MAX IV Strategy regarding Graphical User Interface

Taurus workshop, ESRF

Vincent Hardion, 14/03/2023

Agenda:

Organizational & technical aspects

GUI Architecture

Strategy

Daniel Williams from NYC, USA - The Reactable

Disclaimer:

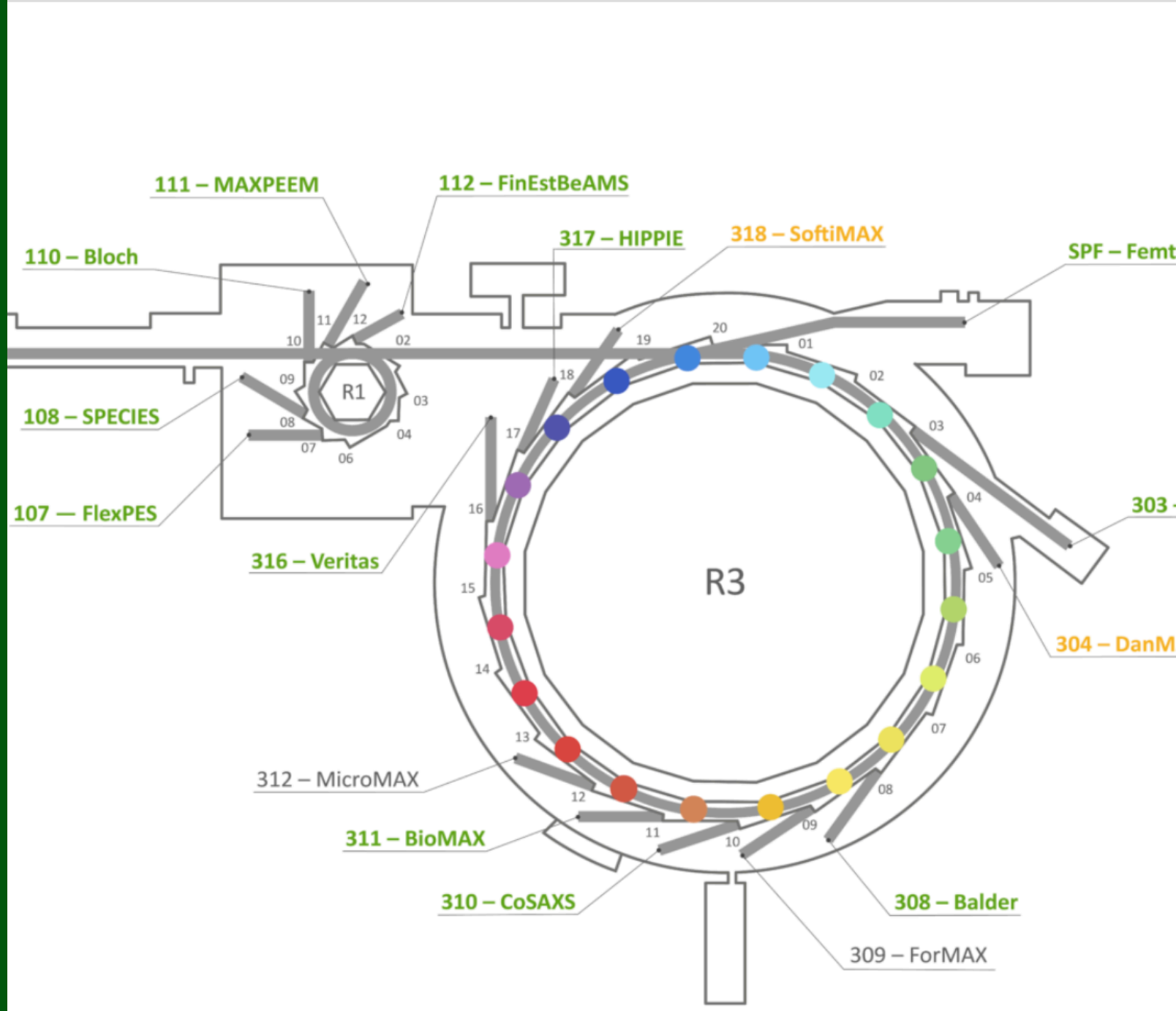
Strongly inspired by CERN GUI workshop but Taurus focused.

