

LIPAc Operation Log – a monitoring and reporting asset for operation supervision

L. Maindive⁴, J. Franco Campos³, J-C Morales Vega^{1,5}, Y. Carin^{1,2}, T. Akagi^{1,3},
A. Jokinen^{1,2}, F. Cismondi^{1,3}, T. Nakayama^{1,3}, M. Sugimoto^{1,3}, K. Kondo^{1,3}, I. Podadera⁵, A. Ibarra⁵

¹IFMIF/EVEDA Project Team, Rokkasho, Japan, ²F4E, Garching, Germany, ³QST, Rokkasho, Japan, ⁴UGR, Granada, Spain, ⁵IFMIF-DONES Granada, Spain.



Introduction

Accelerator-based high energy neutron sources are foreseen to characterize the material of future fusion reactors. The **Linear IFMIF Prototype Accelerator (LIPAc)** is a full-scale prototype aiming at validating the production of a deuteron beam of 125 mA at 9 MeV in continuous wave under the **Broader Approach (BA)** agreement between Japan and Europe.

Just as an orchestra needs a conductor, LIPAc operation team needs an **Operation Supervisor (OS)**. A dedicated monitoring tool was developed to support the person assigned to this role: the **LIPAc Operation Log (LOL)**. This new asset can be summed up as a synthetic and user-friendly **ELog (Electronic Logbook) for operation supervision**. It centralizes the main information and events related to the on-going operation, including objective of the day, assigned operators, beam status, key parameters, alarms, and interlocks.

System objectives

To **ease the reporting** during and after operation shifts.

- Sharing a compact view on the Teams meeting for operation
- Easily open the Event Report Management System



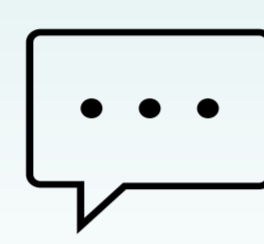
To **display key information** about the ongoing operation, including:

- Title and objectives of the experiment in progress
- Information on operators (names, roles)
- Key parameters of the experiment (type of particles, current, etc.)



To **record key events** during operation, including:

- Beam status (Permission, Destination, ON/OFF)
- Major alarms/interlocks
- Protection Systems statuses (Personal "PPS" and Machine "MPS")
- Key Process Variables (PVs) values when inserted by the OS (ex: Timings etc.)
- Comments submitted by the OS



