Integrating automation and optimisation into the CERN control system

A. Huschauer, N. Burchon, A. Calia, L. Felsberger, J.C. Garnier, R. Gorbonosov, M. Hostettler, F. Irannejad, D. Jacquet, **V. Kain**, N. Madysa, B. Rodriguez Mateos, K. Papastergiou, C. Petrone, M. Remta, M. Schenk, M. Sobieszek, G. Trad, F. Velotti, J. Wulff

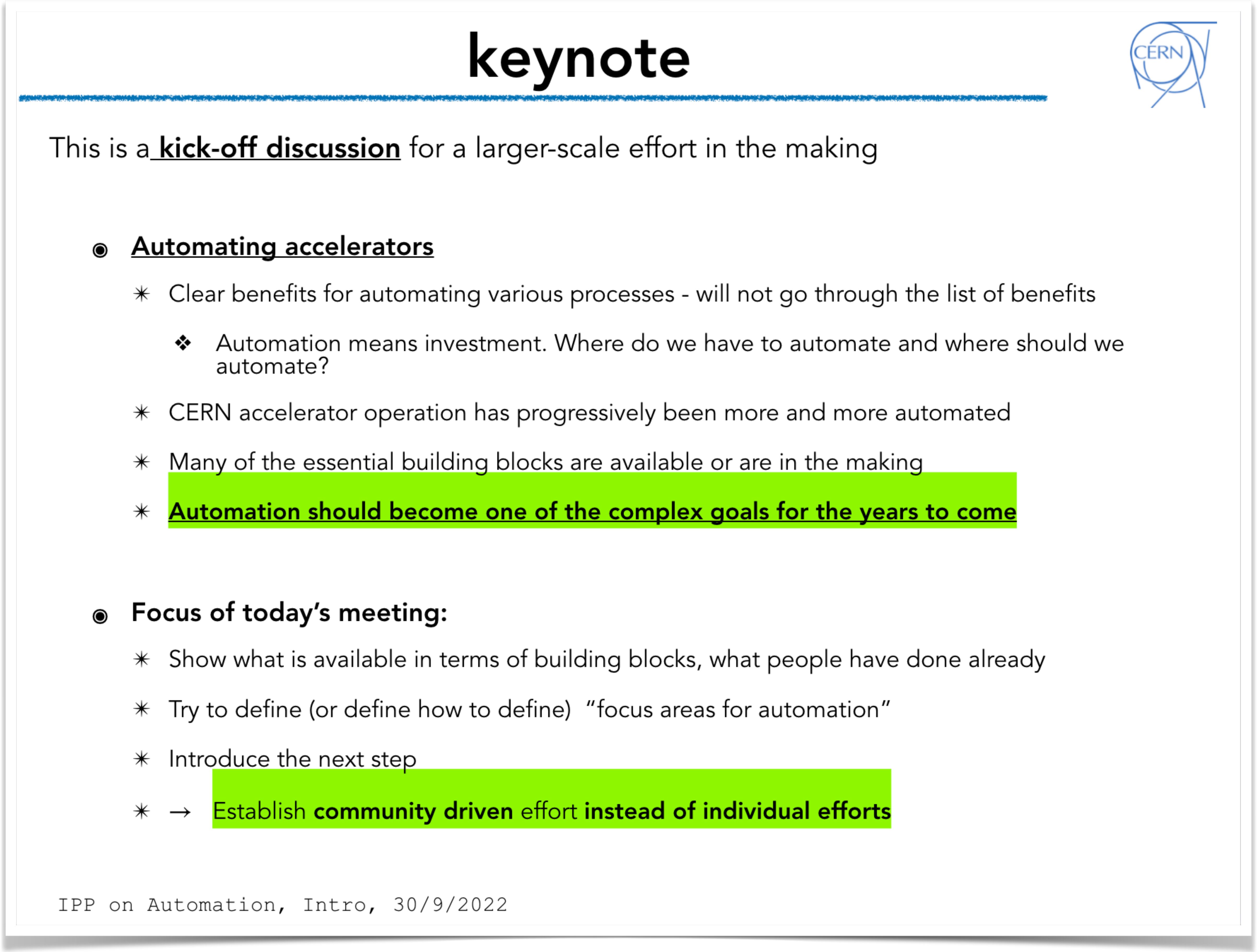
Automation - key challenge for CERN accelerators

Long-term effort at CERN → "**Classical Automation**" in the LHC

Automation - key challenge for CERN accelerators

Long-term effort at CERN → "**Classical Automation**" in the LHC

Revolution in the last years → availability of controls building blocks for full automation, optimisation algorithms, AI/ML



Automation discussion becoming centre-stage:

Example: Introduction slide to Injectors Performance Panel (IPP) mini-workshop on automation on 30/9/2022

The waves of automation @ CERN

The **current** automation efforts are based on three threads

- **Automation wave 1** (2006 -)
  - reduce complexity through models in the accelerator control system (**LSA**)
  - **high level parameter control**, procedures in software (sequencers), software interlock system, classic control algorithms in feedforward and feedback (SVD, COSE,…)

The waves of automation @ CERN

The **current** automation efforts are based on three threads

- **Automation wave 1** (2006 -)
  - reduce complexity through models in the accelerator control system (**LSA**)
  - **high level parameter control**, procedures in software (sequencers), software interlock system, classic control algorithms in feedforward and feedback (SVD, COSE,…)
- **Automation wave 2** (2018 -)
  - provide clever solutions if models are not available. E.g. Learn them…
  - **game-changer:** Python in the control room
  - on-demand execution of **optimisers, ML, etc.**

The waves of automation @ CERN

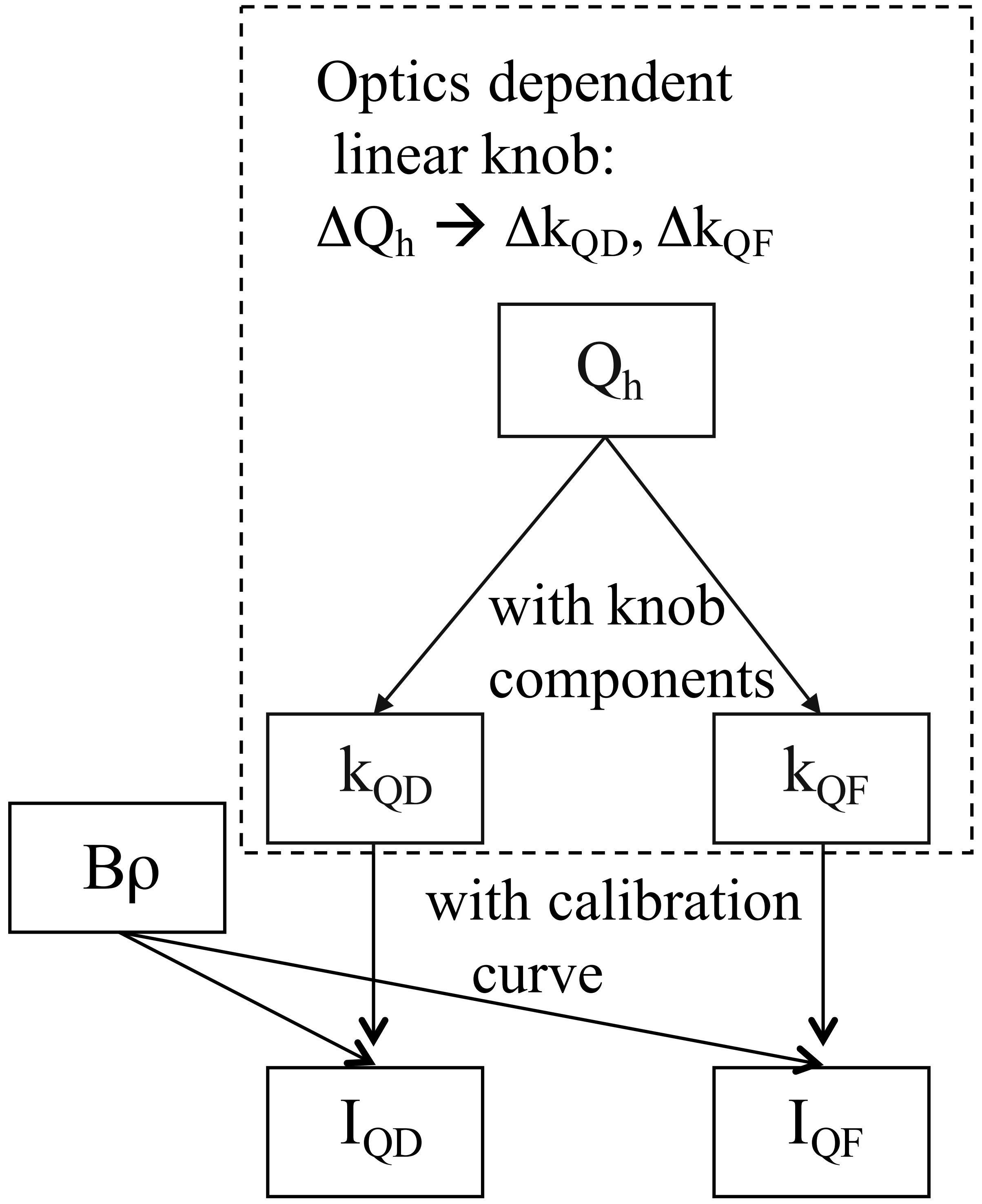
The **current** automation efforts are based on three threads

- **Automation wave 1** (2006 -)
  - reduce complexity through models in the accelerator control system (**LSA**)
  - **high level parameter control**, procedures in software (sequencers), software interlock system, classic control algorithms in feedforward and feedback (SVD, COSE,…)
- **Automation wave 2** (2018 -)
  - provide clever solutions if models are not available. E.g. Learn them…
  - **game-changer:** Python in the control room
  - on-demand execution of **optimisers, ML, etc.**
- **Automation wave 3** (2021 -)
  - Closing the loop…
    - Implementation of **frameworks** (Generic Optimisation Framework (GeOFF), Machine Learning Platform)
    - **auto**-launch correction, **auto**-resets of equipment, **auto/online**-analysis
    - enabling the operational implementation of **auto-pilots**

High Level Parameter Control and Generation

LHC Software Architecture (LSA)

- "only" work with **normalised strengths** k_n and **high level physics parameters**
  - E.g. physics parameter Q_h defined on top of of k_{QD} and k_{QF}
- all parameters are functions of time in "cycle"
- The **lower level** field/current/… functions are **automatically derived**
  - using centrally stored calibration curves, $B\rho$/momentum, other parameters
  - → key concept behind COSE

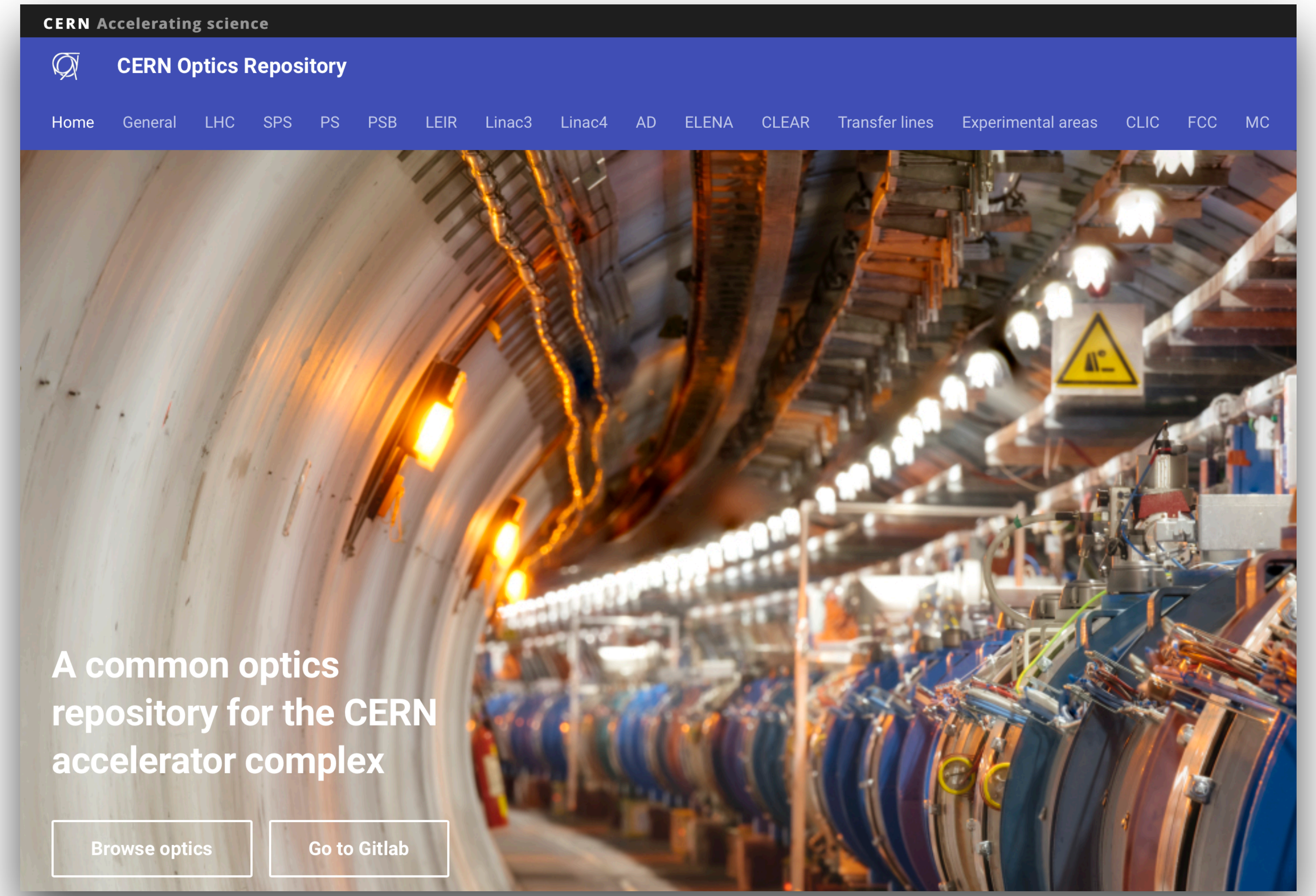


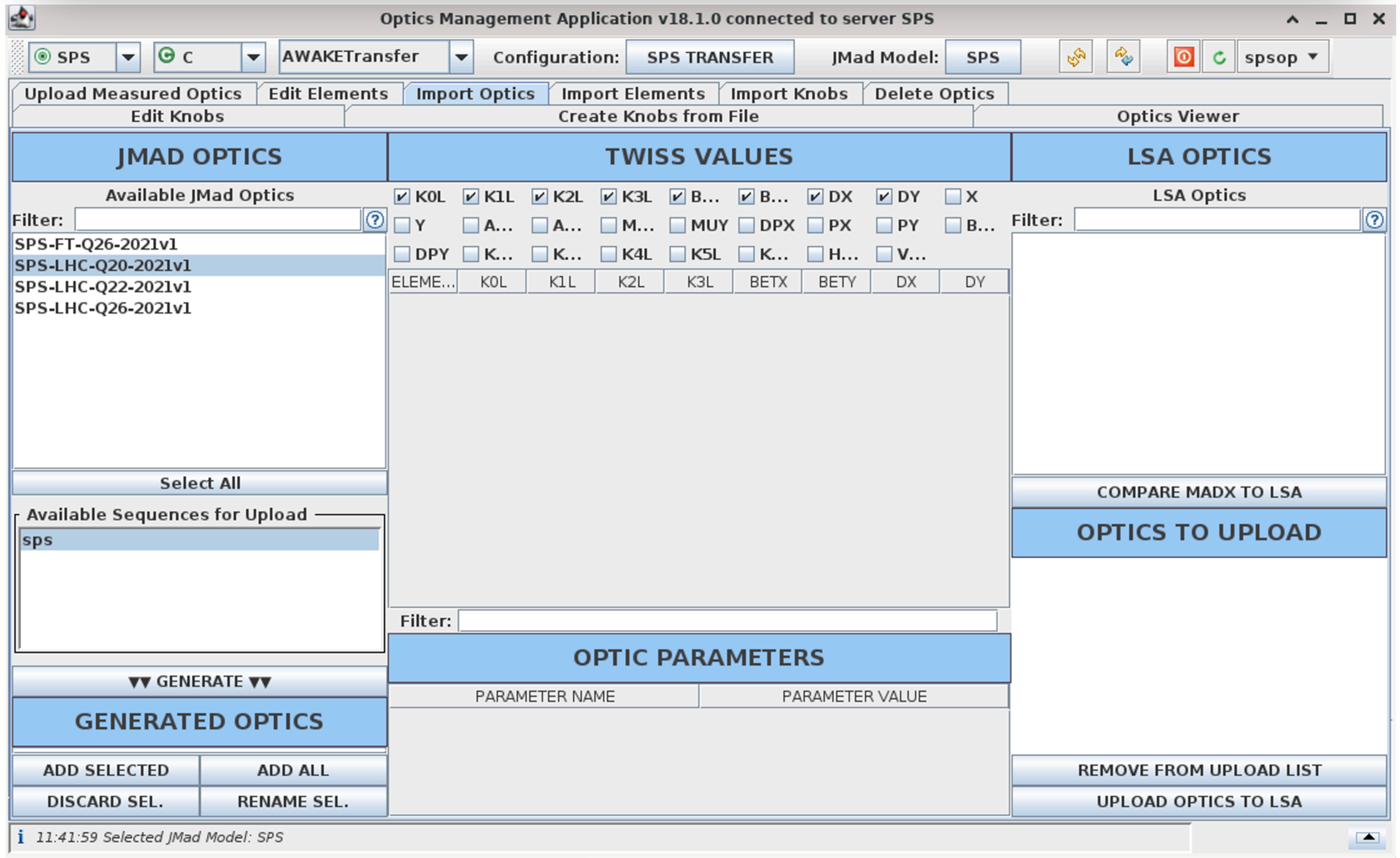
relations/rules/values in LSA DB…

High Level Parameter Control and Generation

Generation: generating parameters according to physics rules

- Parameter/settings generation based on MAD-X optics model rather than empirical definition
- Optics upload from gitlab-based CERN optics repository
  - Efficient upload and modification of optics (even in the control room)





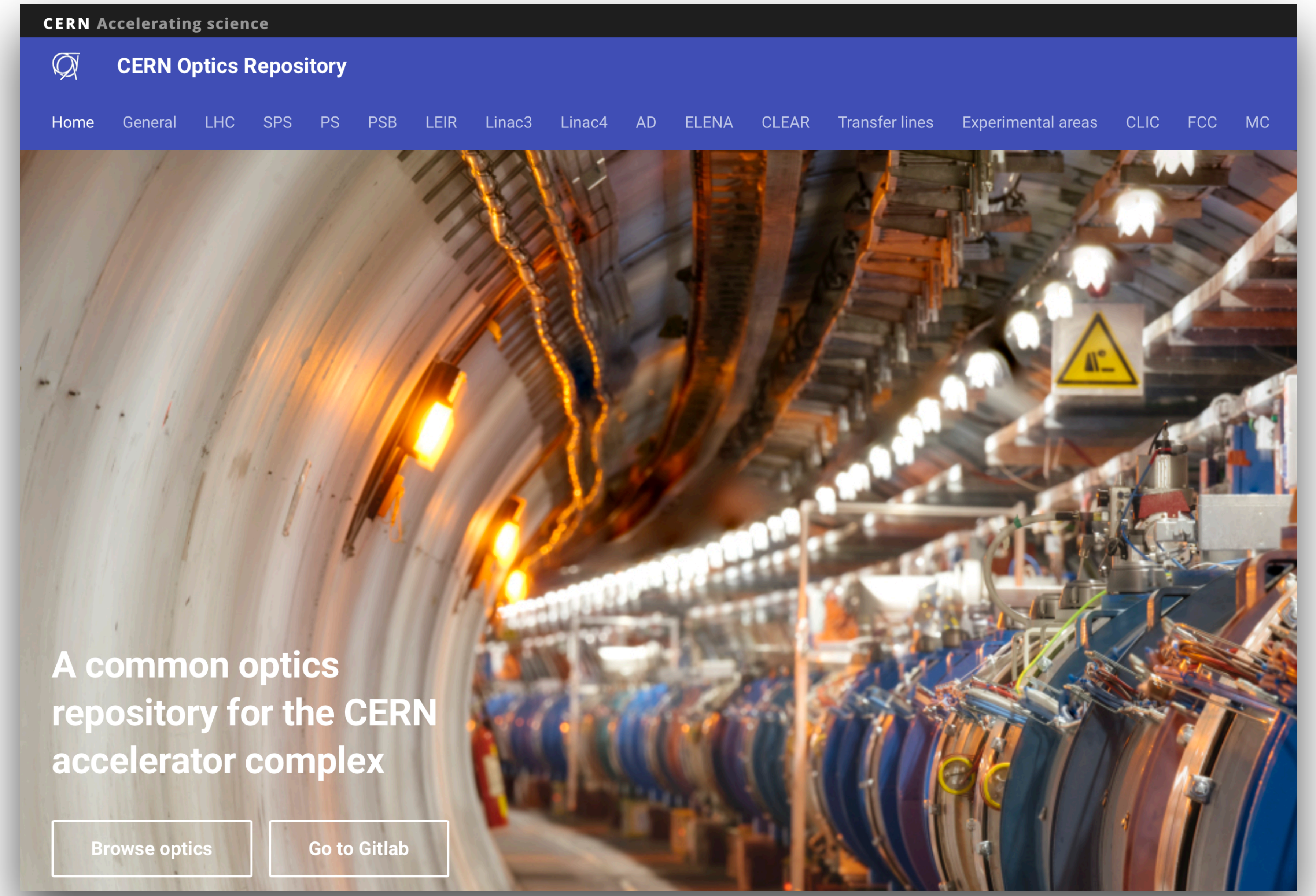
High Level Parameter Control and Generation

