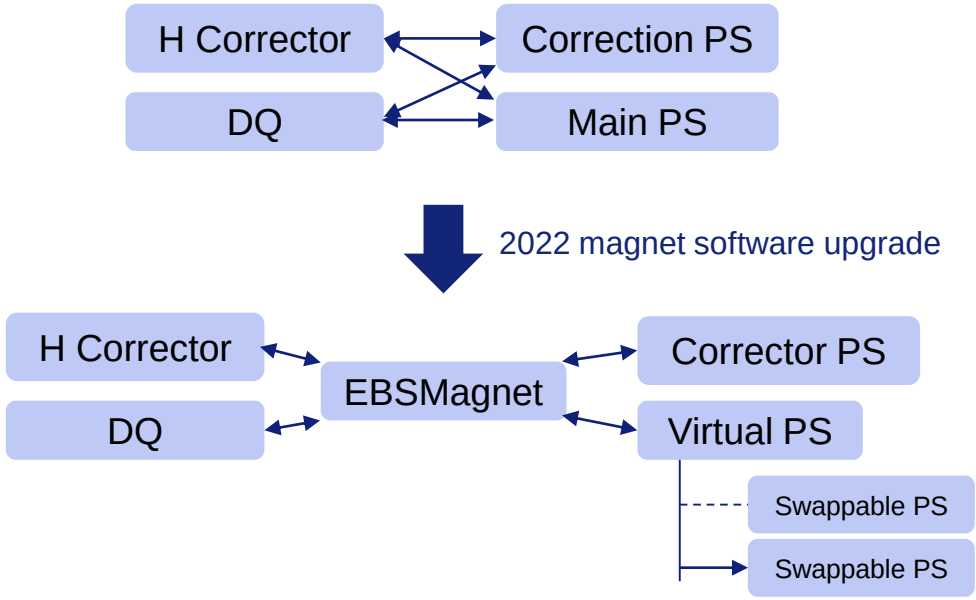
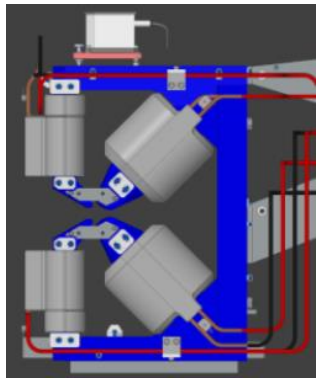
EBS MAGNETS SOFTWARE UPDATE (2022)

- **Improve maintainability/reactivity**
  - Centralize magnet calibration data and source code in a single library
  - Fix conflicts when applying strengths on combined function magnets
  - No #ifdef SIMULATOR directive in the code
  - Minimize hardware access, allow to set H/V at once
- **Power supply virtualization layer**
  - Integrate HotSwap system
  - Handle low level connection between magnet and PS
  - Use a unique device name to access a magnet
  - Allow standalone power supply (QF1E QF1A for injection)
  - Allow HotSwap spare usage for mini-beta optics evaluation (using additional quads)
- **Digital shadow / Simulator**
  - Simulate magnet cycling and electrical power overhead
  - Implement digital twin application
- EBS: 1056 swappable PS / 1156 corrector PS





For each instrument an instance of the pyAML c
necessary resources to respond to the user needs as

```
import pyaml
linac_ml = pyaml('linac.yaml')
linac = linac_ml.live
transferline_ml = pyaml('tl.yaml')
tl = transferline_ml.design
booster_ml = pyaml('booster.yaml')
boo = booster_ml.errors
sr_ml = pyaml('storagering.yaml')
sr = sr_ml.live
```

For the moment, the yaml file format is assumed
are found, this will be changed.

The user of pyAML will have the possibility to inter
setting set points, running commands and getting sta
Each component can be configured with a file.

2.1.1 Element

Each object which needs direct access to control sy
from Element class.Element provides all informatio
simulator.

`hardware_name()` → str returns control syster
a fully qualified tango device name, a process
used by the control system layer to access the