HTTPS://INDICO.CERN.CH/EVENT/1190700/

Daniel Williams from NYC, USA - The Reactable

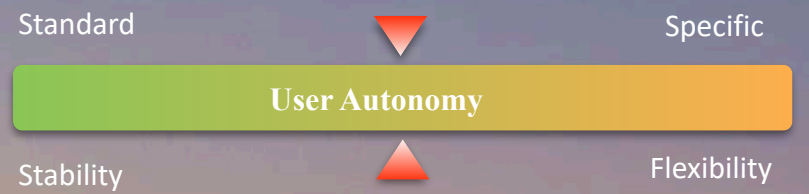
Organizational & technical aspects

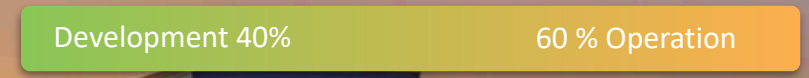
- MAX IV Laboratory: User Research Facility
  - Synchrotron
  - 6/7d, 24/24 h
  - system availability constraints: 99% in user operation



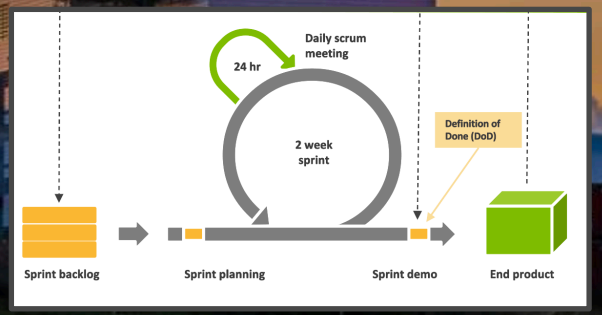
MAX IV

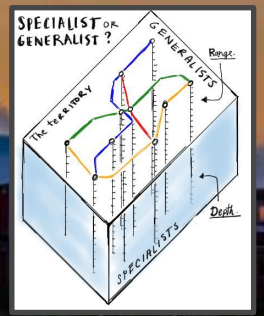
- MAX IV ~300p
- Computing service for Accelerators and Beamlines ~60p
- Visitors: > 1000 p / year
- Beamlines staff facing visitors
- Control System groups involved in the operation of MAX IV
- Software developers: Multi-skilled



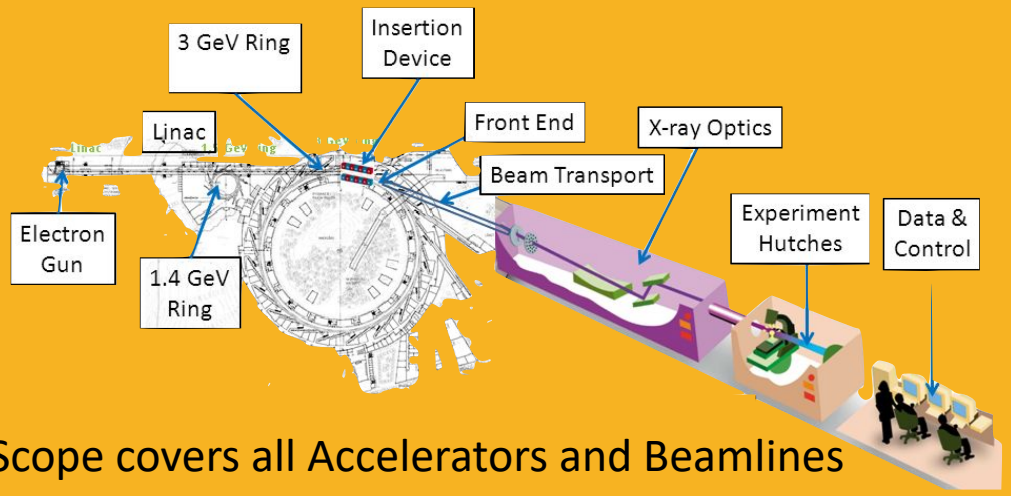


Technical Division				
Software (15p + 4p)	Electronic (13p)	Scientific Data (7p)	Infrastructure (10p)	Engineering I && II (30 + 30 p)