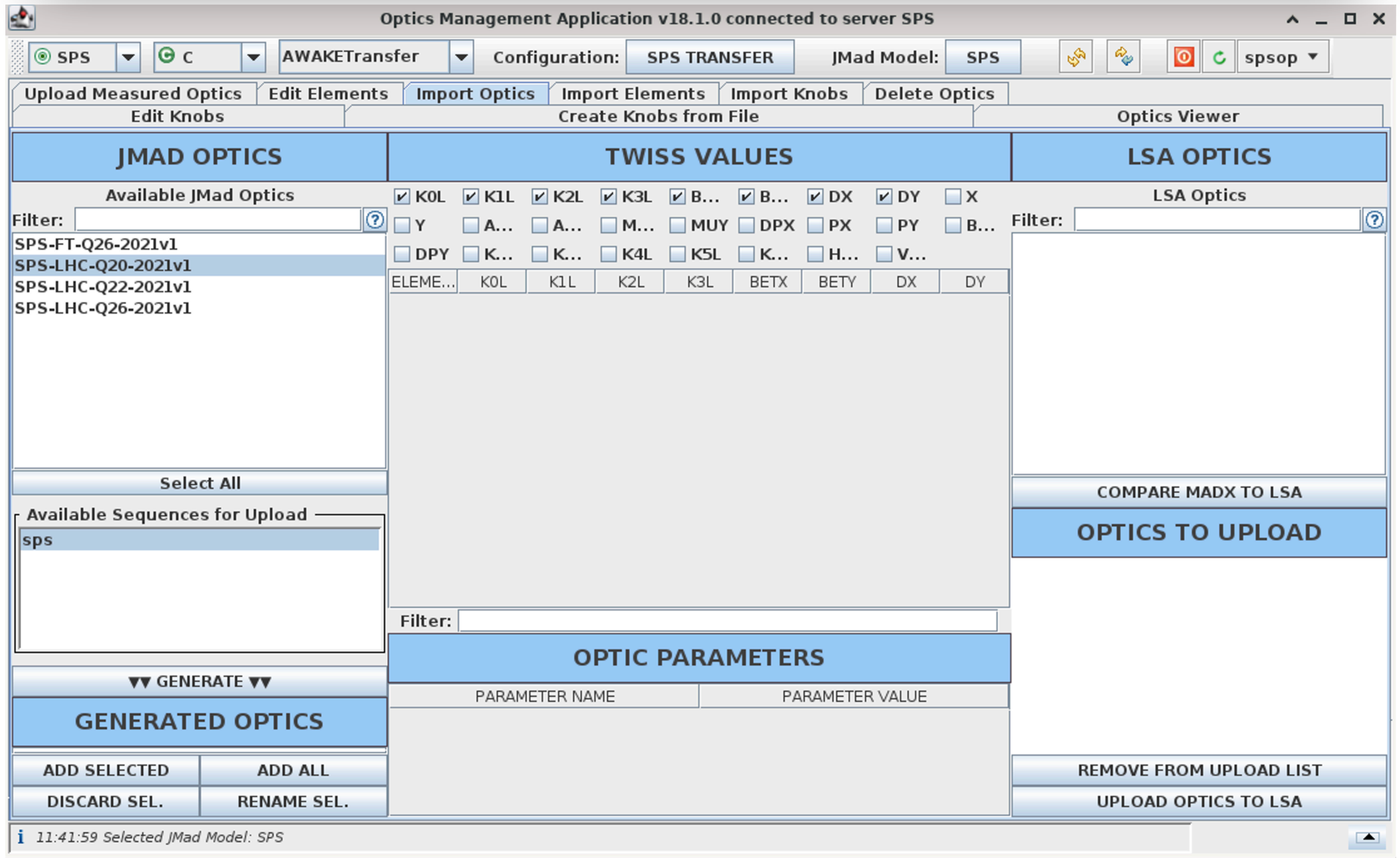
Generation: generating parameters according to physics rules

- Parameter/settings generation based on MAD-X optics model rather than empirical definition
- Optics upload from gitlab-based [CERN optics repository]
  - ✳ Efficient upload and modification of optics (even in the control room)

HL parameter control is natural operating principle of the LHC and the SPS

- Large effort in recent years to roll out in the injectors

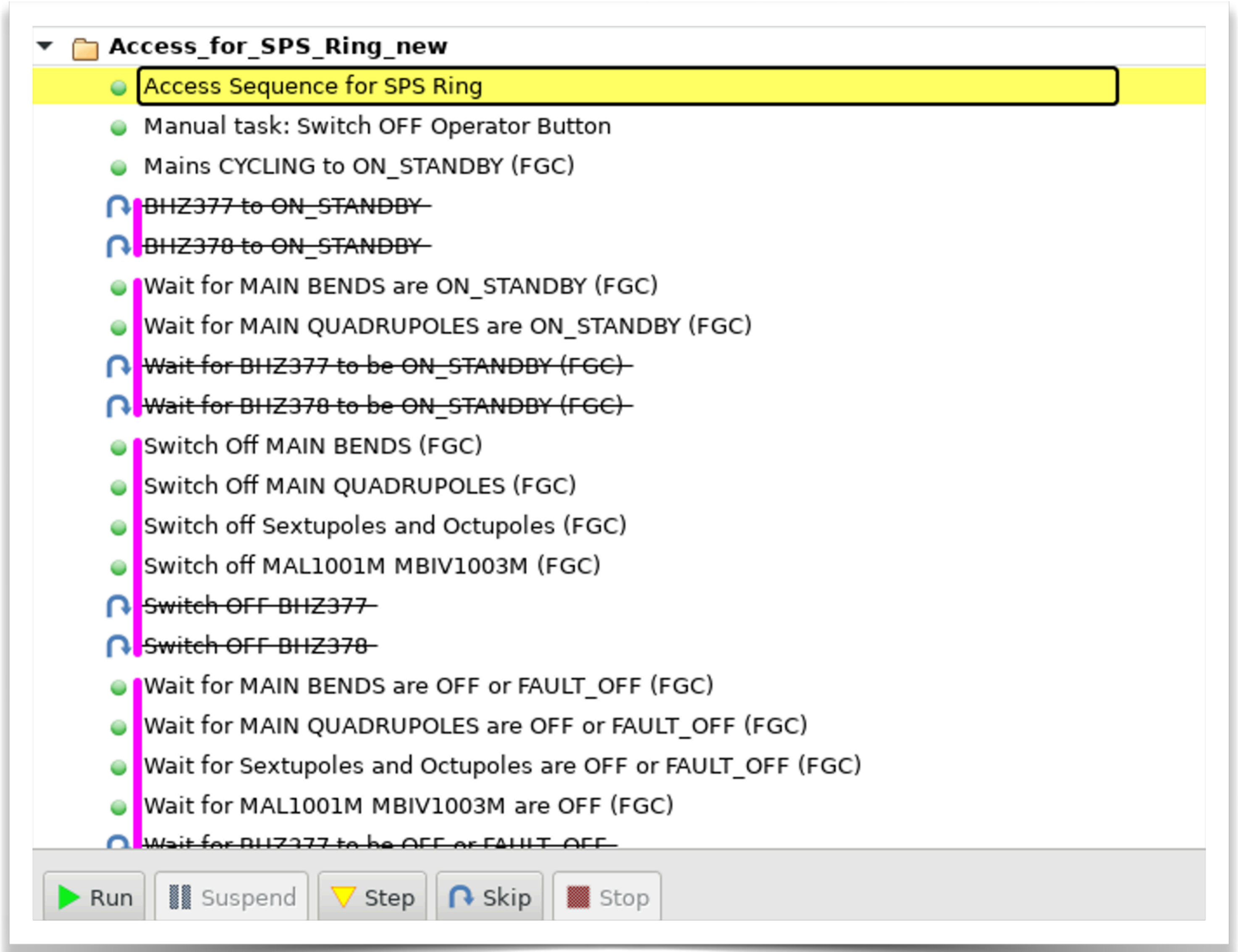


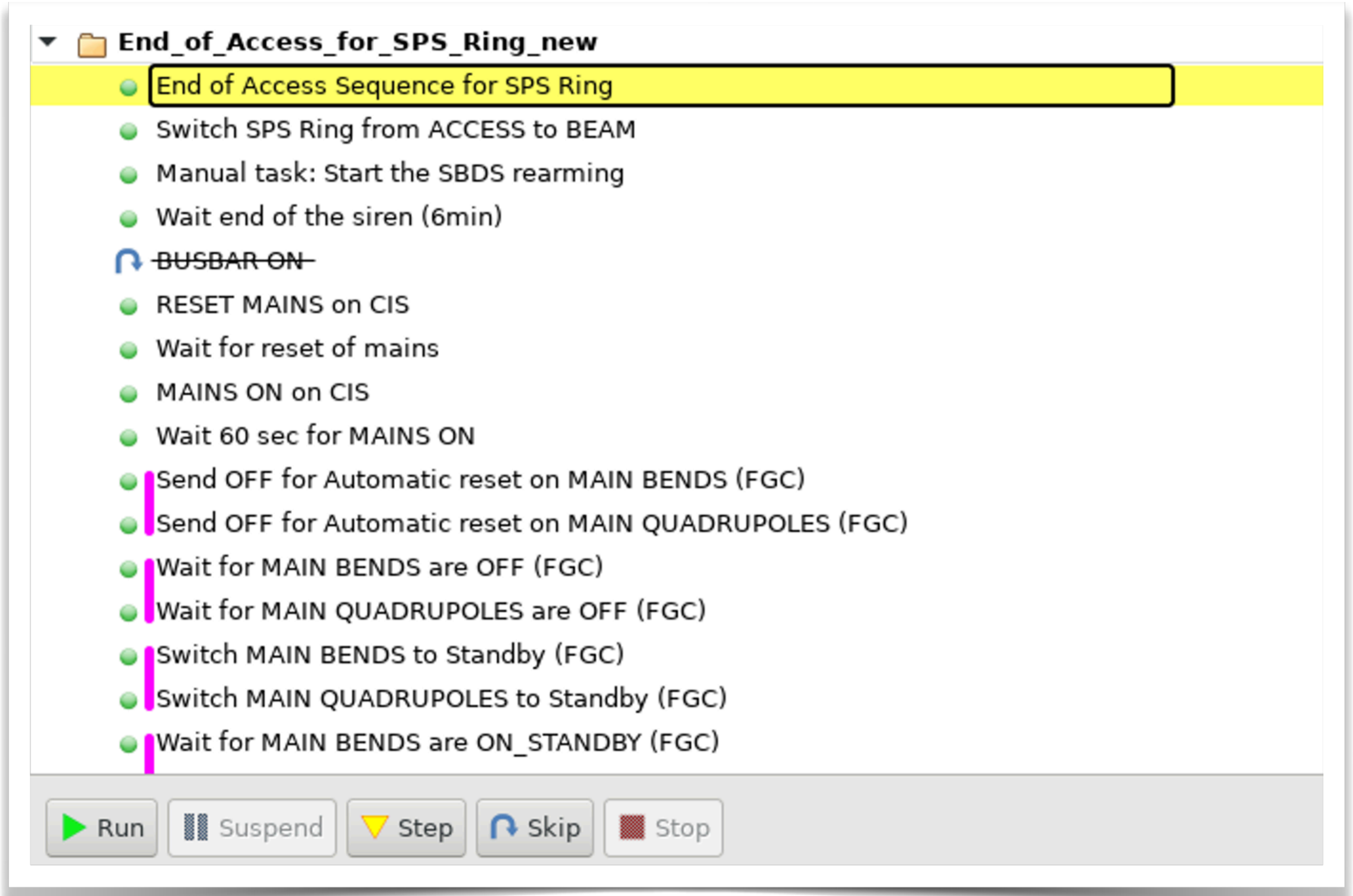


Sequencer

= transforming procedures into code → reusable sequences, sequence tasks,…

- Benefit: clearly defined and hence reproducible sequence of actions
- Example: give and end access to SPS ring



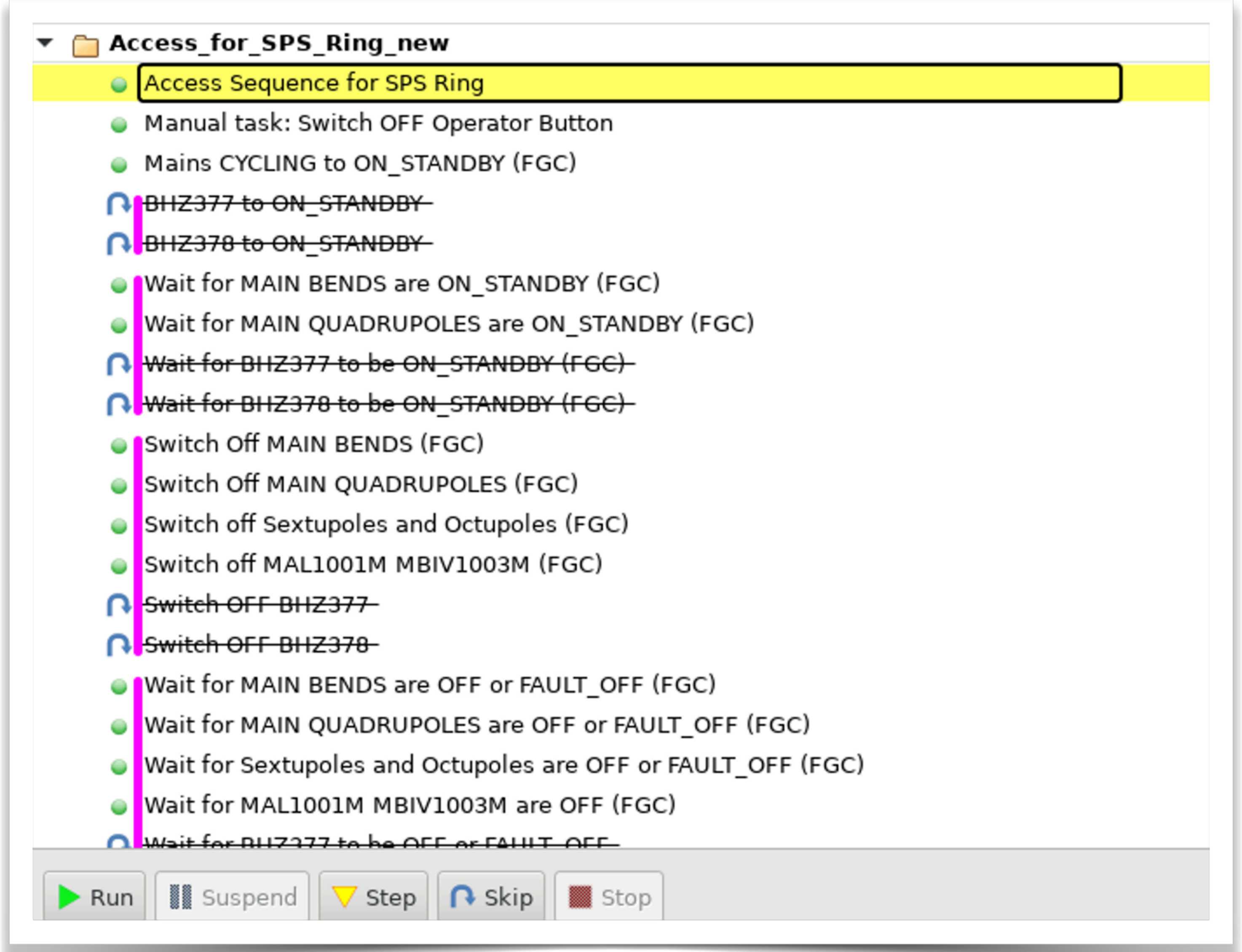


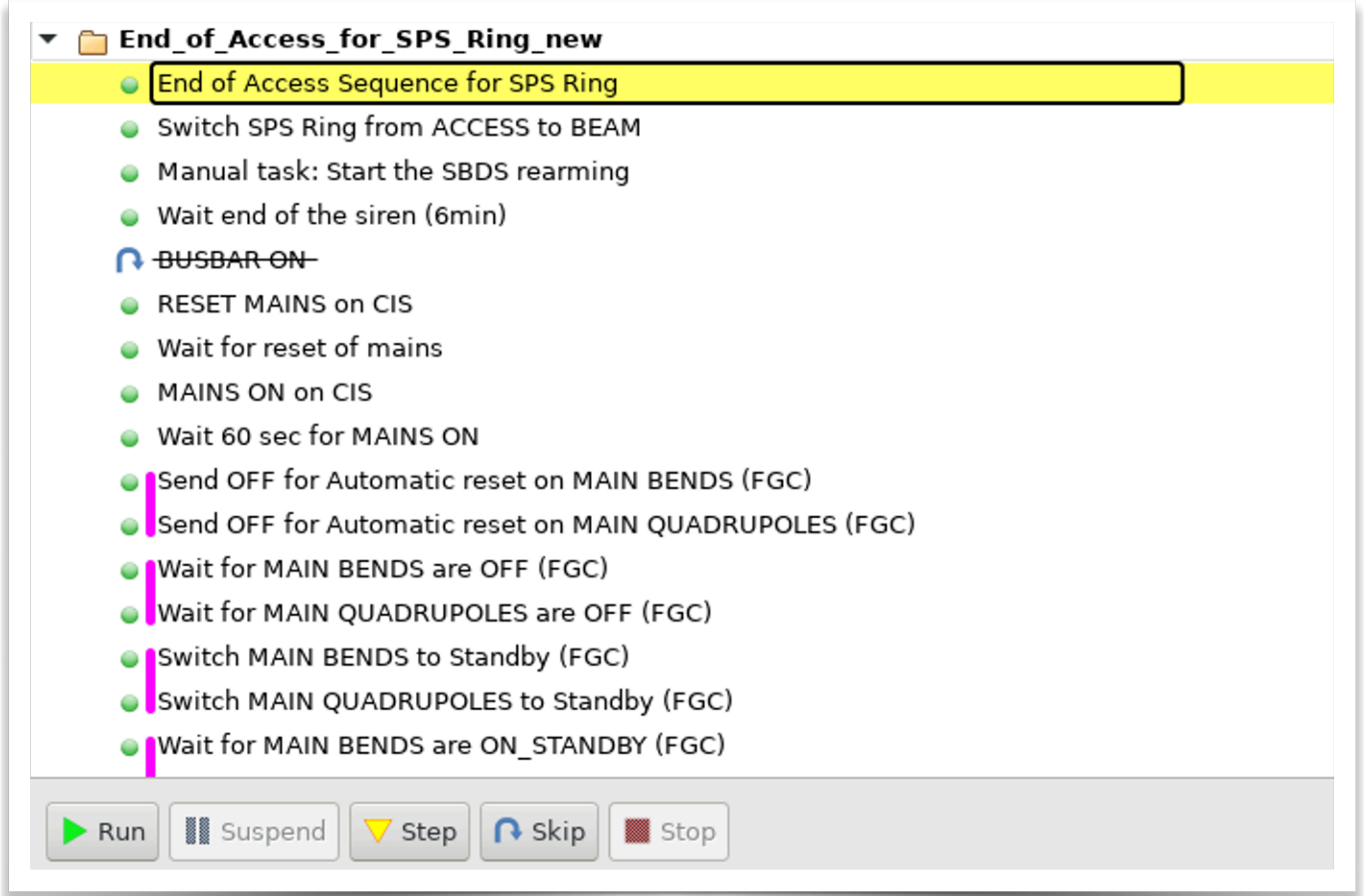
The **entire LHC cycle** is orchestrated by the **LHC sequencer** (GUI-managed)

Sequencer

= transforming procedures into code → reusable sequences, sequence tasks,…

- Benefit: clearly defined and hence reproducible sequence of actions
- Example: give and end access to SPS ring





The **entire LHC cycle** is orchestrated by the **LHC sequencer** (GUI-managed)

Next: sequences/sequence tasks to be **launched programmatically** (no GUI needed)

- key for auto-recovery, auto-measurement and auto-correction.

