SOLARIS
CENTRE

SOLARIS CENTRE

GUI strategy at SOLARIS: status & future plans

TAURUS Workshop, 2023.03.14

Michał Fałowski

SOLARIS National Synchrotron Radiation Centre

NSRC SOLARIS institute

- Synchrotron radiation research + cryomicroscopy.
- Synchrotron accelerator, linear accelerator, 2 diagnostic beamlines, 5 scientific beamlines and 2 microscopes fully operational.
- 1 beamline during commissioning (CS already working).
- 2 beamlines under construction/development.

Co?

Control System in NSRC SOLARIS

- Accelerators and beamlines for control system use PLC (BMS, MPS, PSS) and TANGO (high level, mostly python, sometimes C++).
  - OS: CentOS 6, 7 (mainly), 8, Windows 10 + some embedded custom Linux and 1 Fedora (old one).
  - Python versions: 2.7 (GUIs), 3.6 (devices and GUIs), 3.7 (some web).
  - Tango versions: 9.3.5 (mainly), 9.2.5 (few systems left), 9.1 (embedded) and some 8 (e.g. old archiving).
  - Taurus versions: 3,7 (virtual env), 4.1 (accelerators), 4.4 (virtual env), 4.5 (beamlines) and 4.7 (for Py3.6 and PyQt5).
  - Additionally, some projects use Vue.js, Tango GQL and Taranta (for authentication).
  - Archiving: hdb++ + MySQL InnoDB

- Cryomicroscopes are fully independent and are based on manufacturer software.

Control System in NSRC SOLARIS - numbers

- Around 4000 Tango Devices.
- Around 600 Tango Classes.
- Around 800 Servers instances.
- 9 tango hosts.
- Over 100 hosts.
- 20 workstations.

GUIs in SOLARIS

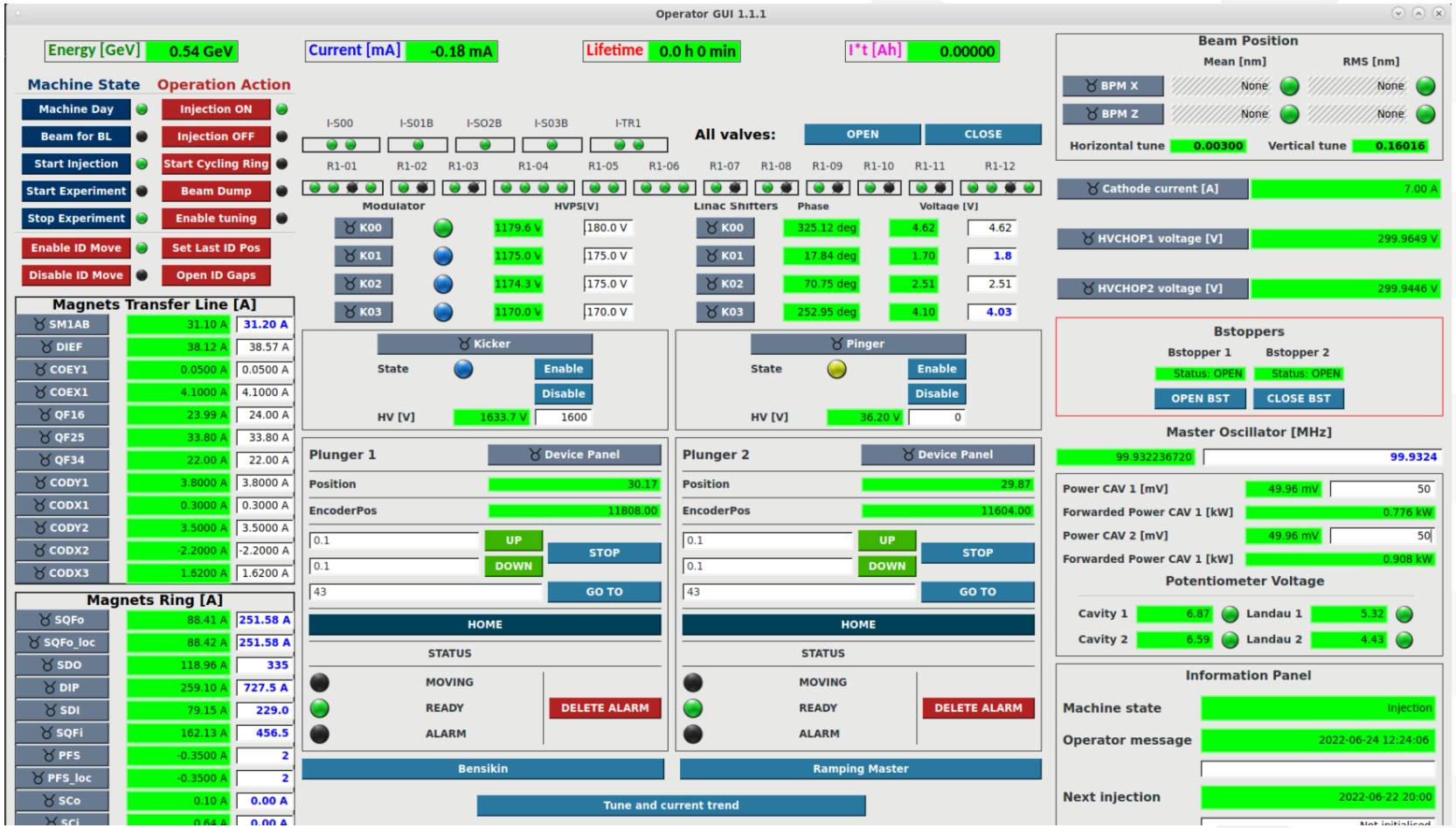
- Most of the GUIs are based on Taurus.
- We have few web applications (Vue + TGQL + custom web archive reader based on PyTA).
- 1 QTango GUI.
- There are some pure PyQt (with custom Tango communication) and MATLAB GUIs, mostly written by operators and scientists.
- Most of the software for research is provided by manufacturers and is Windows based.

