

Experiments orchestration and control software layer for SIRIUS accelerator and beamlines



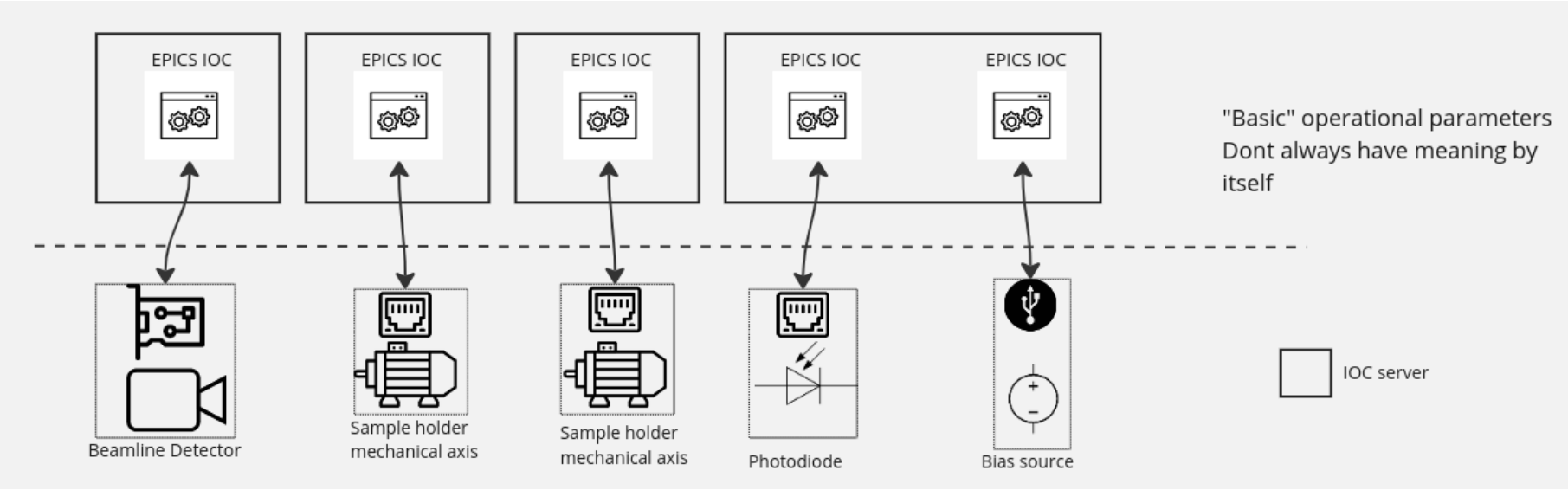
Érico N. Rolim

ICALEPCS EPICS Collab
Meeting 2023

Overview

- Basic control software architecture in Sirius
- Beyond the basic I/O: current challenges in orchestration
- Possible solutions and current implementations