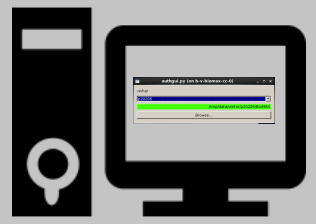




Control System Software responsibilities



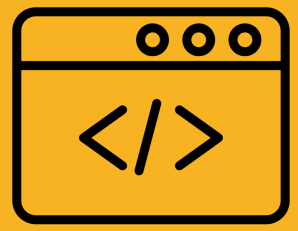
Scope covers all Accelerators and Beamlines
From Electron Gun until the data are recorded



~200 Desktop Applications based on pyQt (taurus)
- Custom UI
- Users and Developers: mainly Accelerator Operators and Beamline staff

Software domains covers from OS to User Application and services

Lasagna architecture



~20 Web Applications mostly based on React
- General Services: Archiving …
- Custom UI: ~80 Dashboard
- Developers: mainly Software Developers

User Communities

Accelerator Operators

Accelerator Physicists

Accelerator Subsystems Expert:
LAS, RF, PS, PLC, DIA, ID, …

400K

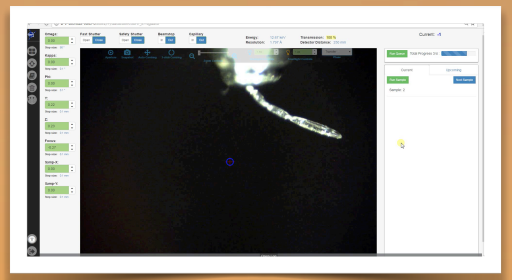
Visitors

Beamline Scientists

Beamline Engineers

Subsystem experts

40K x16



Data collection

User eXperience is important with ~1M Point of Control.

GUI have to be efficient and User Friendly.

Specific and Flexible

User Autonomy

Standard and Stable

Autonomy level	Need	Staff (operators, system experts, scientists)	Software Group	Technology	Example
Full Autonomy	User specific UI	**Design, implement, maintain and deploy**	infrastructure development, training and support	Desktop + Web	SVG Synoptic, Taurus GUI, Taranta Dashboard
Autonomous wo deployment	User specific UI	**Design, implement and maintain**	infrastructure, deploy, training and support	Full Desktop	various commissioning and operation app
Delegated	User generic UI	Specify, Design	**Implement, deploy and maintain**	50% Desktop 50% Web	Camera Application, State grid, …
Provided	Control System and Data Acquisition, generic UI	Help design UI	**Design, implement, deploy and maintain**	90 % Web 10% Desktop	Archive viewer, Scan GUI, Data log

