SECoP Integration for the ophyd hardware abstraction layer
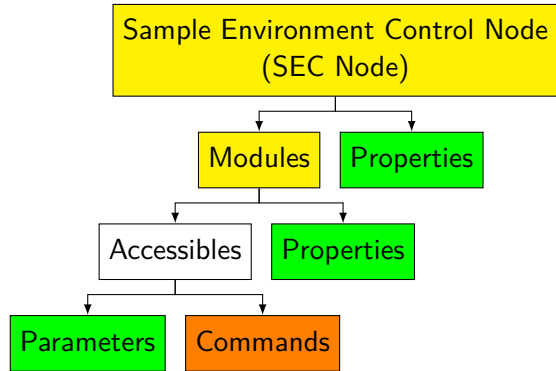
Peter Wegmann
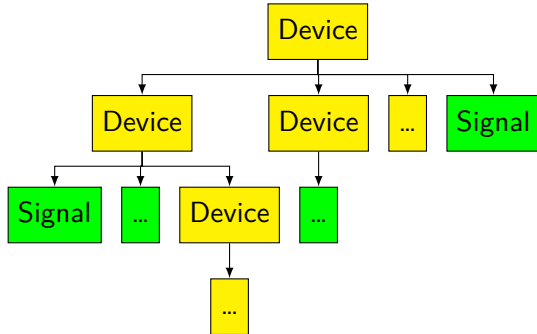
SECoP@‹HMC›

13. Juni 2023

SECoP

SECoP Messages:

- describe
- read `<module>` : `<parameter>`
- change `<module>` : `<parameter>` `<value>`
- do `<module>` : `<command>` `<value>` | null
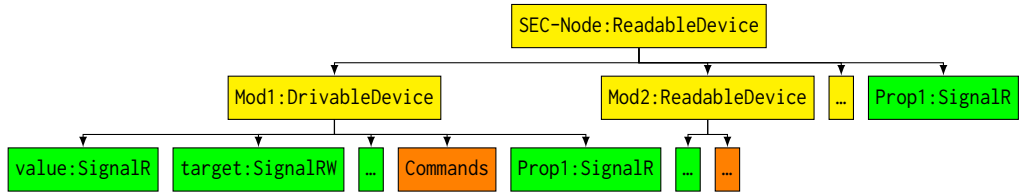
ophyd.v2 Devices:

- async read()
- async describe()
- async read_configuration()
- async describe_configuration()
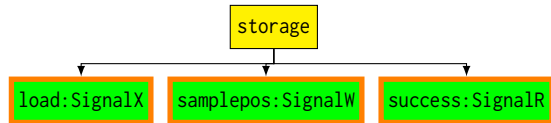- async set(target) -> Asyncstatus
- async stop()

SEC Node structure in ophyd

SECoP Command:

- Name: load

- input: samplepos:int

- return: success:bool
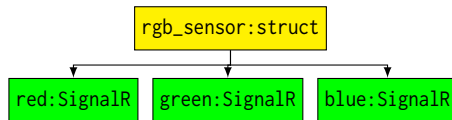
- do `storage` : `load` samplepos

SECoP Struct/Tuple-datatype

- bluesky does not support JSON objects (struct) or arrays (tuple)
- for now structs/tuples are treated as strings

SECoP Struct/Tuple-datatype

- bluesky does not support JSON objects (struct) or arrays (tuple)
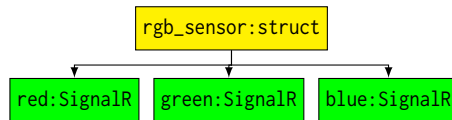- for now structs/tuples are treated as strings

Turn structs into subdevices:

```
        ┌─────────────────┐
        │ rgb_sensor:struct│
        └─────────────────┘
```

| red:SignalR | green:SignalR | blue:SignalR |

SECoP `Struct/Tuple`-datatype

- bluesky does not support JSON objects (`struct`) or arrays (`tuple`)
- for now `structs`/`tuples` are treated as strings

Turn `structs` into subdevices:

- **Arrays containing structs**

SECoP `Struct`/`Tuple`-datatype

- bluesky does not support JSON objects (`struct`) or arrays (`tuple`)
- for now `structs`/`tuples` are treated as strings
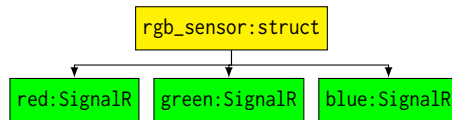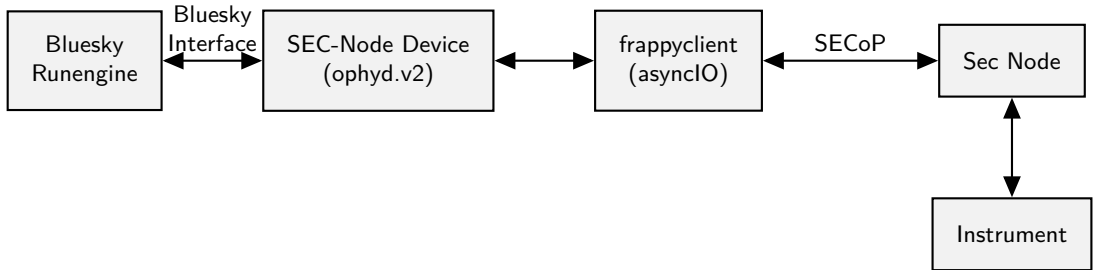
Turn `structs` into subdevices:

- **Arrays containing structs**
- **structs as module value/target**

Architecture

Status

- basic SECoP Device generation is working
- Devices are fully functional read(), describe(), set(), ...
- tired some Plans with the bluesky RunEngine
- Gitlab Continuos Integration Pipeline is set up for running tests (pytest)

Outlook

- not all SECoP features are implemented yet (commands...)
- testing, testing, testing
- Error reporting and exception handling