Acc-Py - Python for CERN accelerators

…with all the devOps one can dream of.

Python fully integrated with control system:

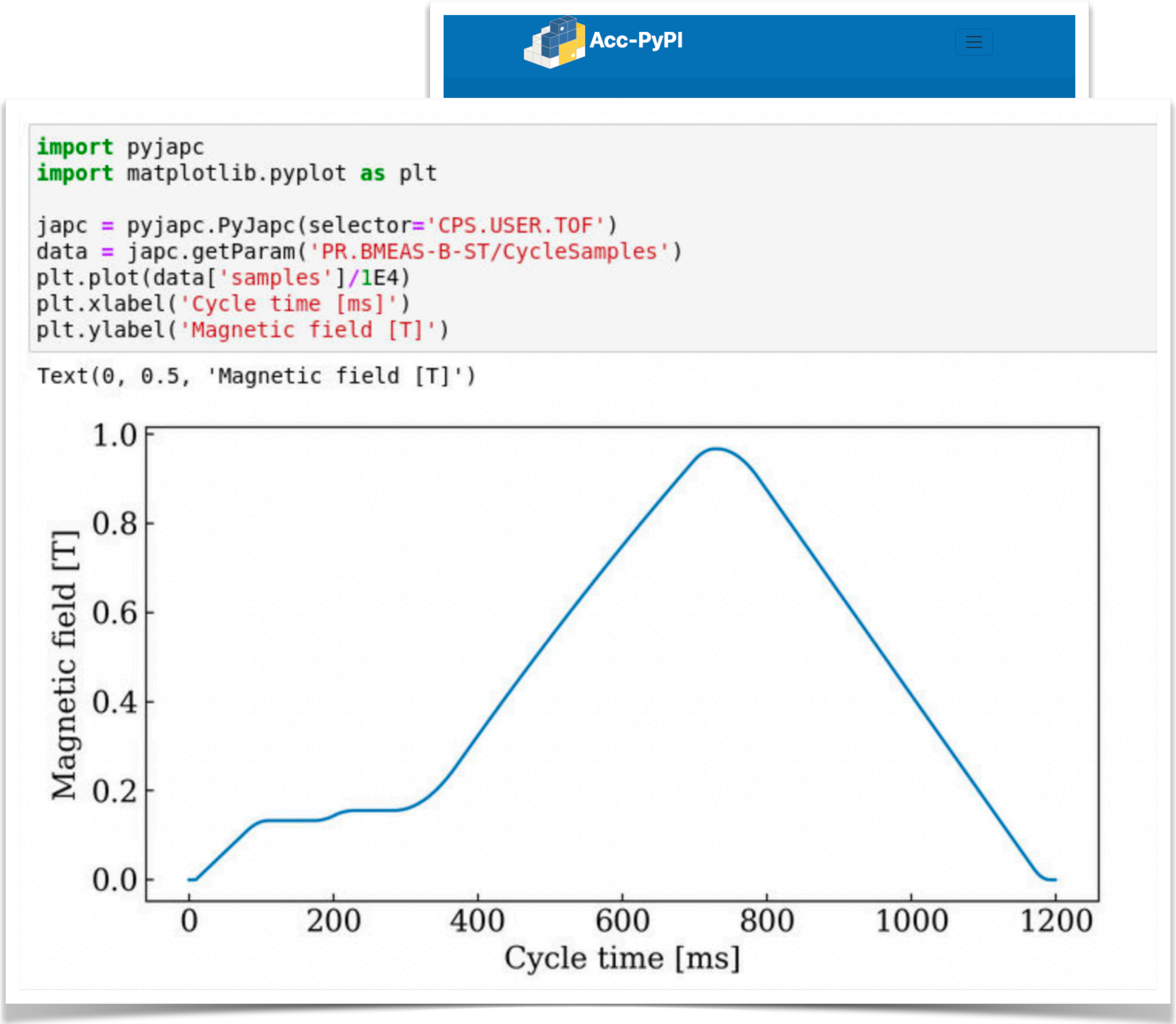
- Equipment access: **pyjapc**, pyda
  - ✳ next: event building

Acc-Py - Python for CERN accelerators

…with all the devOps one can dream of.

Python fully integrated with control system:

- Equipment access: **pyjapc**, pyda
  - next: event building



Acc-Py - Python for CERN accelerators

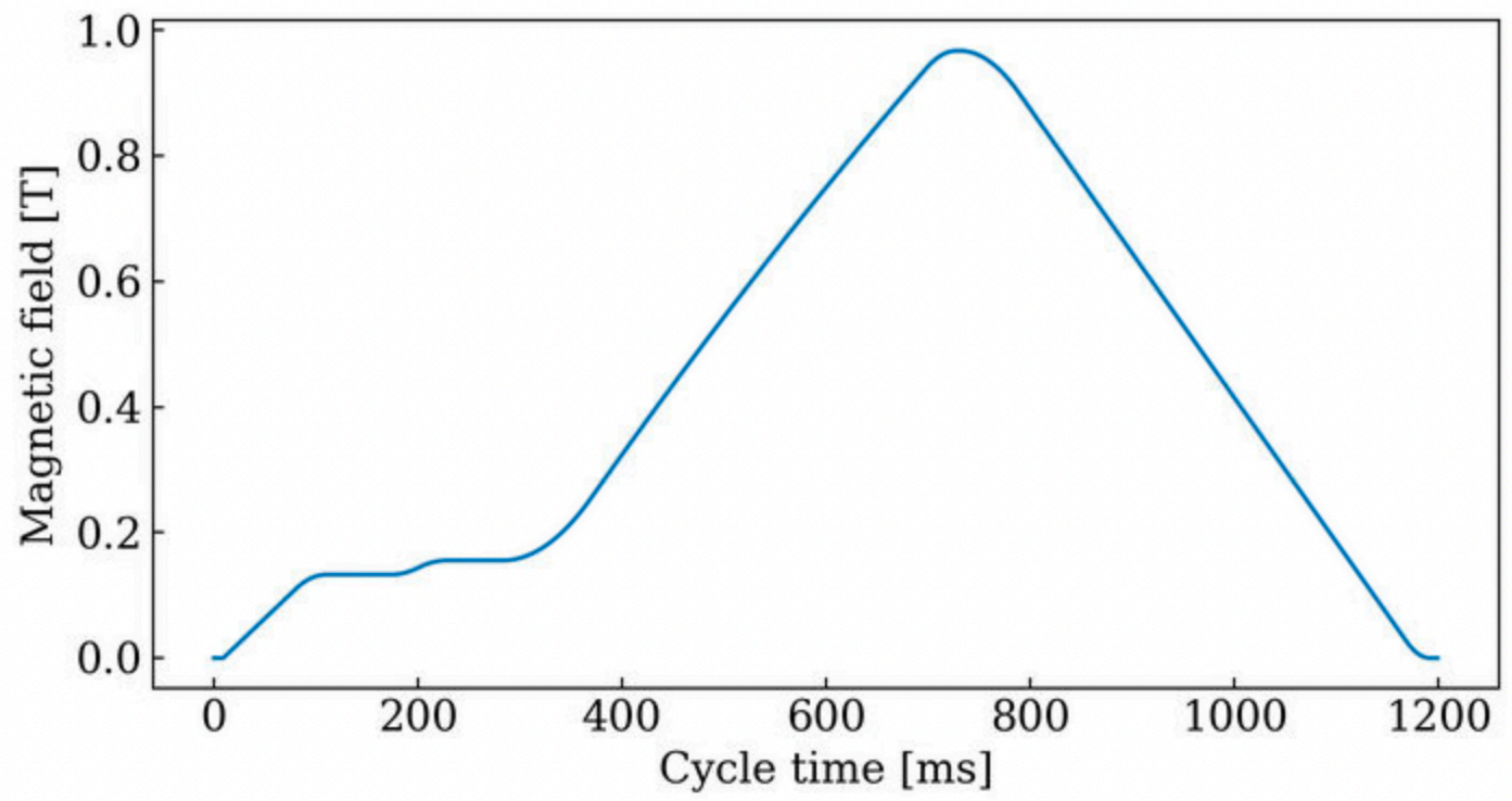
…with all the devOps one can dream of.

Python fully integrated with control system:

- Equipment access: **pyjapc**, pyda
  - next: event building
- access to settings (pjLSA) and archived data (**NXCALS**)

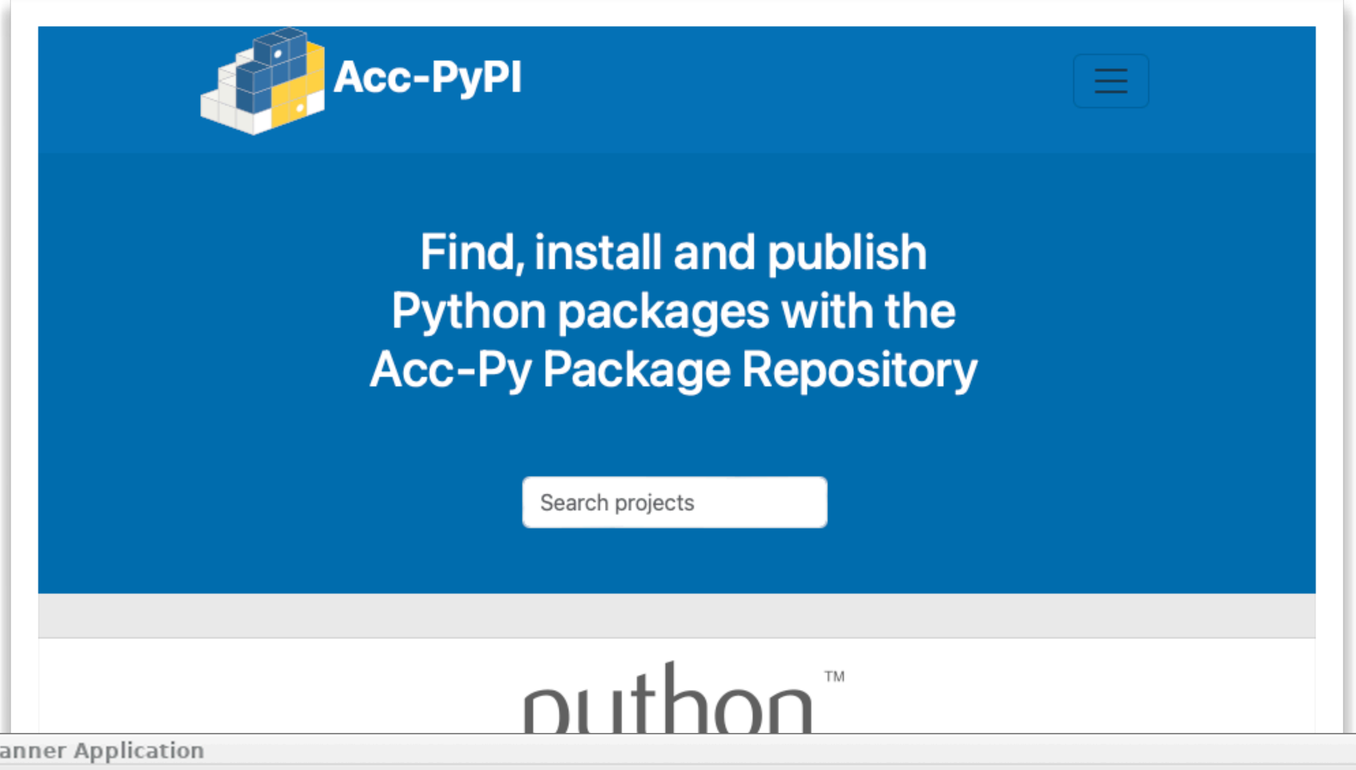
```
ts1 = datetime(2024, 2, 13, 8)
ts2 = datetime(2024, 2, 13, 8, 1)

B_data = DataQuery.builder(spark).byEntities()\
.system('CMW').startTime(ts1).endTime(ts2)\
.entity()\
.keyValue('device', 'PR.BMEAS-B-ST') \
.keyValue('property', 'CycleSamples').build()

df = B_data.toPandas()
plt.plot(df.iloc[0]['samples']['elements']/1E4)
plt.xlabel('Cycle time [ms]')
plt.ylabel('Magnetic field [T]')
```

```
Text(0, 0.5, 'Magnetic field [T]')
```



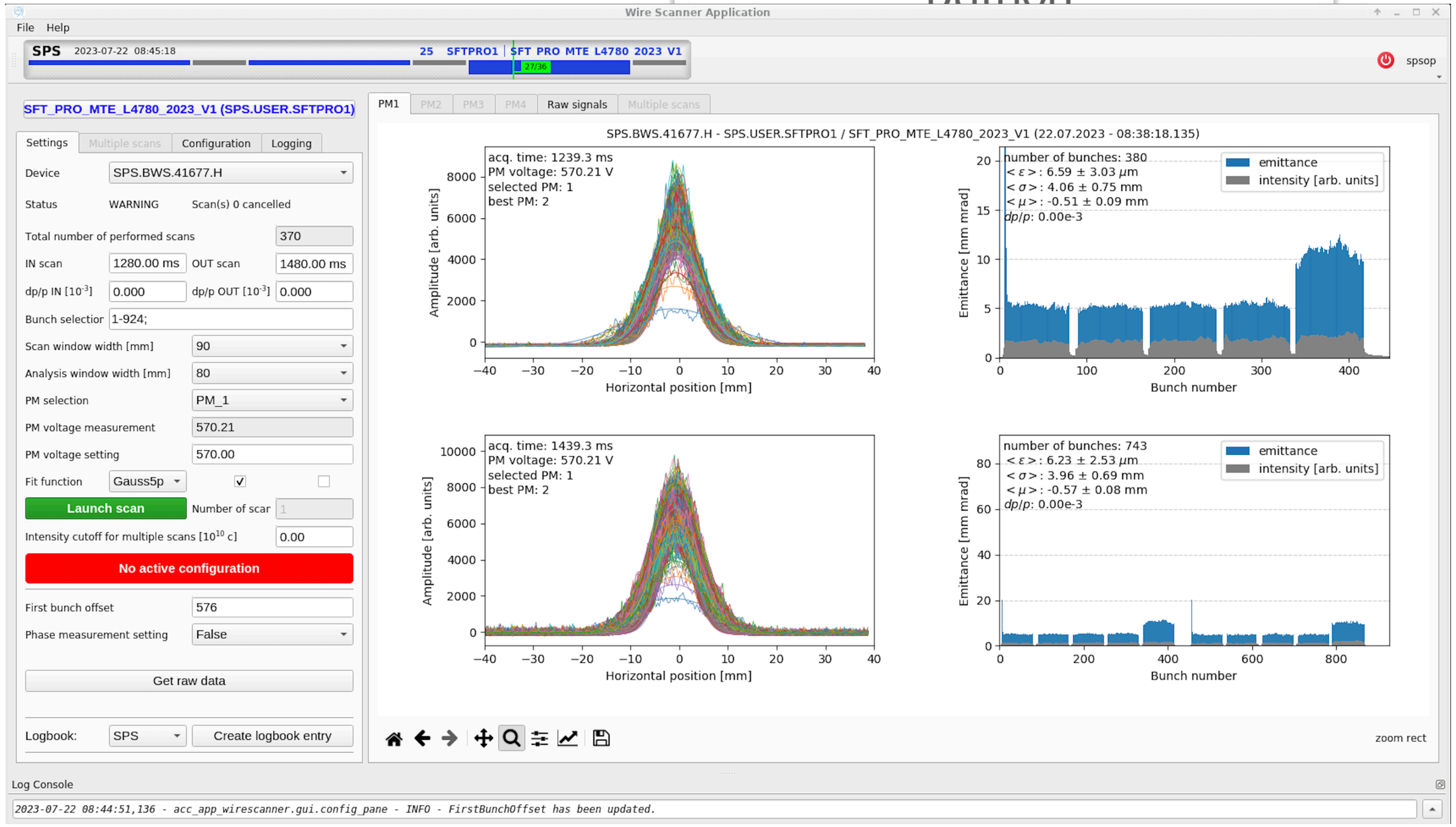
Acc-Py - Python for CERN accelerators

…with all the devOps one can dream of.



Python fully integrated with control system:

- Equipment access: **pyjapc**, pyda
  - ✳ next: event building
- access to settings (pjLSA) and archived data (**NXCALS**)
- GUI framework using **PyQt** and custom widgets
- and more…



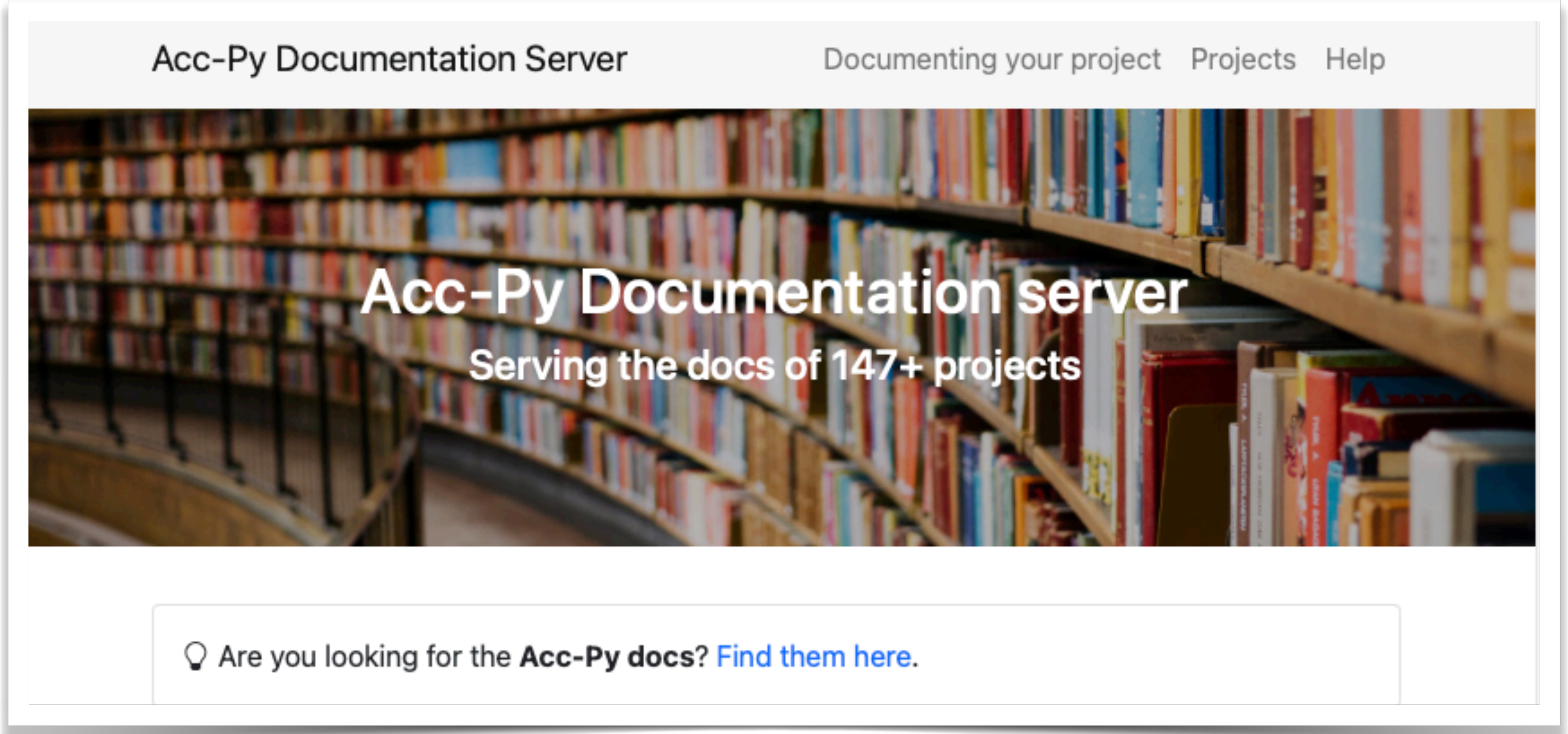
Acc-Py - Python for CERN accelerators

…with all the devOps one can dream of.