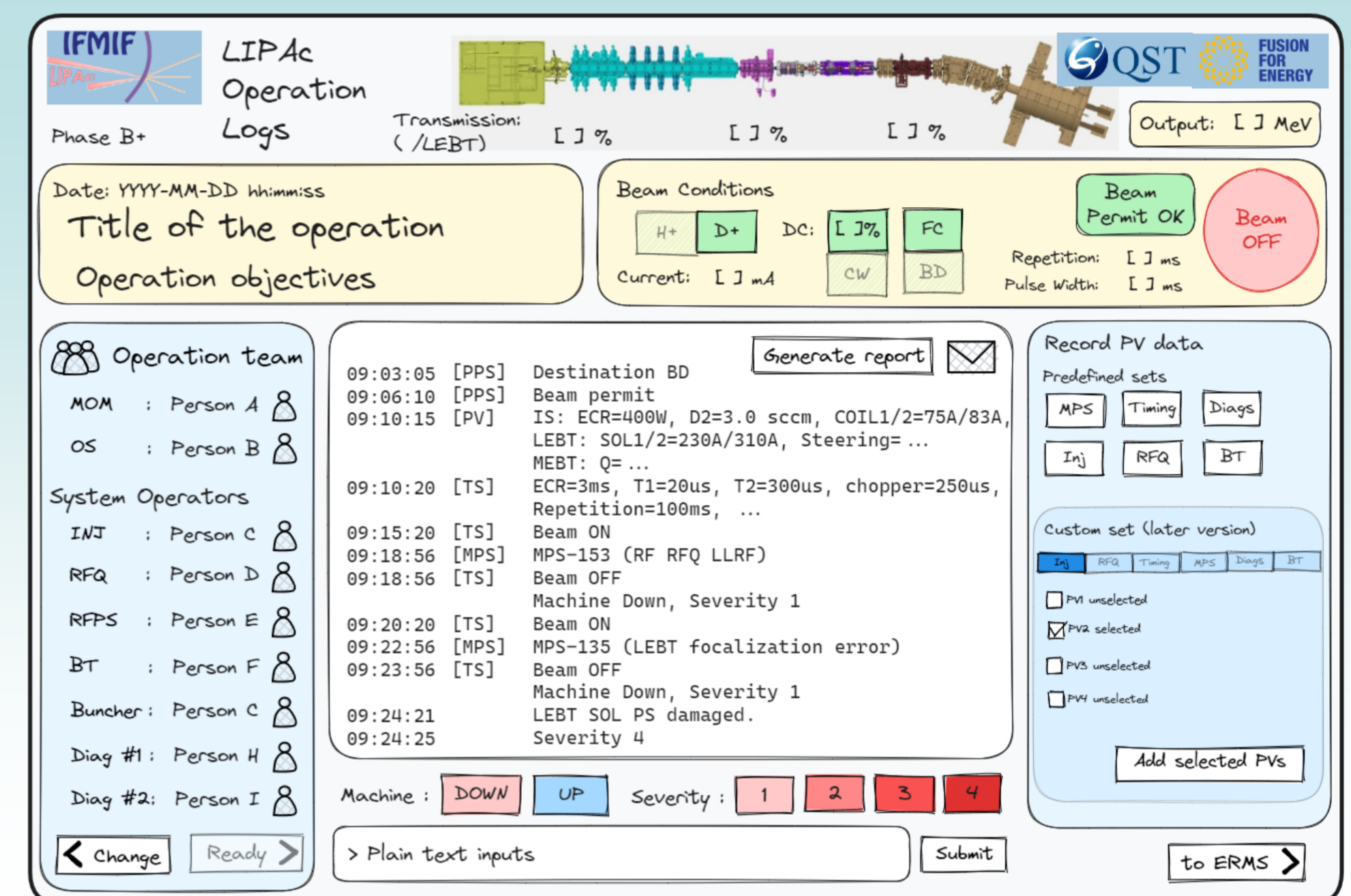
GUI – initial target

Main attributes:

- All-in-one GUI
- View only access for all
- Input access for OS only

Panels dedicated to:

- Operation general info
- Beam condition
- Operation Team
- Event display
- Manual comments/inputs
- Insertion of EPICS PV data
- Links to other tools



Tools

EPICS – Experimental Physics and Industrial Control System, framework often used on large scientific facilities like particle accelerators, including LIPAc.

CS-Studio / CSS Phoebus – Control System Studio, collection of tools and applications for developing and running GUI on EPICS framework

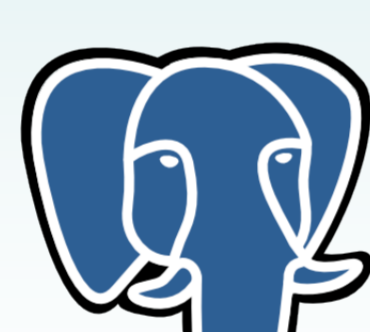
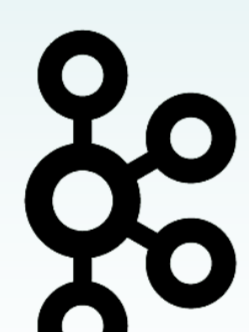
BEAST – Best Ever Alarm System Toolkit, developed by EPICS community

BUZZER – LIPAc Alarm System, in-house dev., intended for customizing alarm sounds

