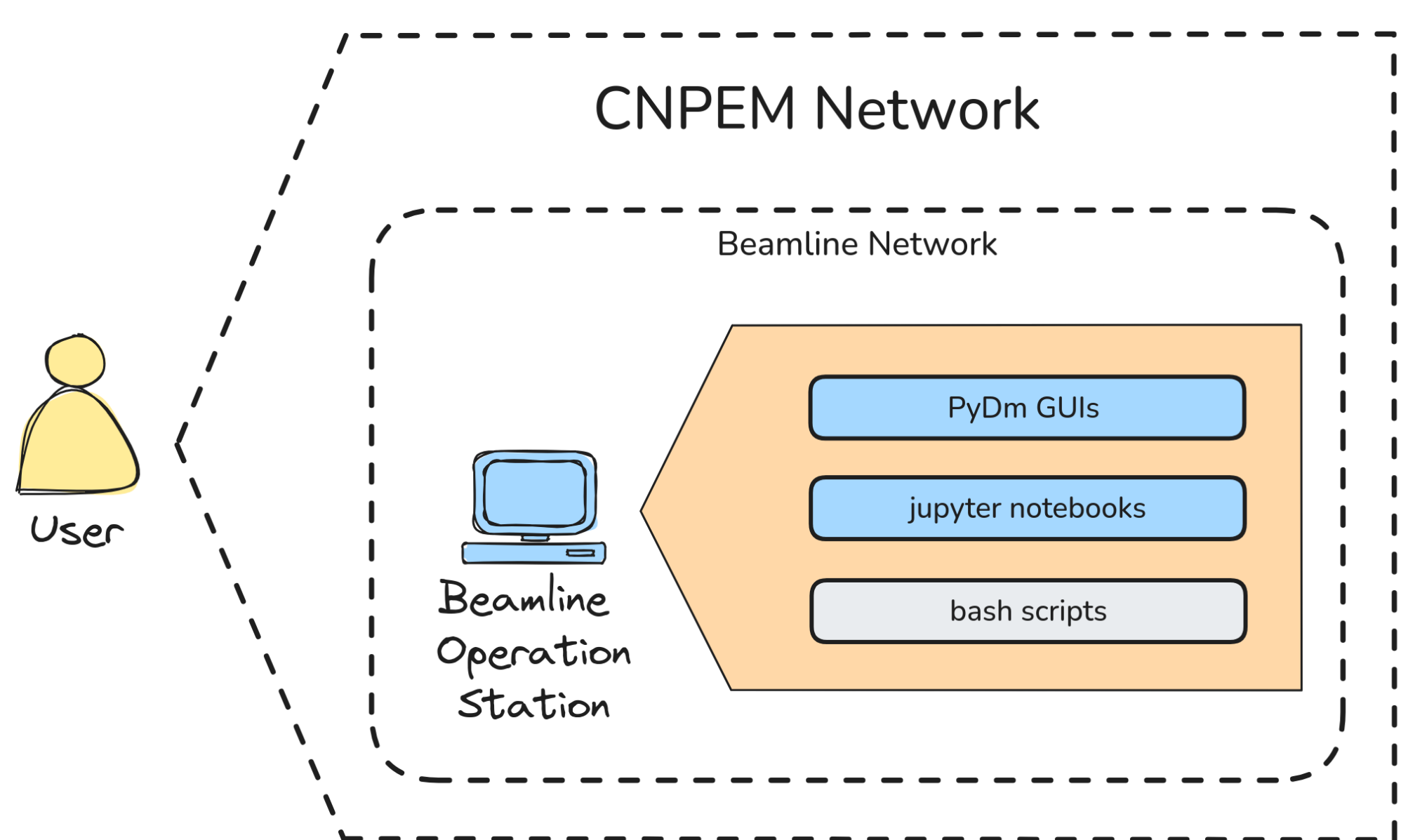


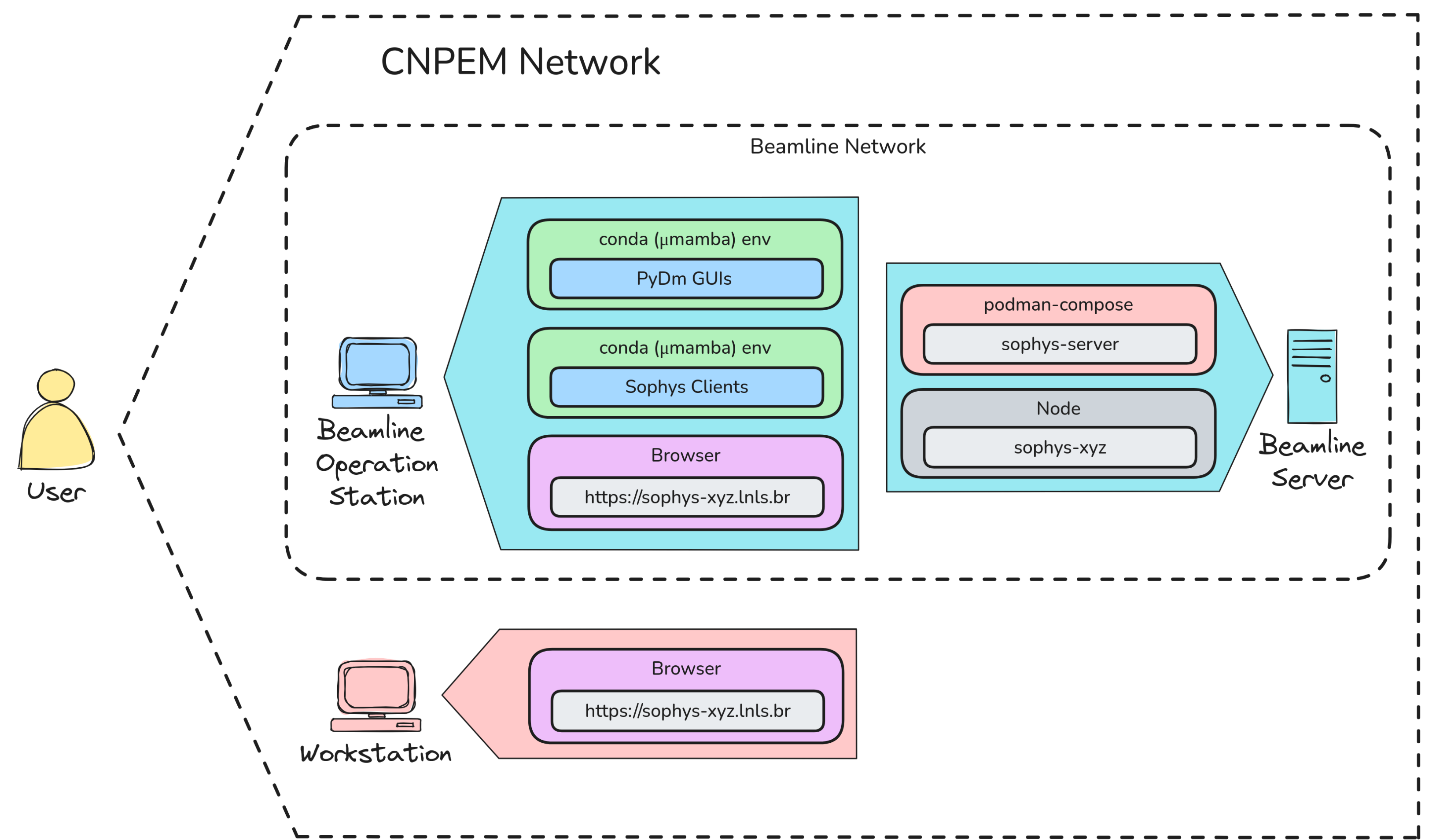
Ana Clara de Souza Oliveira¹, Bruno Vasco de Paula Carlos², Matheus Luís Bernardi³, Igor Torquato⁴, Rafael Lyra⁵, Sofia Donato Ferreira⁶
 ana.clara@lnls.br¹, bruno.carlos@lnls.br², matheus.bernardi@lnls.br³, igor.torquato@lnls.br⁴, rafael.lyra@lnls.br⁵, sofia.ferreira@lnls.br⁶
 Brazilian Synchrotron Light Laboratory (LNLS) - The Brazilian Center for Research in Energy and Materials (CNPEM)

SIRIUS is a 4th generation synchrotron light source facility that was designed, built and is operated by the Brazilian Synchrotron Light Laboratory (LNLS/CNPEM). Currently, SIRIUS has 6 fully operational beamlines and other 8 beamlines in technical commissioning, scientific commissioning or installation phases. Most SIRIUS beamlines currently have their experiment control solutions based on **Python** scripts, **Jupyter** notebooks and desktop graphical interfaces. The desktop graphical user interfaces are mainly implemented in Python and based on the **Qt** framework, using the **PyDM** library on top of **EPICS**. These solutions brought flexibility and the possibility of implementing graphical interfaces with zero or low-code approaches, making development possible by both support groups and beamline staff. Most of the interfaces that were made available as standardized solutions by support groups have as their main objective the monitoring and control of variables, with a focus on equipment control details. Inspired by positive feedback from beamline staff and users of web solutions such as **MxCuBE**, web implementations were explored by the team with the aim of offering standardized solutions for experiment control interfaces and the possibility of integration with the new Bluesky-based experiment orchestration layer.