Python fully integrated with control system:

- Equipment access: **pyjapc**, pyda
  - next: event building
- access to settings (pjLSA) and archived data (**NXCALS**)
- GUI framework using **PyQt** and custom widgets
- and more…

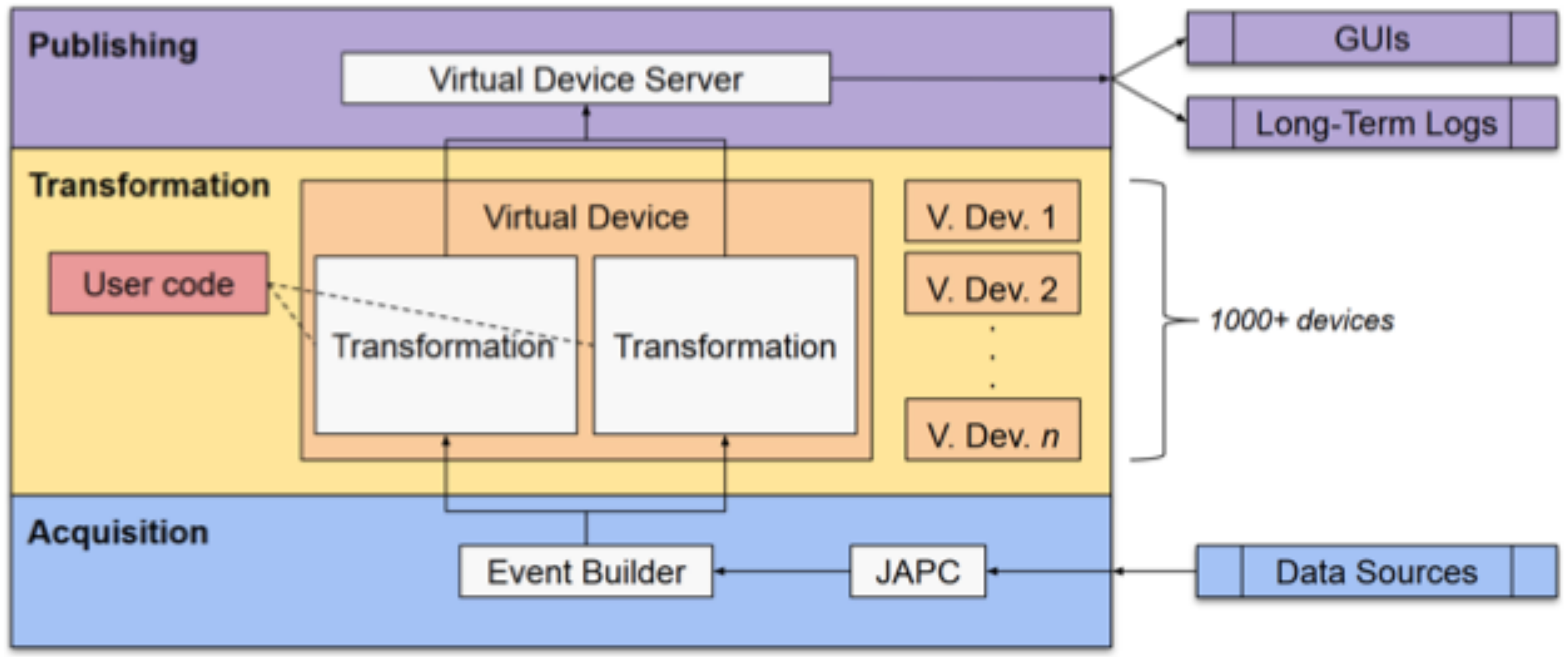
Also: Acc-Py package indexing and central deployment location



UCAP

Unified Controls Acquisition and Processing ("Virtual Device Service")

- servers on-the-fly in JAVA or Python for **online data transformations**
- Plug&Play framework for online analysis (UCAP transformations), i.e. continuously running in the background and triggered by "events"



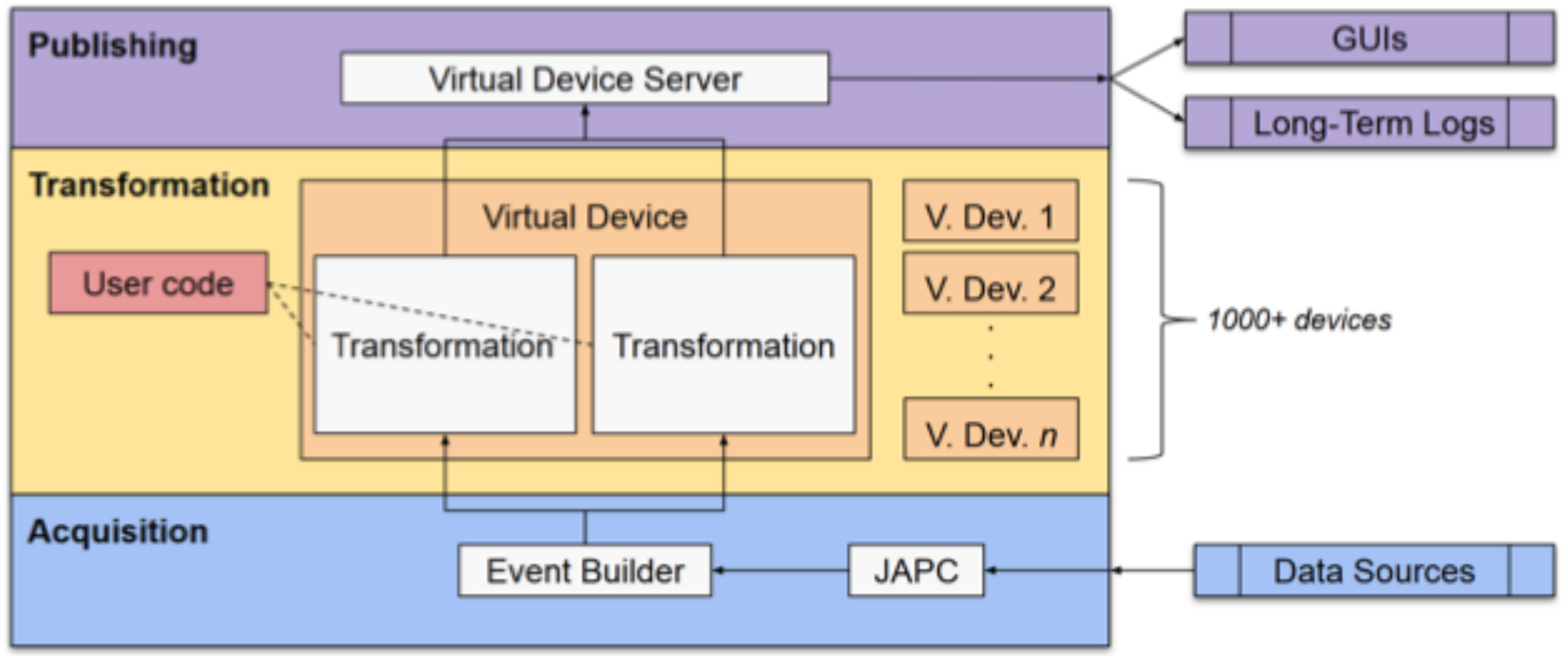
UCAP

Unified Controls Acquisition and Processing ("Virtual Device Service")

- servers on-the-fly in JAVA or Python for **online data transformations**
- Plug&Play framework for online analysis (UCAP transformations), i.e. continuously running in the background and triggered by "events"

Provides:

- tools for development, test and deployment of "transformations" (+actors)
- centrally managed infrastructure to run transformations (GPUs available)



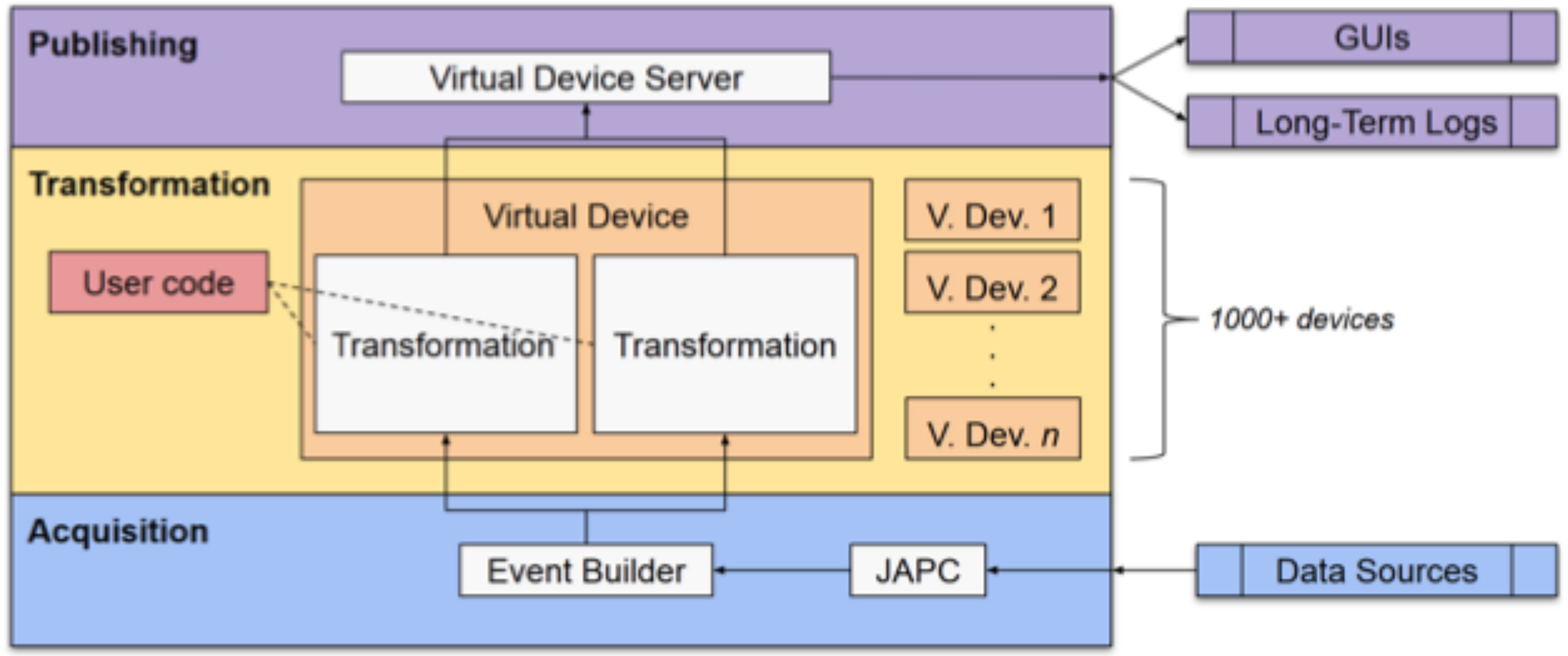
UCAP

Unified Controls Acquisition and Processing ("Virtual Device Service")

- servers on-the-fly in JAVA or Python for **online data transformations**
- Plug&Play framework for online analysis (UCAP transformations), i.e. continuously running in the background and triggered by "events"

Provides:

- tools for development, test and deployment of "transformations" (+actors)
- centrally managed infrastructure to run transformations (GPUs available)



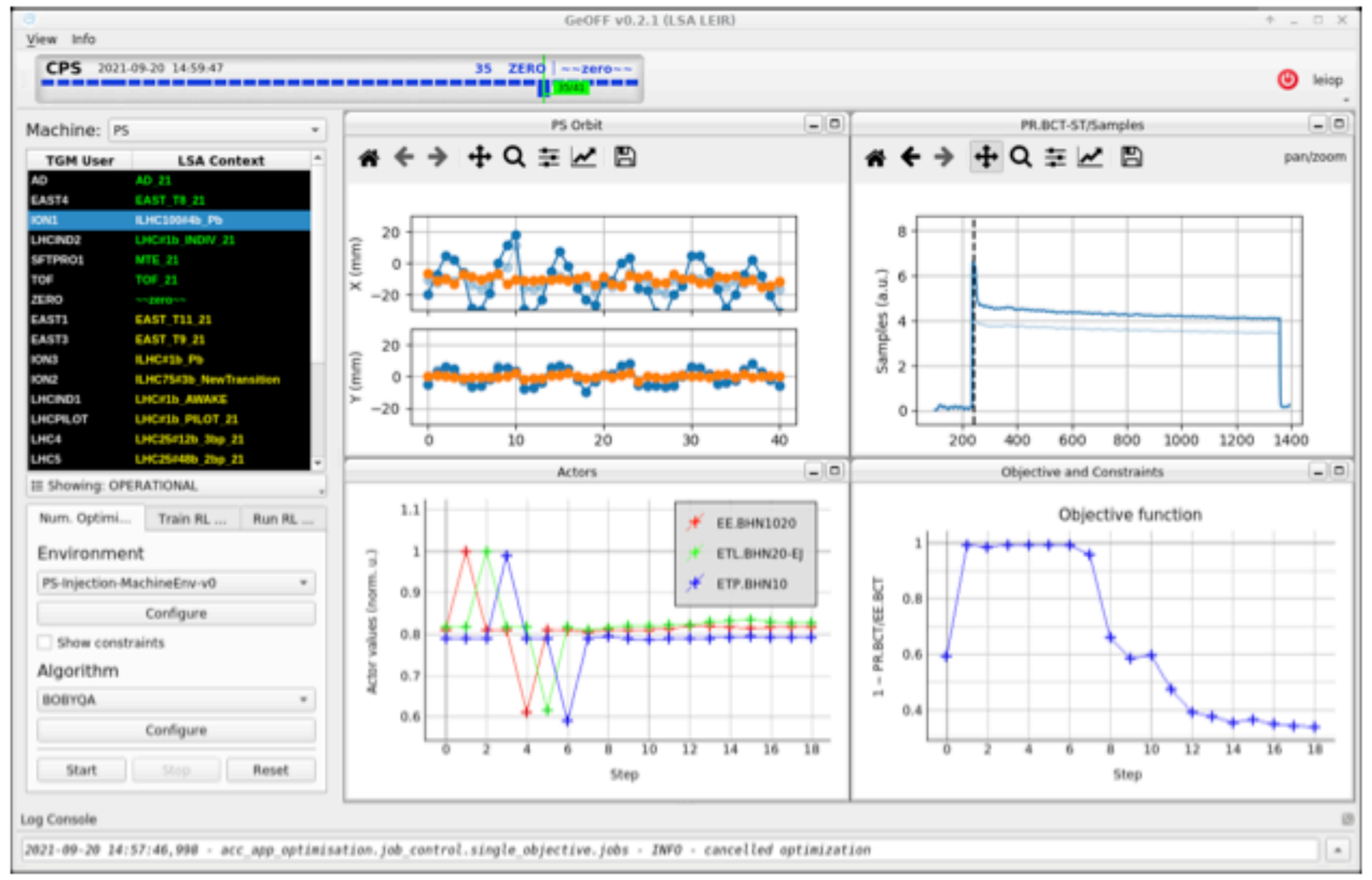
Key features:

- aggregating data: event building (e.g. buffered data,…)
- "transformation" result republished as online input for applications, controllers, logging, software interlock systems, …

Optimisation Infrastructure

Generic Optimisation Framework **GeOFF**

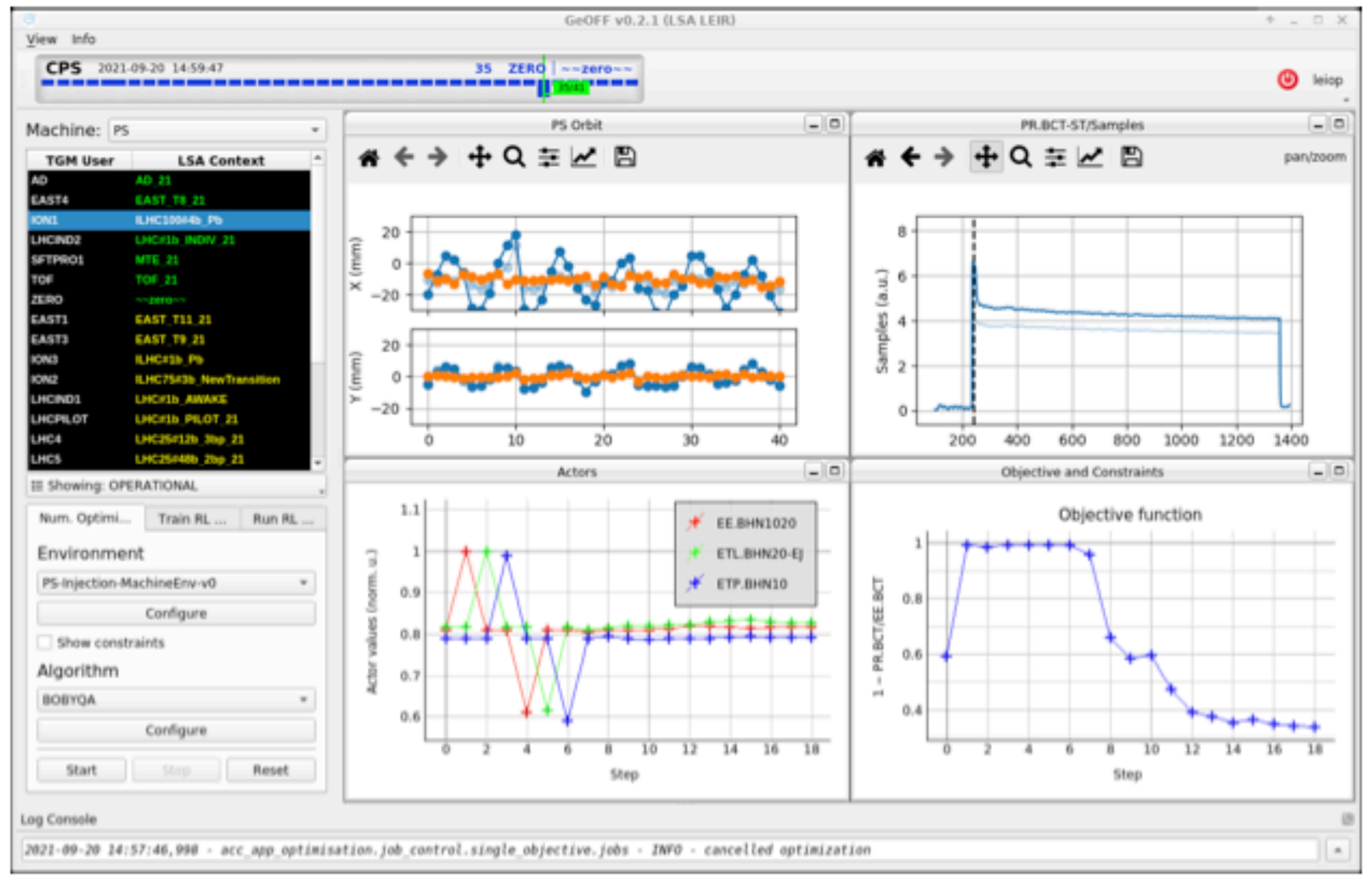
- **Replace manual grid scans** in the control room with **optimisation algorithms**
- GeOFF = flexible parameter optimisation (user-defined problem descriptions)
- To date: > 20 parameter optimisation problems automated across complex



