HELMHOLTZ

ROCK-IT

ROCK-IT GUI STATUS: DESY, HZB, HZDR, KIT

Controls and Acquisition GUI Strategies Satellite Workshop at NOBUGS 2024 | 23 September 2024

Dr. Zeynep Isil Isik Dursun (DESY) | Communications Specialist & Visual Designer @ ROCK-IT

DESY. HZB Helmholtz Zentrum Berlin HZDR KIT Karlsruhe Institute of Technology

THE ROCK-IT PROJECT

Remote, Operando Controlled, Knowledge-driven, IT-based

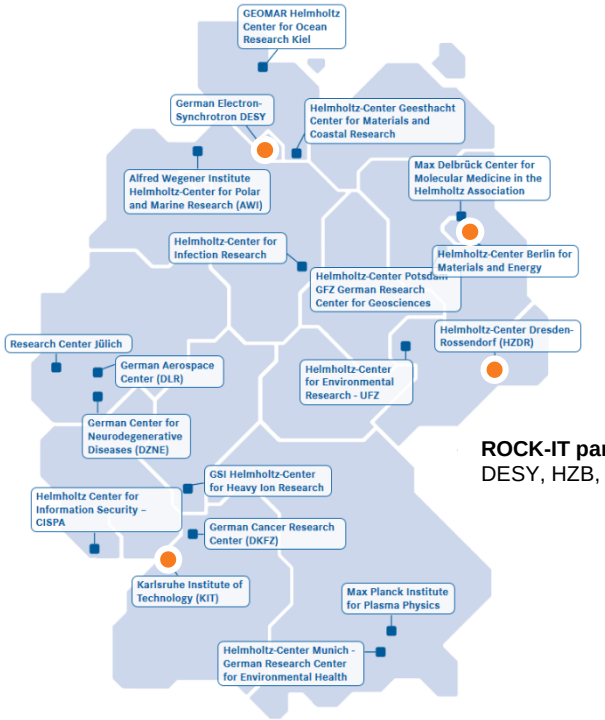
Helmholtz-funded project that aims to develop all necessary tools for the automation and remote access of in situ and operando synchrotron experiments.

Motivation:

- Creating a holistic workflow
- Increasing efficiency of usage, speeding up innovation cycles
- Attracting more industrial users
- Increasing resilience of facility operation
- Reducing CO_2 footprint of user operation via remote access.

Aims:

- A consistent look & feel between institutes
- Create a graphical user interface for experiment control & analysis
- Increased standardization of components & interfaces
- FAIR data lifecycle management
- Advance cyber security to safeguard data and infrastructure.



ROCK-IT partners:
DESY, HZB, HZDR, KIT

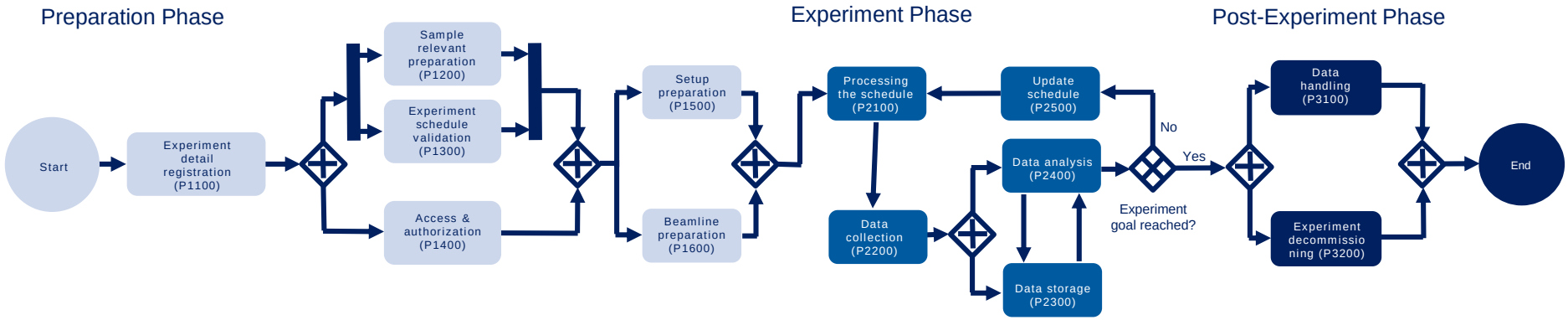
Helmholtz Centers map image source:
https://upload.wikimedia.org/wikipedia/commons/3/3a/Map_of_19_member_centers_of_the_Helmholtz_association.png