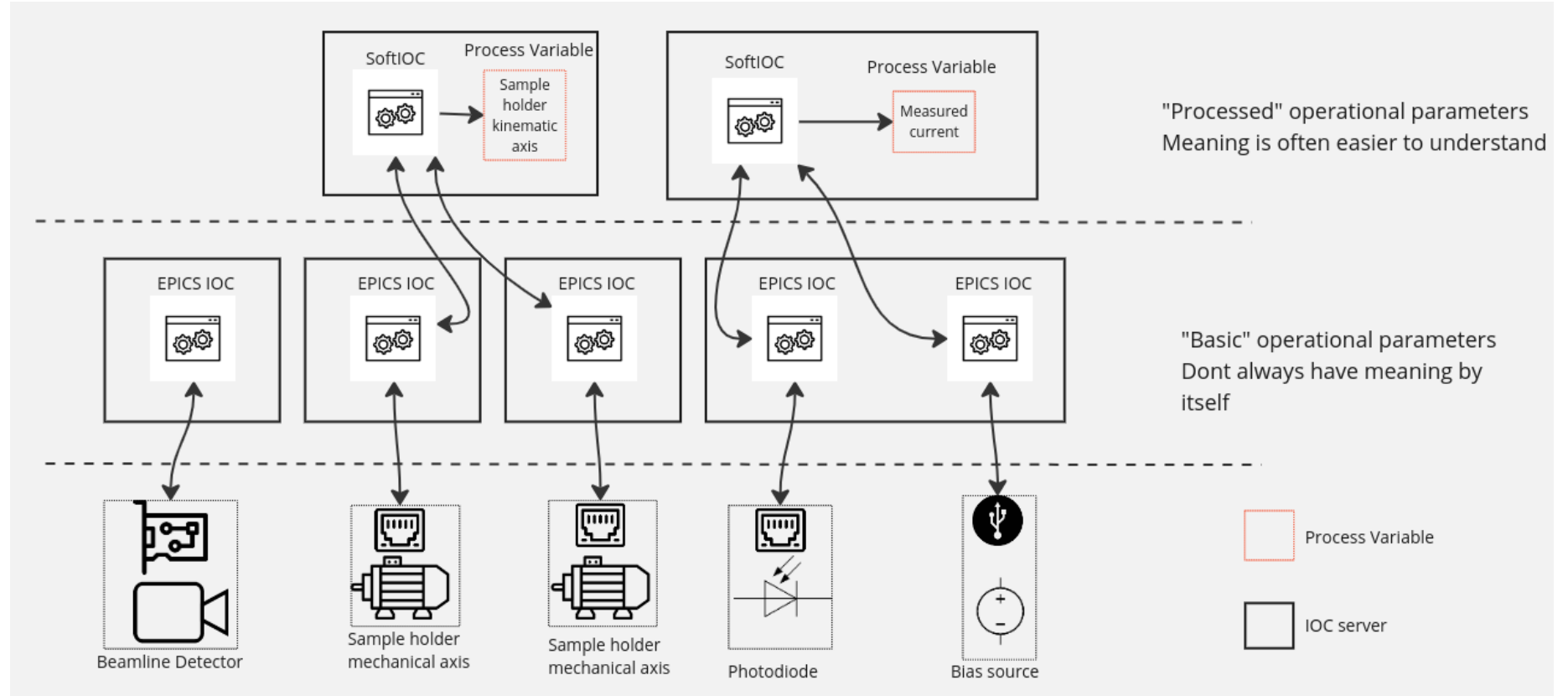
Basic control software architecture



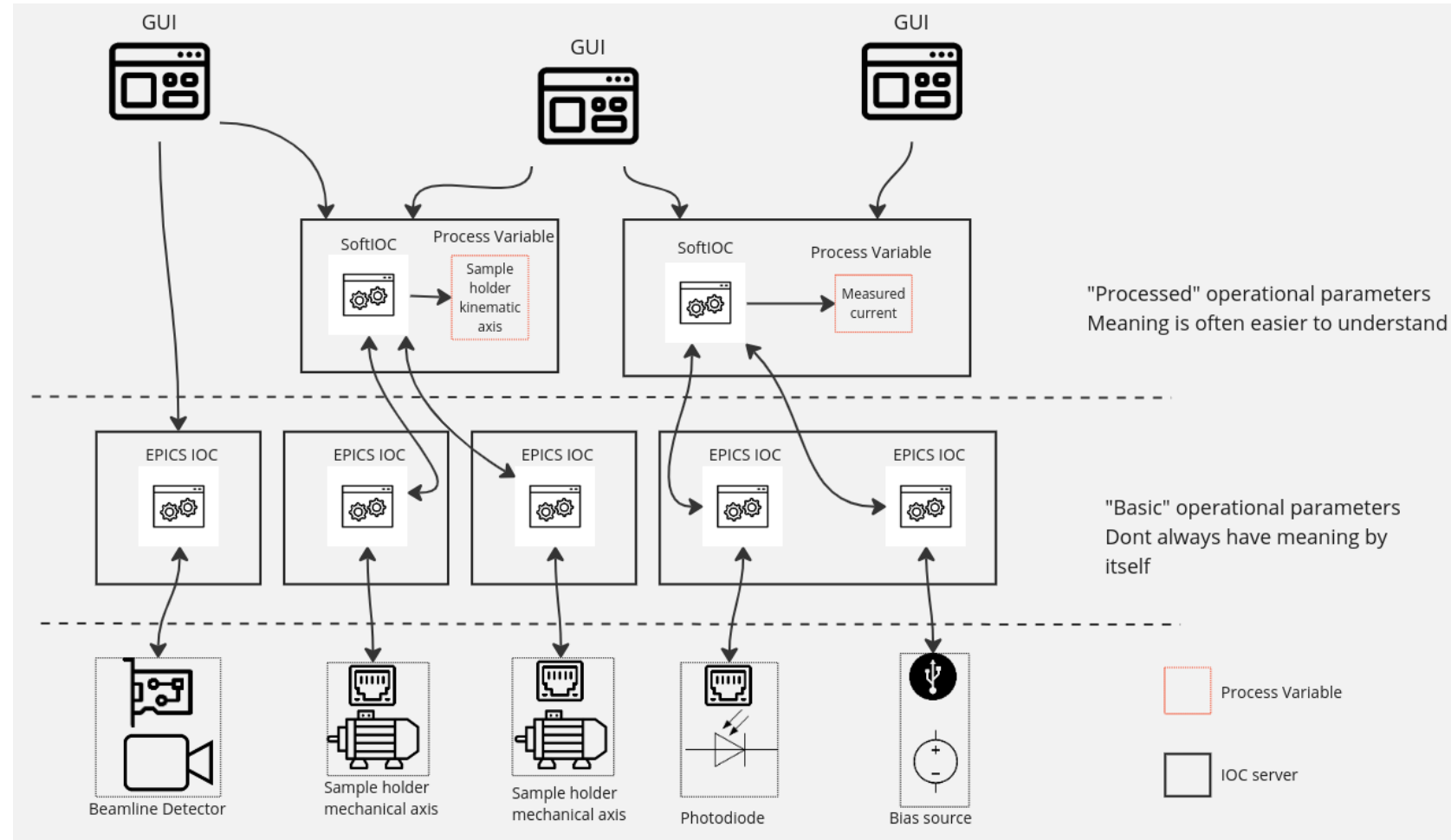
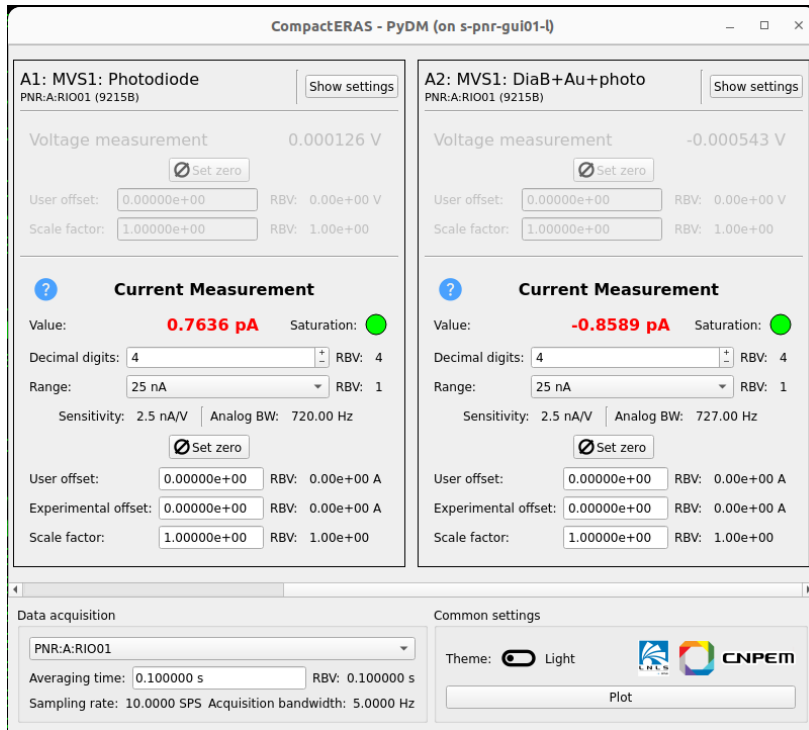
- Multiple equipment controlled by EPICS IOCs
- "Raw" parameters sometimes have little meaning to the end user

Basic control software architecture

- Software-only IOCs process basic operational parameters, extracting meaning;
- E.g.: voltage and sensitivity of photodiode are converted into current, a set of mechanical axes are converted into kinematic axes...