GUIs in SOLARIS - spectrum

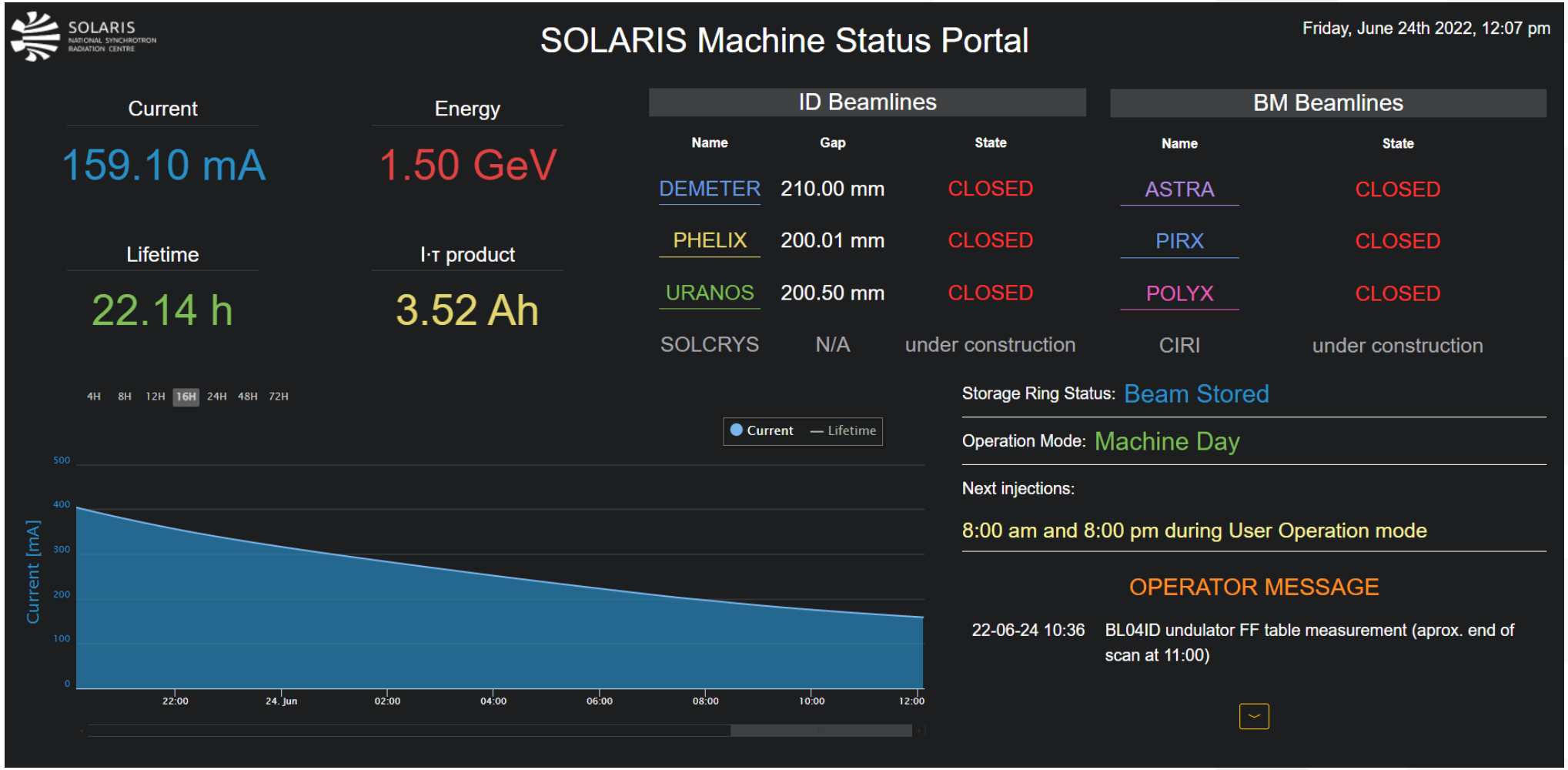
- Over 100 applications in repository.
- The range of applications is very wide:
  - Simple forms with few buttons and labels.
  - Panels to control single devices from power supplies, ion pumps, to Undulators and Cavities.
  - Complicated GUIs to operate entire machine (e.g. RF section, beam injection), setup beamlines and execute sample scans (also based on Taurus GUI).
  - Synoptics (JDraw and svgsynoptic2) of simple sections to entire beamlines.
- Many additional trends and forms on workstations (also based on Taurus GUI).
- Few operators and scientists' GUIs.
- Some GUIs have over 150 attributes connected.
- ATK panels are run from Jive (in some cases to many clients can kill servers).

Halo!

