

SECoP INTEGRATION FOR THE OPHYD HARDWARE ABSTRACTION LAYER

P. Wegmann^a, K. Kiefer^a, W. Smith^a, L. W. Rossa^a, O. Mannix^a, M. Zolliker^b and E. Faulhaber^c

^aHelmholtz-Zentrum Berlin, Hahn-Meitner-Platz 1, 14109 Berlin, Germany - ^bPaul Scherrer Institut, Forschungsstrasse 111, 5232 Villigen, Switzerland - ^cForschungs-Neutronenquelle Heinz Maier-Leibnitz, Lichtenbergstraße 1, 85748 Garching, Germany

ABSTRACT: At the core of the Bluesky experimental control ecosystem the Ophyd hardware abstraction, a consistent high-level interface layer, is extremely powerful for complex device integration. It introduces the device data model to EPICS and eases integration of alien control protocols. This paper focuses on the integration of the Sample Environment Communication Protocol (SECoP) into the Ophyd layer, enabling seamless incorporation of sample environment hardware into beamline experiments at photon and neutron sources. The SECoP integration was designed to have a simple interface and provide plug-and-play functionality while preserving all metadata and structural information about the controlled hardware. Leveraging the self-describing characteristics of SECoP, automatic generation and configuration of ophyd devices is facilitated upon connecting to a Sample Environment Control (SEC) node. This work builds upon a modified SECoP-client provided by the Frappy framework, intended for programming SEC nodes with a SECoP interface. This paper presents an overview of the architecture and implementation of the SECoP-Ophyd integration and includes examples for better understanding.