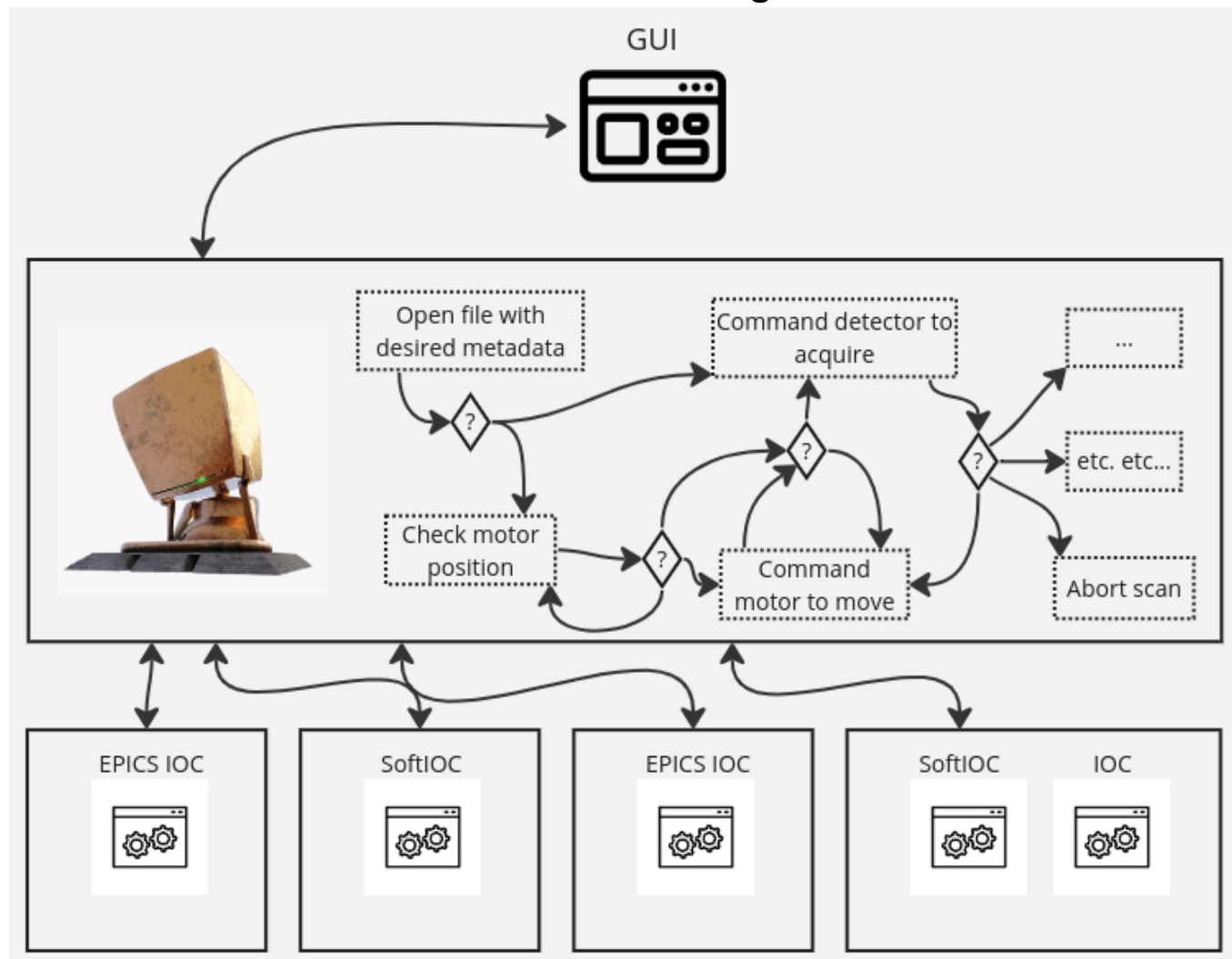


- Finally, users interact with parameters of interest via GUIs.