Previous Systems



New Model



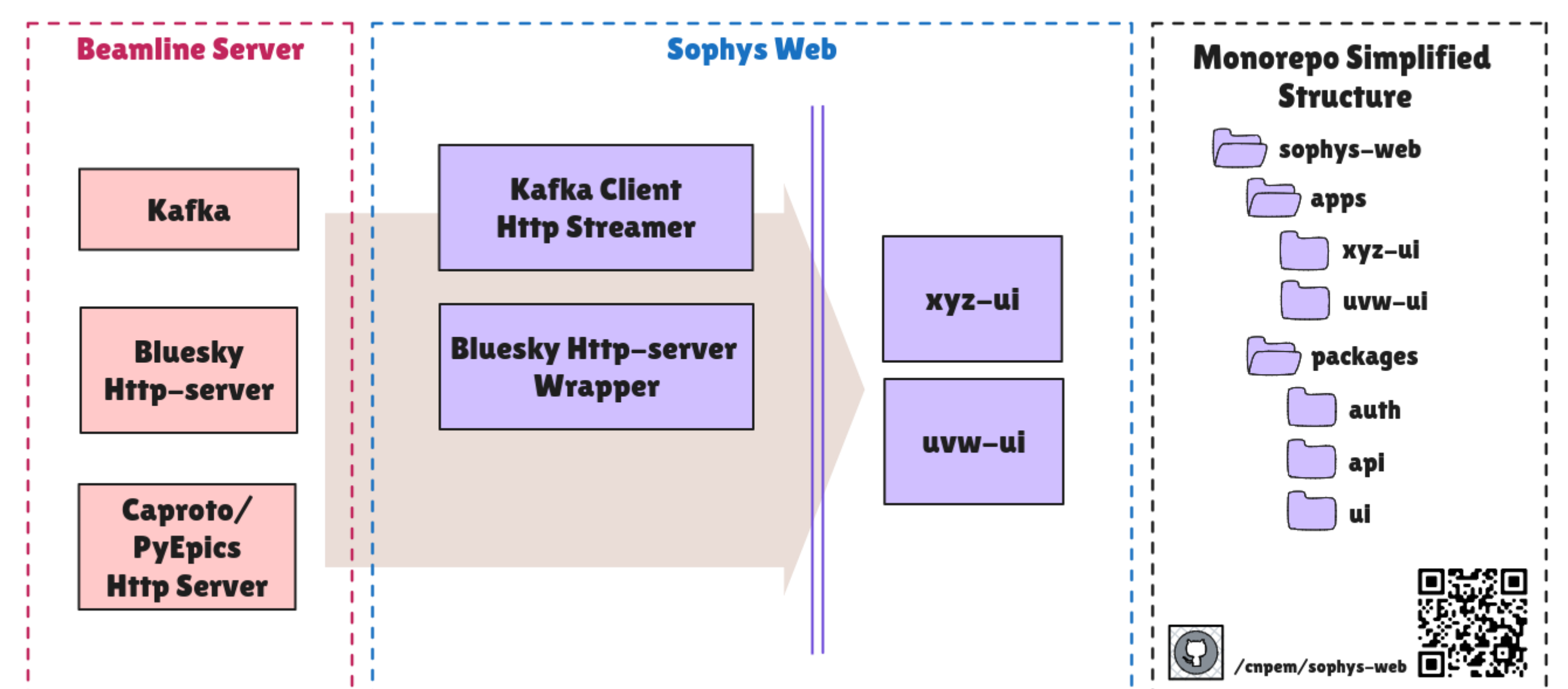
Maintainability strategy for Desktop GUIs

The GUIs developed by the Control Software group at Sirius use a packaging strategy to enhance modularity and reusability by encapsulating all the code into a single, manageable package. For desktop solutions, this approach facilitates distribution and version control, making deployment simpler using the proposed procedure with **micromamba**. In this context, each beamline can create its own environments, whether shared or specific to multiple GUIs, effectively addressing dependency management issues. Previously, the deployment relied on a central server with NFS and a single Python environment configured in the OS, leading to difficulties in maintaining and updating the GUIs for each beamline.

Challenges and Perspectives

- Create an efficient deployment strategy with automated testing and CI/CD for consistent quality and reliability;
- Offering automated deploy and version control tools;
- Enhancing both code quality and internal documentation to ensure better maintainability and clarity for future development.

