PAUL SCHERRER INSTITUT

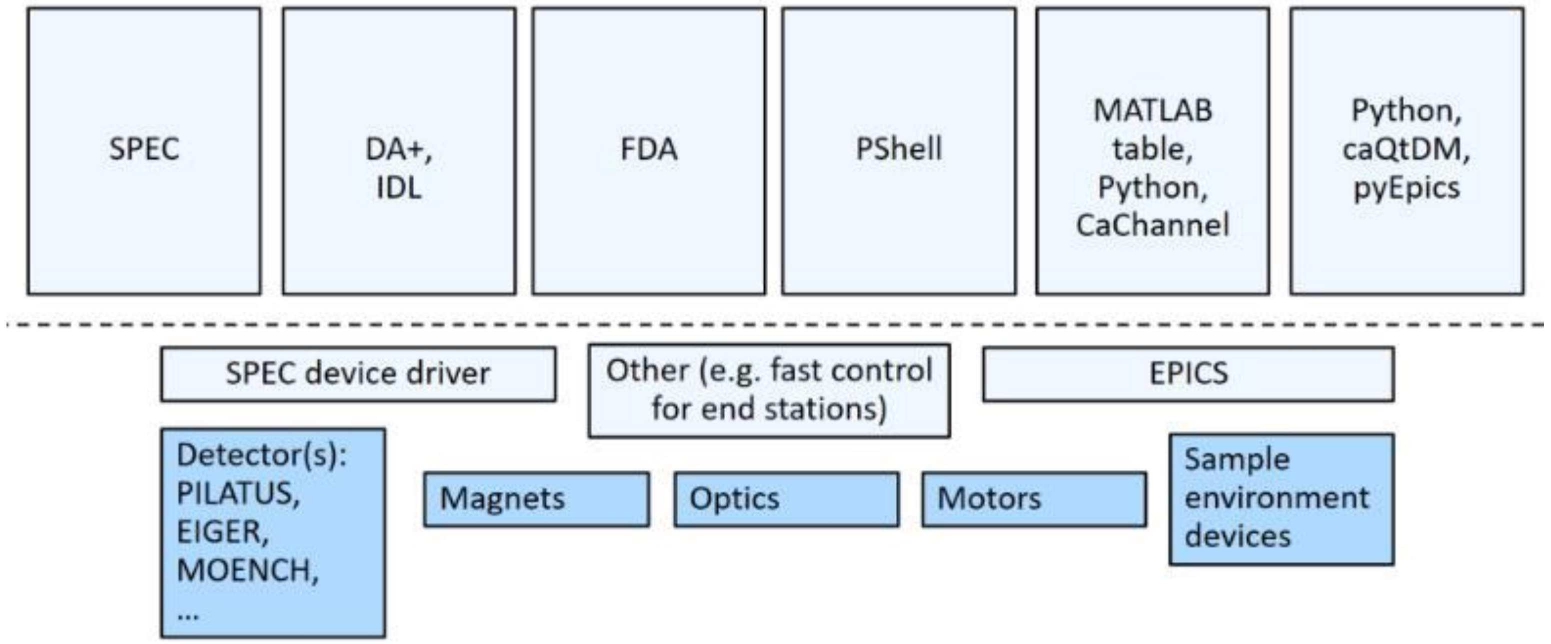
Towards a unified BEC for SLS 2.0

PAUL SCHERRER INSTITUT

What's a BEC?

Beamline (and) **E**xperiment **C**ontrol

"… the layer above the control system tasked with the orchestration of the data acquisition."


SPEC
DA+, IDL
FDA
PShell
MATLAB table, Python, CaChannel
Python, caQtDM, pyEpics
SPEC device driver
Other (e.g. fast control for end stations)
EPICS
Detector(s): PILATUS, EIGER, MOENCH, …
Magnets
Optics
Motors
Sample environment devices

Source: CaSIT CDR

PAUL SCHERRER INSTITUT PSI

A closer look

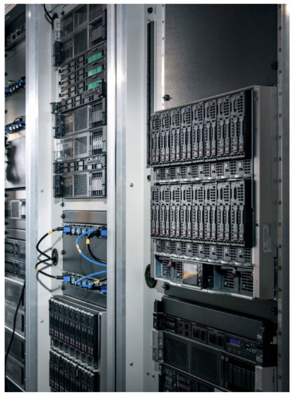
Bliss

bluesky

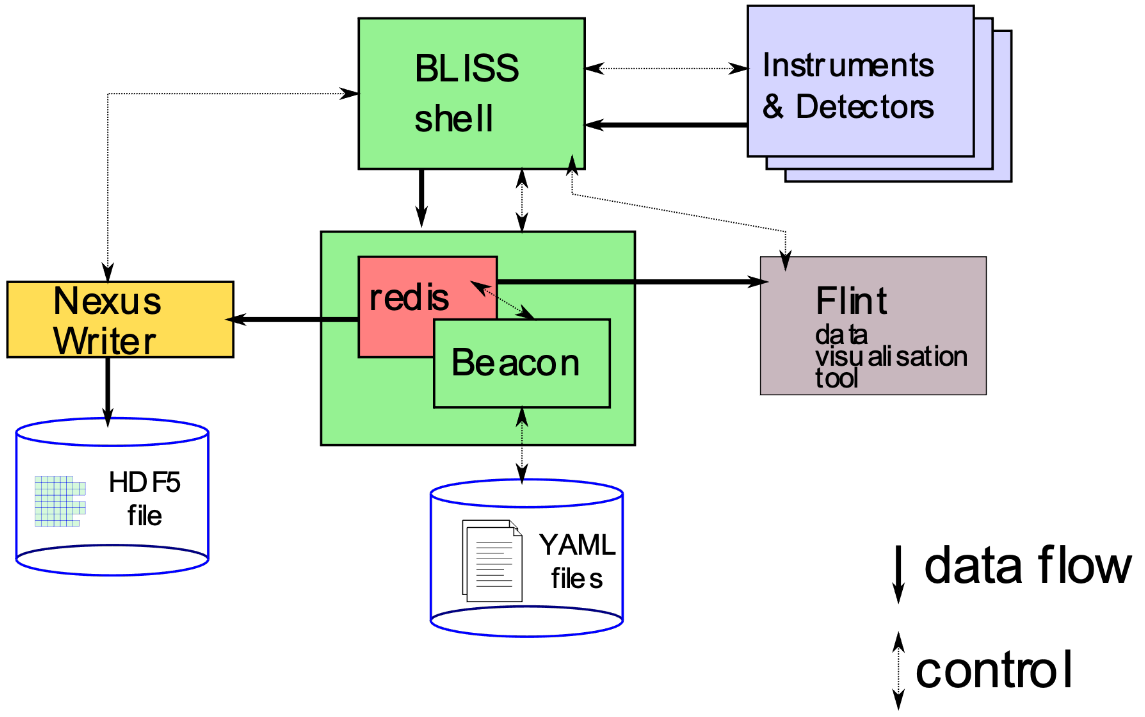
Technical evaluation criteria

- **Architecture**
- **User features and user perspective**
- **Hardware and DAQ support**
- **Stability and maintenance**