MAX IV Software Developer Organisation Model

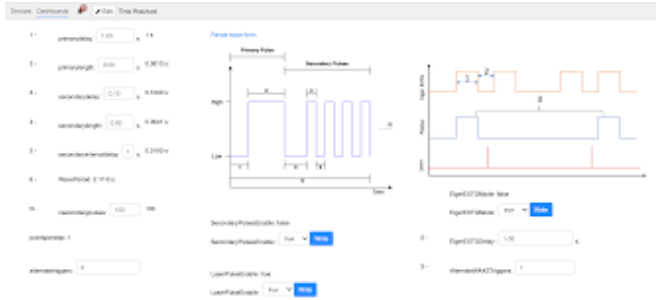
Specific and Flexible

User Autonomy

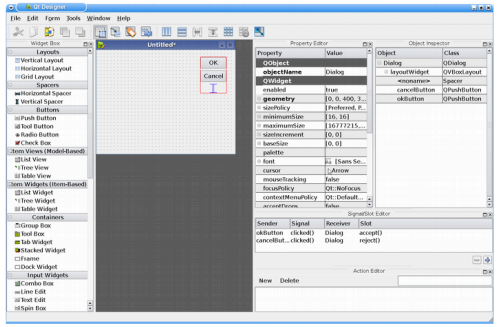
Standard and Stable

UI Framework	Competence
Taranta Dashboard Taurus GUI Synoptic (Taurus)	Programming level not a prerequisite. i.e UI design
Taurus Designer, Taurus	Basic programming level of Qt and python
Taurus	Programming level of Qt and python prerequisite
Web, Taurus	Expert Programming level is prerequisite: Javascript, Python, Qt, …