Sophys-Web



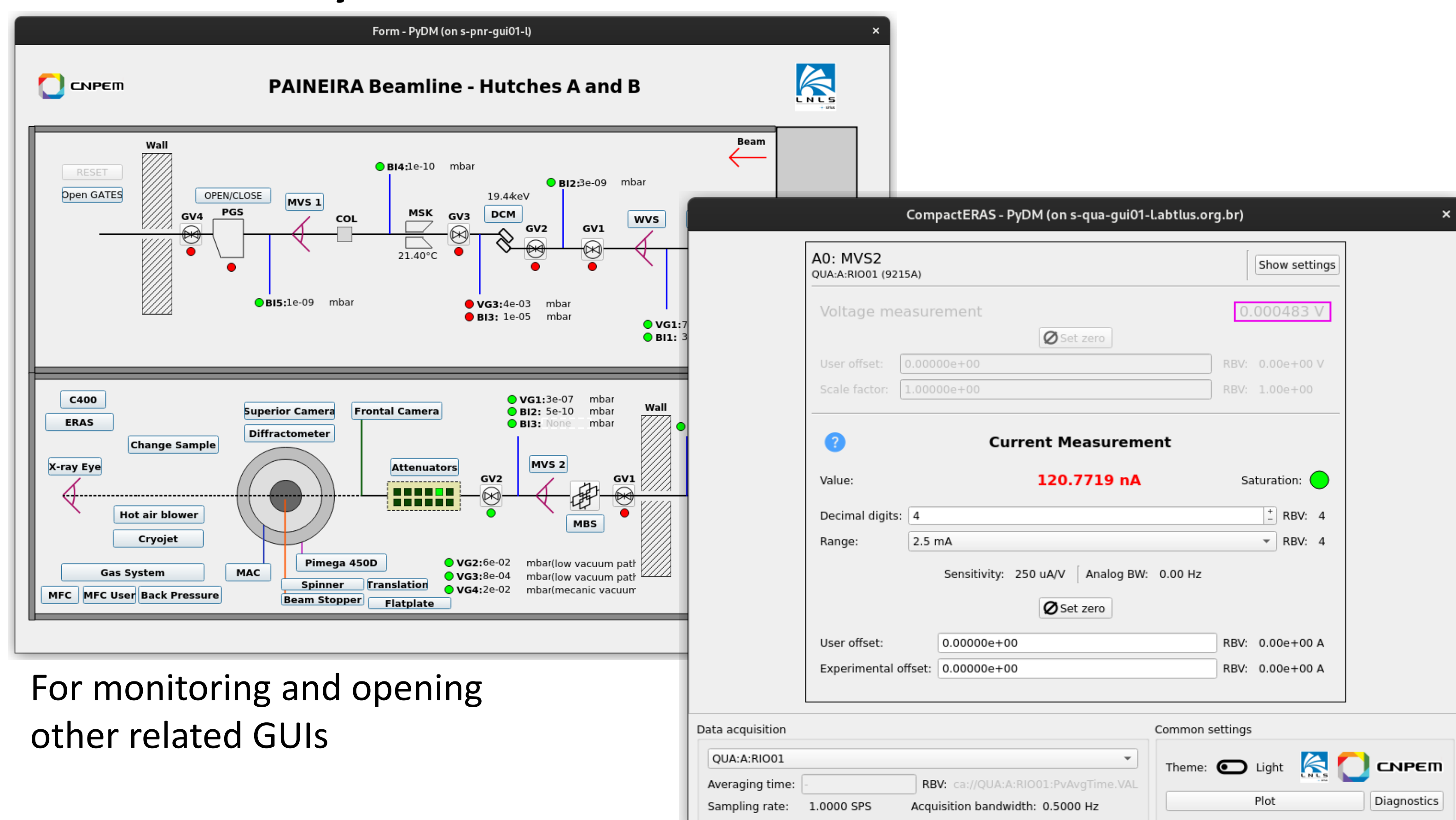
Sophys-web components and monorepo structure

To address the need for remote access, scalability, and maintainability, we are developing web-based control GUI options for SIRIUS beamlines, seamlessly integrating with our distributed architecture for remote data processing and job submissions via **SLURM**. Built as multiple web apps and APIs in a **Turborepo** monorepo with **Next.js** apps, our approach combines type-safe API routes with **trRPC** and **@tanstack/react-query** for efficient query state management and a responsive user experience. We're also exploring Server-Sent Events (**SSE**) to stream real-time data from sources like Kafka, PV (EPICS), and Bluesky, enhancing control and monitoring capabilities.

Challenges and Perspectives

- Develop a strong suite of reusable React components and API procedures for Sophys client apps across different beamlines;
- Reliably stream data from different sources to the client apps, ensuring the close-to-real-time monitoring events and control variables;
- Build a scalable and flexible architecture that allows custom, responsive interfaces tailored to each beamline.

Overview of PyDM GUIs



For monitoring and opening other related GUIs

For monitoring and configure specific devices