WORKFLOW FOR MAIL-IN REMOTE-ACCESS OPERANDO EXPERIMENTS

- Catalysis operando experiments - P65 @ PETRA III (DESY), mySpot @ BESSY II (HZB), CATACT @ KARA (KIT)

  Science Case: $CO_2 + 4H_2 \rightarrow CH_4 + 2H_2O$

- Implementation to the new compact Accelerator Mass Spectrometry (AMS) facility - HAMSTER @ HZDR



CURRENT STATUS OF ROCK-IT GUIS

- All partner institutes have different needs and benefits regarding ROCK-IT.
- Partner institutes are using, implementing and/or developing different user interfaces and control systems, according to their needs.
- Possible solution: Web-based GUI for general users to enter the system, for a consistent look & feel between institutes.

DESY

HZB

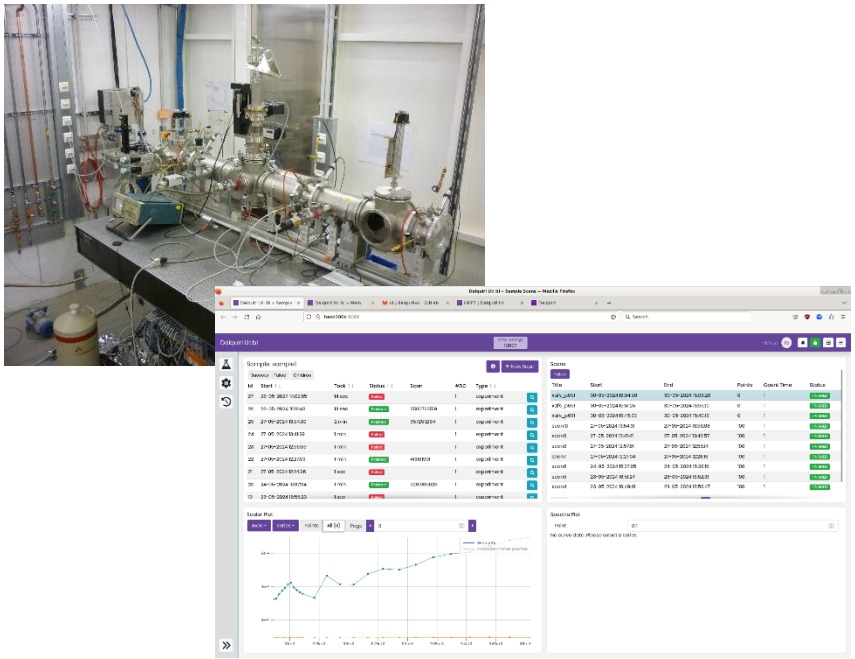
HZDR

KIT

GUI STATUS AT ROCK-IT PARTNERS: DESY

P65 | Applied X-ray Absorption Spectroscopy Beamline

Contact Person: Udai Singh udai.singh@desy.de (at NOBUGS on September 23-24) | TALK: “Daiquiri, Blissdata, Bluesky, and Sardana” – Today @ 16:00-16:20



What technologies are you using and for which types of use cases?

DESY has connected **Bluesky** queueserver to **Blissdata** through **ZMQ_Stream**. This enables them to use ESRF tools like **FLINT** and **Daiquiri** to display data.

