STRUCTURE AND DEVELOPMENT OF SESAME'S CONTROL SYSTEM CLIENT

P.O. Box 7, Allan 19252, Jordan
www.sesame.org.jo

A. Al-Dalleh, A. Ismail, I. Saleh, SESAME, Allan, Jordan

ICALEPCS2017
Barcelona · Spain, October 8-13 · Palau de Congressos de Catalunya

Abstract

SESAME is a third-generation 2.5 GeV synchrotron-light source based in Allan, Jordan. The Pre-injector (Microtron) and Injector (Booster Ring) have been commissioned while the commission of the storage ring began in January 2017 and we expect machine operation in late 2017. The current components of the control systems software side are IOCs developed using EPICS software tools, graphical user interfaces (OPI) designed using Control System Studio (CSS) software tools, process variables archiving using CSS BEAUTY toolkit, alarm handling using CSS BEAST toolkit and tools to help in automation and reporting. This paper will present the current design of the client system which includes what was needed for the active commissioning period as well as upgrades that are under research including EPICS Qt framework as a client replacement for CSS and the pros and cons of this replacement and upgrading the archiver engine to a scalable and higher performance engine.

CONTROL SYSTEM CLIENT STRUCTURE

- The main interface is divided based on the machine's stages, and each stage consists of its own subsystems..