Optimisation Infrastructure

Generic Optimisation Framework **GeOFF**

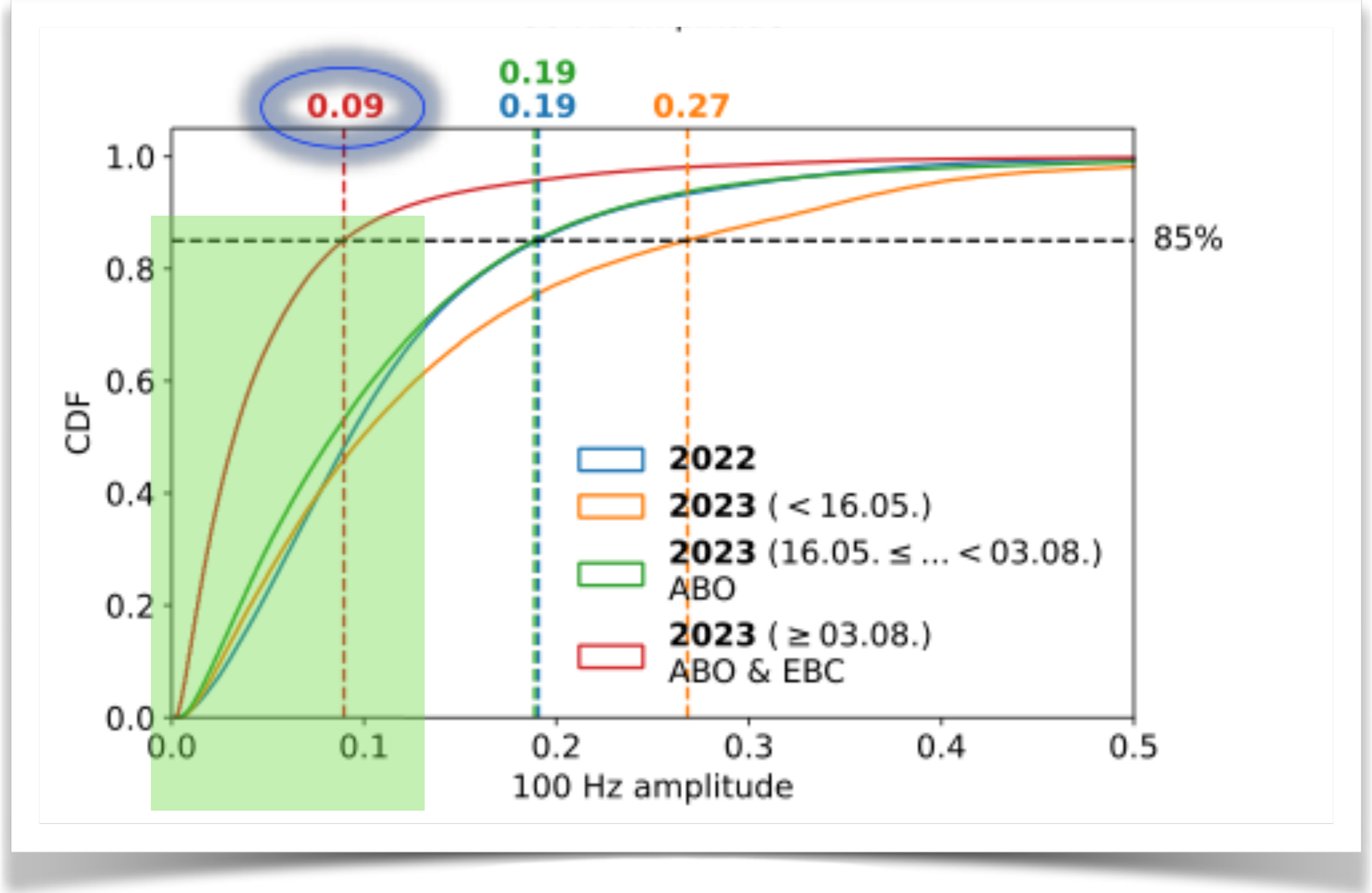
- **Replace manual grid scans** in the control room with **optimisation algorithms**
- GeOFF = flexible parameter optimisation (user-defined problem descriptions)
- To date: > 20 parameter optimisation problems automated across complex



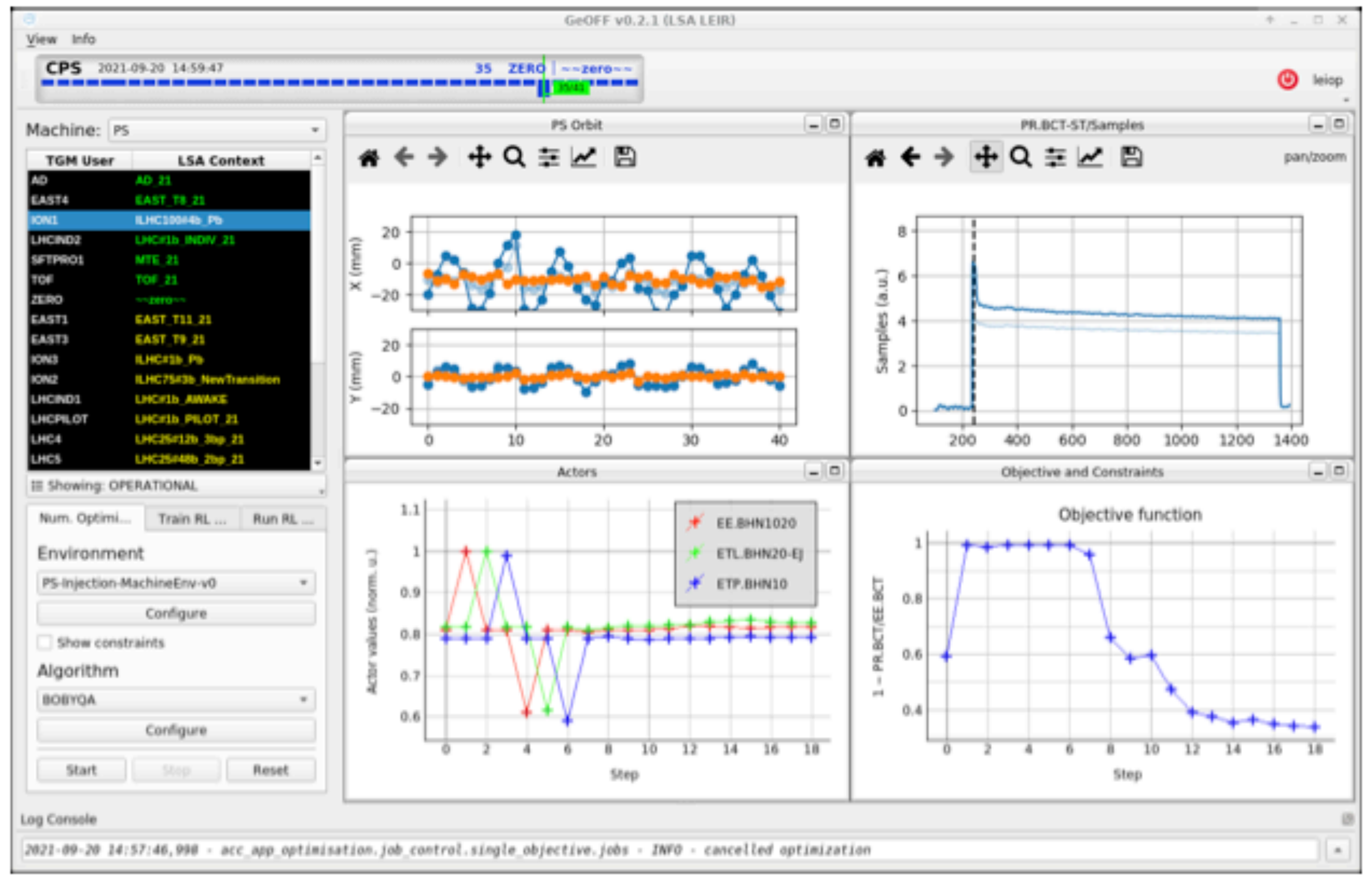
Optimisation framework for auto-pilots

- GeOFF on UCAP → **acc-geoff4ucap** released in summer 2023.
- **Operational**:
  - ✳ $n \times 50$ Hz control with Adaptive Bayesian Optimisation (ABO) for North Area spill with **GPUs on UCAP** (see Francesco's talk)

100 Hz content of NA spill with ABO and EBC



Optimisation Infrastructure



Generic Optimisation Framework **GeOFF**

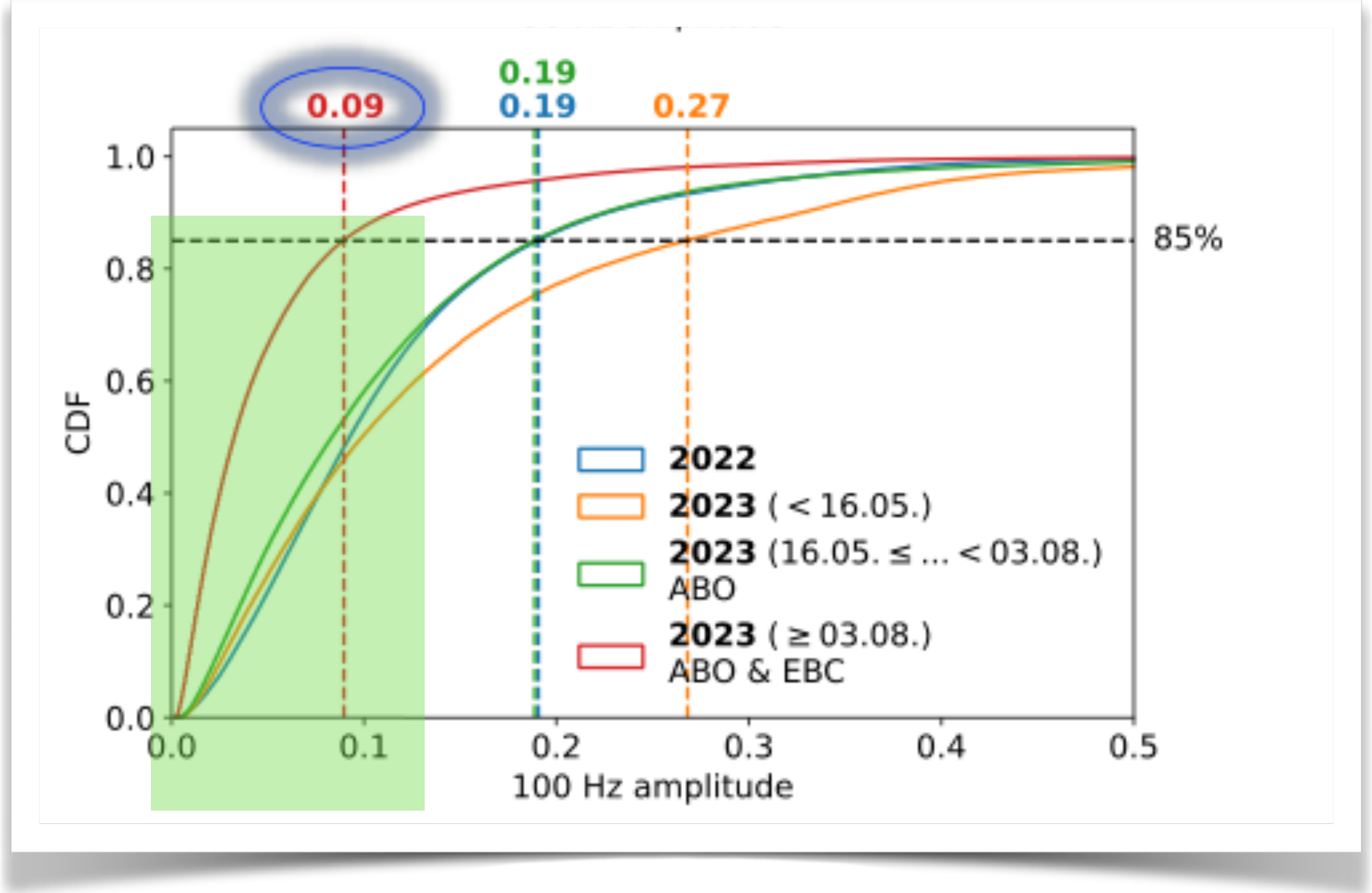
- **Replace manual grid scans** in the control room with **optimisation algorithms**
- GeOFF = flexible parameter optimisation (user-defined problem descriptions)
- To date: > 20 parameter optimisation problems automated across complex

Optimisation framework for auto-pilots

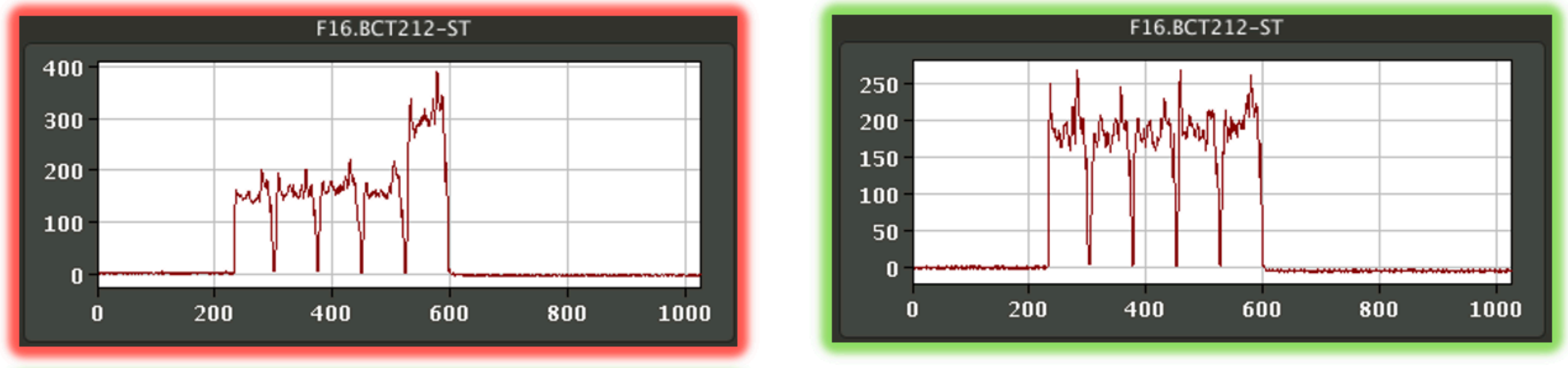
NEW

- GeOFF on UCAP → **acc-geoff4ucap** released in summer 2023.
- **Operational**:
  - $n \times 50$ Hz control with Adaptive Bayesian Optimisation (ABO) for North Area spill with **GPUs on UCAP** (see Francesco's talk)
- To come in **2024**:
  - automated transfer line **trajectory steering**
  - Multi-turn extraction (MTE) efficiency **drift stabilisation**

100 Hz content of NA spill with ABO and EBC



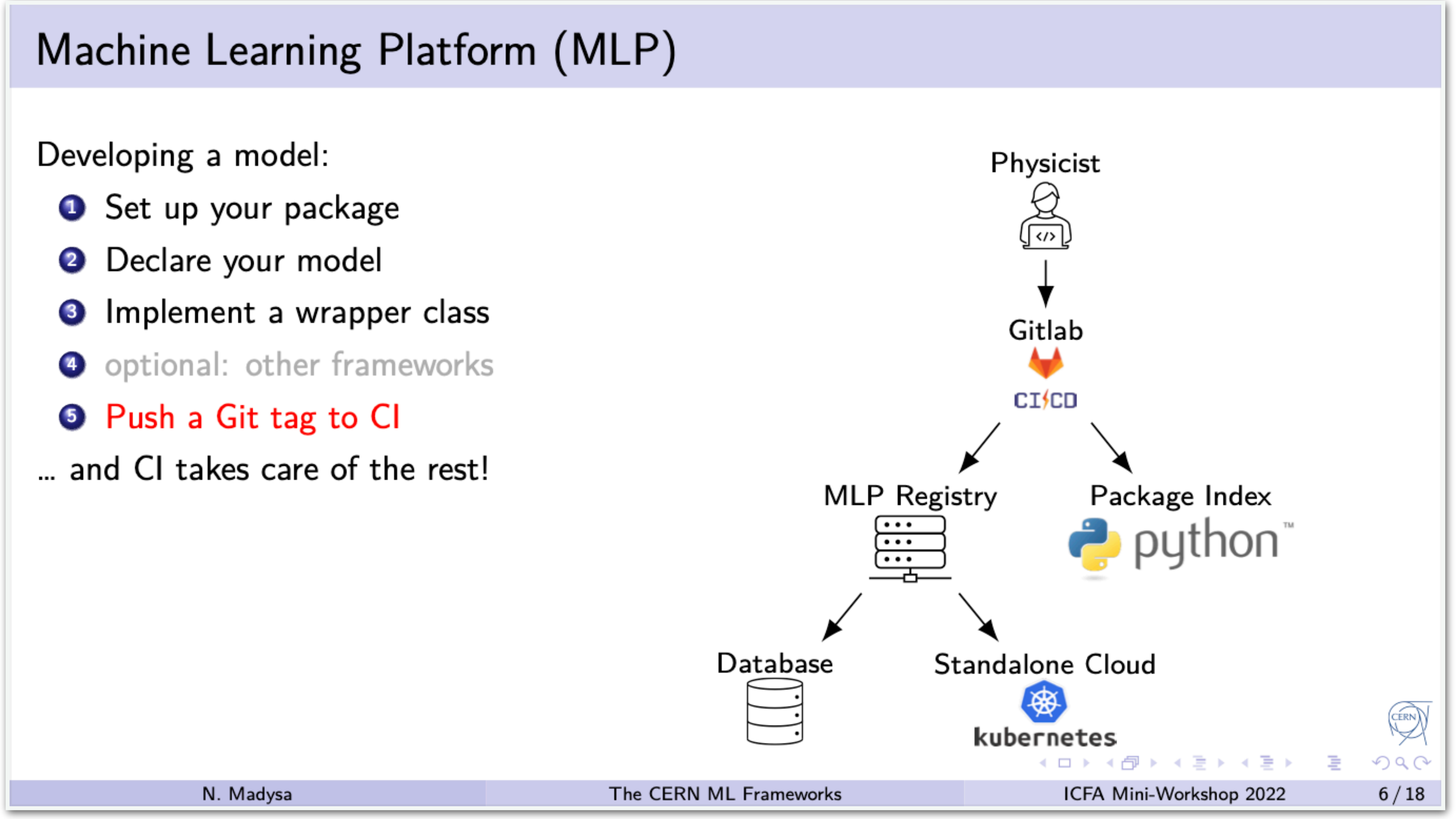
5-turn MTE spill: equal intensity sharing desired

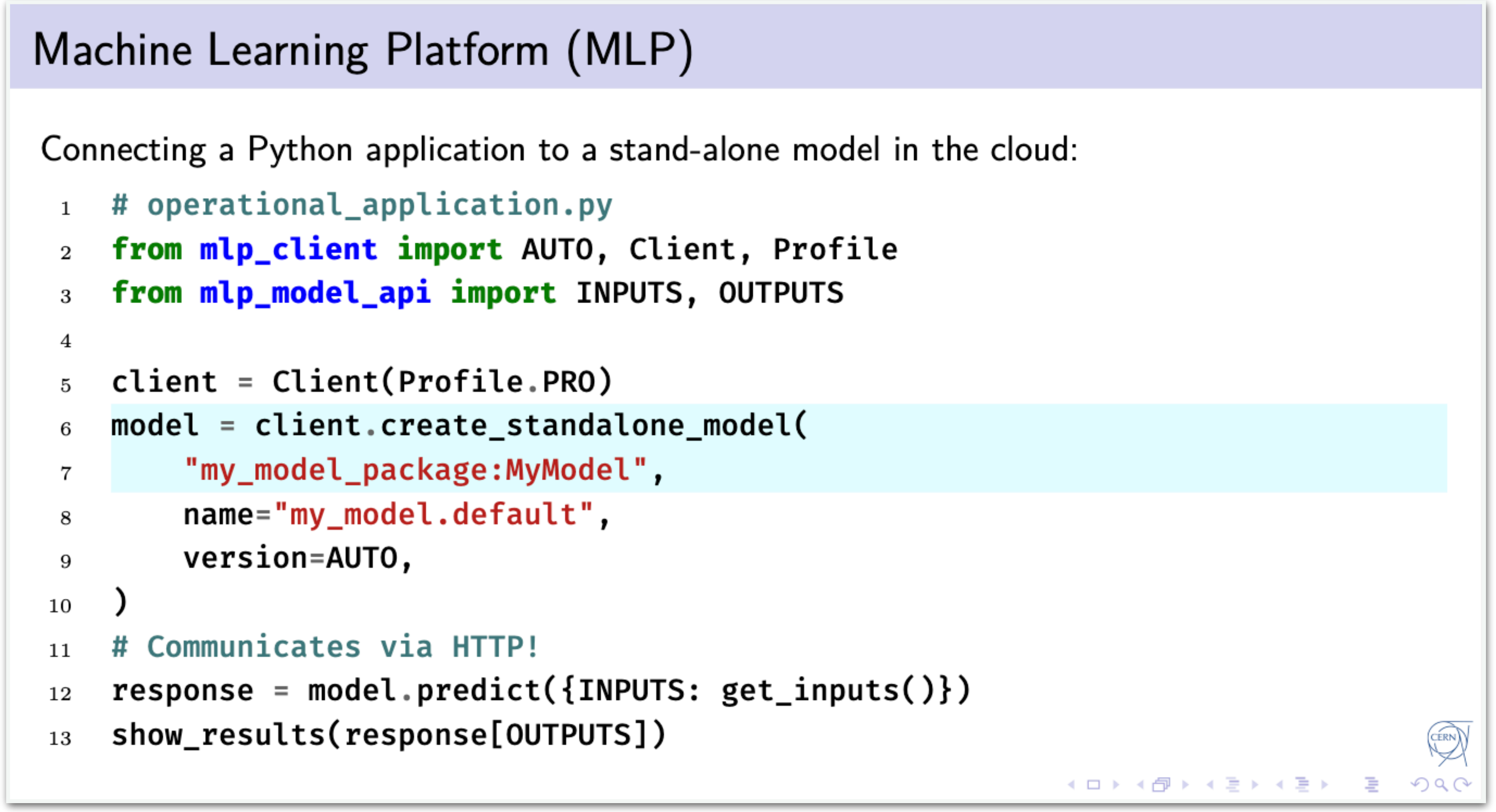


Machine Learning Platform (MLP)

Storing and sharing trained ML models (e.g. ANNs) between users and applications of different languages.