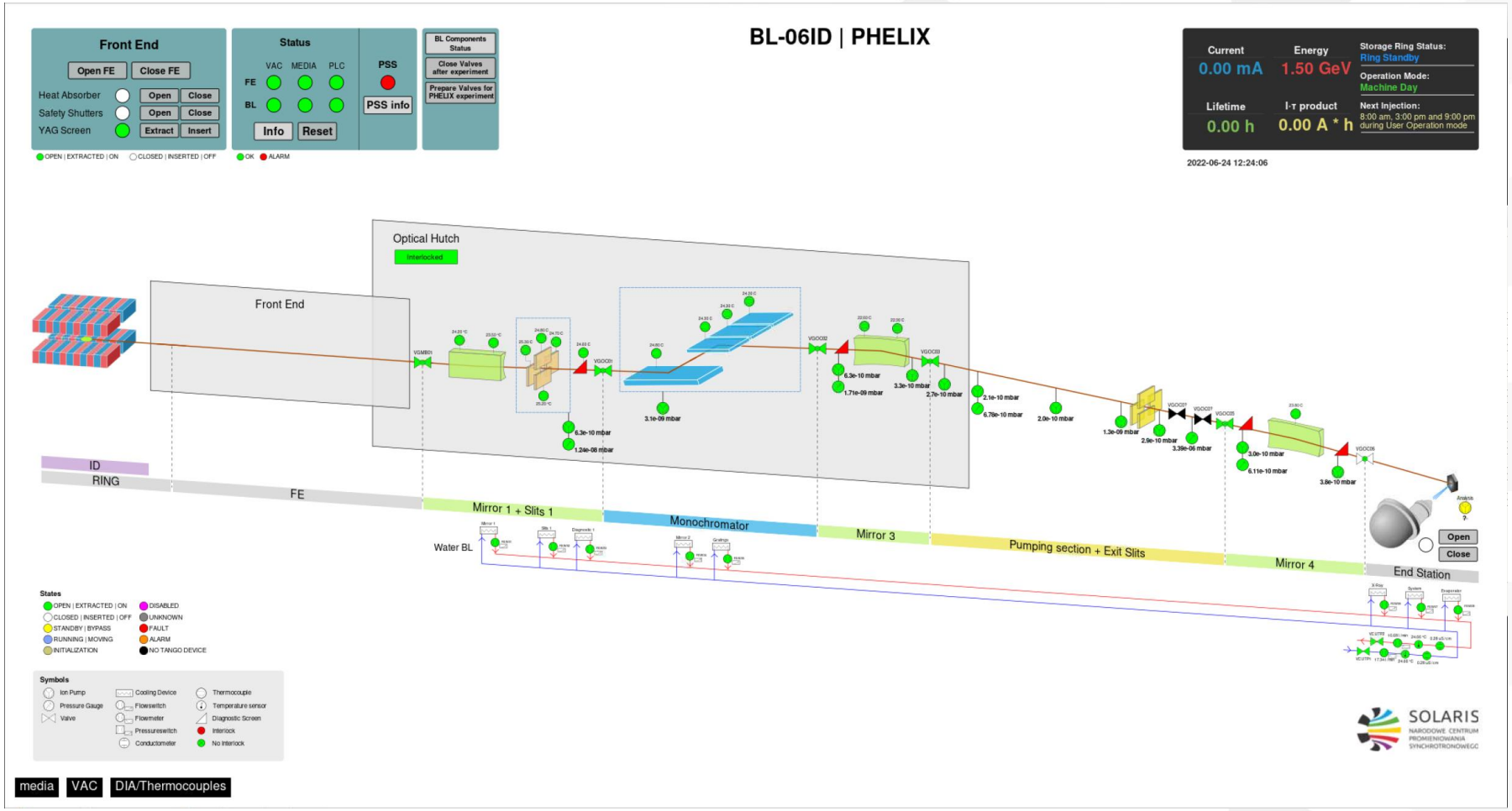
GUIs in SOLARIS – examples



GUIs in SOLARIS – examples



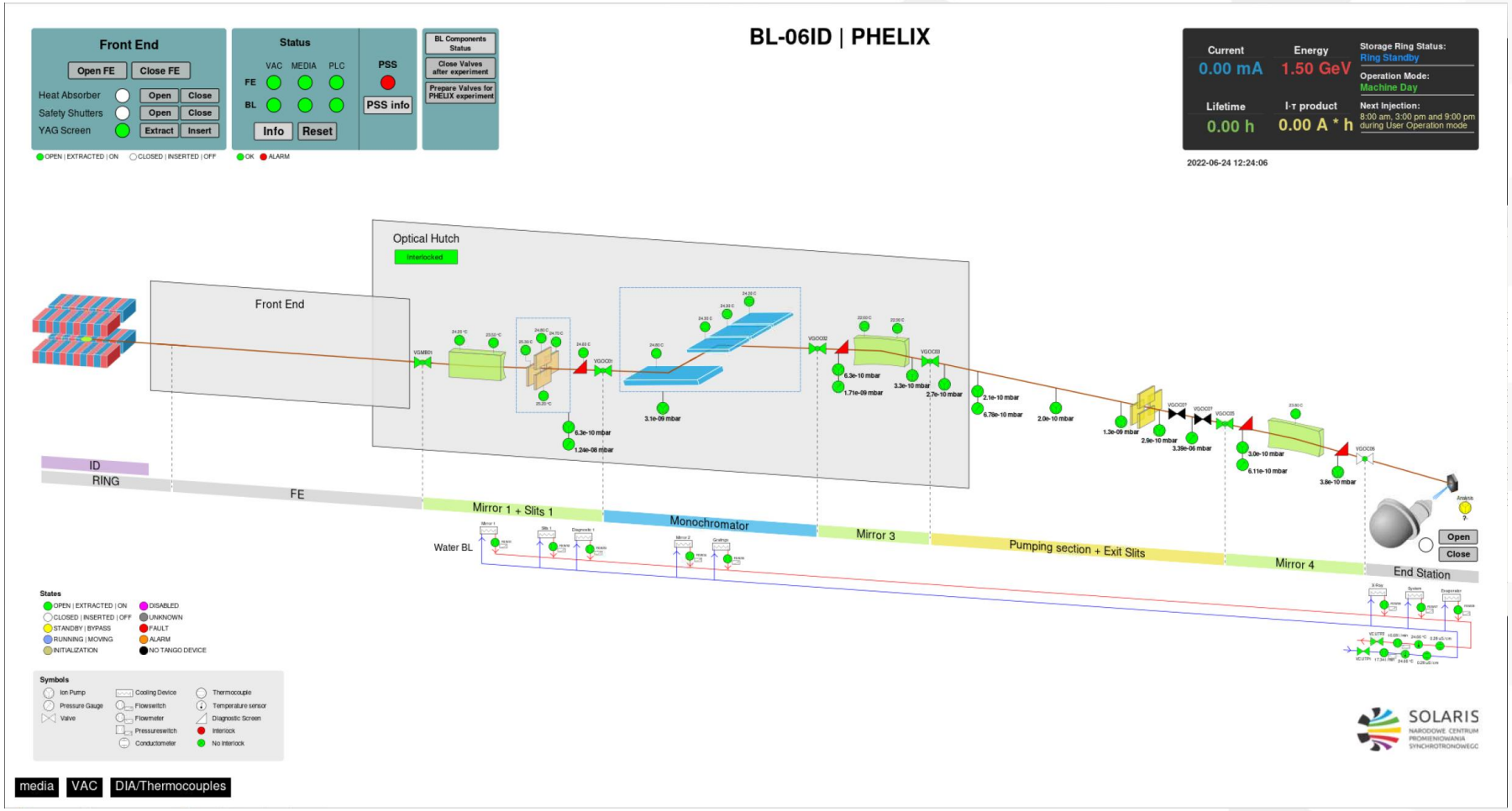
GUIs in SOLARIS – examples



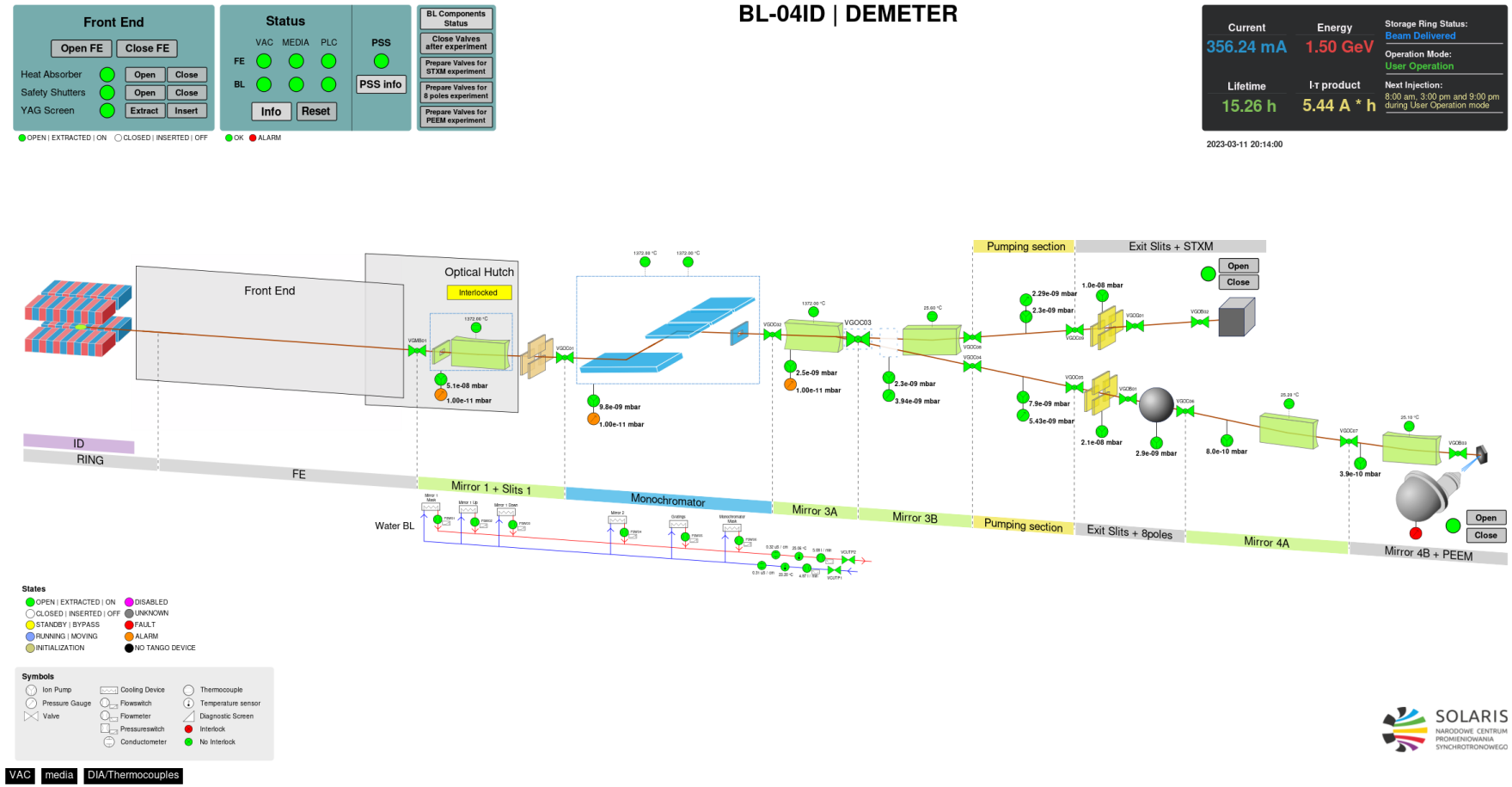
GUIs in SOLARIS - standards

- In every new GUI logic should be moved to server side (e.g. facadedevice, Sardana controllers) and, if possible, also in maintained GUIs.
- Every new GUI should not use older versions than Py3.6 and Taurus 4.7 (unless Qwt or synoptic is needed).
- Taurus is preferred over other frameworks.
- svgsynoptic2 is obligatory for new synoptics.
- Web applications should use Vue.js.
- GUIs and web apps for similar purposes should have similar view and project structure (e.g. beamline operation GUIs are in one project where all configs are stored).
- Synoptics for beamlines and fronted uses same layout, symbols, colours, panels and logic.