Conceptual Design Report on Controls and Science IT for the SLS 2.0 Upgrade Project



BLISS Control and data architecture

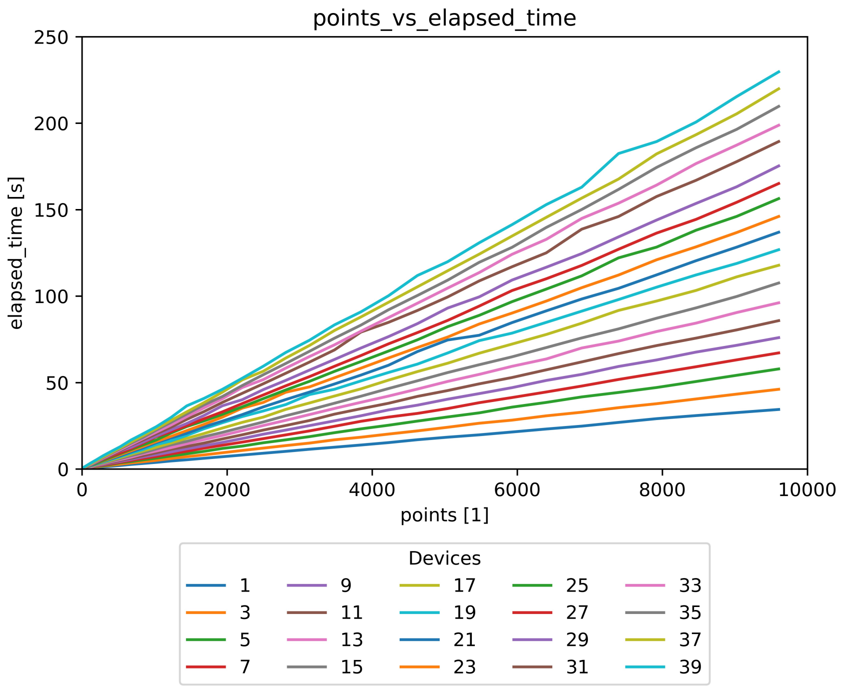


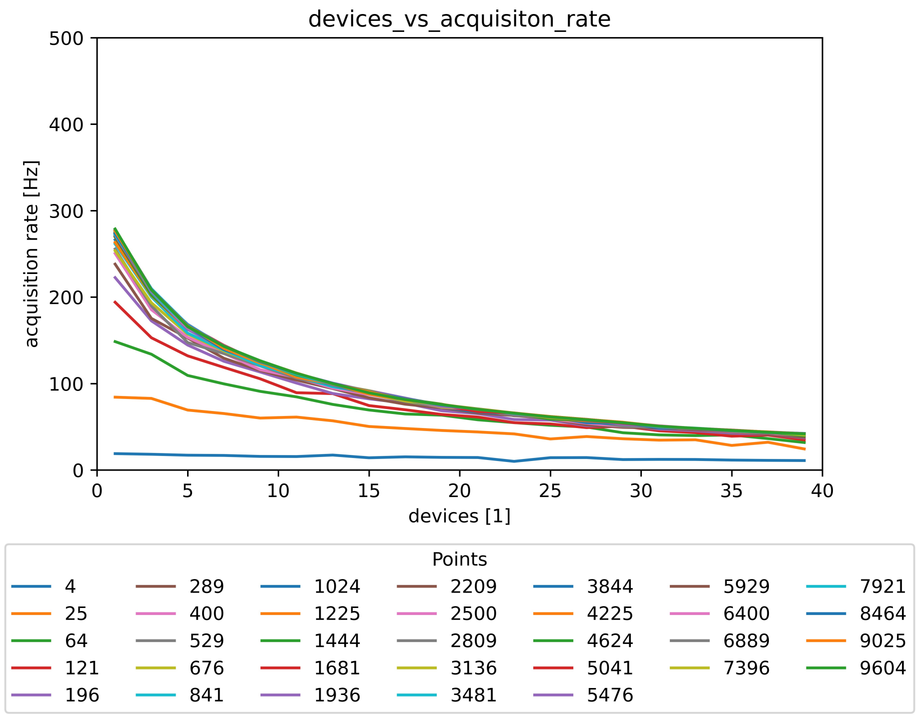
Source: https://bliss.gitlab-pages.esrf.fr/bliss/master/bliss_overview.html

BLISS performance in our test

Performance analysis using a 2D grid scan (mesh scan)

- simulated devices (single value readout)
- removed waiting times (with support from BLISS developers)