The **Daiquiri** user interface to control **Bluesky** queueserver. Bluesky sends data through **ZMQ_Stream** to **Blissdata**.

What issues are you facing (if any) related to these technologies?

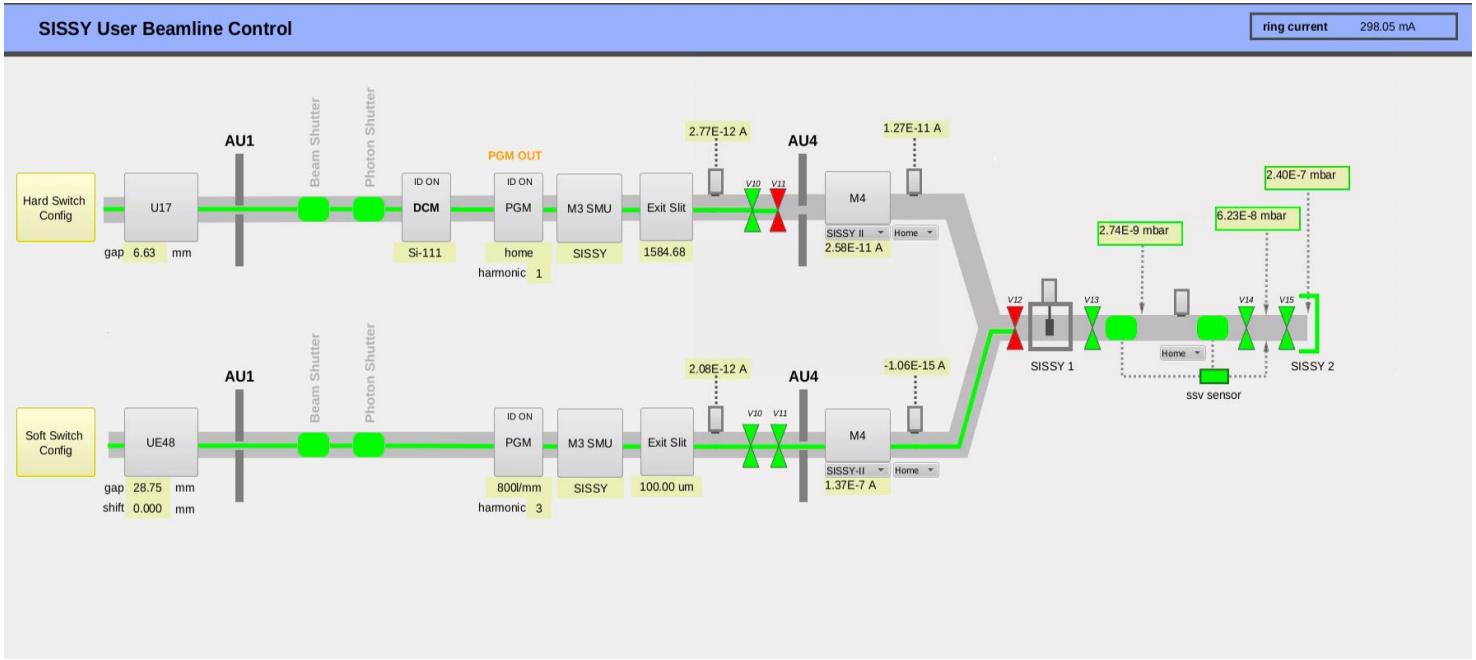
There is more work to be done for the integration of Bluesky with Daiquiri. (More details on Udai Singh’s talk “Daiquiri, Blissdata, Bluesky, and Sardana”, today @ 16:00).