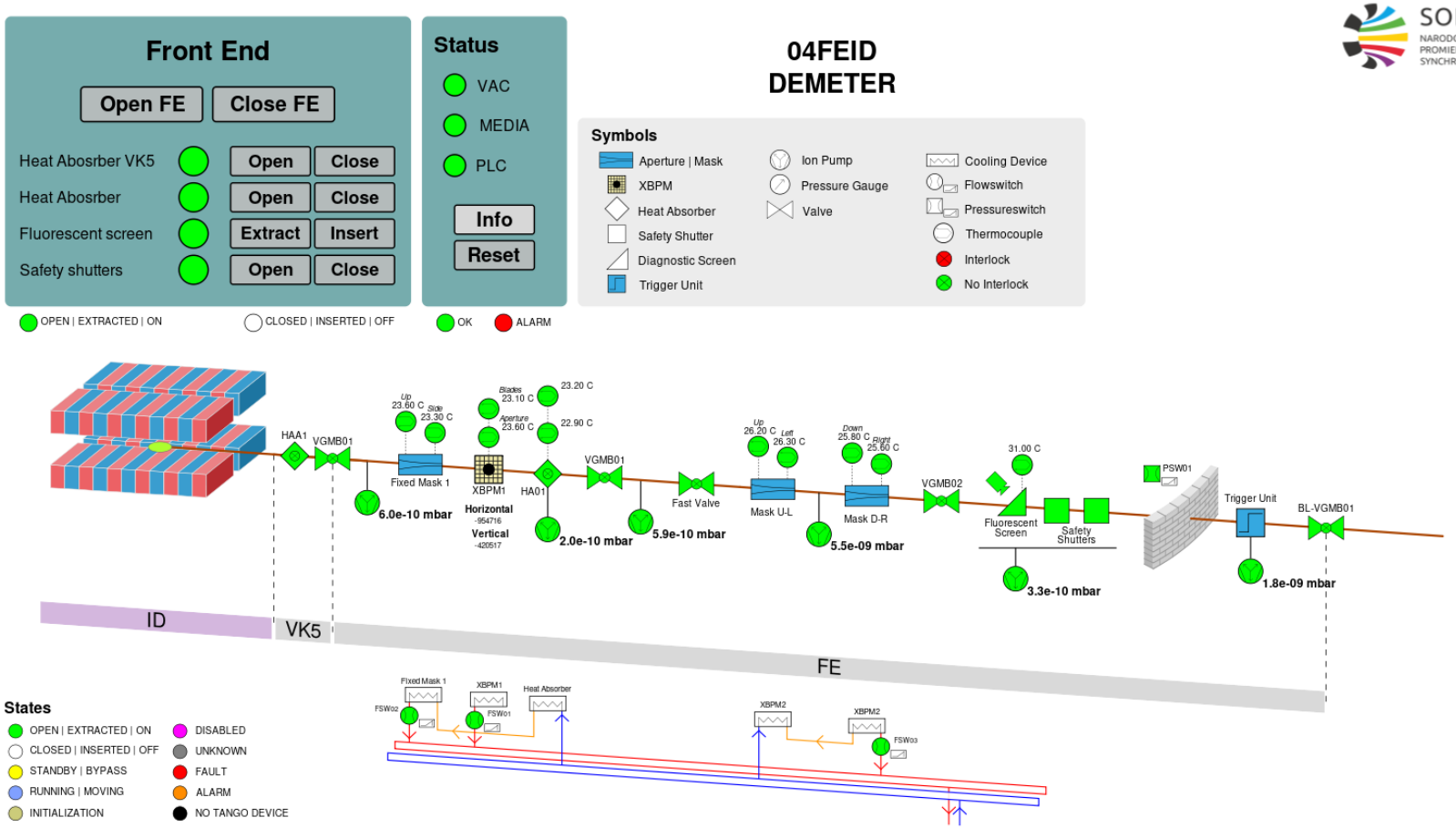
GUIs in SOLARIS – examples



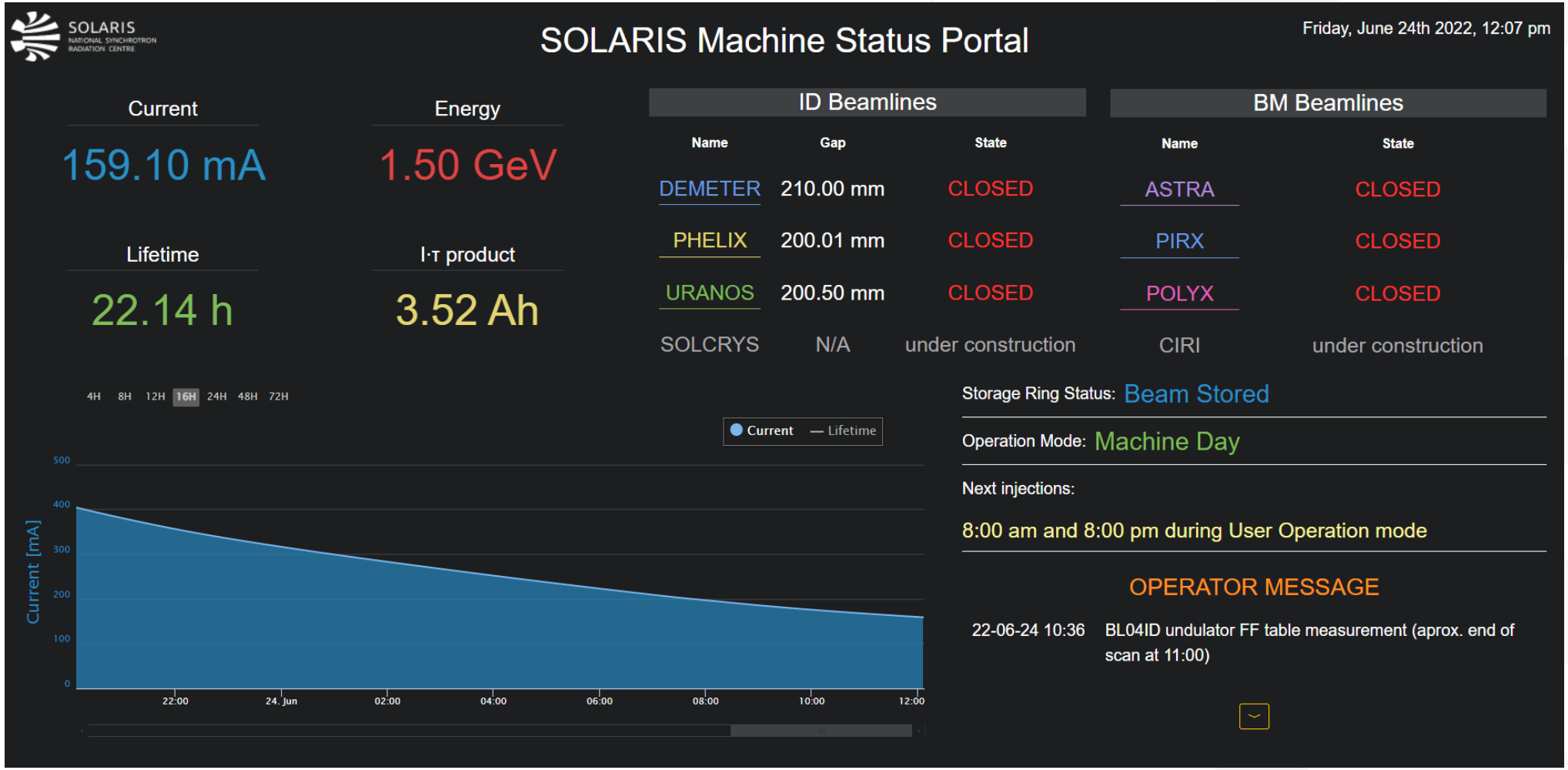
GUIs in SOLARIS – examples



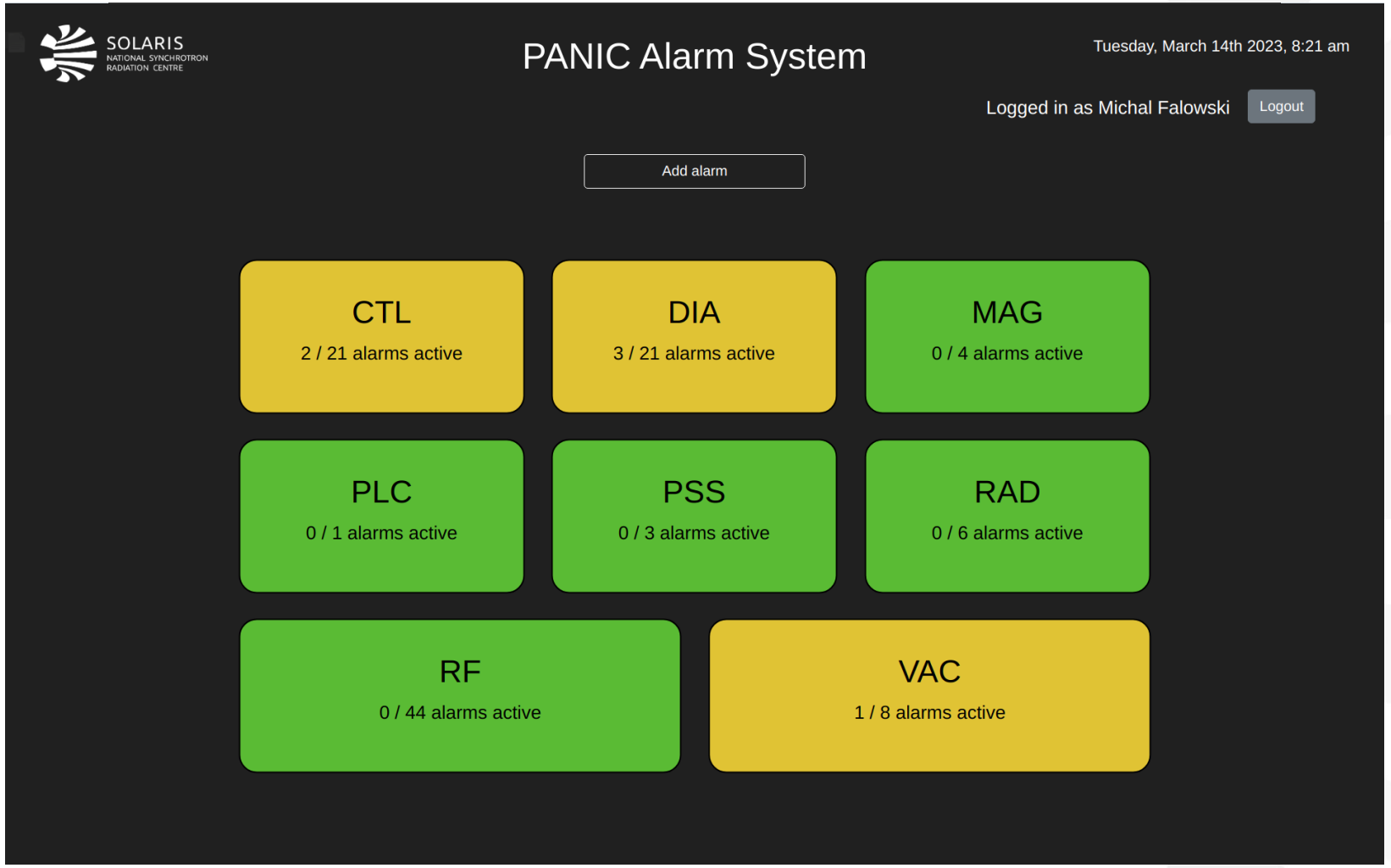
GUIs in SOLARIS – examples



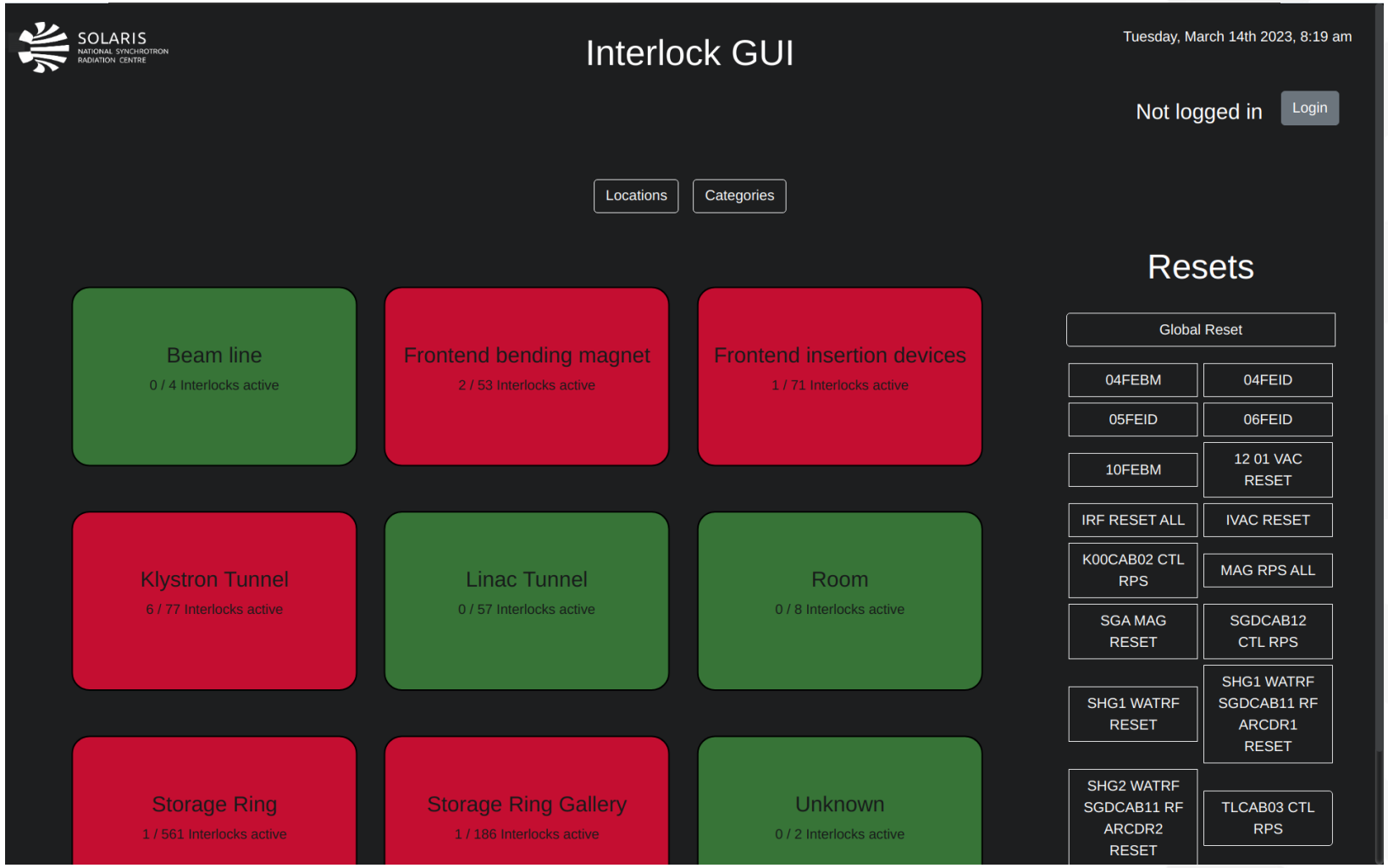
GUIs in SOLARIS – examples



GUIs in SOLARIS – examples

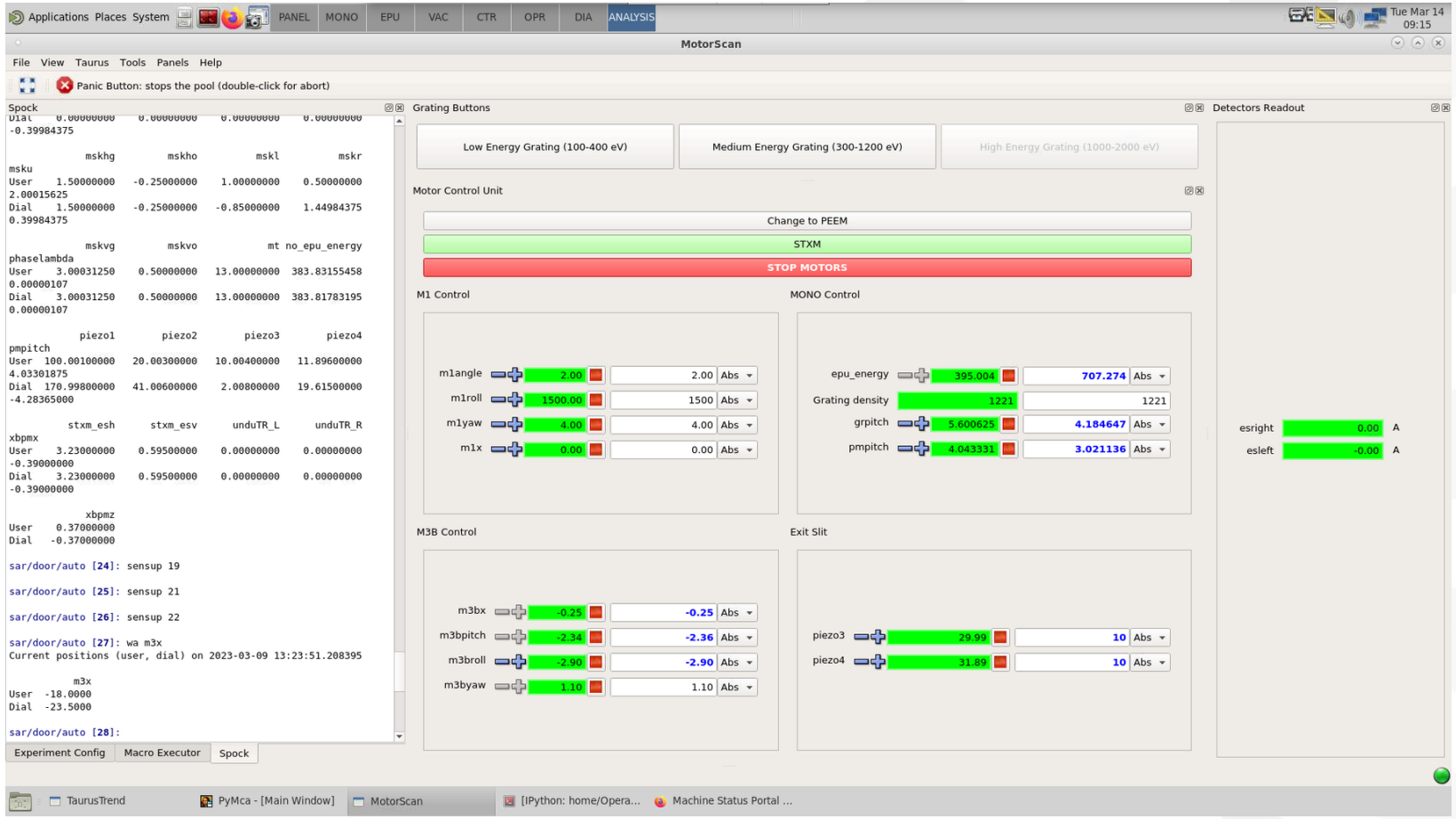


GUIs in SOLARIS – examples

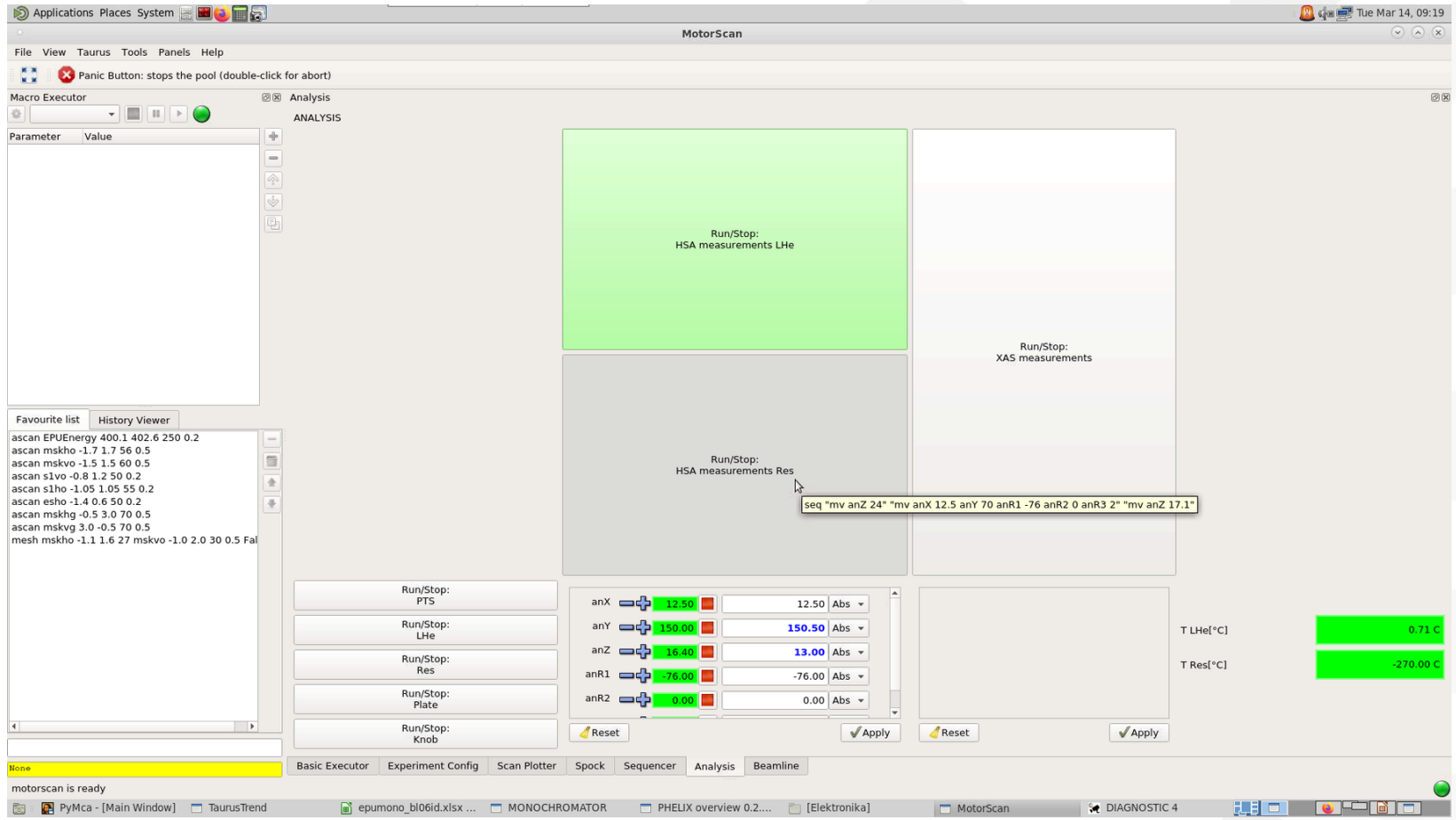


GUIs in SOLARIS – examples

GUIs in SOLARIS – examples



GUIs in SOLARIS – examples



GUIs in SOLARIS - team

- Most of the GUIs are created and maintained by CS team.
- 9 CS employees (7 full-time).
- Almost everyone in CS team takes care of at least a few GUIs (who also takes care of part of the infrastructure, devices etc.).