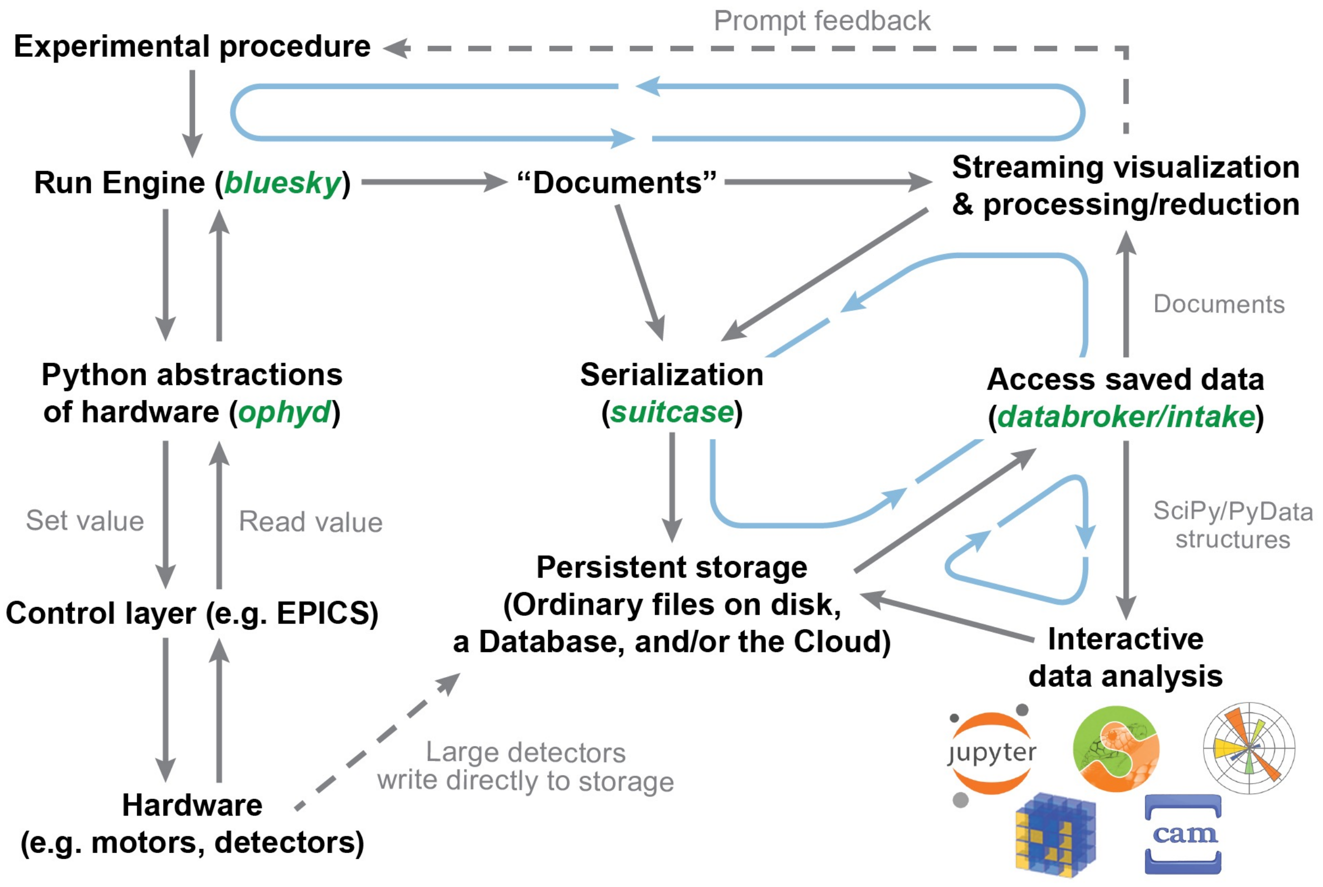


BLISS summary

- Pros:
  - config management
  - data analysis does not influence the performance of the current data acquisition (via redis subscription)
  - checks a lot of boxes on our functional requirements list (multi-user, UIs, data pipelines…)

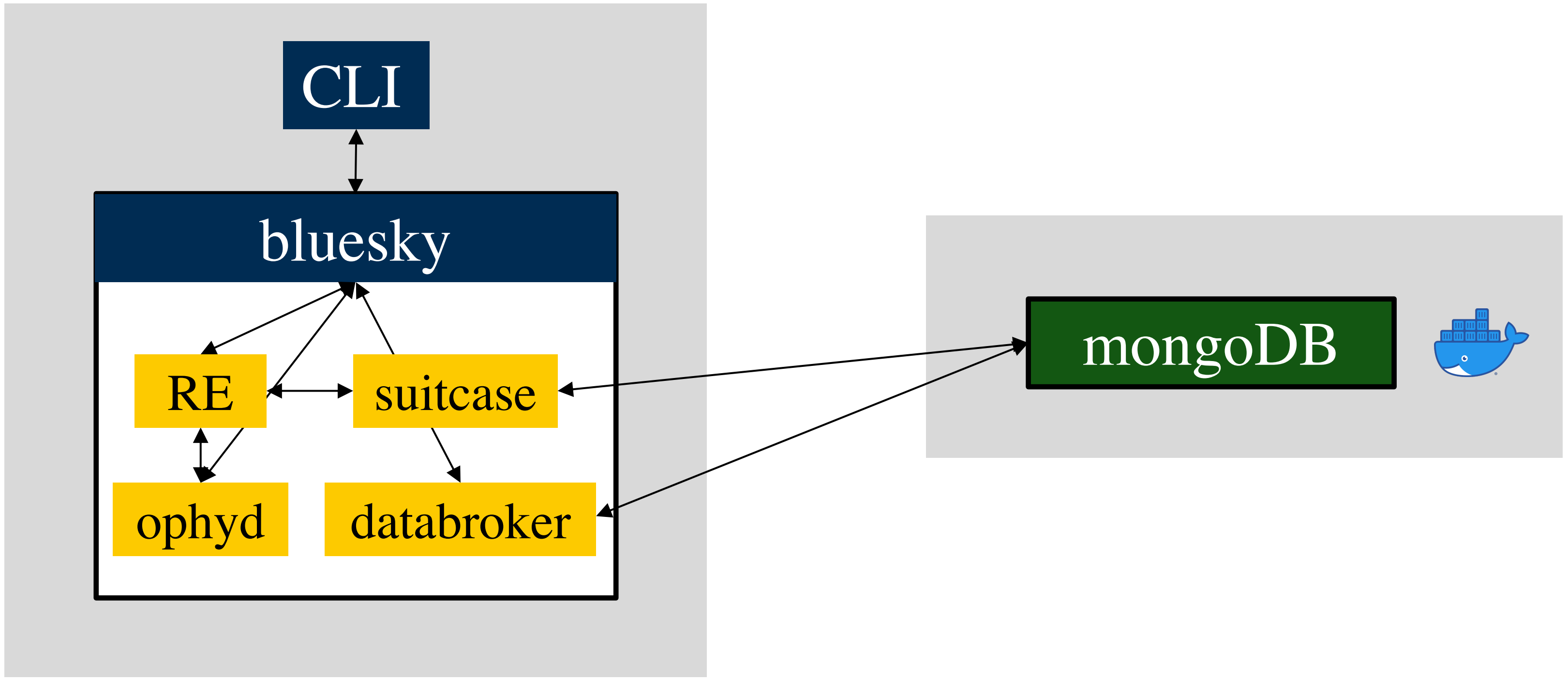
- Cons:
  - “All-in-one” package without clearly defined boundaries of individual components -> difficult to navigate through the code base
  - many dependencies
  - recommended developer onboarding ~3 months
    - unlikely that beamline scientists will be able to contribute to the development