Current challenges in orchestration

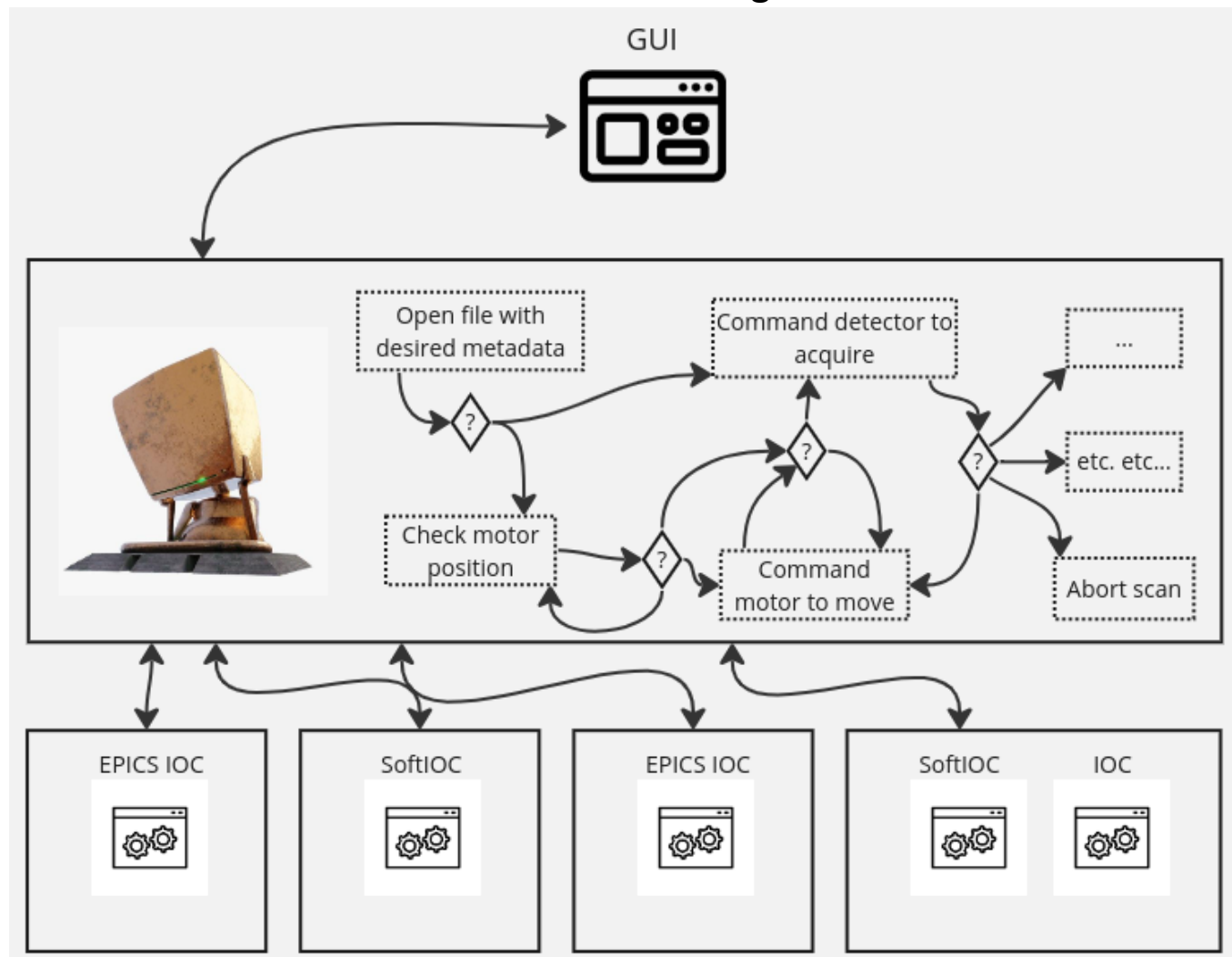
We need something to...



Current challenges in orchestration


Organize all the "atomic" operation parameters into a single, coherent and meaningful set.

We need something to...



Current challenges in orchestration

Four options are currently being used in different scenarios:

 PyDM GUIs (quick, easy and dangerous):

- Easily customized by beamline staff;
- Small developing time;
Yes, but...
- GUI crash/close = process might be left hanging;
- Process is "stuck" to the user running the GUI.