- Operators and scientists can also create GUI (there are already some advanced GUIs made by operators).
- At the request of operators, we can install additional software (mostly python libraries).
- We try to train willing people about TANGO and version control (to use git and our internal GitLab).

GUIs in SOLARIS – organizational aspects

- There are still many old GUIs with many custom views and logic inside the code.
- So many applications make it hard to keep them maintained while there is so high personnel rotation.
- There is no one able to keep track on all GUIs, so review is also limited.
- That is one of the reasons why new logic should be moved to server side, as this would allow to more complete code review.
- But sometimes duplication of functionalities are still occurring.
- GUIs are harder to update, mostly because they depend on more libraries and there are no backward compatibilities more often.

GUIs in SOLARIS – feedback

- So many needs requires independence in creating simpler GUIs by operators and scientists.
- Web applications are warmly received but needs more care and advertisement.
- UX/UI experience is lacking which causes sometimes complaints and disagreements on looks.
- Crashing and load time is the biggest reason for hindering the work.

GUIs in SOLARIS – TAURUS issues

- https://gitlab.com/taurus-org/taurus/-/issues/654 - TAURUS not react to Periodic Events.
- https://gitlab.com/taurus-org/taurus_pyqtgraph/-/issues/112 - Taurus GUI is not restoring trend curves cfg.
- https://gitlab.com/taurus-org/taurus_pyqtgraph/-/issues/104 - Disappearing Y2 axis .
- Very, very, very long startup if there are many attributes connected.
- Crashing with segmentation fault.
- New Taurus not compatible with older config (or no migration scripts at the startup).
- Problem with taurus designer (missing dependency but without information).

GUIs in SOLARIS – future plans

- Migrate most of the GUIs to at least Taurus 4.7 (next Sardana update requires Taurus >= 5), Python 3.6 and PyQt 5.
- Try to move all trends to PyQtGraph.
- Try to migrate all trends to new Taurus.
- Create more web apps.
- Try to introduce Taranta to other users (again).
- Standardize way to deploy custom operators' GUIs to entire system.

Thank You!