MOTIVATION

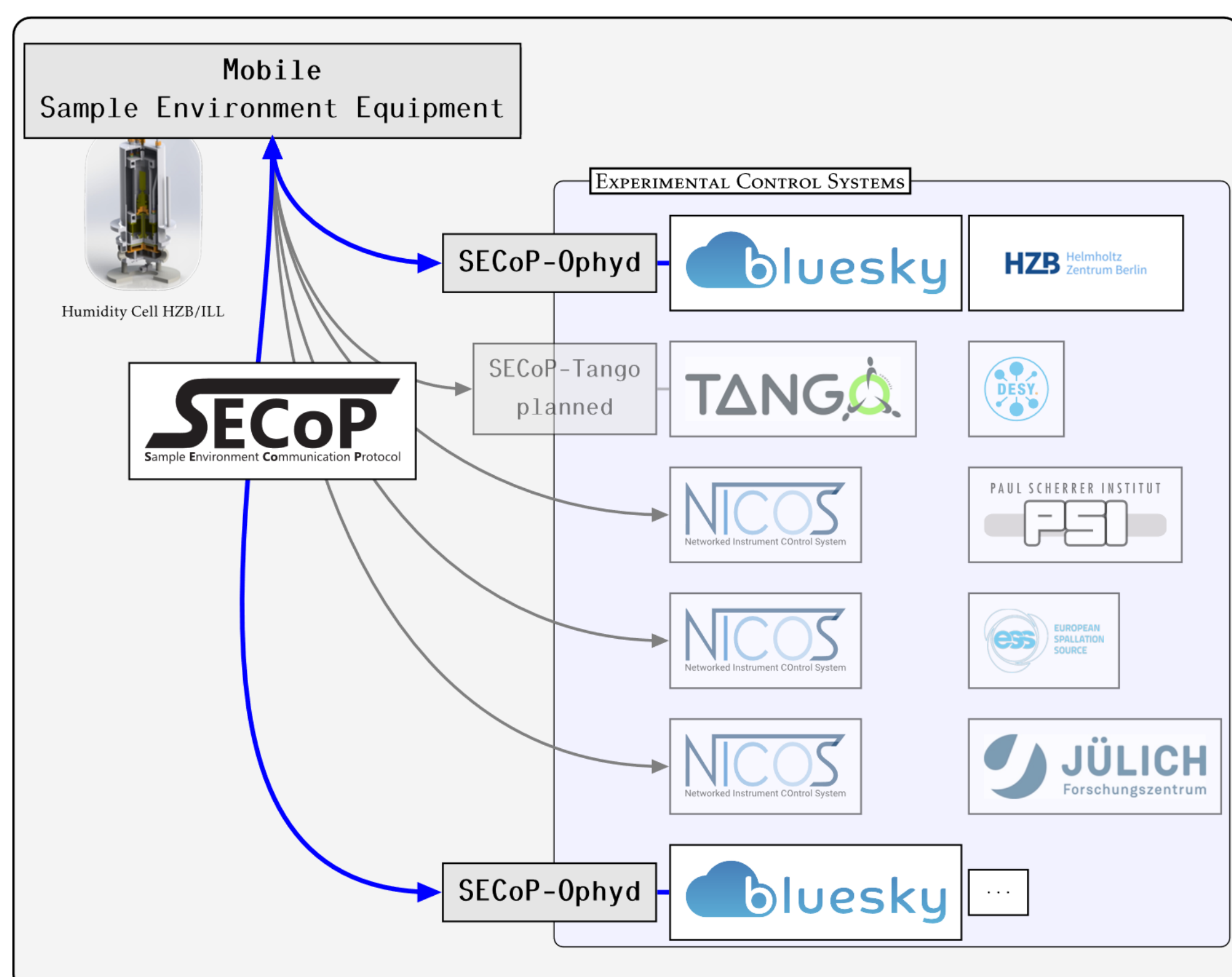


Figure 1: Moving and integrating sample environment equipment between research facilities, with differing experimental control systems