`name()` → str returns simulated object identif
typically used to access the simulator such as

pyAML specification document draft

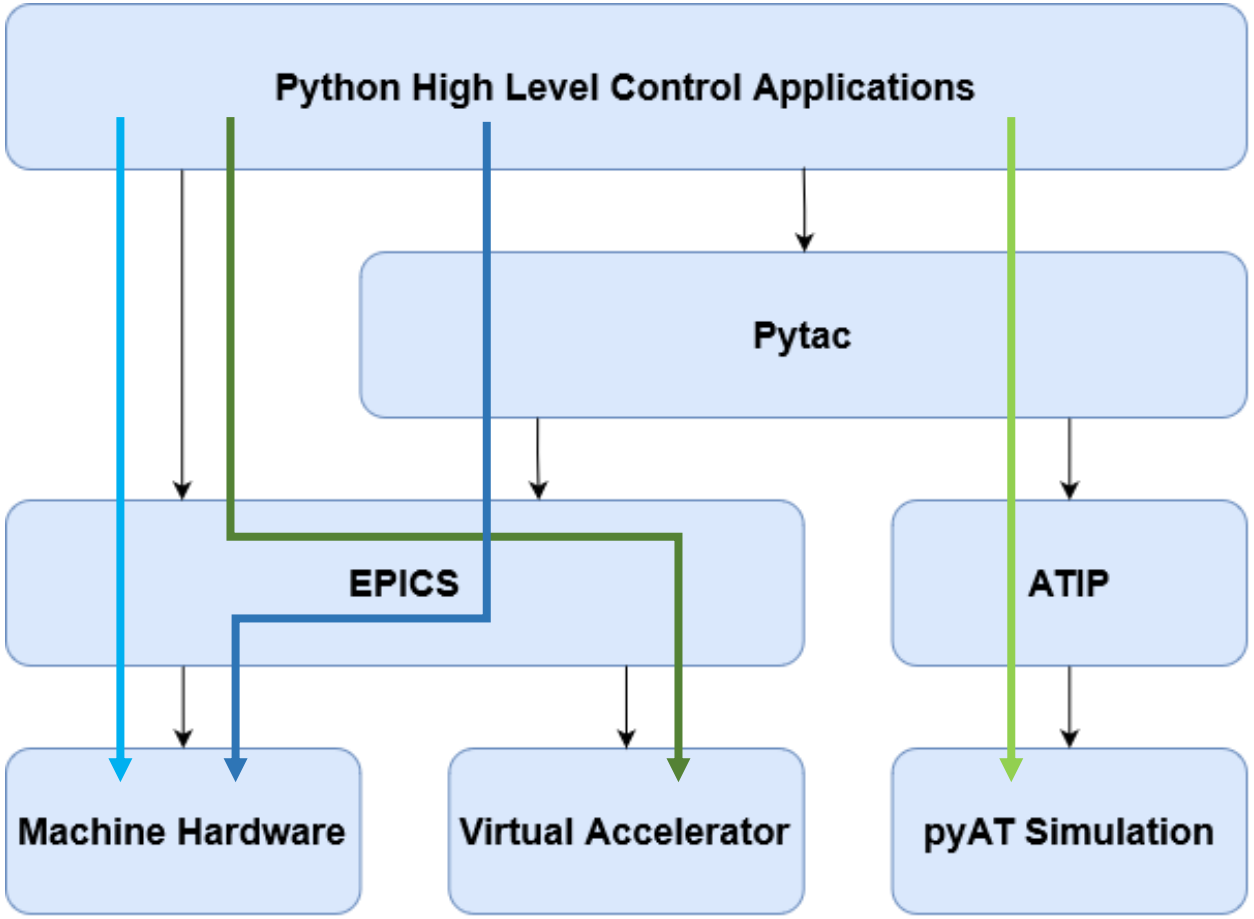
```
sr.live.qf1a.strength.get()
sr.live.qf1a.strength.archive()
sr.design.qf1a.strength.set()
sr.errors.qf1a.strength.units()
sr.shadow.optics.lifetime.get()
sr.live.tuning.run()
sr.errors.tuning.status()
booster.errors.tuning.status()
booster.live.update_optics()
```

WRITE NOW your pyAML

Python for Accelerator Control & Simulation at DLS

Normal operations control flow on live machine

Testing & validation control flow against simulation



PAMILA (Particle Accelerator MIddle LAyer)

- Demonstration of a new prototype middle layer package PAMILA
  - https://github.com/python-accelerator-middle-layer/pamila
  - Demo through example Jupyter notebooks

1. A minimum required environment and configuration setup
2. How to interact with the middle layer and its basic workflow
3. Bluesky wrapper
4. Complex, multiple-input-multiple-output “unit” conversions
5. “Stages and Flows” as parameters of high-level applications
6. Any additional topics from the audience

Modification of ESRF-EBS digital twin for SOLEIL II

Objectives of the Digital Twin (DT) for SOLEIL II:

- Full replication of the future SOLEIL II accelerators control system wired to simulations (of the beam, of some key items, ...) instead of the real hardware.
- Must allow to **design and test most of the control system** (high level devices and graphical interfaces, middle layer, commissioning tools) **before the commissioning period**.
- After commissioning, it is expected to be a fundamental tool for the control group to test new software before being deployed to the real system.