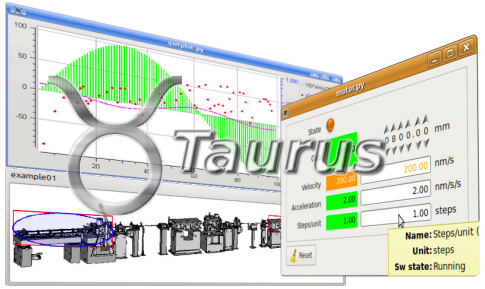
Taurus GUI



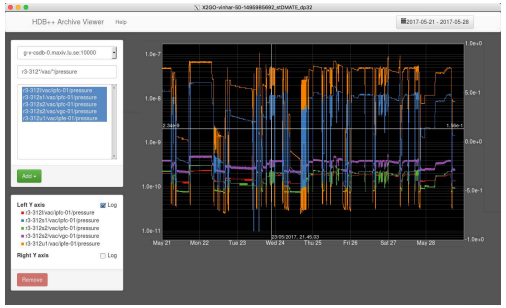
Taranta Dashboard



Qt Designer



Taurus



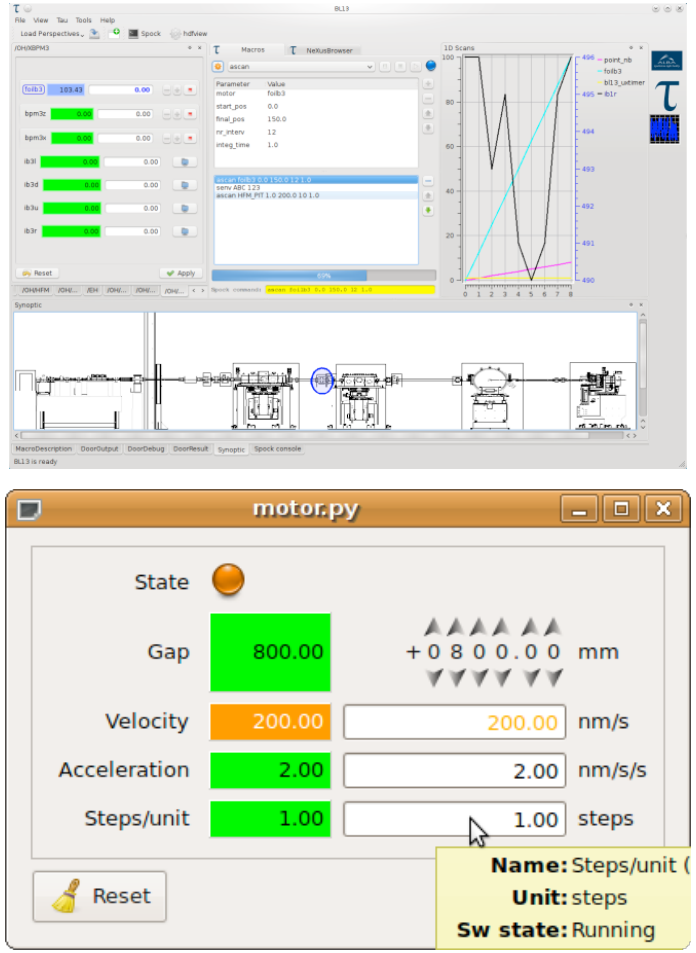
Web tech

MAX IV Software Developer Organisation Model

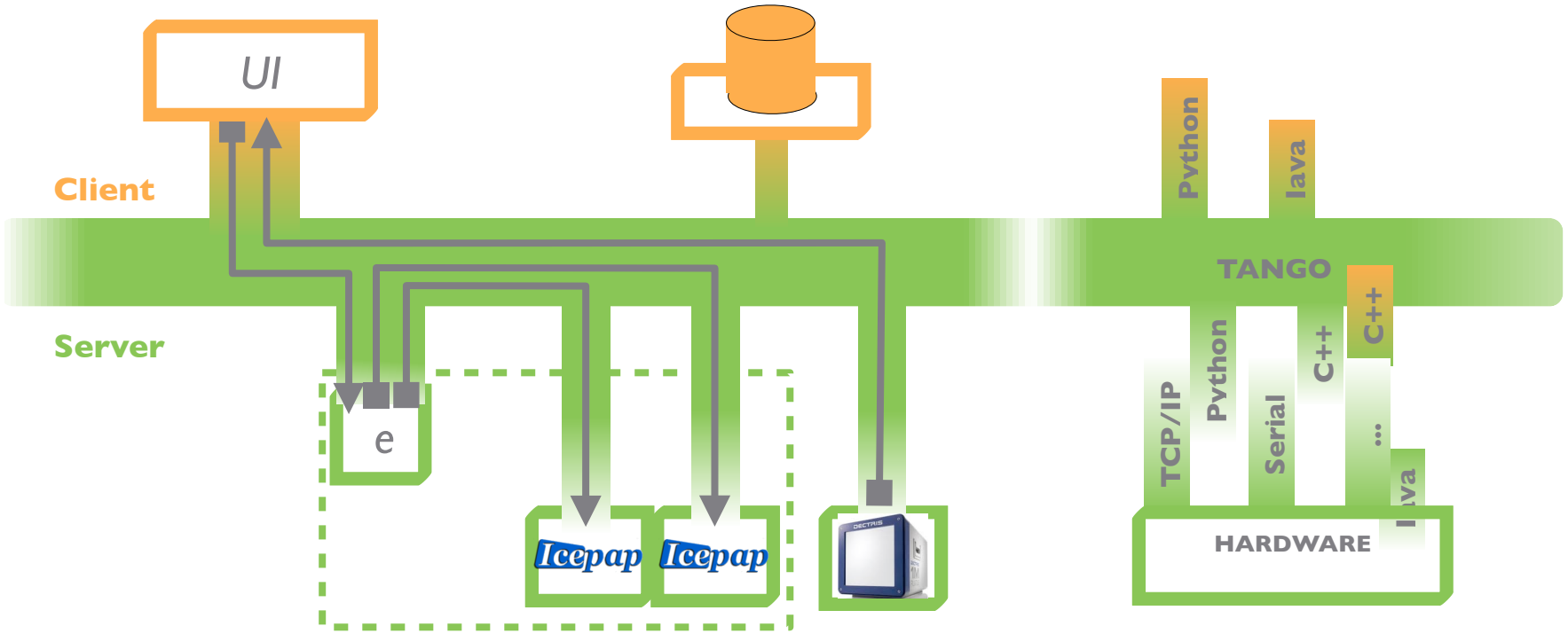
GUI Architecture

Torso Tower, Malmö
Susanne Nilsson, CC BY-SA 2.0

Common Tango Architecture

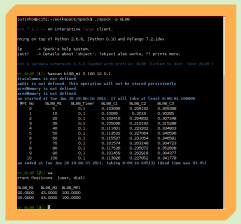
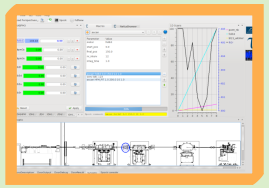


Taurus Rich GUI and form.