Conversion between Hardware abstraction Layers

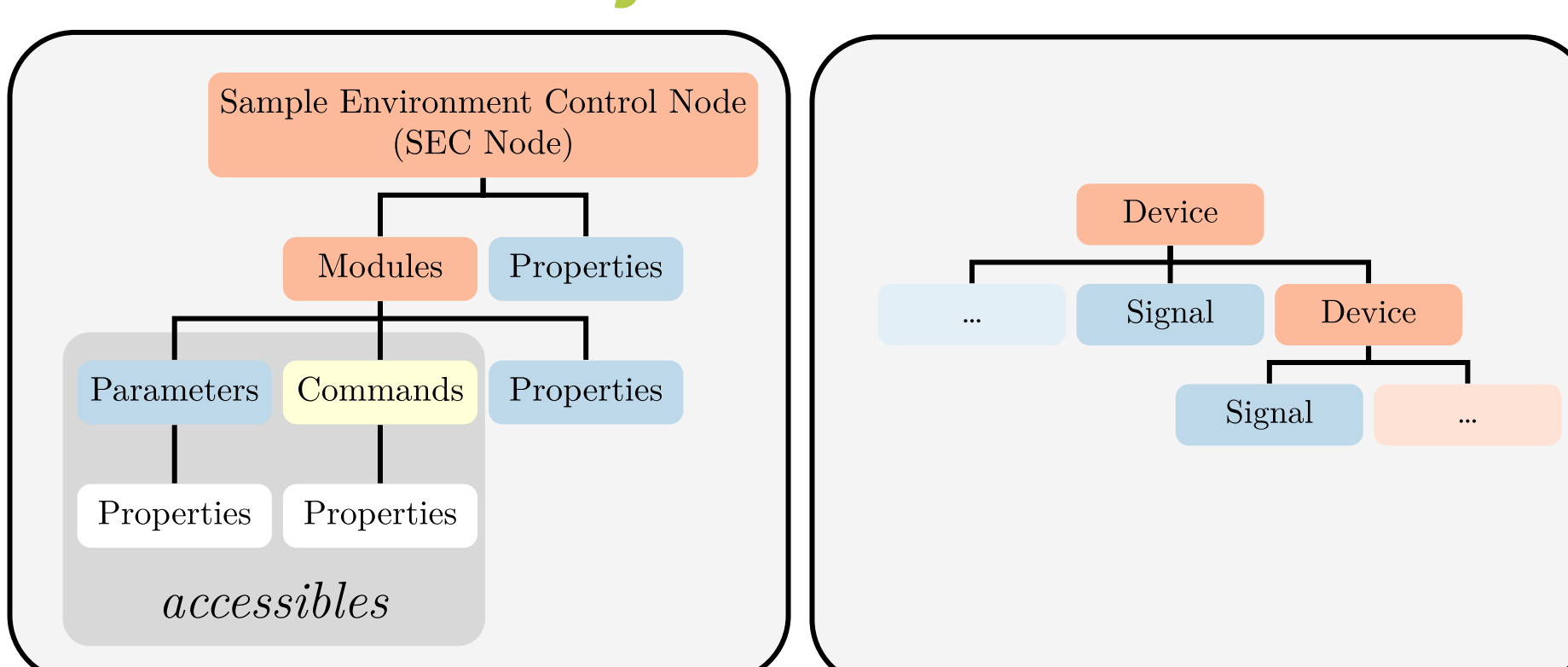
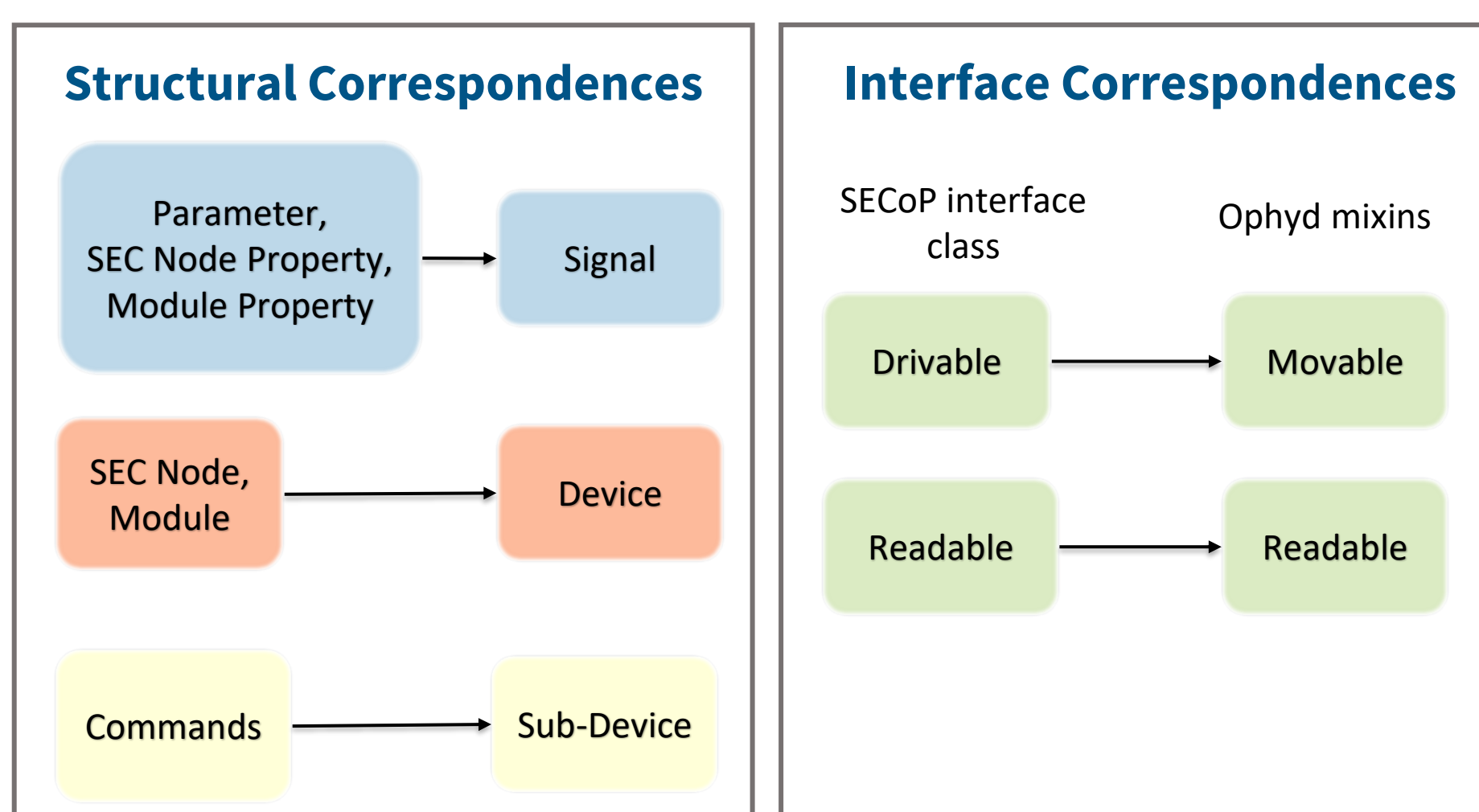


Figure 3: SEC Node structure

Figure 4: Ophyd Device structure



SECoP-Ophyd

- Fully functional ophyd.v2 Devices are generated automatically
- All metadata published by SEC Node is preserved
- Concurrent operation with ophyd.v1
- Concurrent operation with other ophyd devices backed by other control systems like EPICS

Simple interface:

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

Source available at:

<https://github.com/SampleEnvironment/secop-ophyd>

Architecture

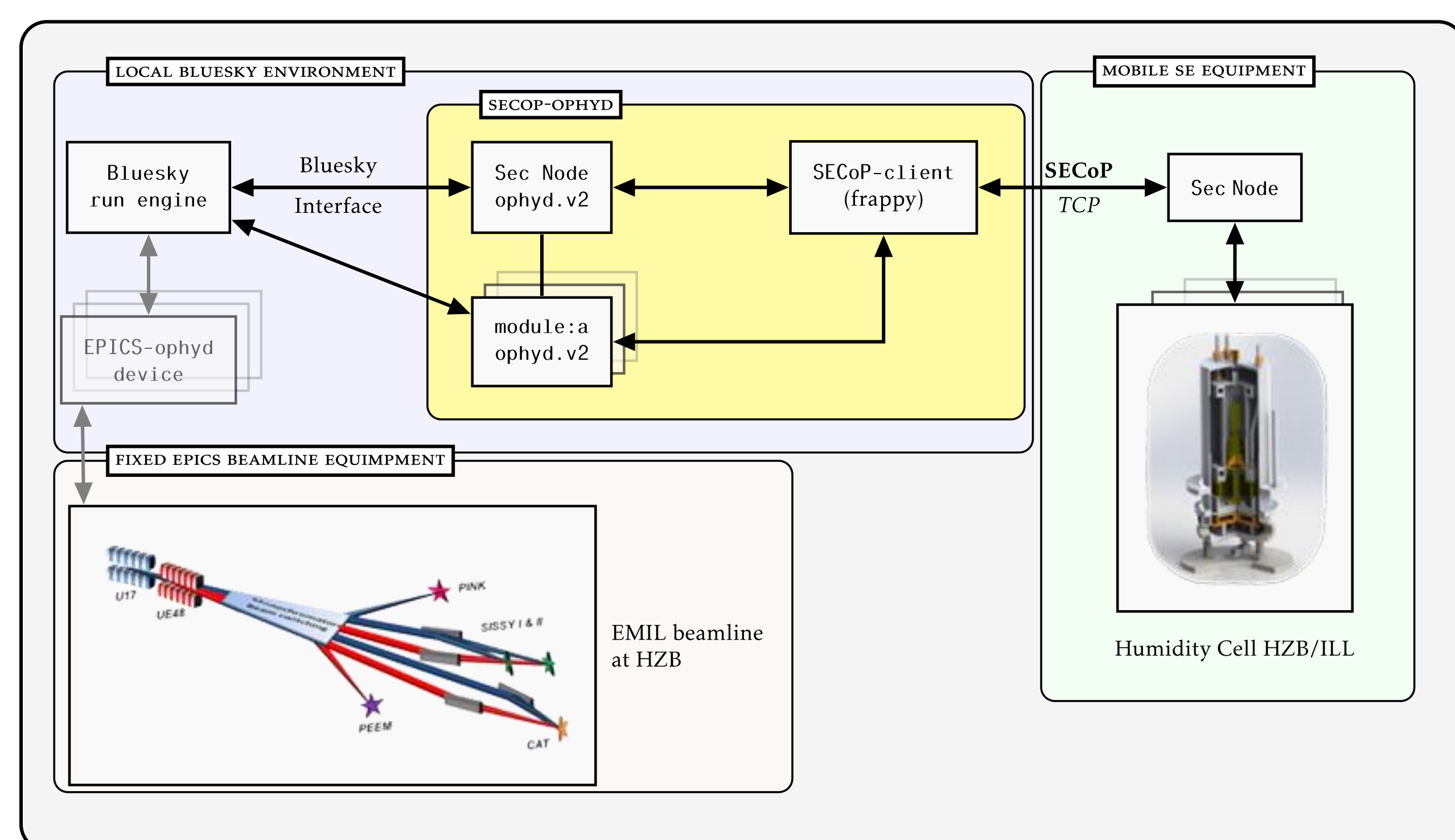


Figure 2: Architecture overview of how the SECoP-Ophyd integration interfaces with the Bluesky run engine and a SEC Node.

ophyd.v2 API

- Python *asyncio* framework
- Concise and clear interface for the backend
- Frappy-client had to be wrapped to work with *asyncio* framework
- API is not yet officially released, but reasonably stable

Next Steps

- Add further optional features supported by Bluesky/SECoP
- Investigate if Ophyd Devices generated by the SECoP-Ophyd integration are compatible with the Nexus Format
- Apply the concept to real use cases at beamlines where an existing Bluesky environment is running

INTERESTED?

- Contact the SECoP working group
- Get started with SECoP

PARTNERS



MORE INFORMATION



Peter Wegmann

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