Bluesky

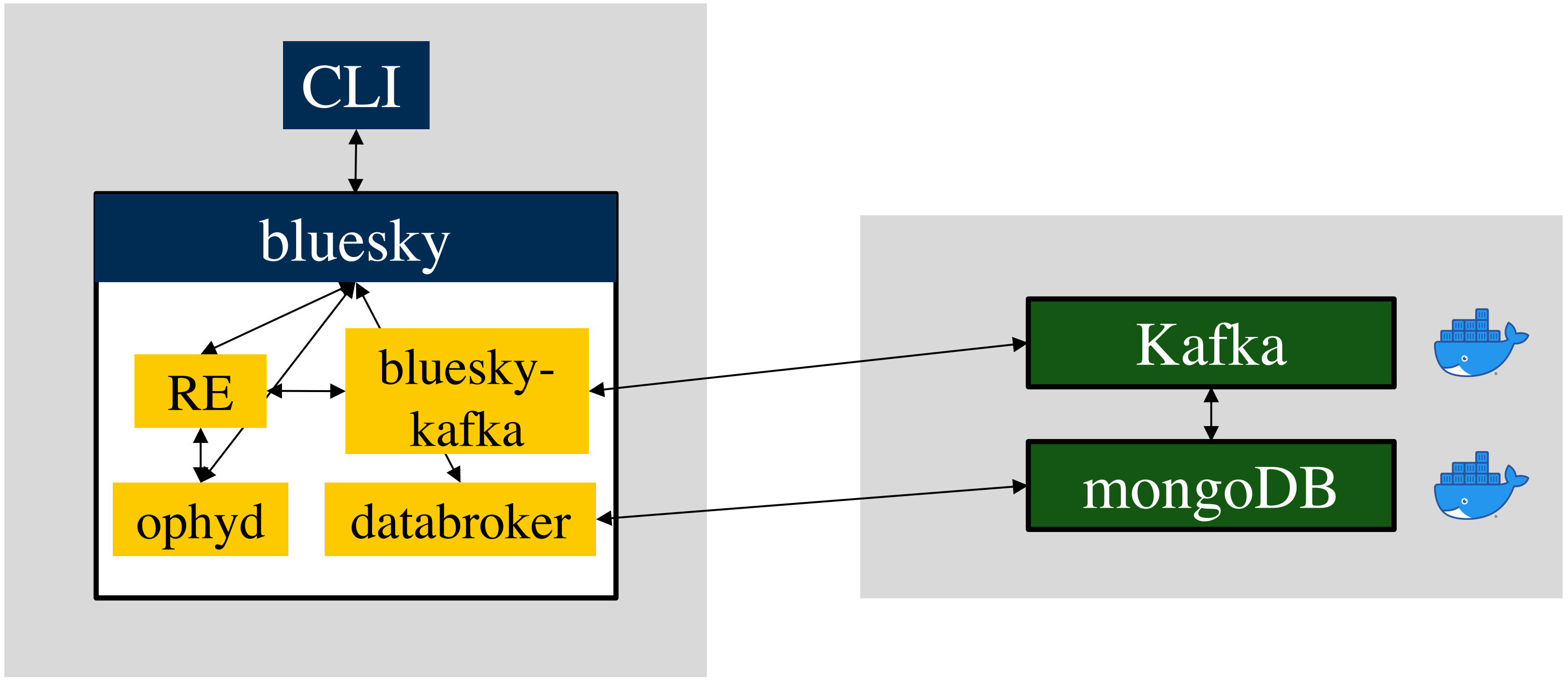

Prompt feedback
Experimental procedure
Run Engine (*bluesky*)
"Documents"
Streaming visualization & processing/reduction
Documents
Python abstractions of hardware (*ophyd*)
Serialization (*suitcase*)
Access saved data (*databroker/intake*)
Set value
Read value
SciPy/PyData structures
Control layer (e.g. EPICS)
Persistent storage (Ordinary files on disk, a Database, and/or the Cloud)
Interactive data analysis
jupyter
cam
Large detectors write directly to storage
Hardware (e.g. motors, detectors)

Source: https://blueskyproject.io

“Vanilla” bluesky



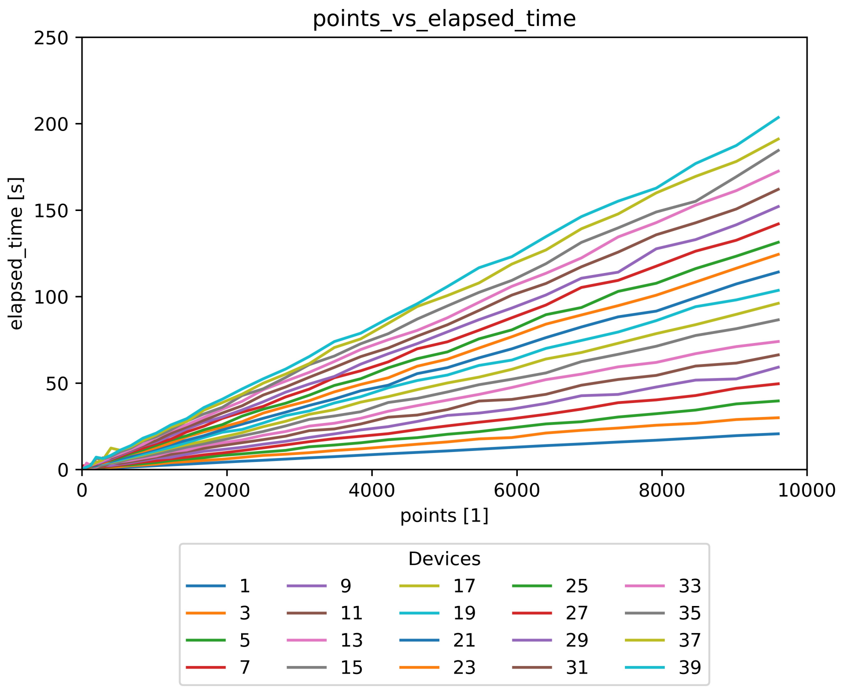
Publishing to Kafka

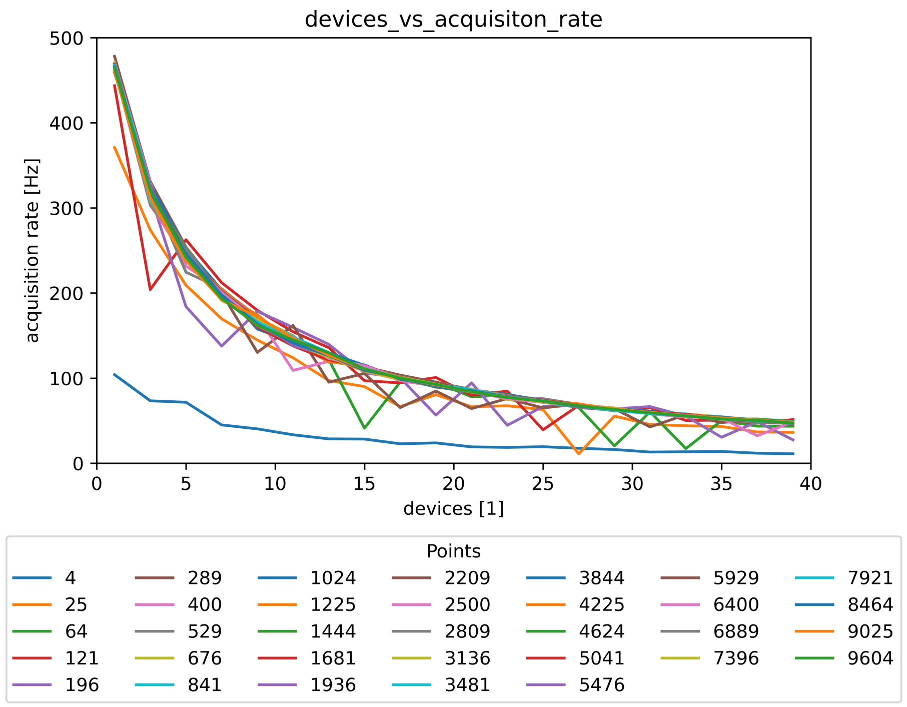


Bluesky performance in our test

Performance analysis using a 2D grid scan (mesh scan)