GUI STATUS AT ROCK-IT PARTNERS: HZB

Contact Person: William Smith - william.smith@helmholtz-berlin.de (at NOBUGS between September 23-26)

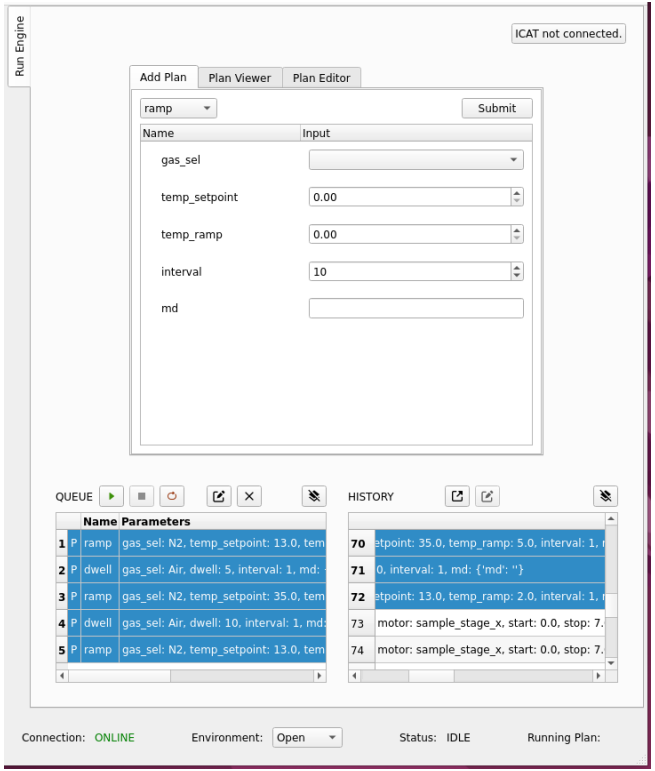
- **Interacting with EPICS devices (Synoptic Screens):**
  - **MEDM:** End of life unsupported software, Easy to produce visually unappealing displays, Extensively used in HZB and at many other labs, very reliable, XML based description of display can be converted to other tools.
  - **CS Studio Phoebus:** has big user and active developer community, Not just a display manager, also has great tools for archiver and alarms, Java -> cross platform, Possible to publish read only displays on the web, moving to using this in the accelerator and beamlines.
- **Running an Experiment:**
  - Various bespoke or vendor specific tools
  - **Igor, Metrixs, PREVAC, SpecsLab, PEAK** etc
  - **Labview** (expensive, not open source but widely used, has licensing problems)
  - Standardazing where possible on using **Bluesky**, Graphical user interfaces talk to a server, **REST API**. They have a **PyQT based GUI** for interacting with **Queueserver**. **PyQT** is easy to develop and interate in. Making use of web technologies. **Bluesky** enables platform independence and suitable for remote access. They looked at using **REACT** to make a **web front end**, as other labs make good progress with this (HZDR, ANSTO). **Daiquiri** is attractive because we can write it with **Python**.
- **Archived Data:** **EPICS** archiver service to archive the history of PV's, also using **Prometheus** to archive other parameters. They use the **CS Studios Phoebus** tools to interact with the history of PV's, **Grafana** is used to create dashboards from various data sources.
- **Interacting with experiment data:** Different user groups use different tools; i.e. **Igor, PyMca, Silx; Python, matplotlib, Matlab.**
- **GUI Strategy:** Standardize on tools; **CS Studios Phoebus** → Synoptic Overview; **Bluesky with Daiquiri**/ some **web** front end → Running Experiments

GUI STATUS AT ROCK-IT PARTNERS: HZB



Example Phoebus Beamline Control Panel

GUI STATUS AT ROCK-IT PARTNERS: HZB



PyQT GUI for interacting with Bluesky Queueserver

GUI STATUS AT ROCK-IT PARTNERS: HZDR

HAMSTER

Contact Person: Nicole Wagner - n.wagner@hzdr.de (at NOBUGS between September 24-26)

Image source: https://www.hzdr.de/db/Cms?pNid=3110&pOid=39579#&gid=1&pid=6

HAMSTER (Helmholtz Accelerator Mass Spectrometer Tracing Environmental Radionuclides) - New compact, dedicated AMS facility based on a pelletron tandem accelerator with a maximum terminal voltage of 1 MV **(IN PROGRESS).**

Two layers of user interfaces at the current AMS-demonstrator:
1) (Detailed) Expert-views → EPICSQt (detailed interaction with beamline components).
2) Web-interface → React + **Next.js** framework to interact with the **BlueSky QueueServer**.
The web-interface use cases **control the AMS-experiment via Bluesky plans**, **visualizing live data (coming from Ophyd)** and **historical data (coming from Tiled)**.