Kafka – Event Streaming Platform, used by CSS Phoebus Alarm System

PostgreSQL – Database System already used in most applications for the project

Grafana – data analytics and visualization platform



CS|studio

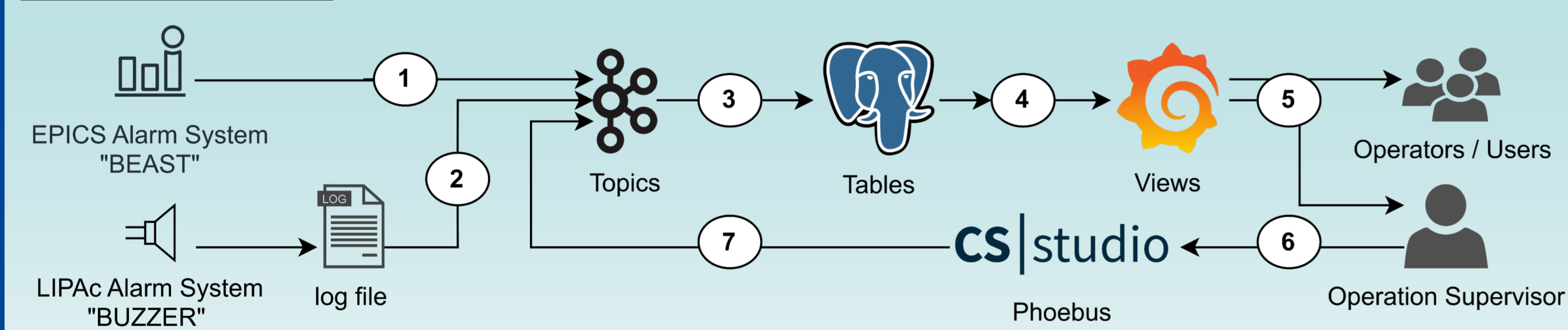
kafka

PostgreSQL

Grafana

Application Dataflow (current)

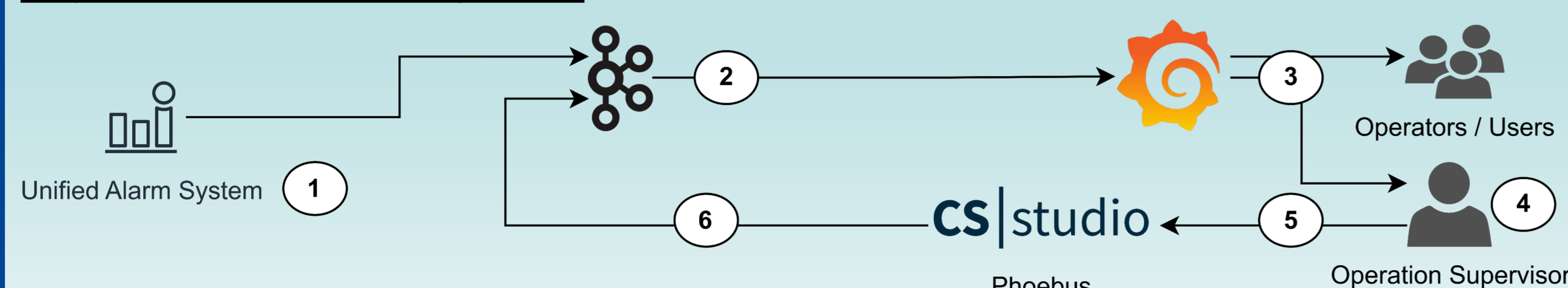
Current dataflow



1. BEAST writes directly to Kafka topics
2. Ingestion of BUZZER output (raw log file) into Kafka
3. Kafka connectors transfer the data to PostgreSQL tables
4. A Grafana dashboard queries and aggregates the data from PostgreSQL
5. The Grafana server is accessible both from the LIPAc network and the Internet
6. Acquisition of OS inputs from a Phoebus display in the Control Room
7. Python scripts embedded in the display to stream data into Kafka

"Ideal" dataflow (long term)

Objective dataflow (long term)



1. If LIPAc was to ever use a unified Alarm System, it would simplify data ingestion.
2. Ideally, Grafana could directly read data from Kafka topics.
3. Better queries and dashboards will ease users' access to information.
4. An export stage could support the OS to easily generate a final report.
5. Experience and feedback will greatly improve the content of the Phoebus display.
6. Embedded scripts are to be avoided. We should develop a Phoebus application.

