python apps
Matlab apps

Accelerators as user of

Tango GUI (Jive, ATK...)
PANIC
Databrowser
Mambo / Bensikin
MachineStatus
Grafana (under evaluation)

IT teams

IT teams as owner of

Tango generic GUI (Jive...)
Comete framework
GUI based on Comete
PANIC
COOX
Spyc
Tango Bindings
Grafana (under evaluation)

*Support on deployment of products
(Matlab, Labview, Python...)*

Tango

Nexus

Beamlines

Beamlines as owner of

Python apps
Igor Pro apps
Labview apps

Beamlines as user of

Tango GUI (Jive, ATK...)
PANIC
COOX
Salsa / FlyScan GUIs
Datastorage/Databrowser
Mambo/Bensikin
MachineStatus

- Feedback on user autonomy
 - Positive feedback from COOX users
 - Positive feedback on apps with Tango bindings
 - Positive feedback on Grafana
 - Initial feedback on Taurus is also positive
- Balancing concerns between IT teams and Business teams :
 - IT teams: Focus on stability and reliability
 - Business teams: Focus on flexibly and changes ... or not ...
- Long term maintainability
 - How to position support between IT teams and business teams?
- IT Expertise
 - As some point, IT expertise becomes crucial for operating GUI applications, as they interact with various IT components (Tango devices, OS, network, etc.).



Future plans



Upgrade program has a project dedicated to UI

- Strategy definition is under construction in the IT & Data management program
- Accelerators, beamlines and computing teams working together on the project

UI requirements

- Shared development between business & computing teams
- Well-known technologies are important -> Possibility to sub-contract
- Existing framework with a strong community to be able to share experience/dev
- Control layer between Tango device servers & GUI:
 - Add security/authentication: to allow only approved clients
 - Add requests tracking: to track clients requests and abnormal usage

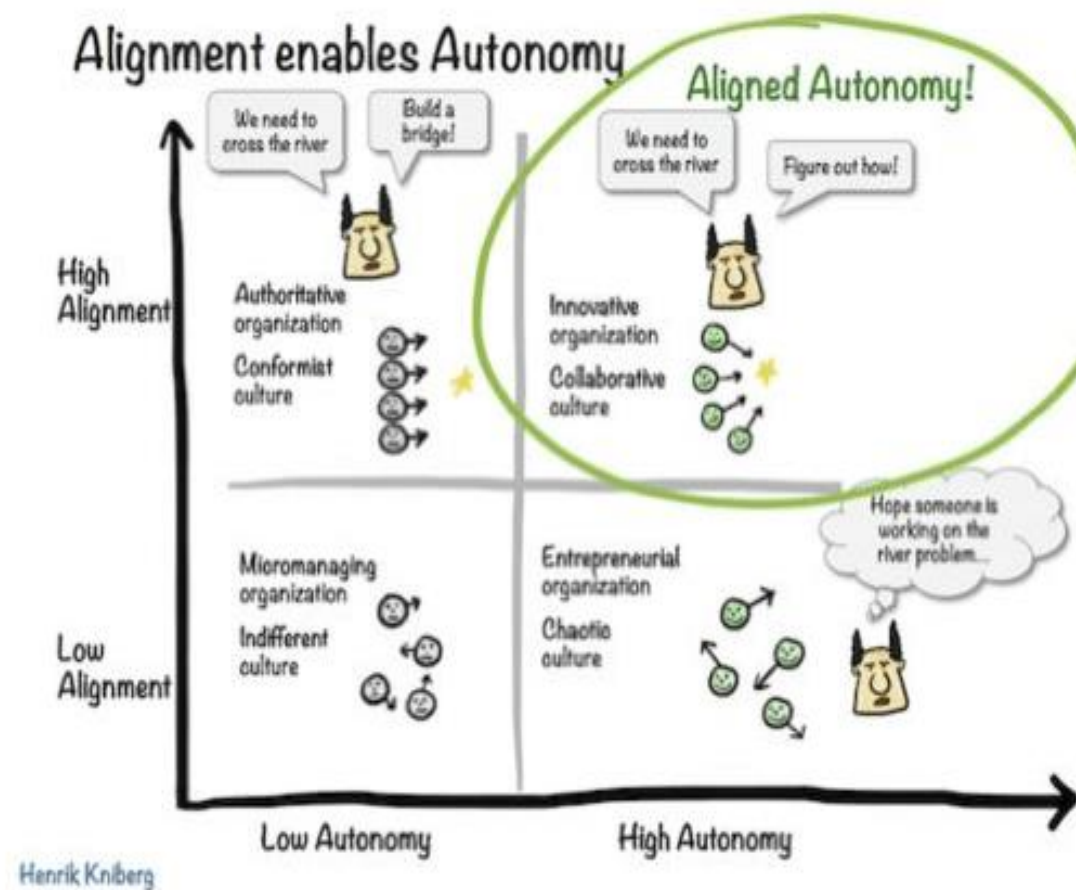
UI perspectives:

- Maintain existing Java Swing
- Deploy Grafana for monitoring on the control systems
- Organize transition to new GUI technologies
- Provide Python ecosystem

Accelerators & Beamlines requests:

- Python/PyQt and other Python packages
- Web solutions

IT teams & business teams work together to share the strategy

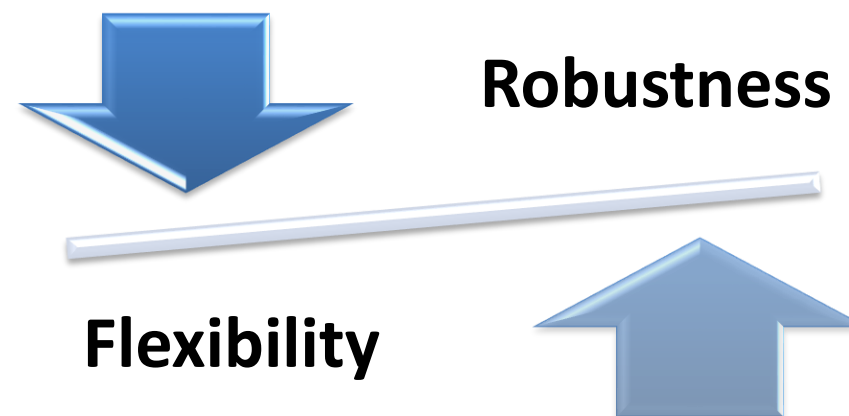


GUI strategy under construction in the context of SOLEIL II

Current challenge is to manage the transition

Balance between desktop and web

Looking for partners to share UI frameworks and experiences



Architecture and technology Transformation

complexity/reliability/maintenability

**Collaboration
Strong community**



Understanding organization around UI

- Who do what? How? Which policy did you choose?
- Experience of migrating to new GUI technologies: skills, complexity
- What about long-term maintenance of the GUI?
- UX Expertise?

Understanding operational architecture up to the desktop

- Cloud technology approach
- What about security for these upcoming solutions?
- What about dependencies management?
- How to transition smoothly?

Feedback about Taurus and Taranta

- Users' feedback
- Technical feedback: complexity, maintainability