Main facility expected to go into operation next month - **a lot of user-interaction will be tested** at the test stand. The web-UI is a new interface for the scientific users – functionality to be extended according to the scientists' extended use of the demonstrator.
Challenge → to **catch unknown use-cases** by the scientists, and implement them to the machine commissioning in parallel.

"I would like to get a feeling about recruitment of personnel in the UI-development sector:

1) Isn't it easier to get new personnel work on more standardized UI technologies, than in complex frameworks (like daiquiri)?

2) How about ramp up time of those new employees to come to a good performance in developing (comparing standardized UI-technologies and complex frameworks)?"

GUI STATUS AT ROCK-IT PARTNERS: KIT

CATACT Beamline

Contact Person: Anna Zimina - anna.zimina@kit.edu

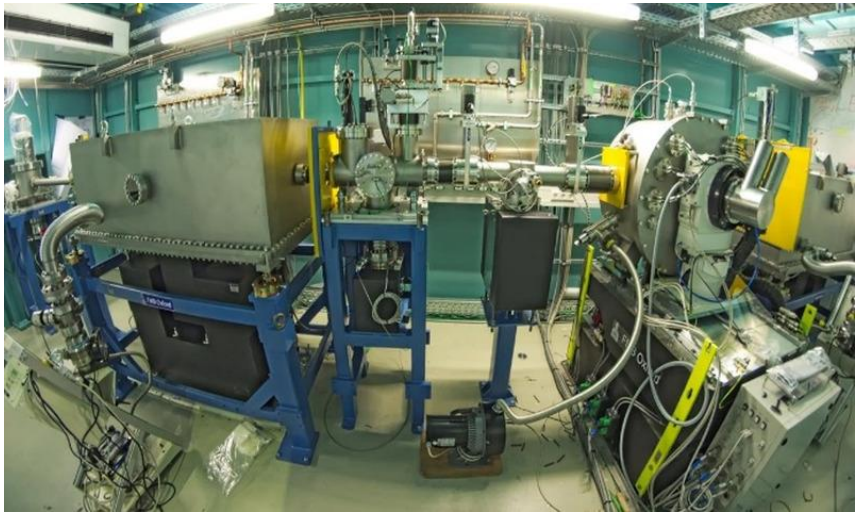


Image source: https://www.itcp.kit.edu/catact/english/67.php

CATACT - Catalytic research using X-ray spectroscopic techniques.

Doesn't use GUIs for the beamline controlling. Their SPEC is the command line based system. The beamline scientist codes the commands simultaneously. KIT's benefit from ROCK-IT: automation system.

No plans to create a GUI for operating - the **experiments change every week** and it is impossible to build a unique GUI for changing setups.

For the optics control - not found necessary. Only very experienced users can try to improve the setting of the optics in a safe manner (not often at CATACT). Every necessary optimization done together with the beamline scientist.

So far, working with **Linux** and no plans so far to find and implement a software to be used for GUIs and that can communicate with SPEC.

