CONTROL OF SAMPLE ENVIRONMENT VIA SECOP WITH BLUESKY, WHILE PERFORMING MEASUREMENT PROCEDURES ON THE BEAMLINE

P. Wegmann, K. Kiefer, D. Wallacher, W. Smith, M. Bjadel, N. Grimm, S. Vadilonga, Z. Chen

Helmholtz-Zentrum Berlin, Hahn-Meitner-Platz 1, 14109 Berlin

ABSTRACT: The easy integration of sample environment hardware into Bluesky represents a crucial step for achieving the goals for automation of in-situ and operando experiments at photon sources required by the ROCK-IT project. The use of Ophyd as a common hardware abstraction layer facilitates the integration of sample environment hardware into an existing Bluesky environment. Based on the metadata and structural data provided by the Sample Environment Communication Protocol (SECoP), the SECoP-Ophyd Integration, developed by us, is capable of automatically generating and configuring Ophyd devices upon connection to a Sample Environment Control (SEC) node. A successful integration test for a catalysis experiment at BESSY II demonstrated the control of a SECoP enabled sample environment and performing measurement techniques on the beamline within the same Bluesky environment. This poster presents an introduction to the SECoP-Ophyd integration, as well as an overview of the control of the sample environment at the experiment.

SECoP-Ophyd

- Fully functional ophyd-async device construction for a given SEC node
- All metadata published by SEC Node is preserved
- Uses SECoP's self-describing mechanism to instantiate sub-devices and signals for SEC node modules and parameters.
- Complex datatypes, such as dictionaries are transported as structured numpy arrays
- Concurrent operation with ophyd.v1
- Concurrent operation with other ophyd devices backed by other control systems such as EPICS

Simple interface:

```
SECoP_Node_Device.create(
   host: str, # SEC Node IP
   port: str, # SEC Node Port
   loop        # Eventloop
)
```

Source available at:

SECoP-Ophyd
https://github.com/SampleEnvironment/secop-ophyd

Sample Environment

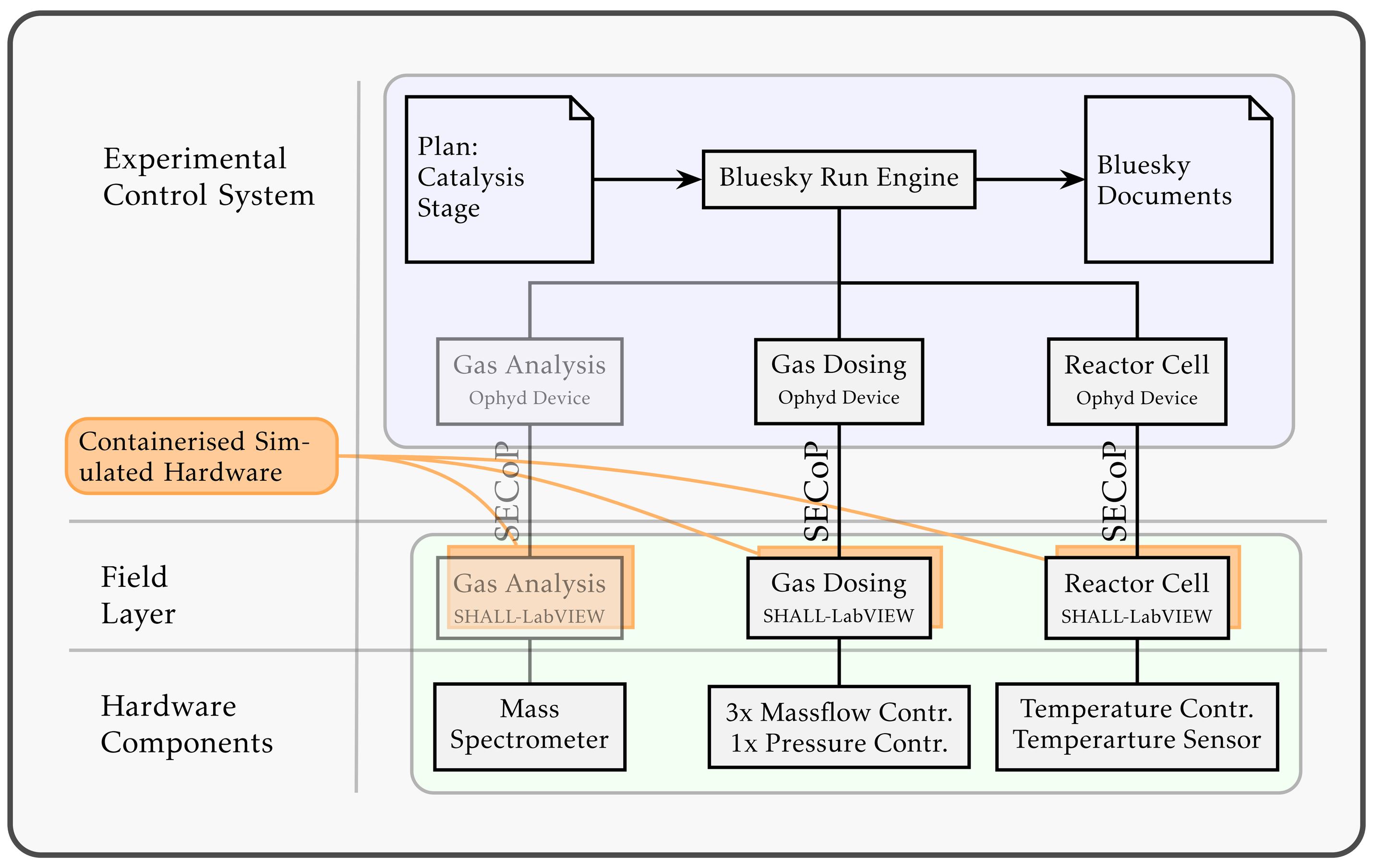

Figure 2: Overview of the sample environment

- SEC nodes initially specified in Yaml
- Integration testing, plan writing and testing of upstream software components supported by containerised sample environment simulation
- Beamline is controlled with a separate RunEngine

For more Information refer to: W. Smith: ROCK-IT Beamline and Experiment Control [Poster 60]