- simulated devices (single value readout)
- removed waiting times





Bluesky project

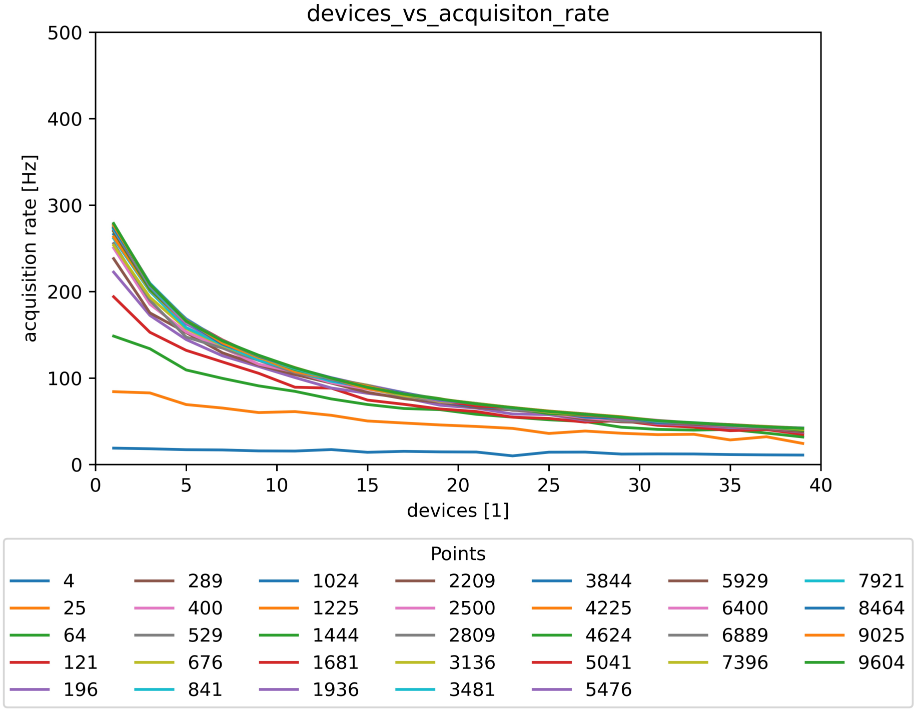
- Pros:
  - simplicity
  - modular structure (ophyd, bluesky, suitcase, databroker…)
  - very few dependencies

- Cons:
  - high memory usage
  - state objects propagate through the entire system (difficult to debug)
  - no config management
  - suggested user-management layer comes with a new CLI (and sacrifices most of bluesky's features)
  - no data management; everyone has access to everything