- Stand-alone models hosted in the CERN cloud, ready for inference.



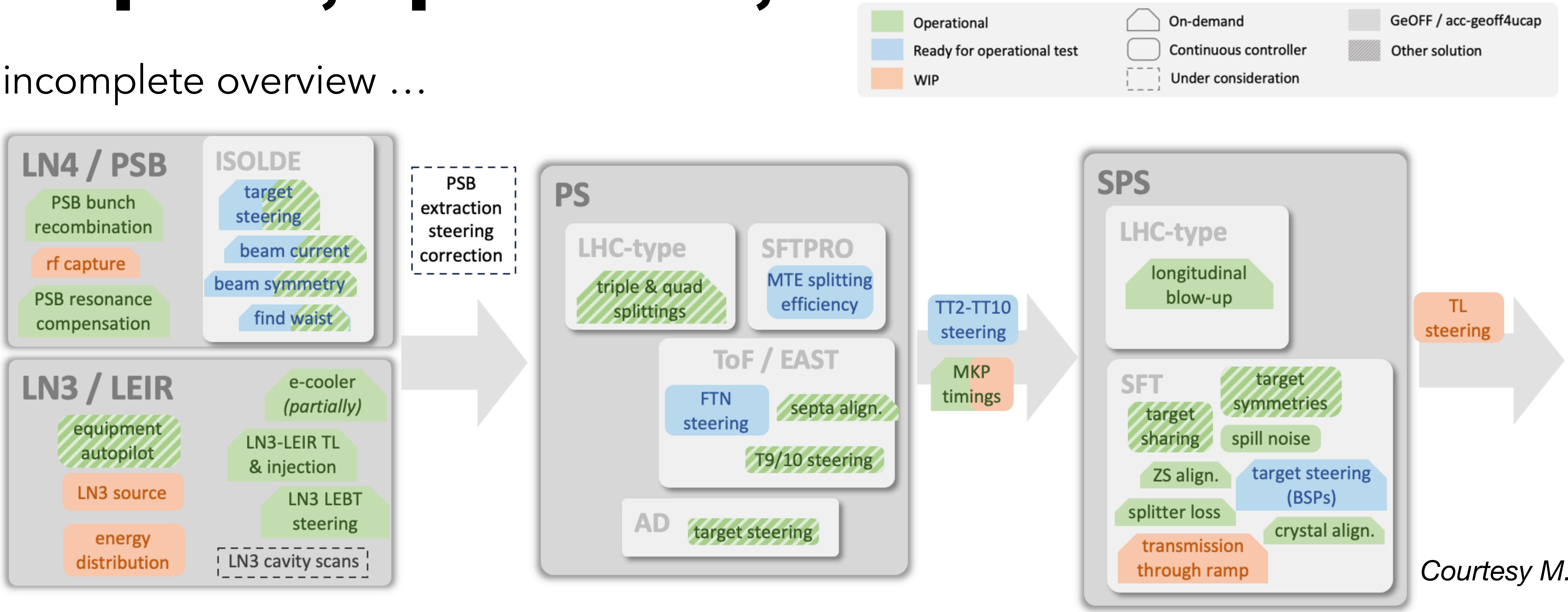


Successfully used for deploying AccGPT

- challenge of very large Llama 2 model

Auto-pilots, optimisers,...

An incomplete overview …

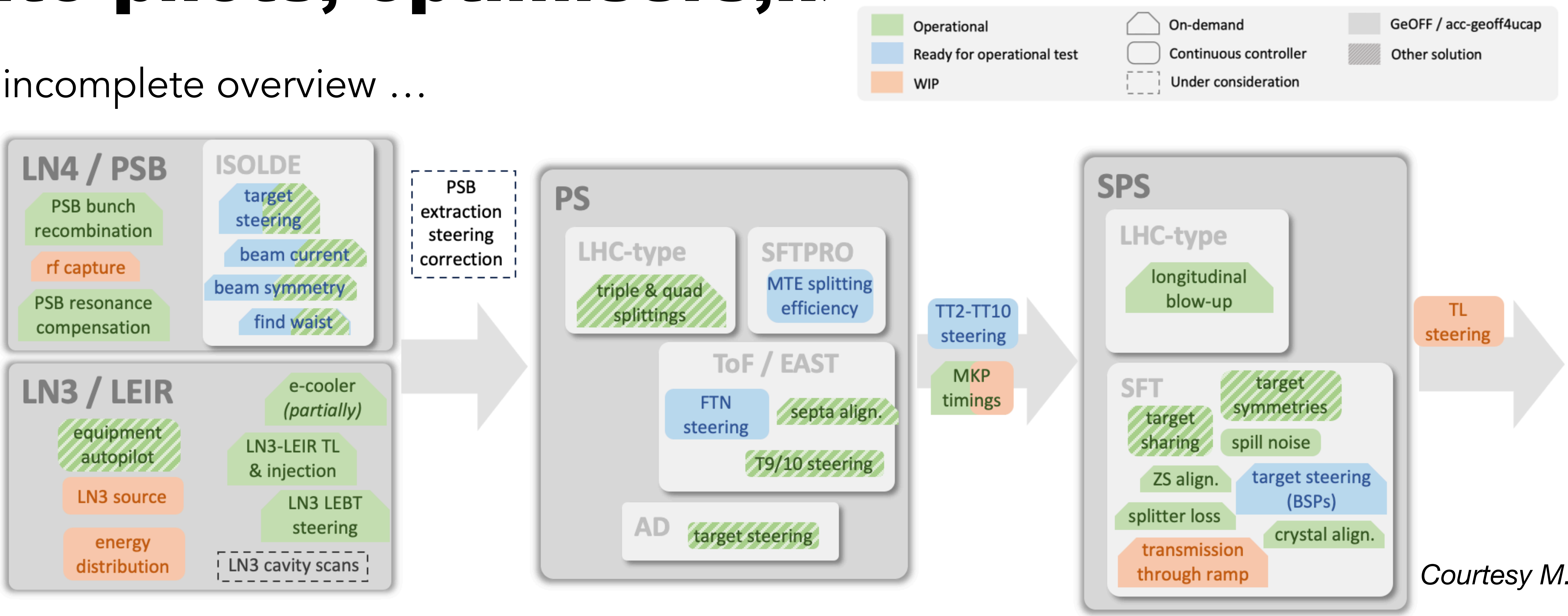


Courtesy M. Schenk

Status 2023: multiple auto-pilots / optimisers used operationally and by experts during commissioning

Auto-pilots, optimisers,...

An incomplete overview ...



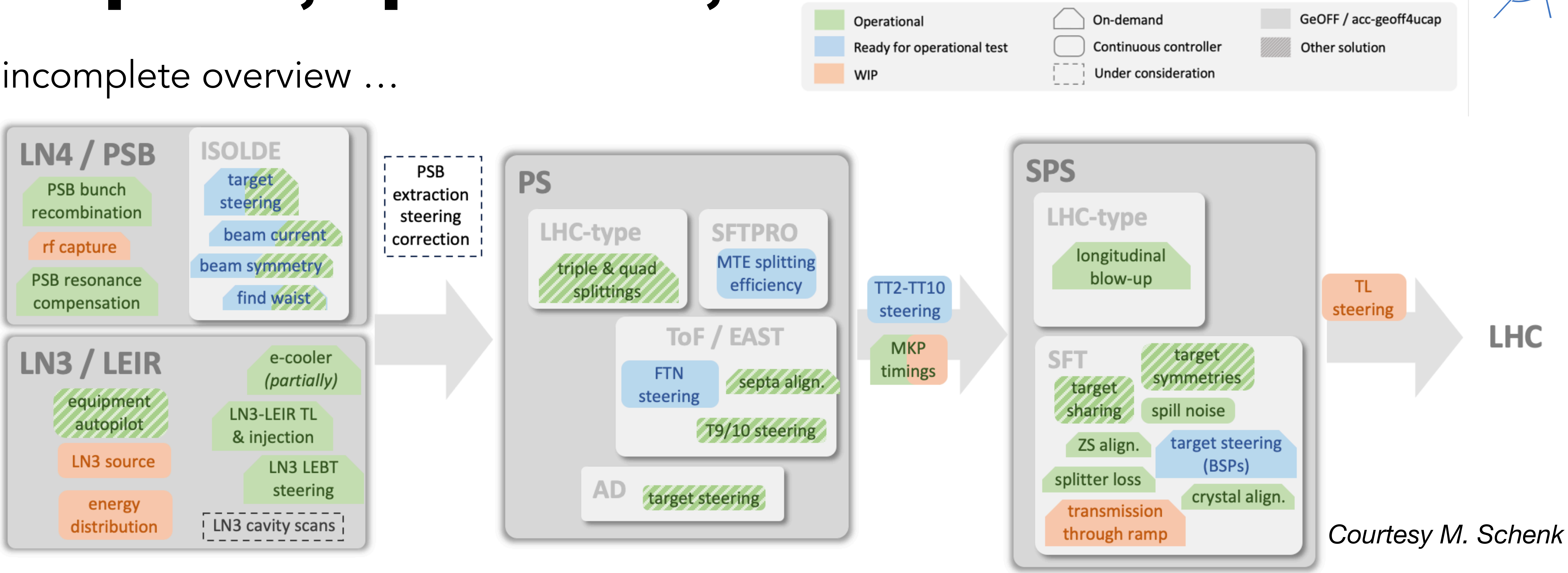
Courtesy M. Schenk

Status 2023: multiple auto-pilots / optimisers used operationally and by experts during commissioning

Trends 2024: move on-demand to acc-geoff4ucap and implement some new auto-pilots

Auto-pilots, optimisers,…

An incomplete overview …



Courtesy M. Schenk

Status 2023: multiple auto-pilots / optimisers used operationally and by experts during commissioning

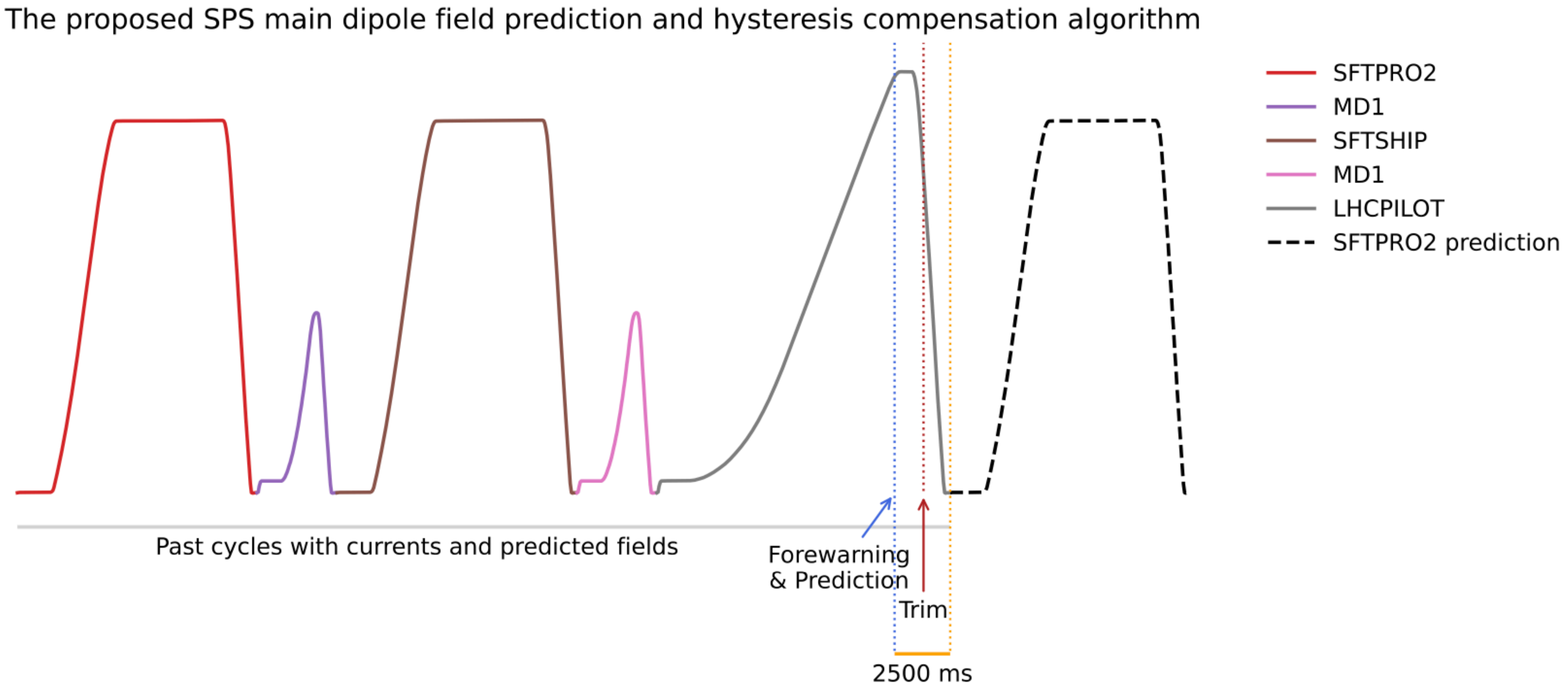
Trends 2024: move on-demand to acc-geoff4ucap and implement some new auto-pilots

Goal until end of run 3: automation of all typical optimisation and continuous control problems

Predicting magnetic hysteresis and eddy current effects

Training NN to predict future field and set correction $\hat{\boldsymbol{B}}_{t+1} = \boldsymbol{NN}(\boldsymbol{I}_{t,t-1,\ldots}, \hat{\boldsymbol{B}}_{t,t-1,\ldots})$

- Cycling the SPS is very expensive —> potentially game-changing
- **feedforward correction** triggered before every cycle



Courtesy A. Lu

Predicting magnetic hysteresis and eddy current effects

Training NN to predict future field and set correction $\boldsymbol{\hat{B}_{t+1} = NN(I_{t,t-1,\dots}, \hat{B}_{t,t-1,\dots})}$

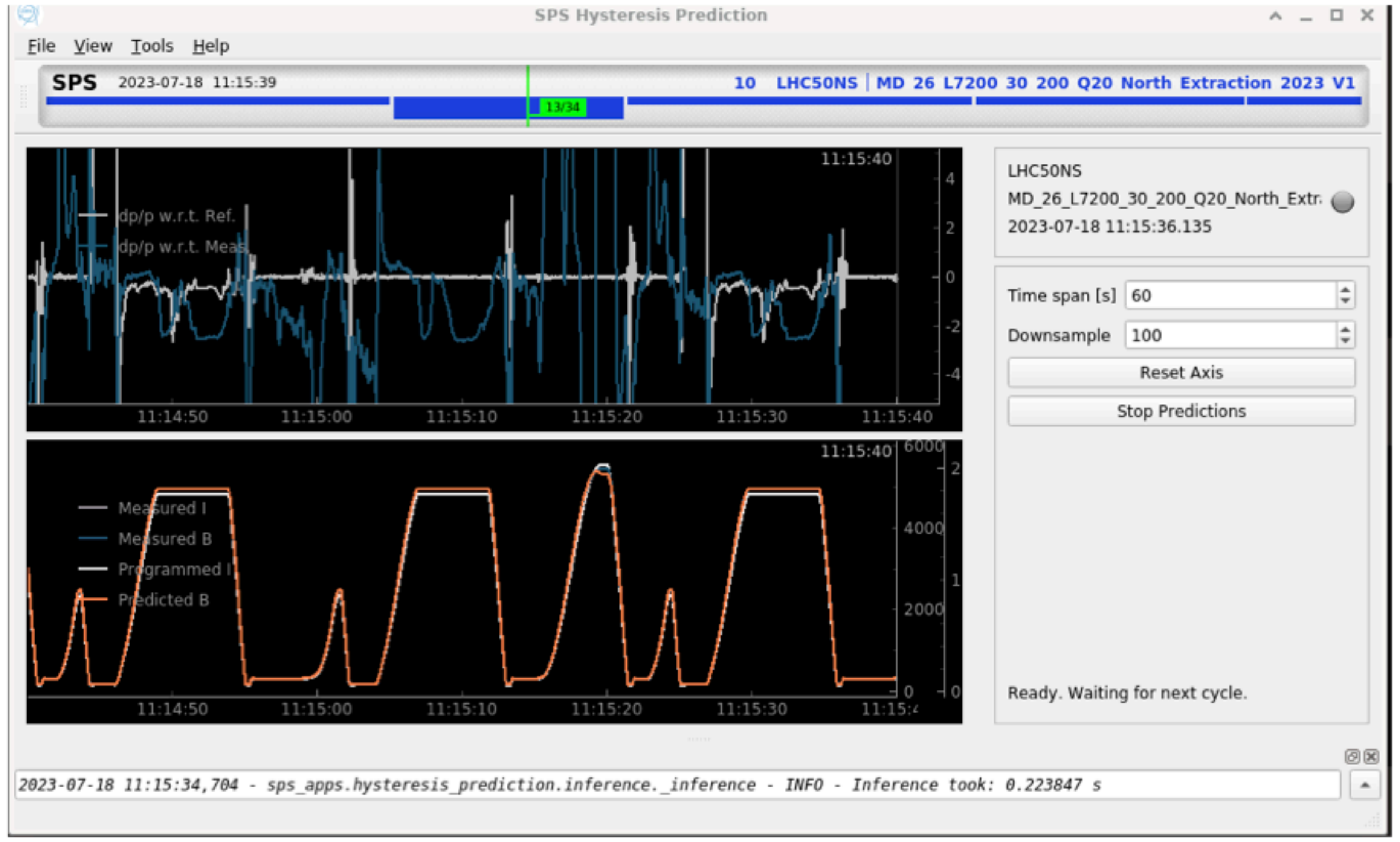
- Cycling the SPS is very expensive —> potentially game-changing
- **feedforward correction** triggered before every cycle

First operational experience

- first results using physics-inspired NN PhyLSTM $\ddot{B} + g = \Gamma\, i(t), \qquad g = \left[b(B,\dot{B}) + r\left(B,\dot{B},B(\tau)\right)\right]/a$
- **accuracy not yet sufficient**

Next:

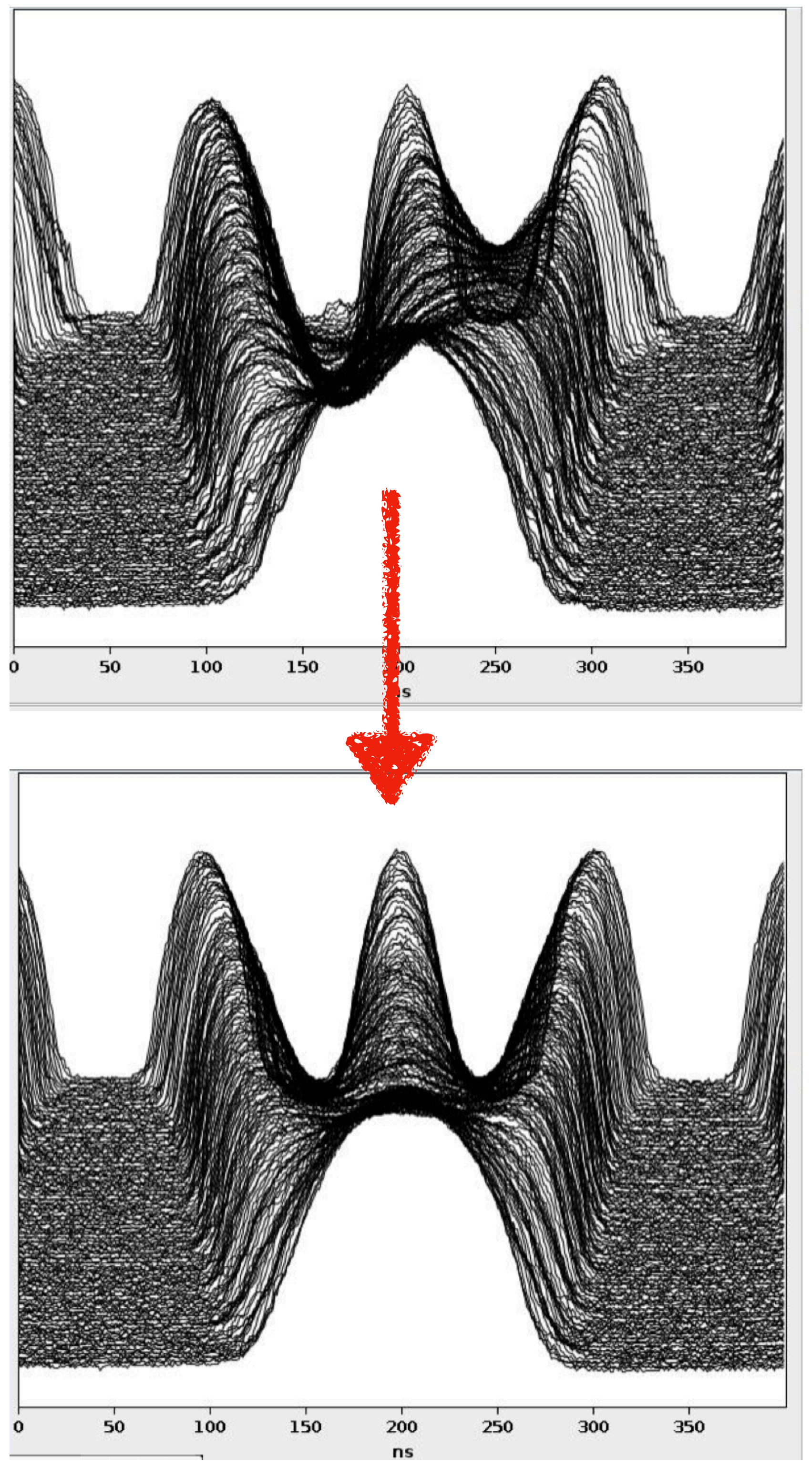
- Investigate usage of **transformers**
- Collect **more data** of different super cycle configurations



Courtesy A. Lu

RL in the control room

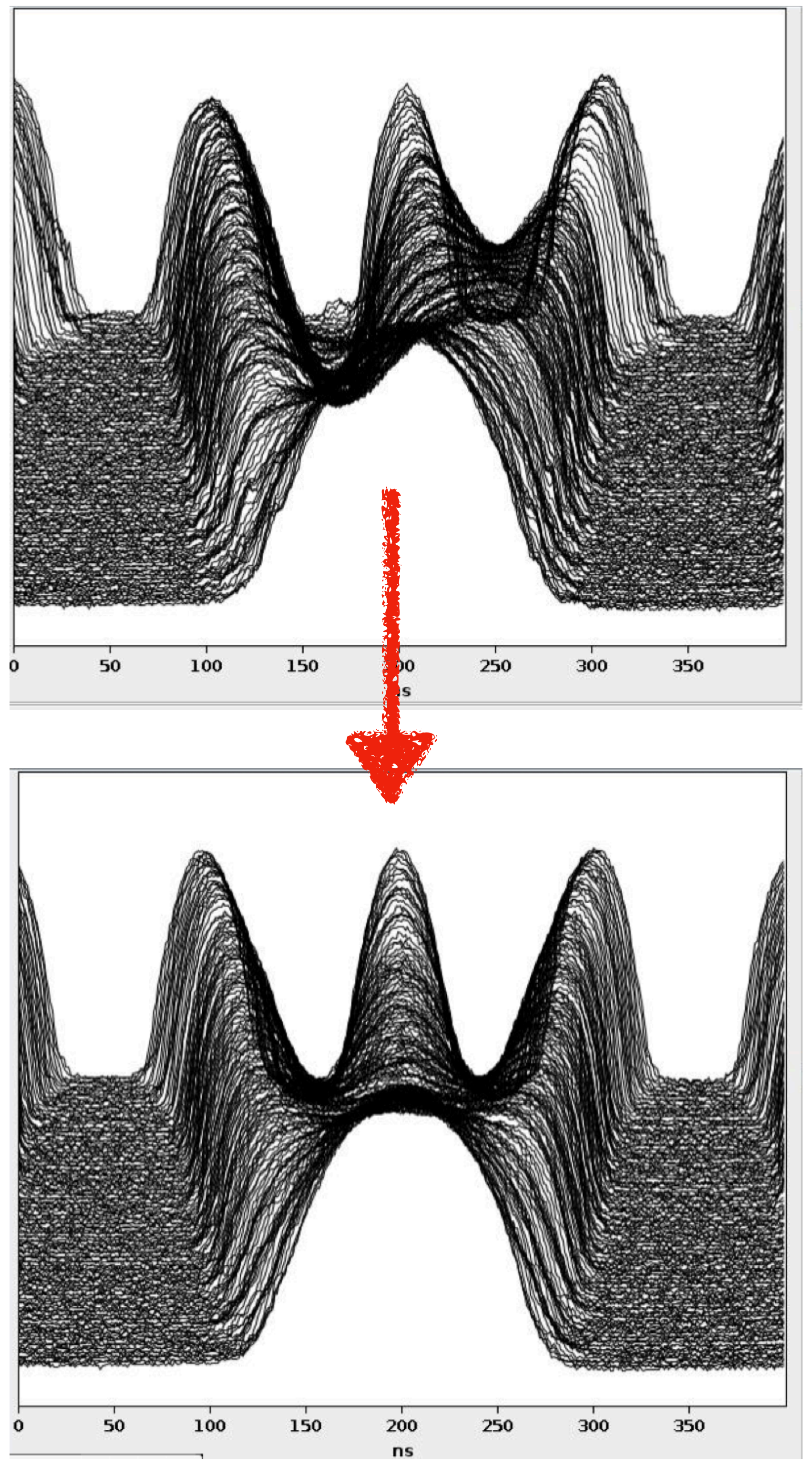
RL agent to correct **RF phase and voltage** to produce uniform RF splitting in PS for LHC beams

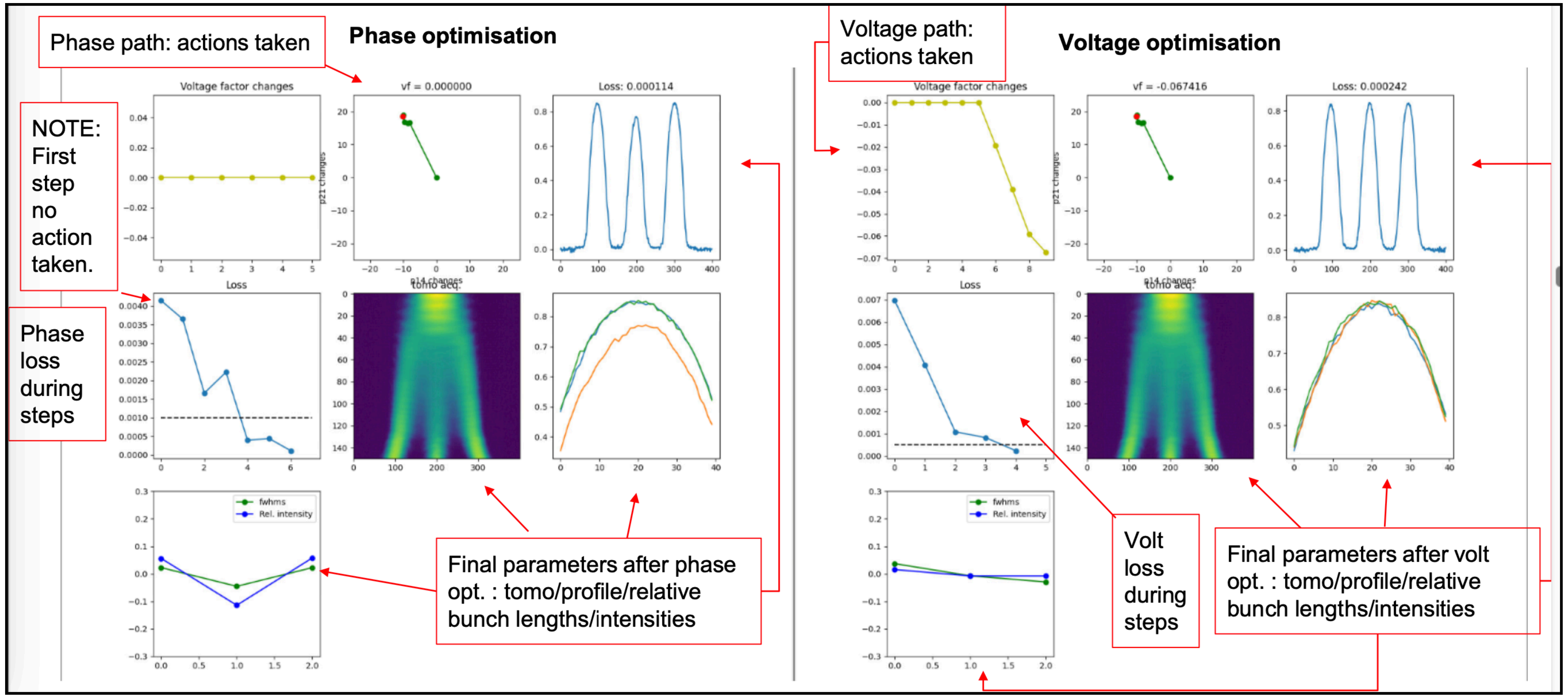


RL in the control room

RL agent to correct **RF phase and voltage** to produce uniform RF splitting in PS for LHC beams

- Trained on simulation and successfully transferred to the controls room —> **fully operational**
- RL algorithm: Soft Actor-Critic (SAC); multi-agent algorithm using CNN to define initial set point
- Ongoing: from on-demand to continuous on UCAP

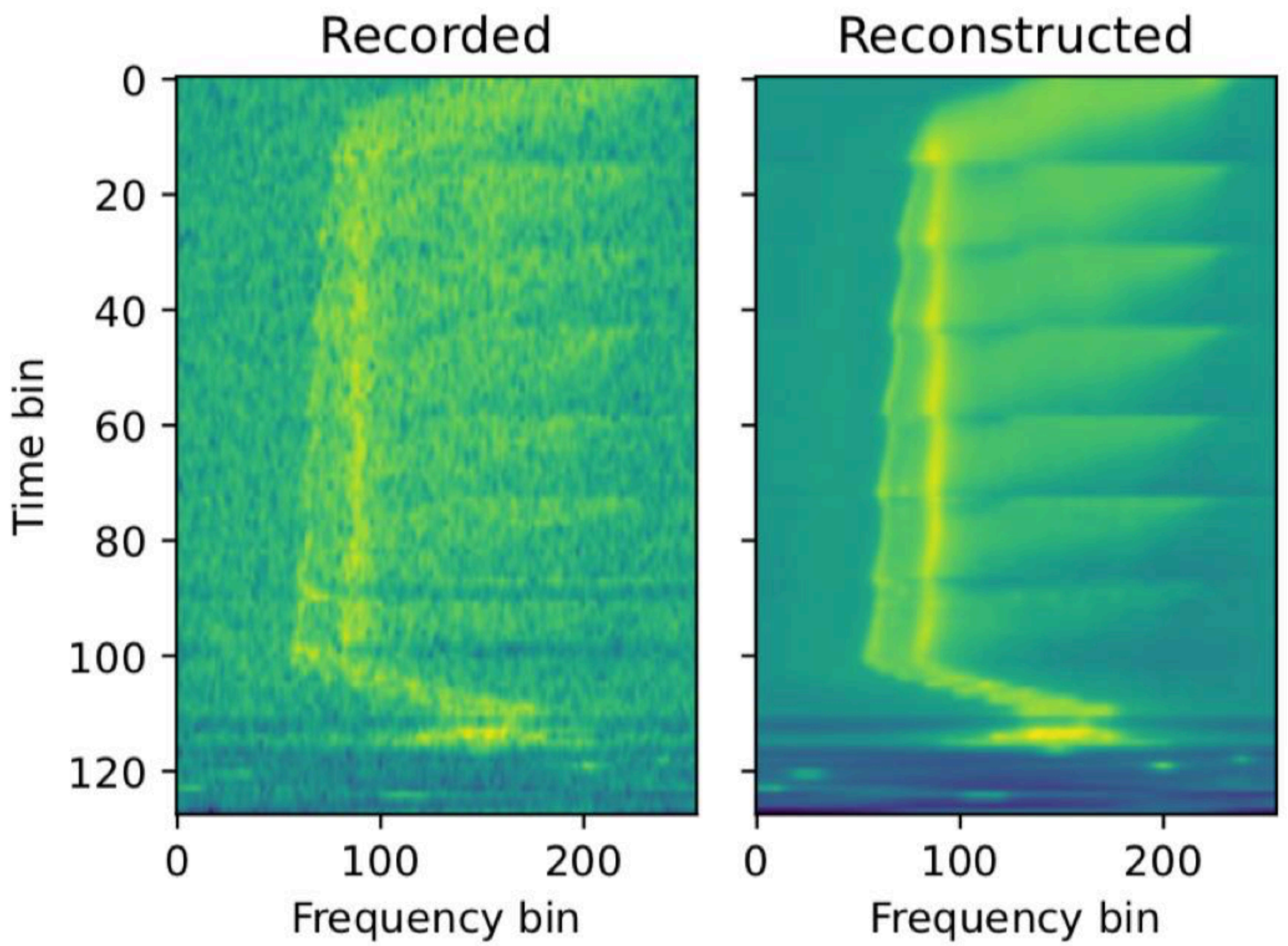




RL in the control room

WIP: PhD ongoing to optimise injection efficiency into LEIR

- Control of ramping and debunching cavities in LINAC3
- Observation based on **longitudinal Schottky spectrum**
- Trained on data-driven dynamics model

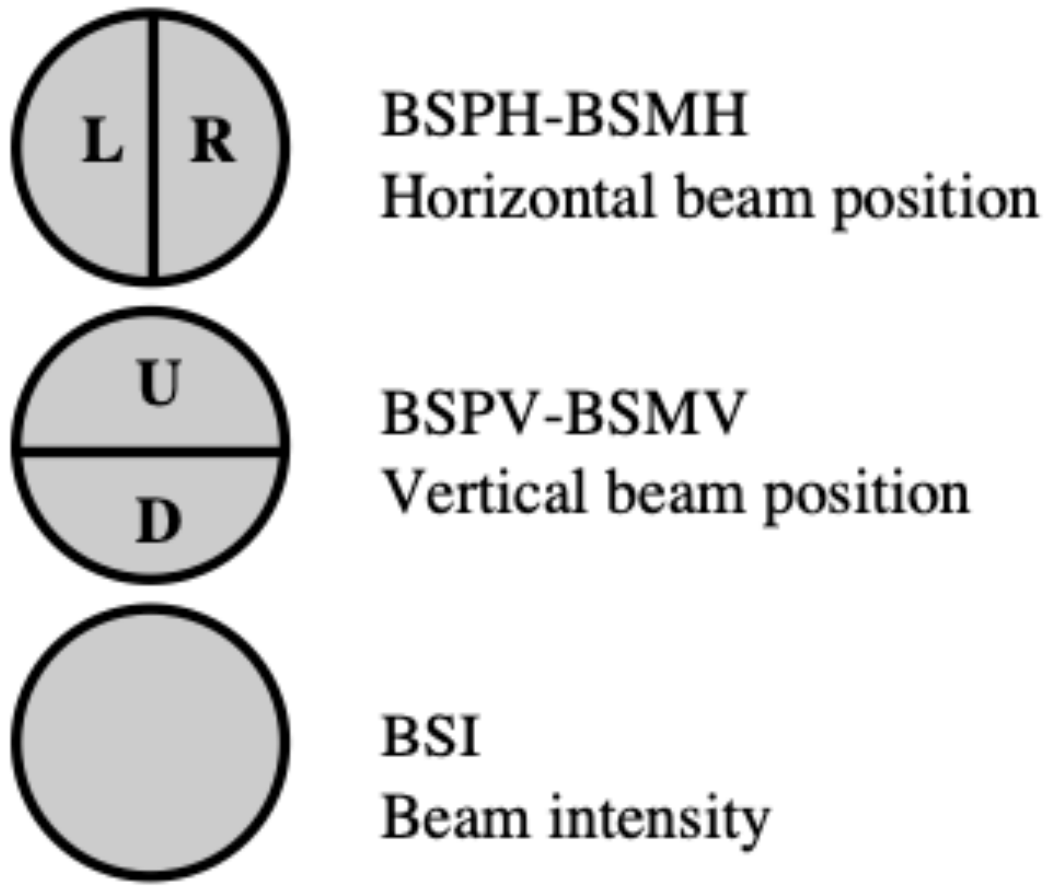


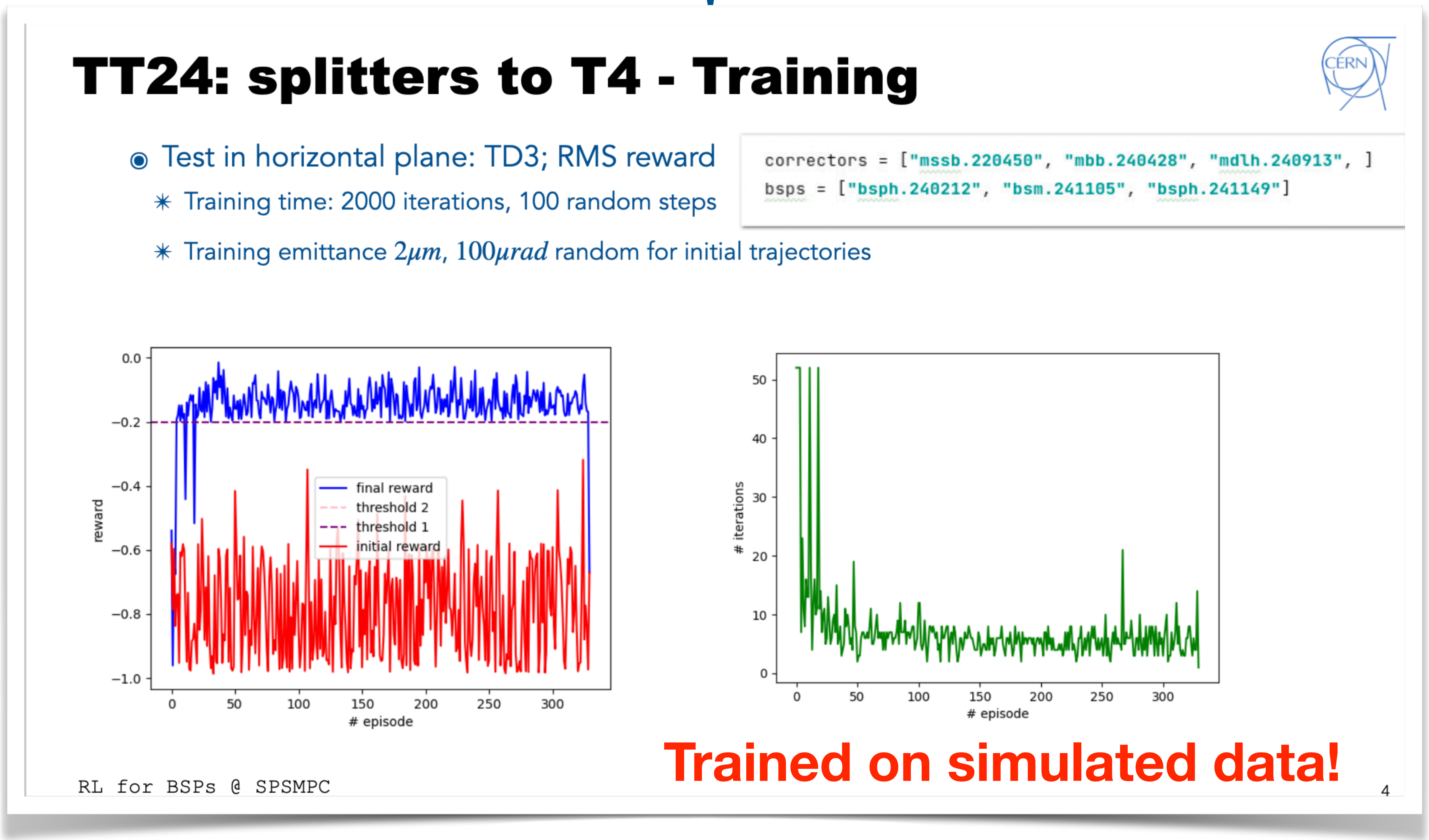
Schottky spectrum from training data set (left) and its reconstruction by an variational auto-encoder(right)