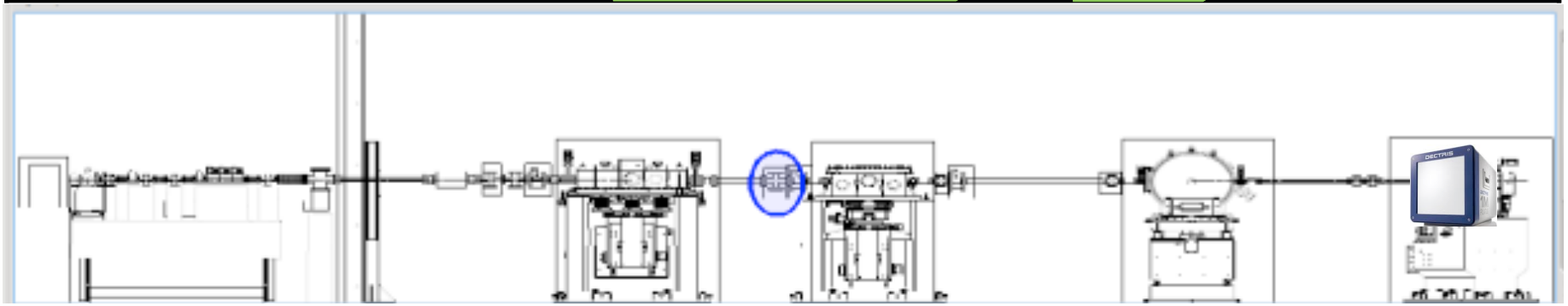


Tango is distributive control system acting as a software bus.
Each object has a self-descriptive API (Reflectivity) which make it very GUI - friendly.

Lasagna Architecture

SOFTWARE	User Interface				
	High Level Device	Undulator	Mono	Acquisition	
	Low Level Device		icepap	DAQ	
HARDWARE		Icepap		DAQ	

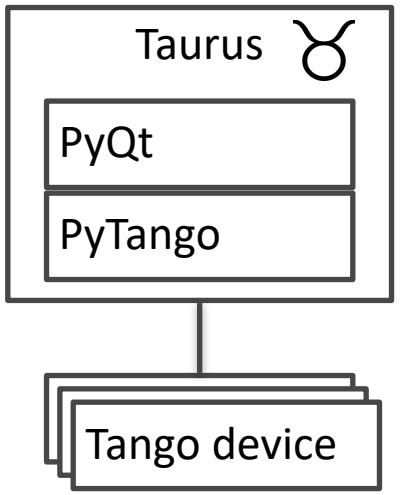
} 320 Classes of Tango Device



MAX IV hybrid GUI Architecture
Standalone and Central Application

Standalone Application

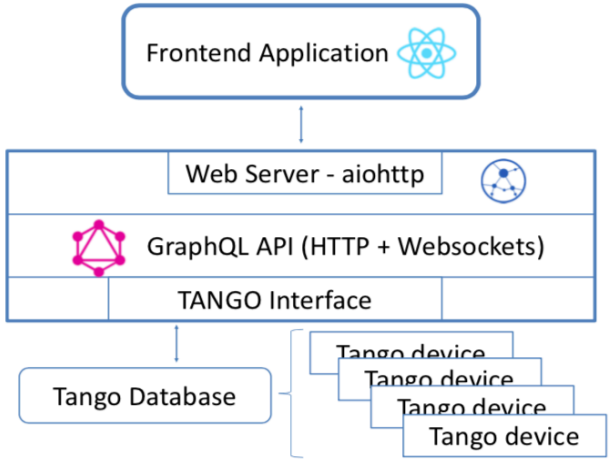
Taurus framework
- based on PyQt
- widget oriented
- UI designer
- generic Form
- Rich Client Application
- plotting based on PyQtGraph
- Running RockyLinux
- Deployed with conda



MAXIV could not be built on time without Taurus as it offers a very complete and ready GUI.