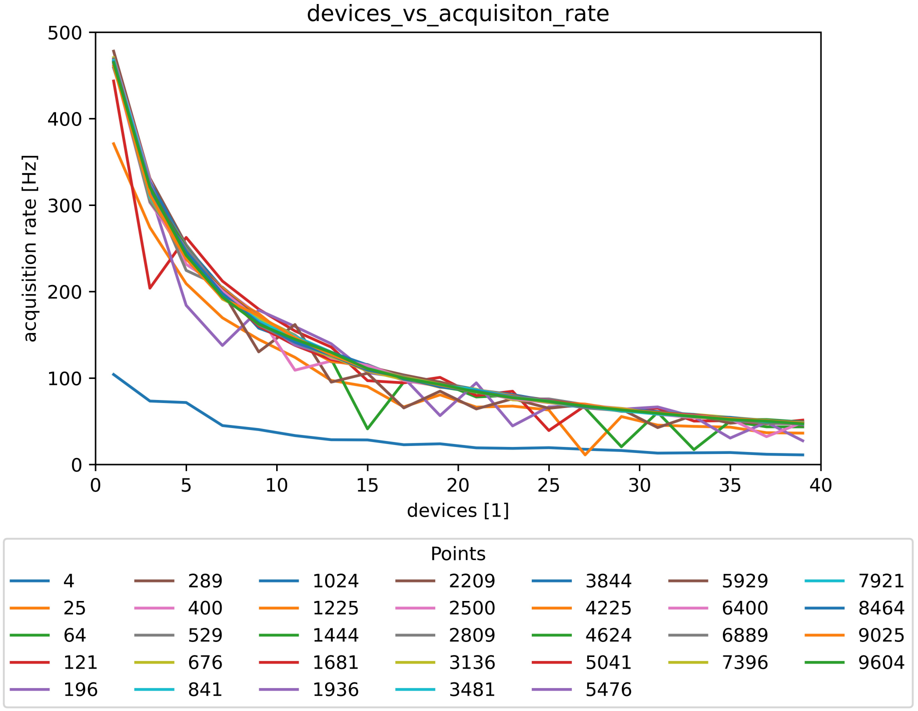
BLISS and Bluesky

BLISS / Bluesky

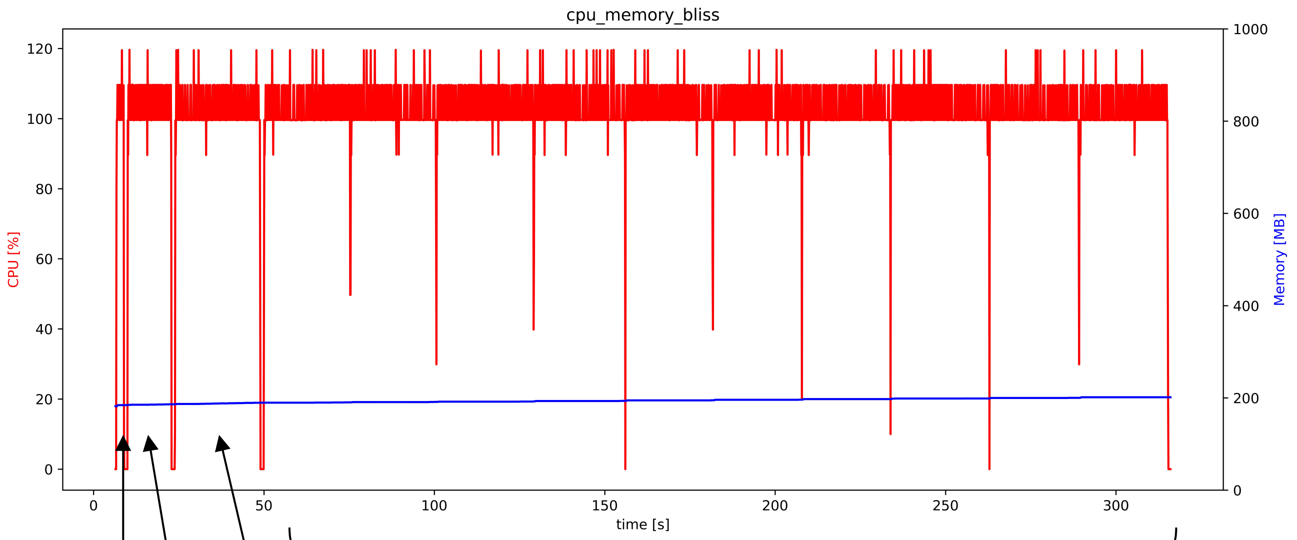
BLISS



Bluesky



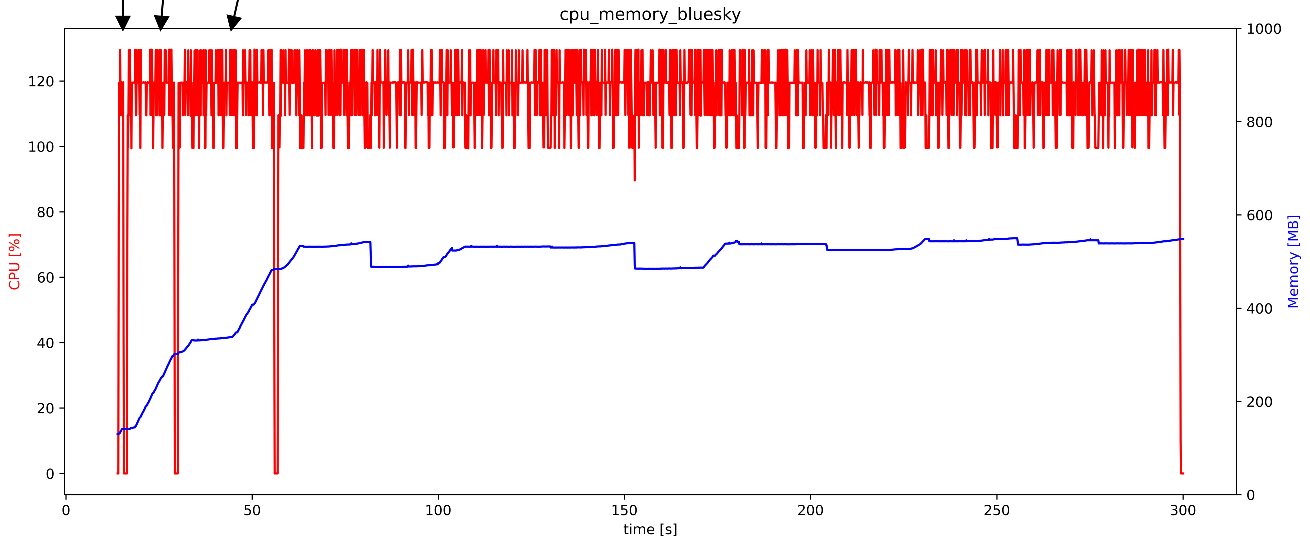
BLISS / Bluesky



BLISS

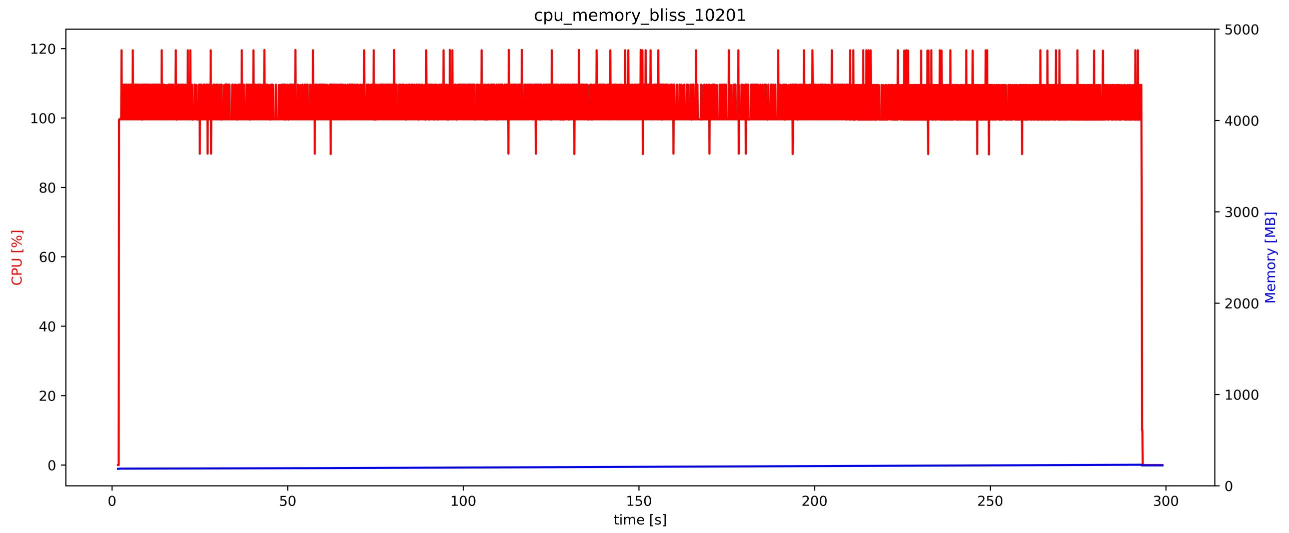
devices: 2, 50, 100

10 x grid scan, 441 points, 100 devices



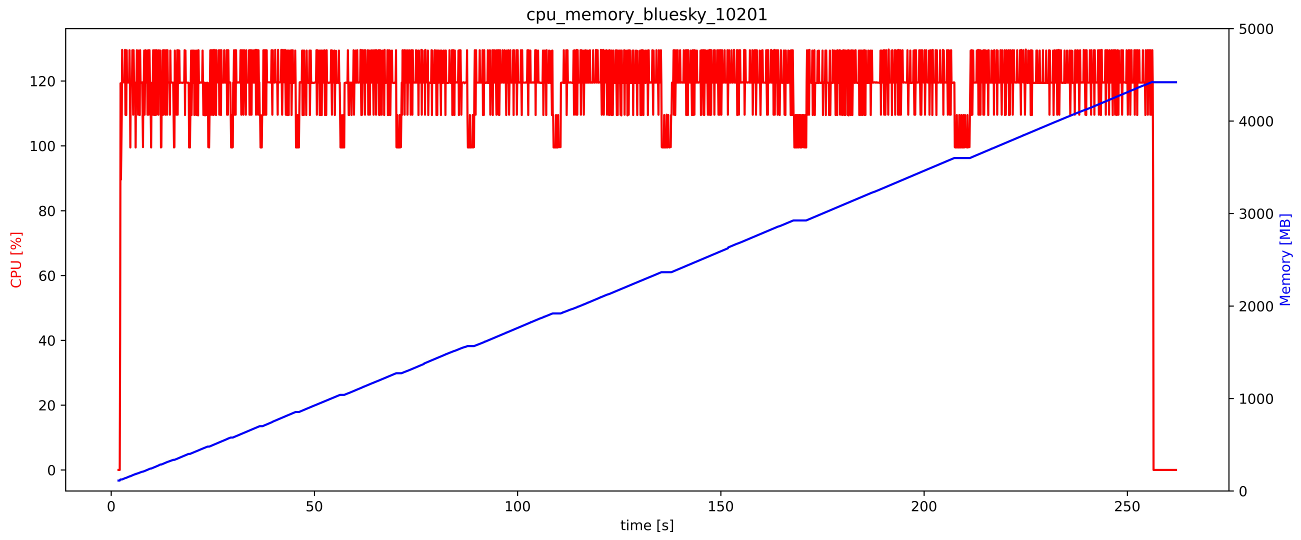
Bluesky

BLISS / Bluesky



BLISS

grid scan, 10201 points, 100 devices



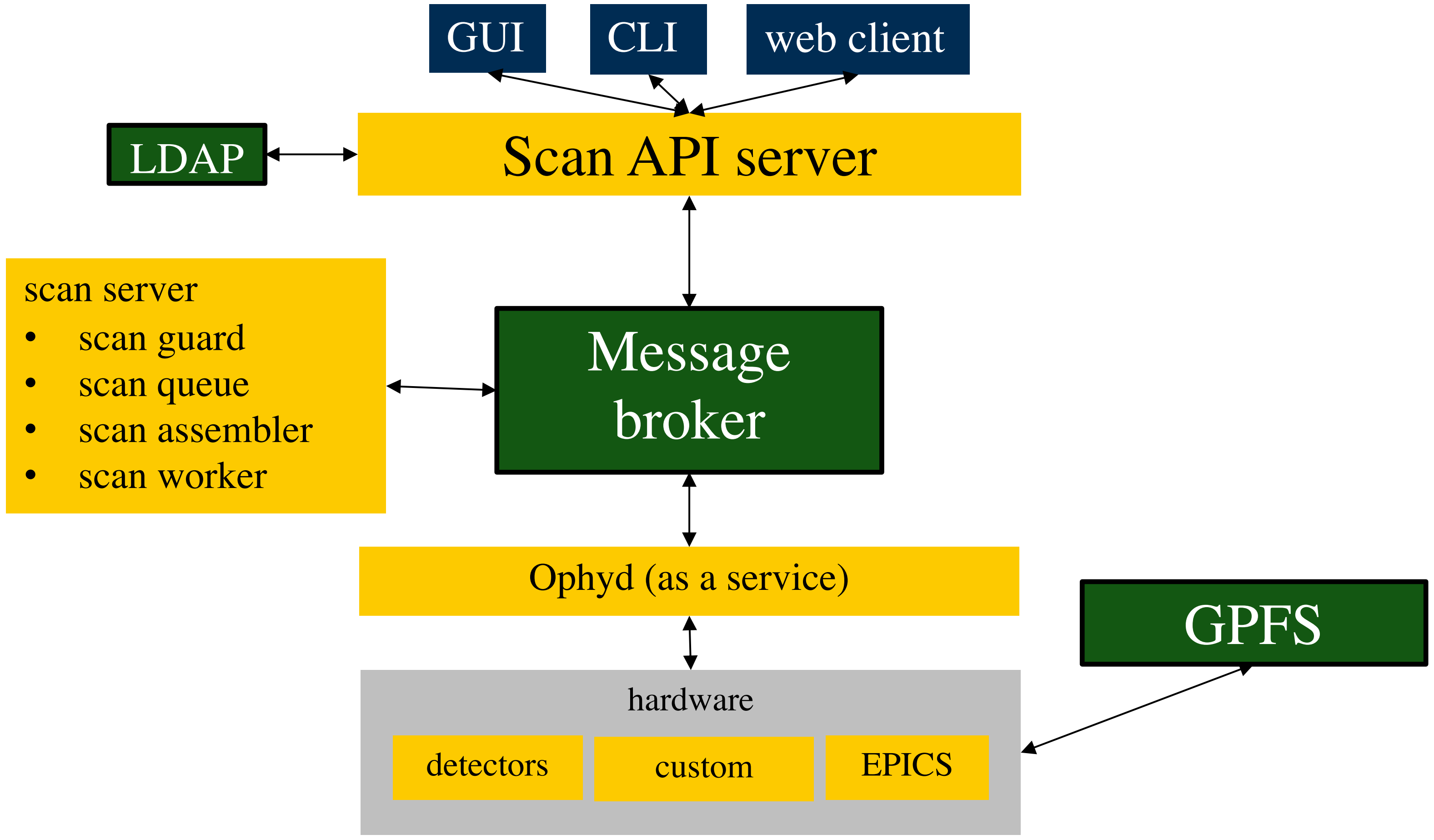
Bluesky

Lessons learned from the tests

- *A dedicated hardware abstraction layer may lead to a cleaner solution.*
- *Components should be kept small and clearly separated (-> microservices).*
- *Avoid propagating state objects through the entire system.*
- *Embed authorization and authentication from the beginning.*
- *Be aware of the external dependencies. Remove unnecessary dependencies.*
- *Data analysis should be structurally decoupled from scan orchestration.*