PCASpy softIOC:

- Easily customized;
- Small development time;
- Easily managed by several users;
- Made to be constantly running;
Yes, but...
- Performance is far from ideal;
- Doesn't get CA server updates from EPICS_BASE;
- How to "port" PCASpy code to PVAccess?

Sequencer softIOC:

- Easily managed by several users;
- Made to be constantly running;
- Highly performatic;
Yes, but...
- Bigger developmenttime;
- Learning curve for non-epics developers is considerable.

Bluesky scans:

- Easily managed by several users;
- Server constantly running, clients can connect whenever they want;
Yes, but...
- Learning curve is considerable, not as much as sequencer;
- No defined GUI situation



Examples and use cases...

The screenshot displays the SI-SOFB control interface with several key components:

- SI - Horizontal Orbit:** A plot showing orbit deviation in micrometers (µm) versus BPM Position in meters (m). The y-axis ranges from -12 to 8, and the x-axis ranges from 0 to 500. Two data series, Line 1 (blue) and Line 2 (red), show oscillatory behavior.
- SI - Vertical Orbit:** A plot showing orbit deviation in micrometers (µm) versus BPM Position in meters (m). The y-axis ranges from -10 to 10, and the x-axis ranges from 0 to 500. Two data series, Line 1 (red) and Line 2 (orange), show oscillatory behavior.
- Orbit Registers:** A list of seven registers, all currently empty.
- IOC Control Panel:** Contains control parameters for Orbit, Correction, and Matrix. The Orbit section includes SOFB Mode (SlowOrb), RefOrb (Out of Date), Num. Pts. (20 / 20), and Smooth Method (Average). The Correction section includes Auto Correction State (checked) and a Loop Performance graph. The Matrix section includes Sel. (Full) and use RF (checked).
- Annotations:** A red box highlights the Orbit and Correction sections, with an arrow pointing to the text "Pcaspy SoftIOC to correct orbit". A black box highlights the Loop Performance graph, with an arrow pointing to the text "Sequencer SoftIOC to take an orbit snapshot".



Examples and use cases...

Sequencer SoftIOc to identify and transition between desired and current slits states

The screenshot displays the 'Slits - PyDM' control interface. At the top, there are logos for LNLs and CNPEM, a 'Beam' status indicator with x-y axes, a red 'Stop All' button, and a 'Light' toggle switch. The main area is titled 'Single motors' and contains four motor control panels: 'motor 12' (50.00000 steps, 50.00000 degrees), 'motor 10' (40.00000 steps, 40.00000 degrees), 'motor 9' (100.00000 steps, 100.00000 degrees), and 'motor 11' (15.00000 steps, 15.00000 degrees). Each panel includes a 'Step' input field, a 'STOP' button, and a numerical display for degrees. A central black square with a white rectangle is also visible. A watermark 'developing' is overlaid on the interface.

Possible solutions

Bluesky vs. SoftIOC

- Any specific criterion for choosing one vs. another?
- Is the states machine concept well embedded into bluesky?

Sequencer: PVAccess support?

- What is the future of sequencer with PVAccess?
- Currently, sequencer has many tools to directly work with C.A. in .stt files. Will this also be true for P.V.A?

Legacy PCASpy SoftIOCs:

- We need to port old PCASpy code to more modern solutions. What are our options?
- Possibly: Python SoftIOC, PVAPy, any more...?

In conclusion

- We seek solutions for experiments orchestration and old softIOCs code portability for P.V.A. Both demands seem to ask for similar solutions;
- Several attractive tools are available. Which are the ones most aligned with the community goals/perspectives?

Thank you!

These discussions have been in happening in SIRIUS's Data Acquisition and Processing Division (DAP), Control Software Group (SwC), and in the Accelerator Physics Group (FAC).

Daniel De Oliveira Tavares – Head of Data Acquisition and Processing Division

Eduardo Pereira Coelho – Leader of Control Software Group

Ana Clara de Souza Oliveira – (Soon to be) Leader of Control Software Group

Marco Montevechi Filho – Control Software Maintainer

Érico Nogueira Rolim – Control Software Maintainer

Please get in touch!

erico.rolim@lnls.br