Proof of concept (POC) digital twin experiment started in July 2024 at SOLEIL:

- Team mixing software engineers and machine physicists.
- Gain experience in design and usage of DTs.
- **Modified version from ESRF-EBS DT** [1,2] to meet the needs of our POC.

What we will show:

The DT that now runs with SOLEIL II lattice in a separated environment from SOLEIL control system

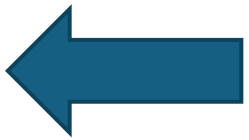
- Beam physics and diagnostics correctly modelled and accessible from the TANGO control system.
- Main modifications from ESRF DT version (use pyAT element types instead of nomenclature to detect controlled elements, use of generic 'Publisher' device to hold magnet strengths)
- **Automated procedure to generate a DT from a given lattice and TANGO nomenclature.**
- More than 1000 magnets simulated and controlled in the DT.
- Middle layer tests plugged to the DT (MML version from current SOLEIL, pyacal).

[1] Liuzzo, S.M., et al. "The ESRF-EBS Simulator: A Commissioning Booster." *Proc. ICALEPCS'21* (2022).
[2] Liuzzo, S.M., et al. "Update on the EBS Storage Ring Beam Dynamics Digital Twin." *Proc. ICALEPCS'23* (2023).

The problems

- Anomaly detection package for „on the fly” diagnostics (RF Cavity problem, mechanical movement)
- Accelerator stability forecast
- Fit optimization for physical analysis

The possible „**solution**”

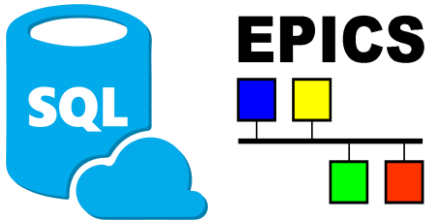


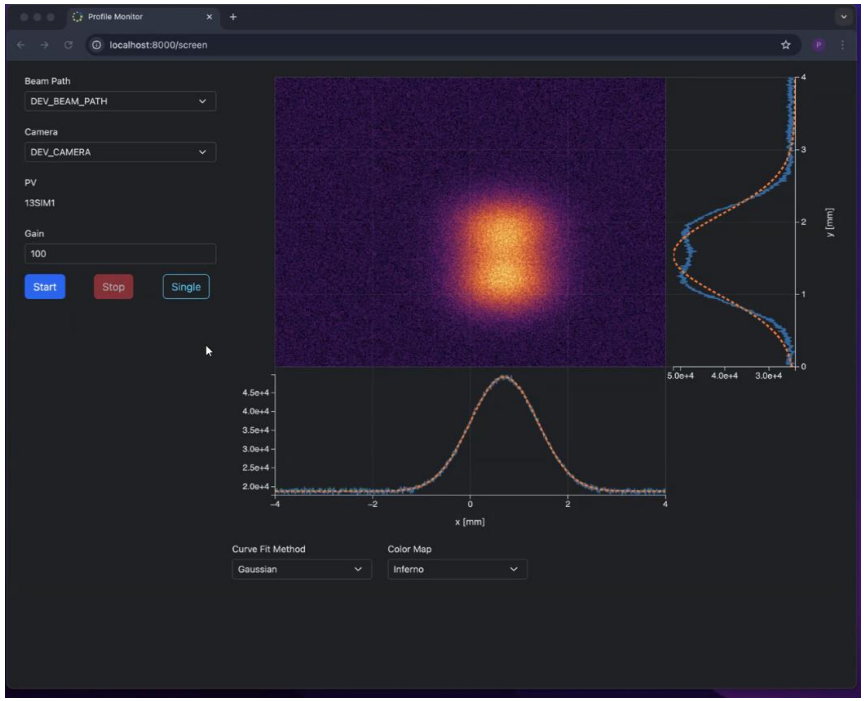
- **Fast deployment**
- **Engineering** non-scientific approach (plug & play)
- Lots of ready to use easy packages
- In the future one can switch to **Theano** for custom neuron design, **possible publication**

Web Applications in the Accelerator Control Room

Software Stack

LCLS-Tools



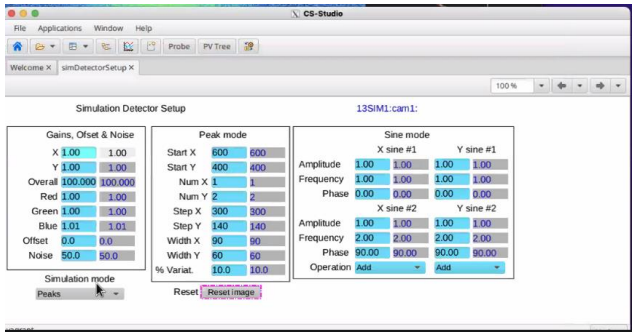


Webapp Advantages:

- SW Industry Standard GUI Platform
- Web Browsers Are Fast

New Middle Layer Advantages:

- Local Device Database
- Dynamic Device Object
- Composition Over Inheritance



Bonus Advantages:

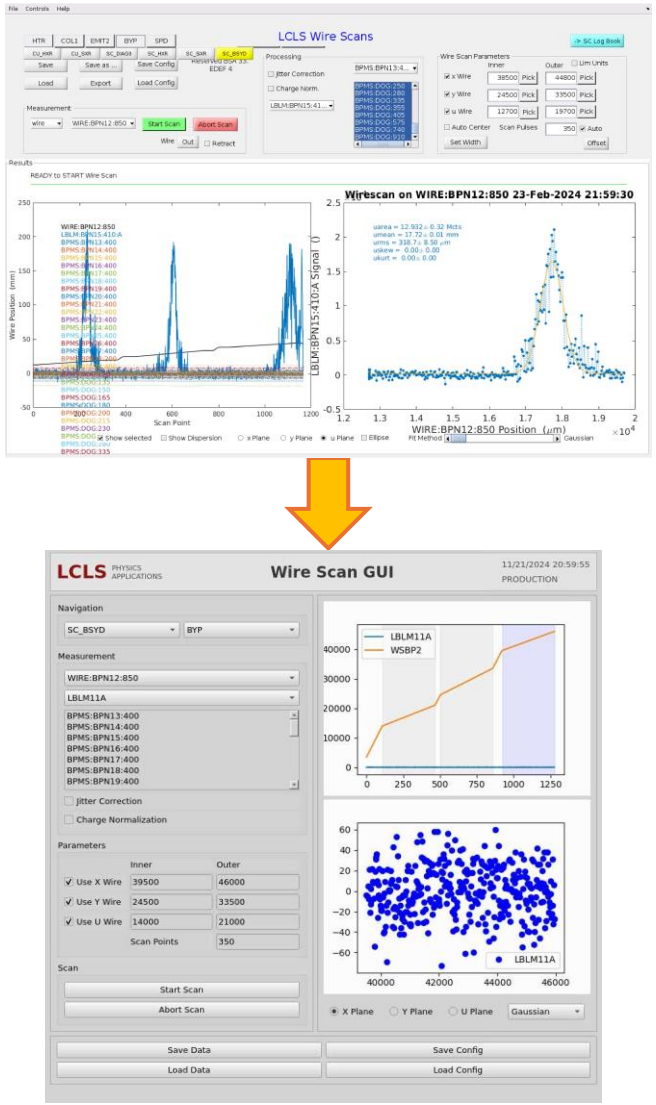
- Containerized Environments
- EPICS Simulation

SLAC + radiasoft

Evan Carlin (Radiasoft) & Eloise Yang (SLAC)

Developing a Python Wire Scan Measurement

- Objective
  - Replace **legacy MATLAB implementation** with a modular, Python based middle-layer approach
- Key Components
  1. Wire Device Object
     - Encapsulates EPICS PV interactions, allowing **attribute-based control**
       - *wire.position*
     - Provides Python methods to execute wire scans
       - *wire.perform_measurement()*
  2. Beam Loss Detector Device Object
     - Reads beam loss data **from timing buffer**
  3. Measurement Process
     - **Set scan parameters** via EPICS or attribute calls
     - **Reserve timing buffer** before data collection
     - **Run scan:** move wire, collect synchronous beam loss data
     - **Retrieve and process data,** then **release the buffer**
- Impact
  - **Decouples GUI from core logic** enabling script-based operation
  - Ensures synchronized measurements using BSA buffers
  - **Modernizes wire scan execution** improving maintainability and performance



CATAP (Controls Abstraction to Accelerator Physics)