Central Application

Web Application
- mainly based on REACT
- widget oriented
- GraphQL
- Websocket, SSE
- plotting based on plotly datashader, etc
- Running on K8S



MAXIV North Star

And CLI like Jupyter?

Strategy for the future

Taurus is part of our core tech
mainly user of.

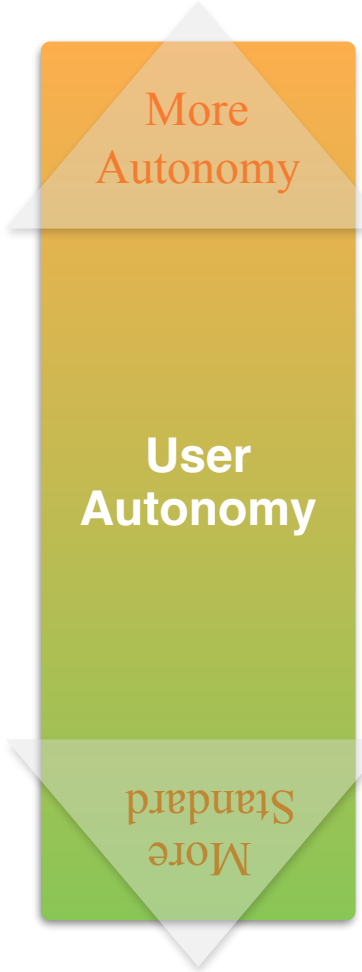
Taurus has always provided
95% of our need.

Web for standard application

Previously more CLI oriented

Strategy for the future

MAX IV UI Long-term Strategy



Strategy (in discussion)
More No-Code / Low-Code SW

Middle Earth Challenge :
USER-FRIENDLY SUPPORT
AI AUTOMATION
STABILITY LEVEL (TRL)

Collect Specific need => Extend Standard
2 UI projects:
REMOTE EXPERIMENT
GUIDED EXPERIMENTAL CONTROL

Desktop tech. challenges vs web

Strengths	Weaknesses
Most software developer knows Desktop development.	Web browsers are more and more ubiquitous
Qt and Python are popular in Science world.	Distributed application increased workload in deployment and resource management
Desktop can handle high performance (GPU, …). 1 client / application	Less and less taught to junior Developers for UI
Security is given for free by the Desktop Environment of the OS	Security at the application level is difficult/non existing i.e login with OS
UI style accepted by every one. Qt adapt well to the OS.	Execution dependant on the environment OS, …
Qt framework is quite the GUI standard for Desktop	Other users don't profit of the local change in UI.
Can perform computation of data in the same application	Debugging, logging and monitoring less standard than web front end
Specific development or adaptation does not impact others computer's users	UI mixed with logic can become quickly messy I.e monolithic application
Advanced users can just fork and change the app	Remote operation less practical
Opportunities	**Competition**
Reactivity of desktop application is better	Web: New UI design are modern, stylish..
Advanced User can program Qt application easily from their desktop	Responsive UI to work on smartphone and tablet
Taurus has a richer component collection for Tango	Central Resource offers better access to different Control Systems.
OS better at windows management than web browser	More Technology progress on Web (data analytic, …)
Desktop is simpler in architecture	Web proposed a better integration schema (link, frame, …)
Distributed means more reliable	Centralised infrastructure can propose automatic recovery

Conclusion

- Love Taurus and fully satisfied
- Logic in Tango devices allows to use simpler Taurus widget
- A lot of Expert UI
- Few standard complex UI (Synoptic, Camera, Scan, …)

- “hybrid” desktop and web UI strategy mainly due to maturity of competence and technology
- More Development in Web technology, part of the MAX IV strategy
  - Centralised applications with more No/Low-code

Question?

- Do you want to replace Taurus by web tech?
- Fall between chair?
- Responsiveness of Control group to deploy?
- level of support & training provided to non-expert developers?
- TO YOU: What about JupyTango with interactive widgets?