Conversion between Hardware abstraction Layers

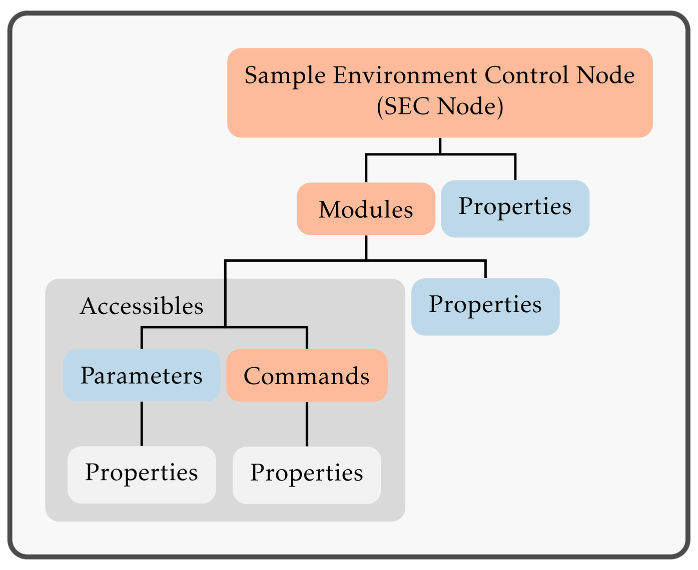

Figure 3: SEC node structure

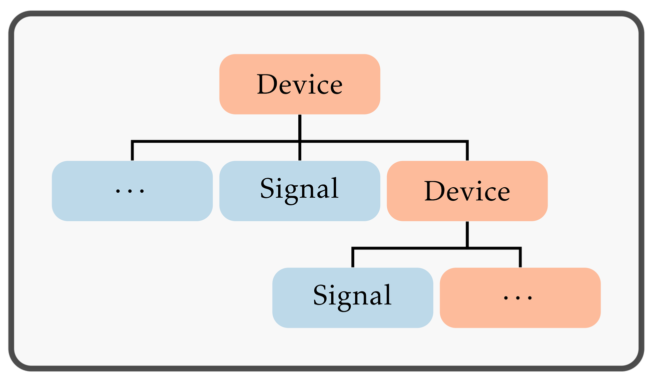

Figure 4: Ophyd device structure

SECoP-Ophyd Architecture

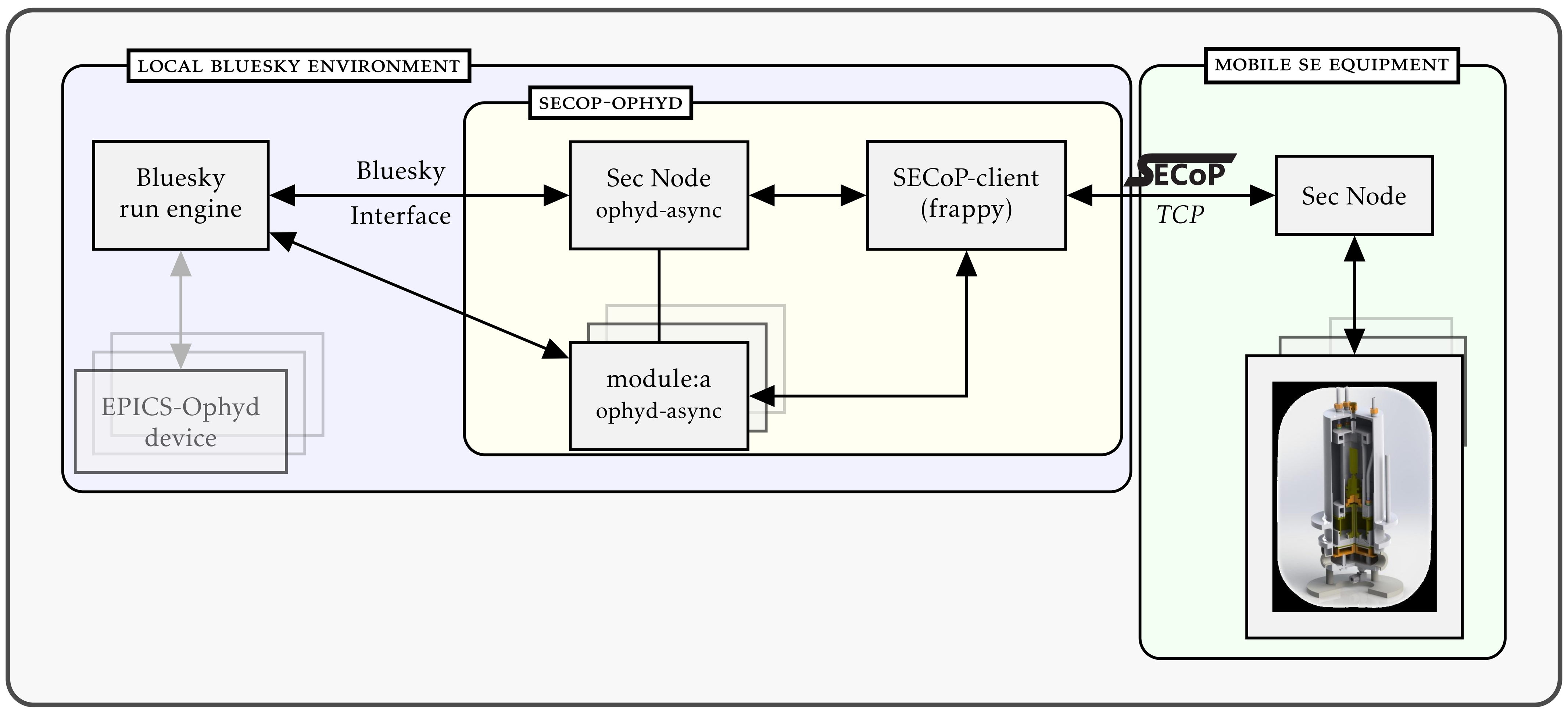

Figure 1: Architecture overview of how the SECoP-Ophyd integration interfaces with the Bluesky RunEngine and a SEC node.

SECoP-Ophyd development:

- Using the Python *asyncio* library
- Ophyd-async provides a clean and concise backend interface.
- Frappy-Client had to be wrapped to work with the *asyncio* library.

Next Steps

- Add support for detector-like devices into SECoP-Ophyd
- Finish work on automatic NEXUS file generation for data generated by SECoP-Ophyd devices according to SECoP → Nexus mapping
- Investigate instantiation of SECoP-Ophyd devices via HAPPI

INTERESTED?

- Contact the SECoP working group
- Get started with SECoP

An introduction to SECoP
[1] K. Kiefer, et al. (2020). An introduction to SECoP – the sample environment communication protocol. Journal of Neutron Research, 21(3-4), pp.181–195
https://doi.org/10.3233/jnr-190143

SECoP on GitHub
https://github.com/SampleEnvironment/SECoP

International Society for Sample Environment (ISSE)
https://sampleenvironment.org/

Helmholtz Metadata Collaboration:
https://helmholtz-metadaten.de/de

ROCK-IT Project
https://www.rock-it-project.de/

PARTNERS

MORE INFORMATION

Peter Braun (Wegmann)

peter.wegmann@helmholtz-berlin.de
Helmholtz-Zentrum Berlin für Materialien und Energie
Hahn-Meitner-Platz 1
14109 Berlin, Germany

The project SECoP@HMC is part of the HMC project cohort 2021
and receives funding by the
Initiative and Networking Fund (INF) of the Helmholtz Association