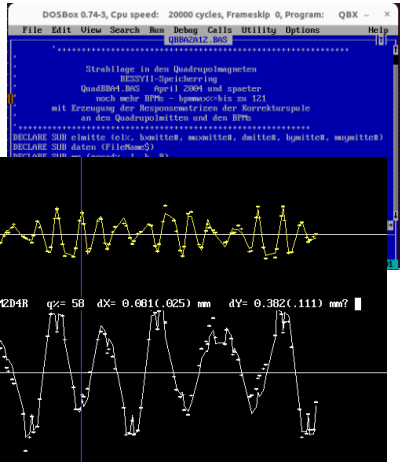
- Previously a C++/Python library (2019-2023), migrated to a pure-python library as of 2024
- Provides a user-focused way of interacting with CLARA control system for:
  - *Operators*
  - *Experimental Users*
  - *Non-controls technical staff*
- Aims to reduce the barrier of entry for accelerator control and data acquisition
- Configurable files (YAML) are generated for each piece of hardware (PVs, controls information, metadata)
- YAML files are used to generate hardware-specific objects using pydantic models
  - *ensures type-checking amongst other validation*
- **Hardware instances**:
  - *Created and accessed via a **Factory***
  - *Allows users to perform the same routine for multiple hardware objects*
  - Also created/accessed via a High-Level System
    - *LLRF, Modulator, Protection classes can be grouped into a Cavity hardware class*
    - *Mirrors, Shutters, Diagnostics, Energy Meters can be grouped into a PI Laser class*
- **Snapshots:**
  - Since Hardware instances are pydantic, we can:
    - *Output "snapshots" of settings into JSON/YAML*
    - *Load "snapshots" back into Hardware class and apply settings*
    - *Perform diffs of snapshots*

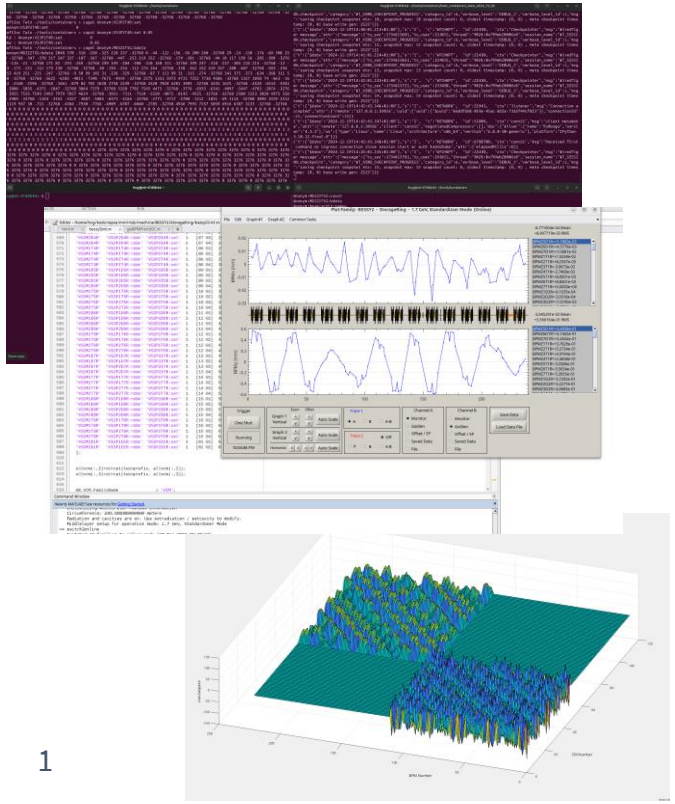
BESSY II CONTROL ROOM

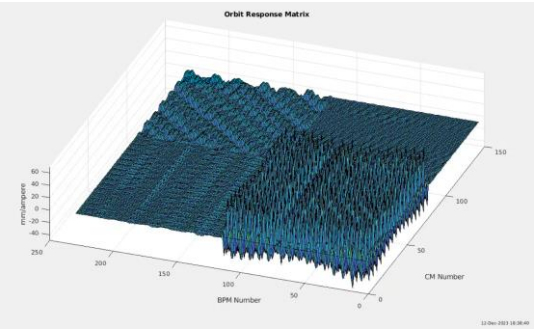
- user operation & development BESSYII & MLS
- machine time!! → middle layer in action
- high level physics applications
- outlook towards more complex machine commissioning & automated operation

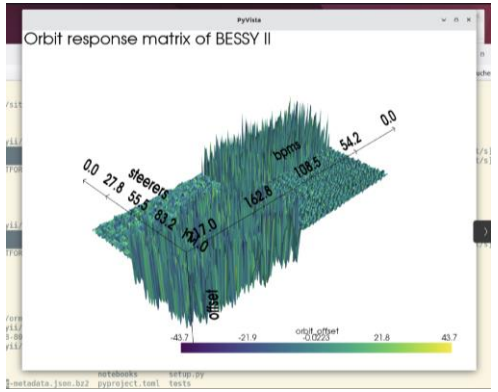
DIGITAL TWIN

- twin time!! --> middle layer in action
- talk about show cases
- performance and architecture









Preliminary Integration of the Python-based Optics and Beam Tracking Tools into Low Energy Beam Transferline Design for the Sarajevo Ion Accelerator

SIRIUS PYACAL evaluation at other facilties