"I see **challenges on the experimental site**, where we have **to establish more standards how the measurements need to be done** and **how the data can be evaluated in a fast and secure manner**. The beamline and the detection procedures are already optimized**."**

GUI STATUS AT DESY: SELECTED BEAMLINES

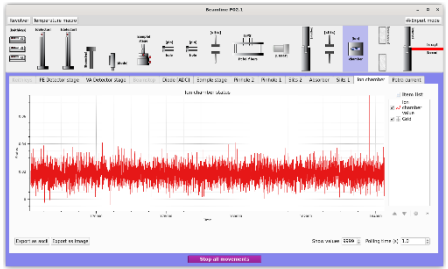
P02.1 – Powder Diffraction and Total Scattering

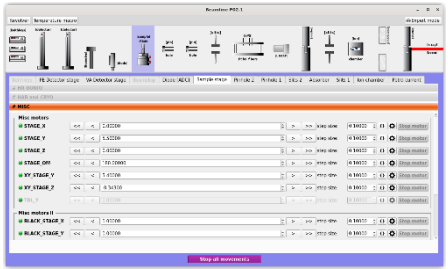
Contact Person: Alexander Schökel - alexander.schoekel@desy.de

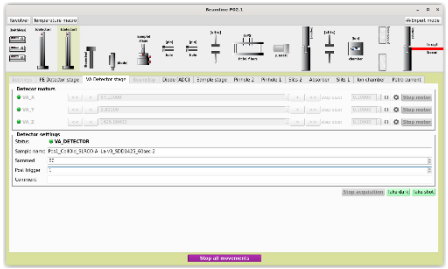
- Beamline control: Most of the standard experiments are run by the **Python**- and **PyQT**-based GUI, developed by the (former) beamline scientists. More complicated experiments are run by **Python scripts**. Detector visualization: Windows program **QXRD** software. Further processing: **PyFAI** and **DAWN**.

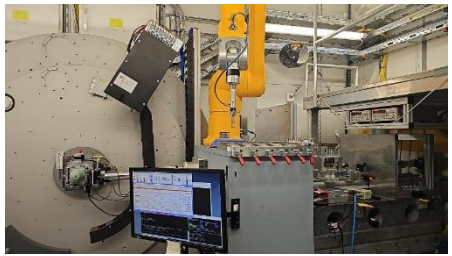
- Issues with self-written programs (Python/PyQT) are **mostly due to problems with** the **underlying libraries** and **dependencies**. Updating a library can introduce compatibility issues; this is most severe when upgrading the operating system, because this changes almost all components incl. the Python version.

- What is your **GUI strategy** going forwards (if any)? What developments are planned or are in progress? What challenges are you facing?
  "There is no explicit strategy; the codebase of our current GUI solution is not so well written (several authors, no software development background of the authors and not much time for proper development) and we are considering an overhaul to bring the code to a better shape; time for development is our major obstacle here as we don't have a dedicated programmer in our team and there is currently no central support for development, the programming is mostly done by the beamline scientists; **if we find a little time, we would do the overhaul; should we find "a lot" of time or get help from a dedicated software developer, we might consider a complete new beamline control GUI**".









GUI STATUS AT DESY: SELECTED BEAMLINES

P04 – Variable Polarization XUV Beamline

Contact Person: Frank Scholz - frank.scholz@desy.de

- Beamline completely automated, motorized.
- GUI aim: Getting everything on one screen. Software based on **Python** and **PyQt5**.
- **PyTango** – all values are in Tango servers. If server is not running well, this effects the servers, which makes the user think that the GUI is not running well. The problem is mostly in the backend.
- Communication with motors and detectors is via **Taurus** (Python GUI library), which has defined widgets inside.
- Electronic logbook – **DOOCS**, e-log accessible via web. The Distributed Object-Oriented Control System - DOOCS - provides a versatile software framework for creating accelerator-based control system applications. This is a general solution from DESY. https://doocs.desy.de/
- **Issues:** The right libraries should be installed on all PC‘s, so the versions are all the same. One may not have direct access to user libraries. Previously (until 3 years ago) they observed memory leak - memory consumption increased after 8 hours.

Thank you for your attention

More information on ROCK-IT:
rock-it-project.de

Zeynep Isil ISIK DURSUN | **DESY**
zeynep.isik.dursun@desy.de

PROJECT STRUCTURE