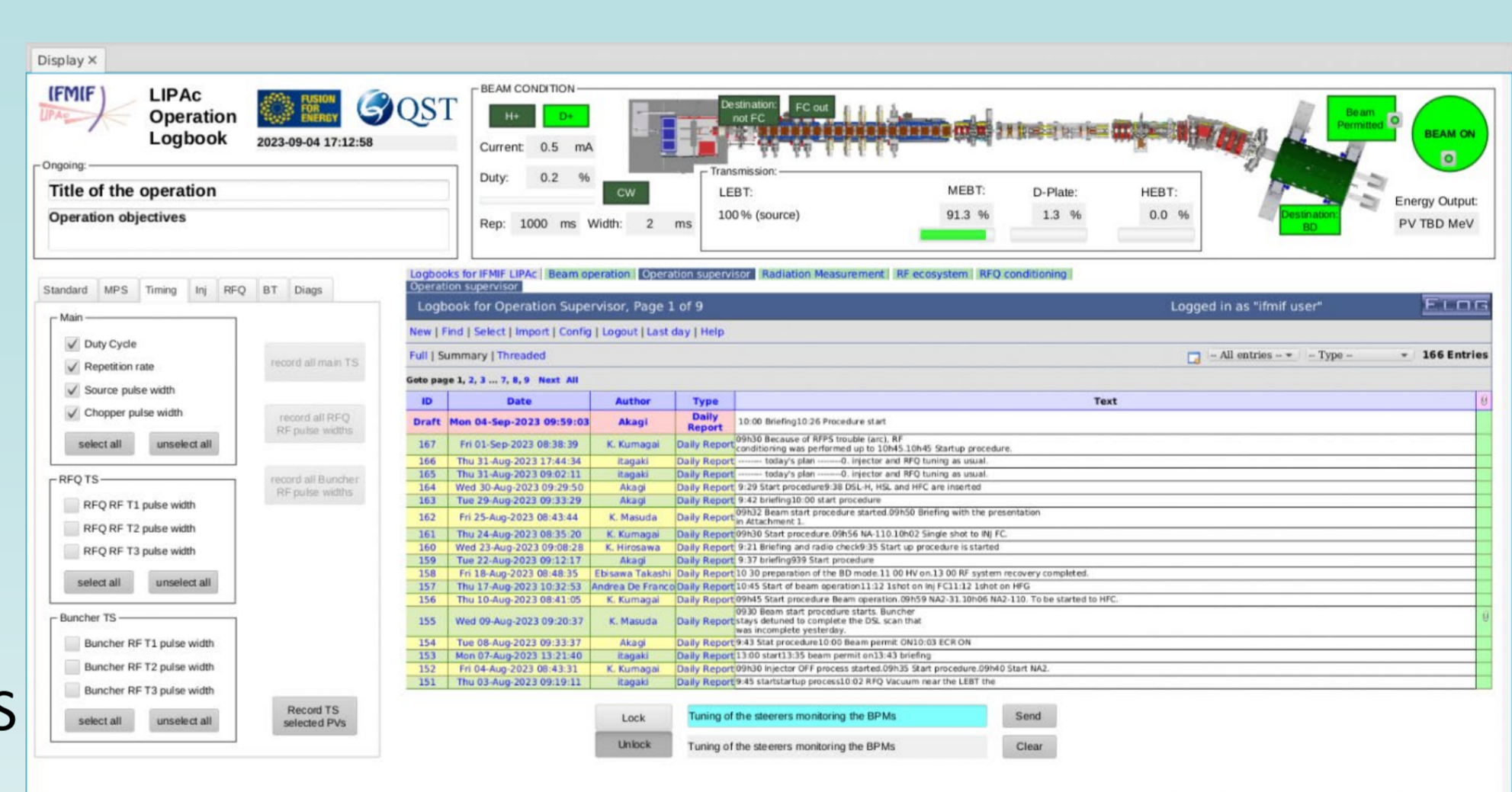
Two interfaces:

CS|studio

Phoebus display (OS side) :

- Summary of the operation
- Automatic data recording
- PV record functions
- Manual inputs from the OS

GUI – today's status



Grafana dashboard

- read-only
- customizable queries
- multiple boards

Defects/Problems to be solved in the current system

Phoebus

- Automatic data recording runs on the Phoebus display: no information recorded if the GUI is not opened.
- No direct feedback about the outputs (Grafana could be embedded but is too heavy).

Grafana

- Maximum refresh rate of 5 seconds.
- More than one hour time window for the query is too heavy: should be optimised.
- Only raw information from the alarms: they should be sorted, filtered and displayed more pertinently.

Alarm Systems

- Two very different systems to integrate.
- Non-homogeneous use of Warnings and Alarms among accelerator subsystems.

Conclusion

The LIPAc Operation Log tool is still at a development stage, but several shortcuts were chosen to quickly provide a prototype for current LIPAc operation stage. Even if the overall design is well defined, there is still a lot of evolution to come, in a near and further future. Yet, it could be interesting to reach an alpha version that could be tested by LIPAc OS before the end of the Operation stage.