RL in the control room

Steering of DC beams in the CERN TT20 transfer line using split-foil secondary emission monitors (BSPs)

- RL state $\vec{s}$: $[(I_1 - I_2)_i]$ for each monitor; all intensities are normalised.
- Our metric: symmetries per monitor: $S = \sqrt{1 - \frac{|I_1 - I_2|}{I_1 + I_2}}$; Goal: S > 0.8





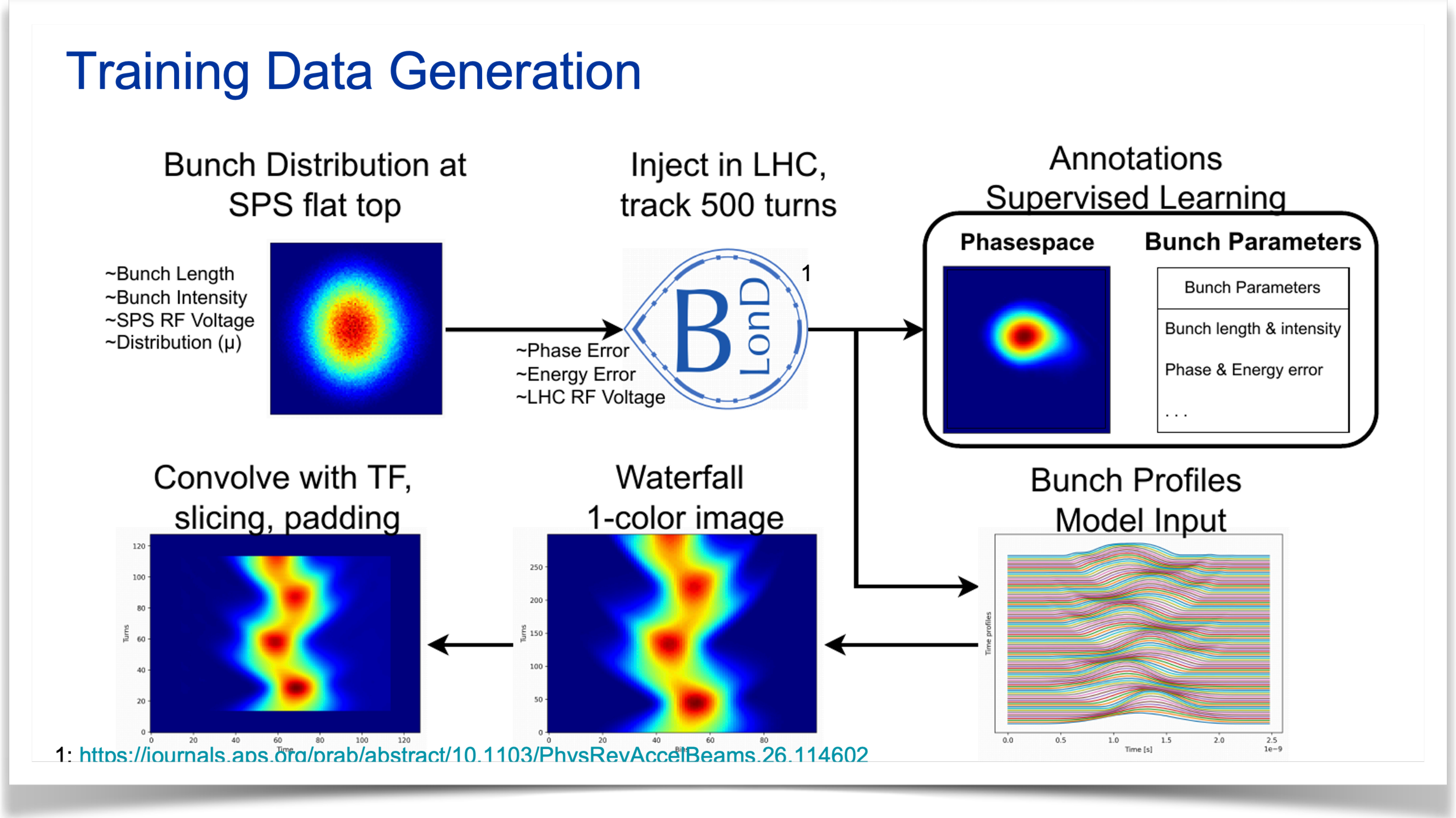
Test of transfer foreseen for 2024 startup.

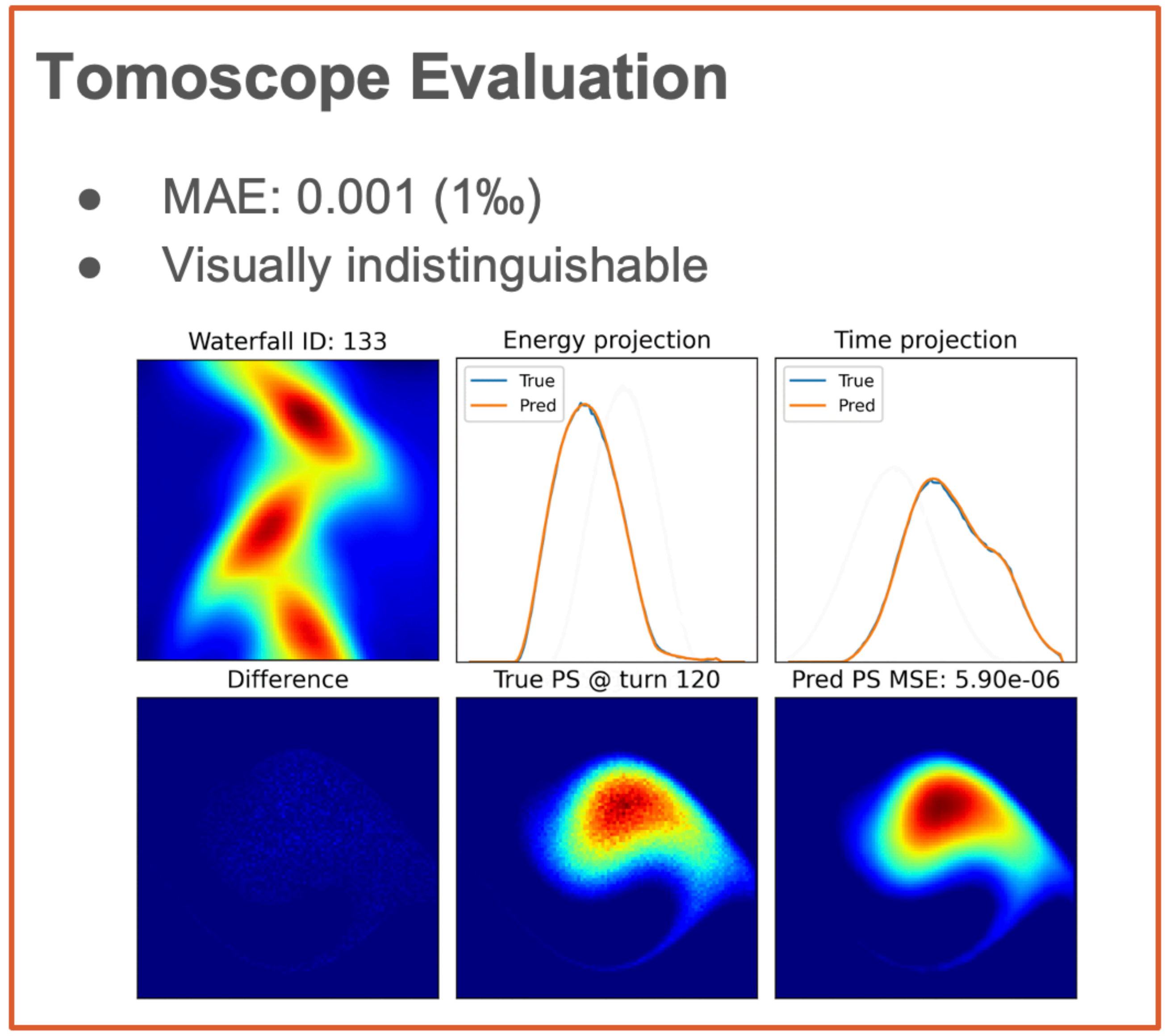
Enhancing diagnostics - ML tomoscope in the LHC

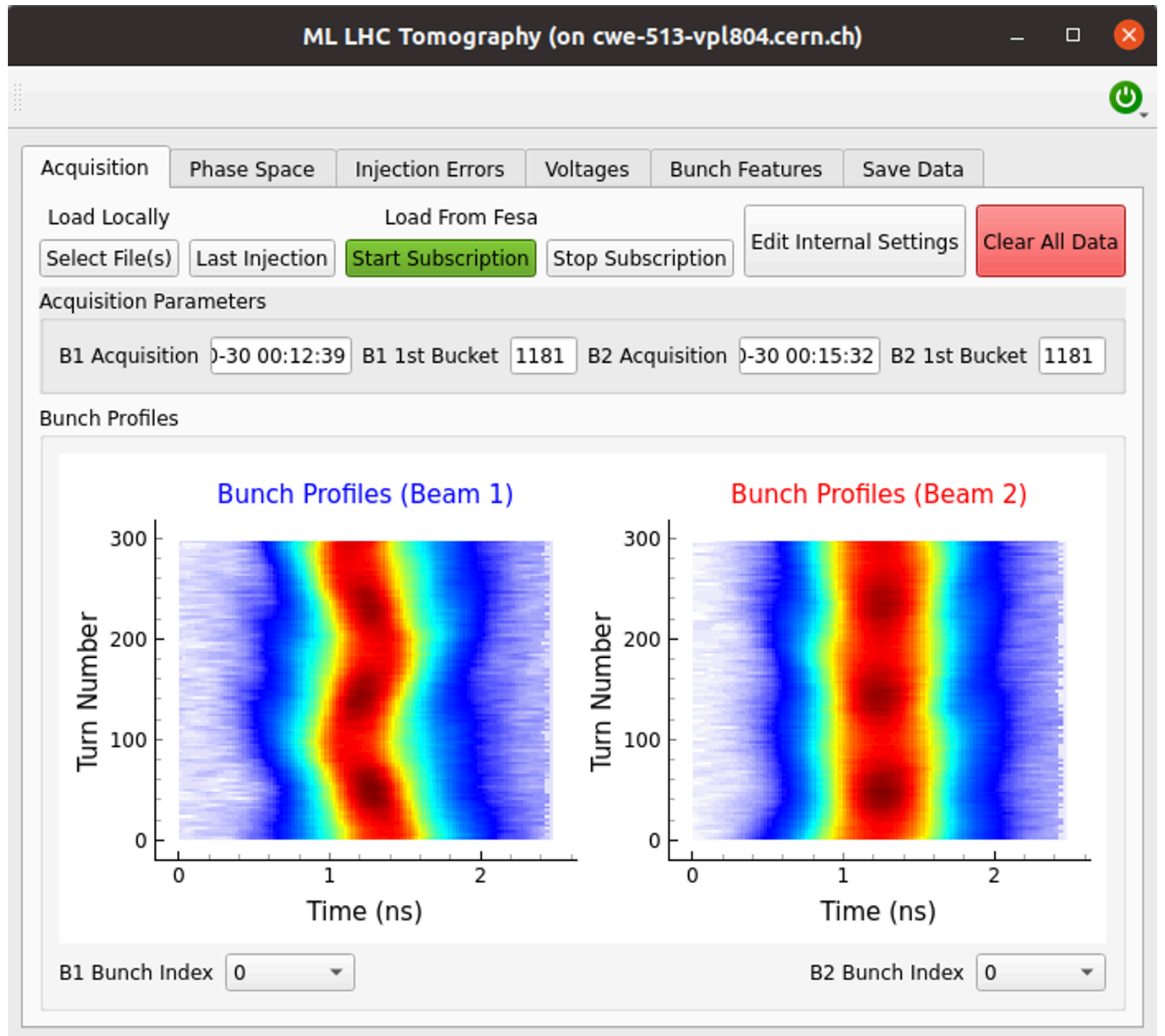
Speeding up tomographic bunch-by-bunch reconstruction in the LHC: using Auto-encoder ensemble: without AI bunch-by-bunch **not** possible.

- Fully operational on MLP
- Ready to be used in the LHC control room to measure **injection errors**

Trained with simulator







Courtesy K. Iliakis, T. Argyropoulos

Next steps: Efficient Particle Accelerators (EPA) project

Providing a body for **synergetic and streamlined development**

10 work packages with interdependencies: work packages addressing **technical topics** and **controls infrastructure evolution**.

Each WP is high-level deliverable.

WP0
Project Management
Lead: V. Kain
Deputy: A. Huschauer

WP1
Dynamic Beam Scheduling
Lead: F. Irannejad

WP2
Automated LHC Filling
Lead: A. Huschauer, G. Trad

WP3
Automated Parameter Control & Optimisation
Lead: M. Schenk

WP4
Hysteresis Compensation
Lead: C. Petrone

WP5
Next Generation Sequencer
Lead: R. Gorbonosov

WP6
Efficient Settings Management
Lead: M. Hostettler

WP7
Automated Equipment Testing
Lead: A. Calia, J.C. Garnier

WP8
Automate Equipment
Lead: F. Velotti, K. Papastergiou

WP9
Data Processing Framework
Lead: M. Sobieszek

EPA goals

Focus is on automation of accelerator operation and equipment management

- to increase efficiency, reproducibility, flexibility and hence performance

EPA goals

Focus is on automation of accelerator operation and equipment management

- to increase efficiency, reproducibility, flexibility and hence performance

WP1 **Dynamic Beam Scheduling:** **remove fixed supercycles** by algorithmically and dynamically scheduling beams

WP2 **Automated LHC Filling:** **automate** and standardise **LHC** beam preparation and **filling**, and reduce impact on fixed target users and LHC turn-around time

WP3 **Automated Parameter Control and Optimisation:** **automation** of parameter **optimisation** and automatic containment of drifts

WP4 **Hysteresis Compensation:** deterministic **field control** with the aim of decoupling cycles

WP7 **Automated Equipment Testing:** AccTesting for "all" equipment for injectors and LHC to **automate hardware commissioning**

WP8 **Automate Equipment:** automation of **equipment setup**, **fault analysis**, and **recovery**; preparing the grounds for **preventive maintenance**

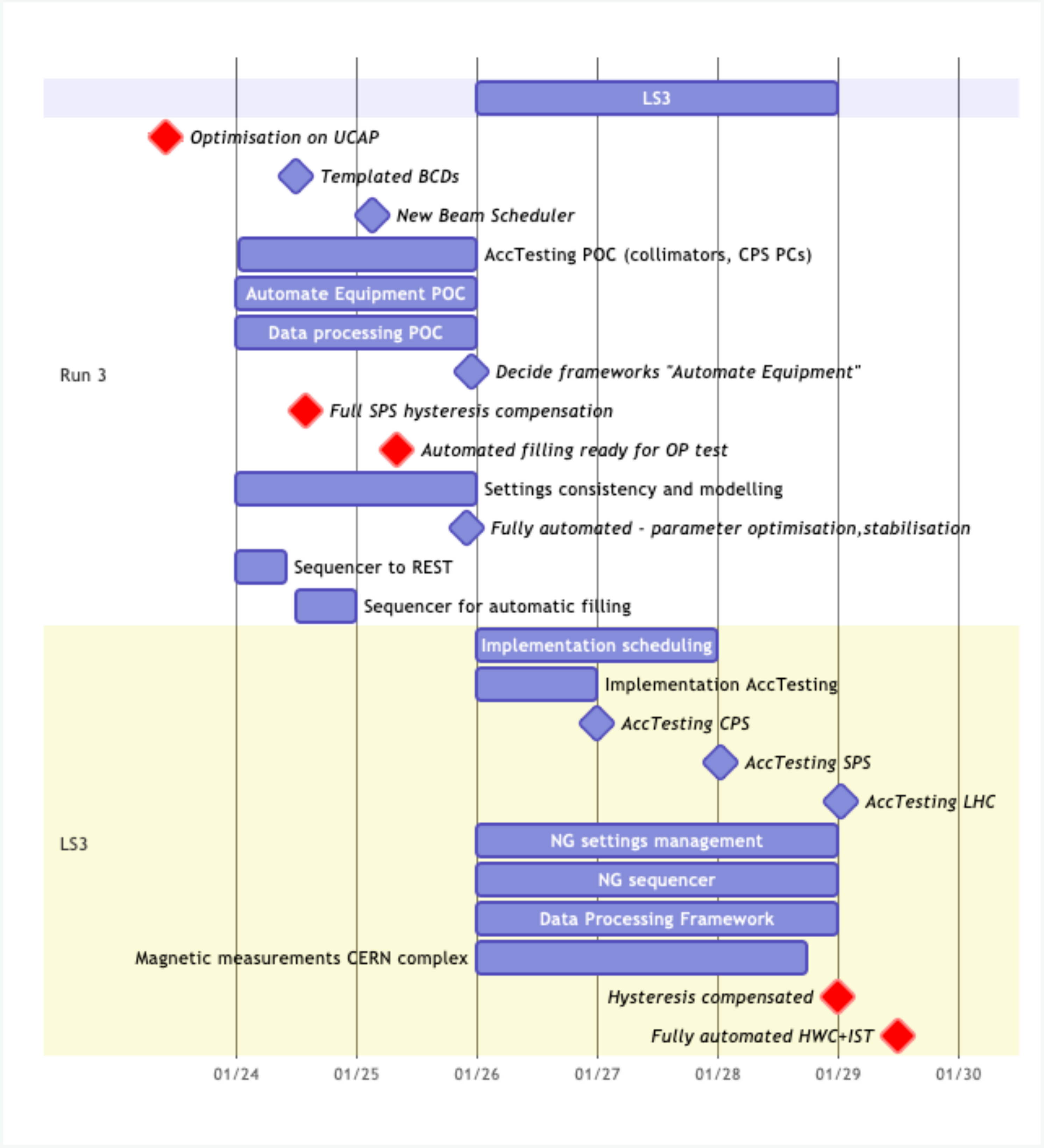
EPA timeline

Time-bound project for 5 years

- Improvements ready for Run 4 starting in 2029

Project with 2 phases:

- **Prototyping** and first operational tests in Run 3 (now!)
- **Full implementation** during LS3 and sequential commissioning in Run 4



Conclusions

Automation has long tradition at CERN

Conclusions

Automation has long tradition at CERN

The recent automation wave is radically different: **data-driven**!

Conclusions

Automation has long tradition at CERN

The recent automation wave is radically different: **data-driven**!

CERN has been investing into automation infrastructure and is working towards a paradigm shift with the **Efficient Particle Accelerators (EPA)** project

EPA was given 5 years to address various efficiency bottlenecks through

- Automation of **accelerators and equipment**
- Improved modelling
- Use of AI/ML at scale