A potential direction…

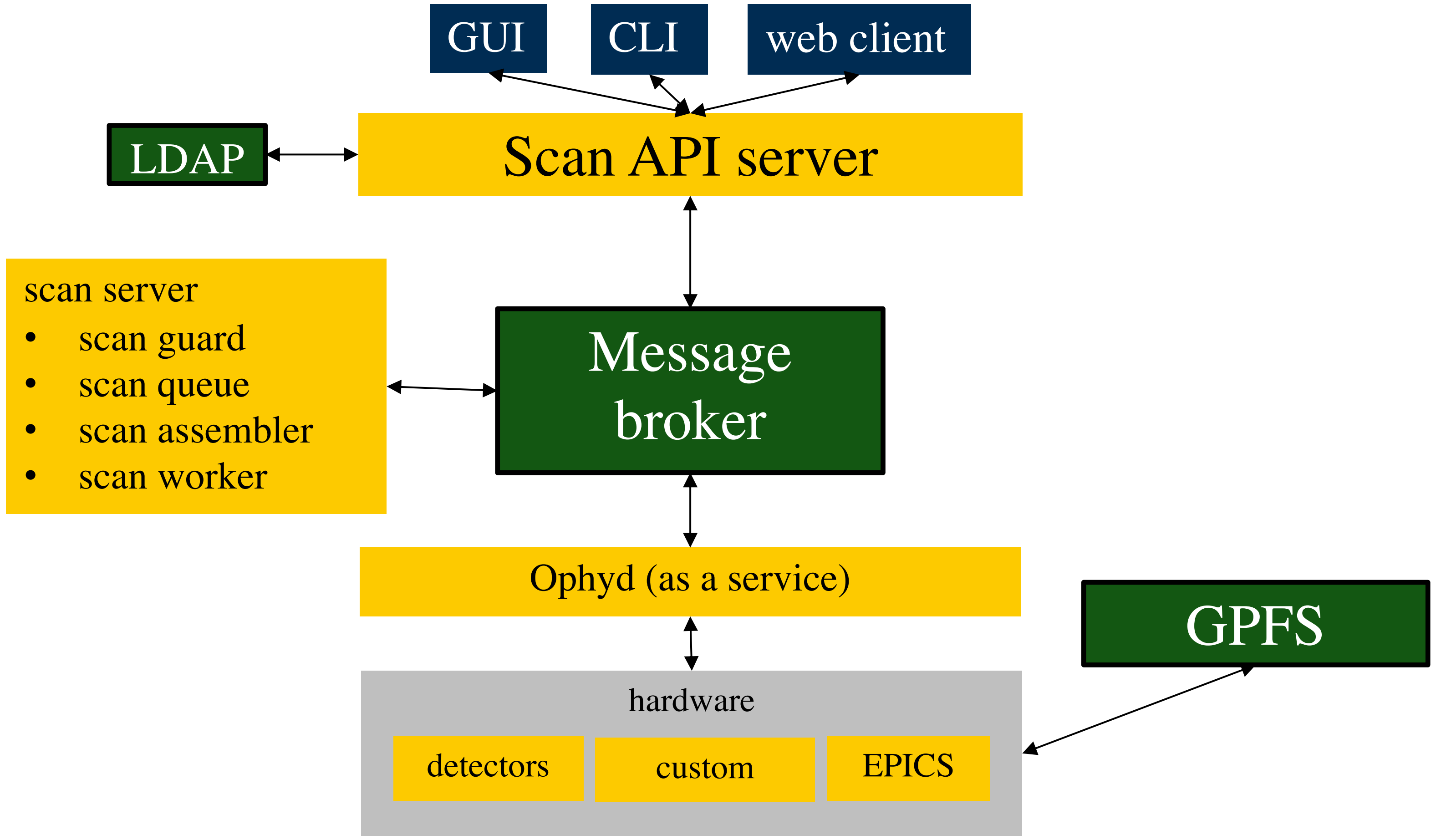
Bluesky "enhanced"



Advantages:

- Unified API between clients and the BEC system
- If needed, the scripting language can be easily changed in the future
- Easy to integrate with facility-specific DAQ components
- Smaller, more manageable and better maintainable services
- If needed, the system is highly scalable

Bluesky “enhanced”



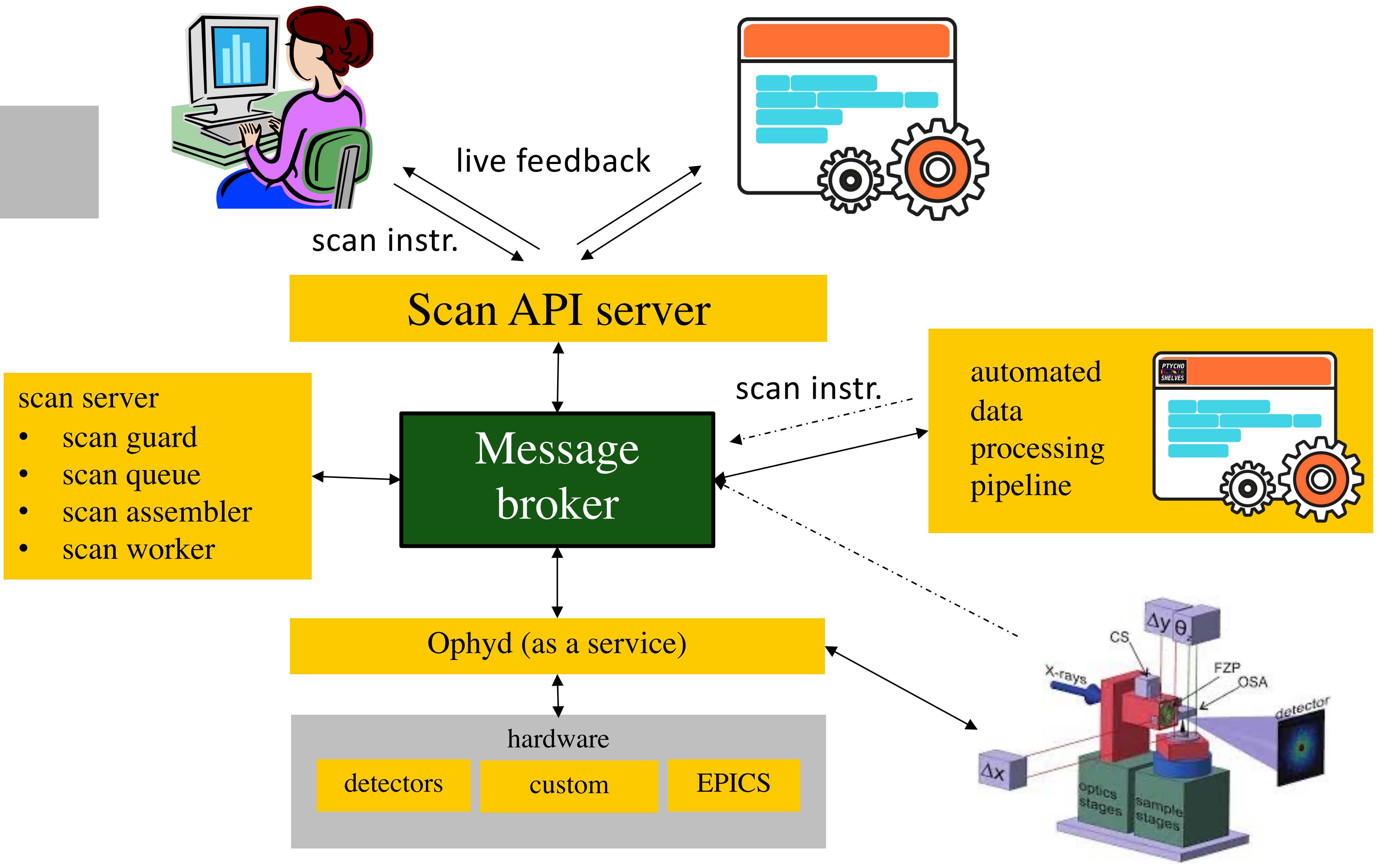
Statements by the Bluesky core dev team

The Bluesky Collaboration is moving its development focus from modular *libraries* to modular *services*. The work proposed by PSI would be beneficial to that effort.

In retrospect, we should have worked in a service-client design on the acquisition side earlier […]

PAUL SCHERRER INSTITUT PSI

Potential direction: Ptychographic tomography



PAUL SCHERRER INSTITUT

Questions?