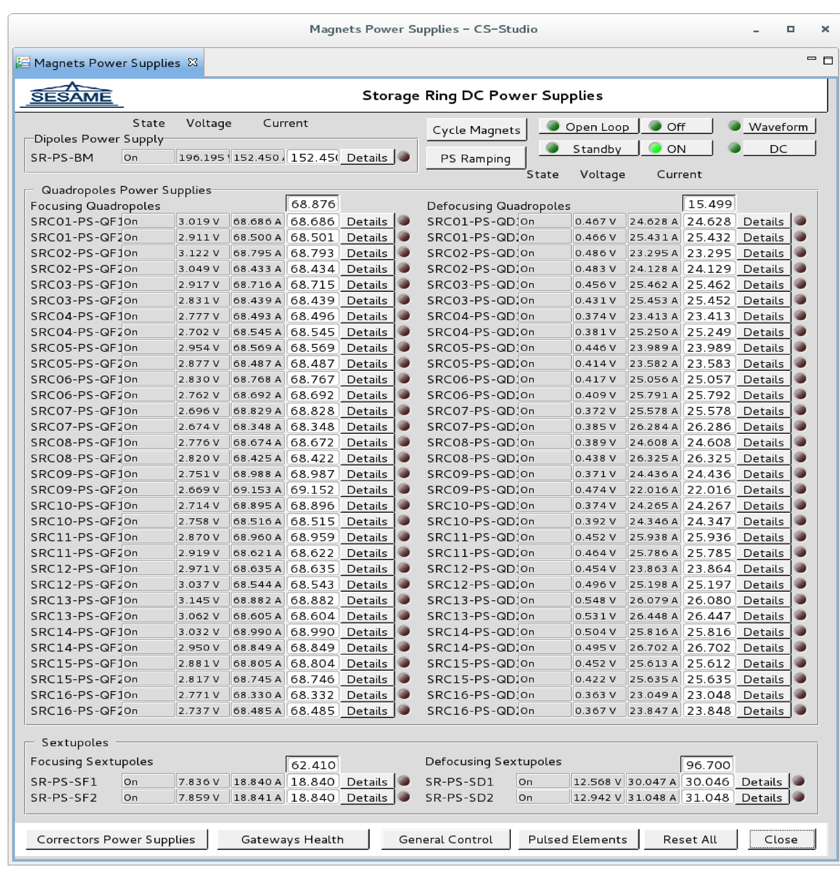

Figure 1: Storage-Ring DC Power Supplies OPI

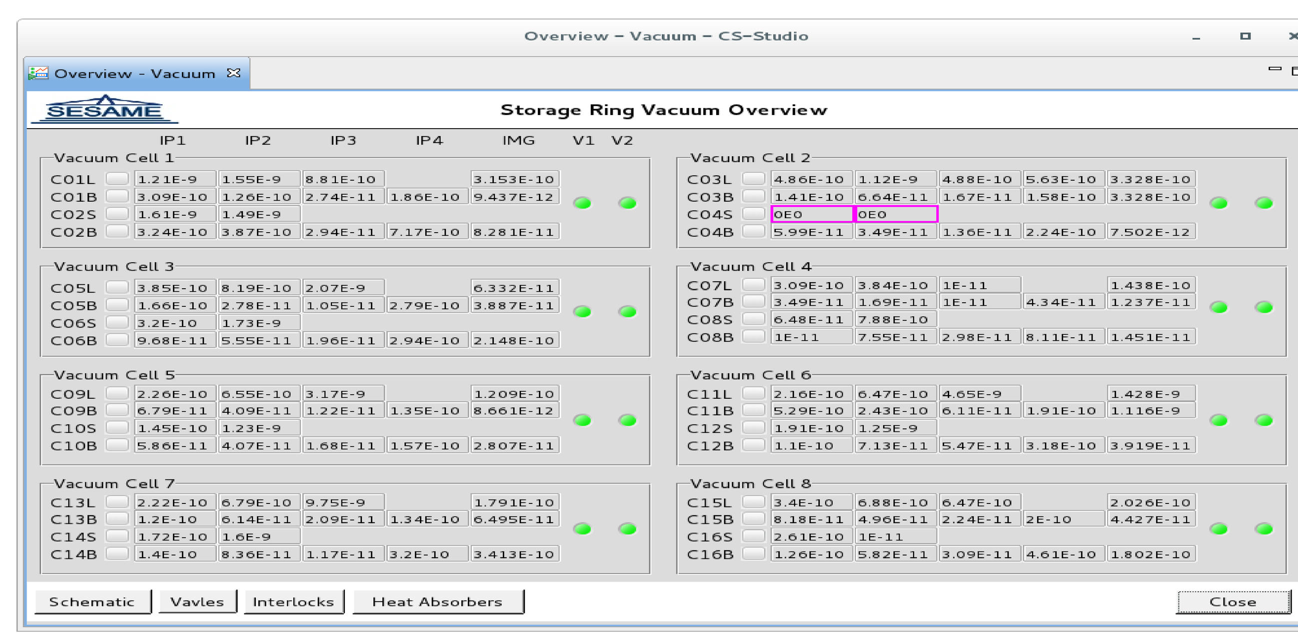

Figure 2: Storage-Ring Vacuum OPI

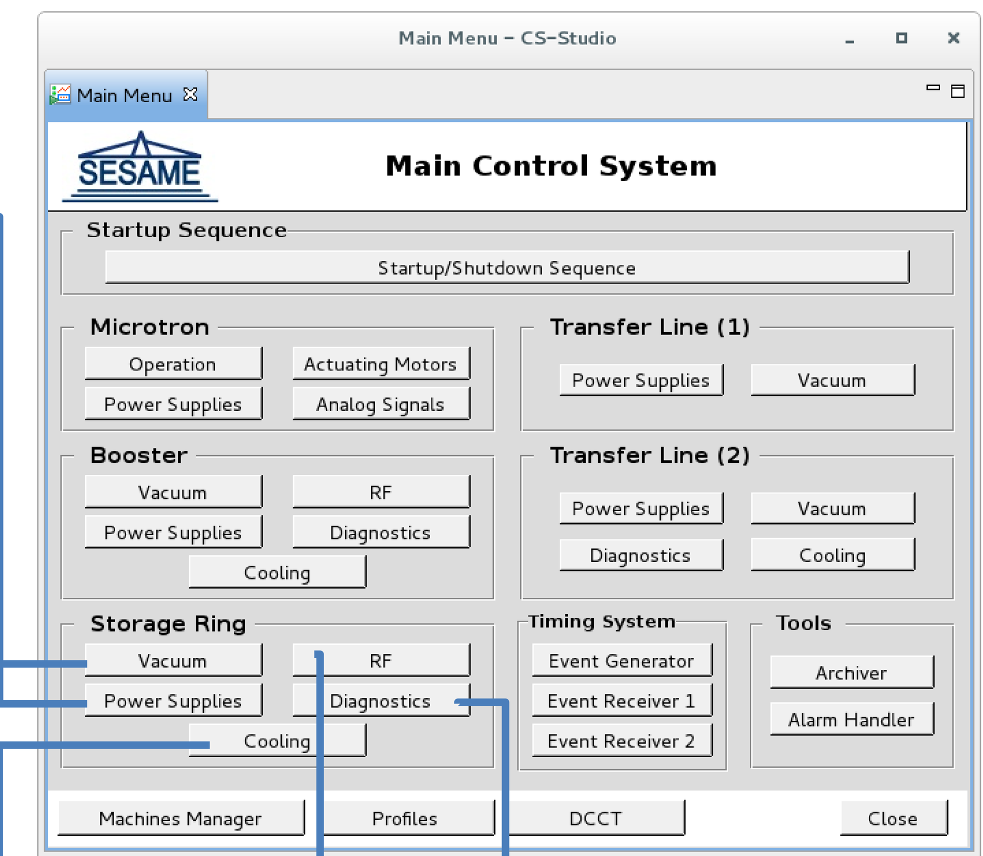

Figure 3: Main Interface for the Control System Client

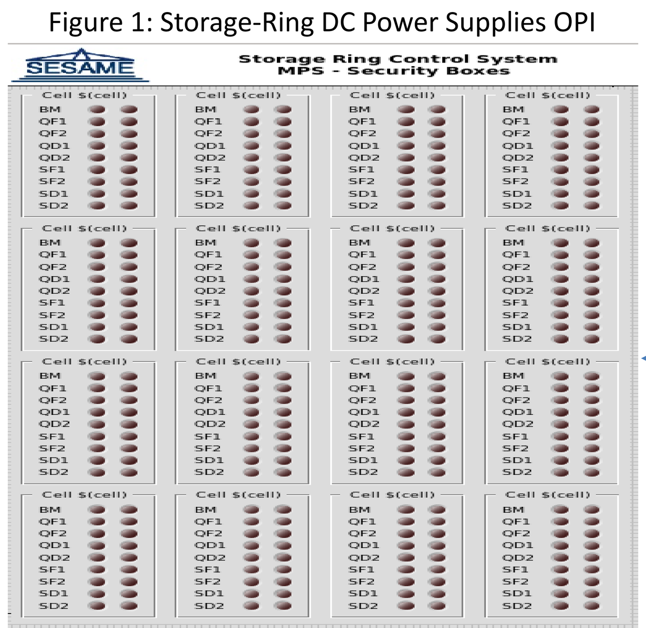

Figure 4: Storage-Ring Cooling OPI

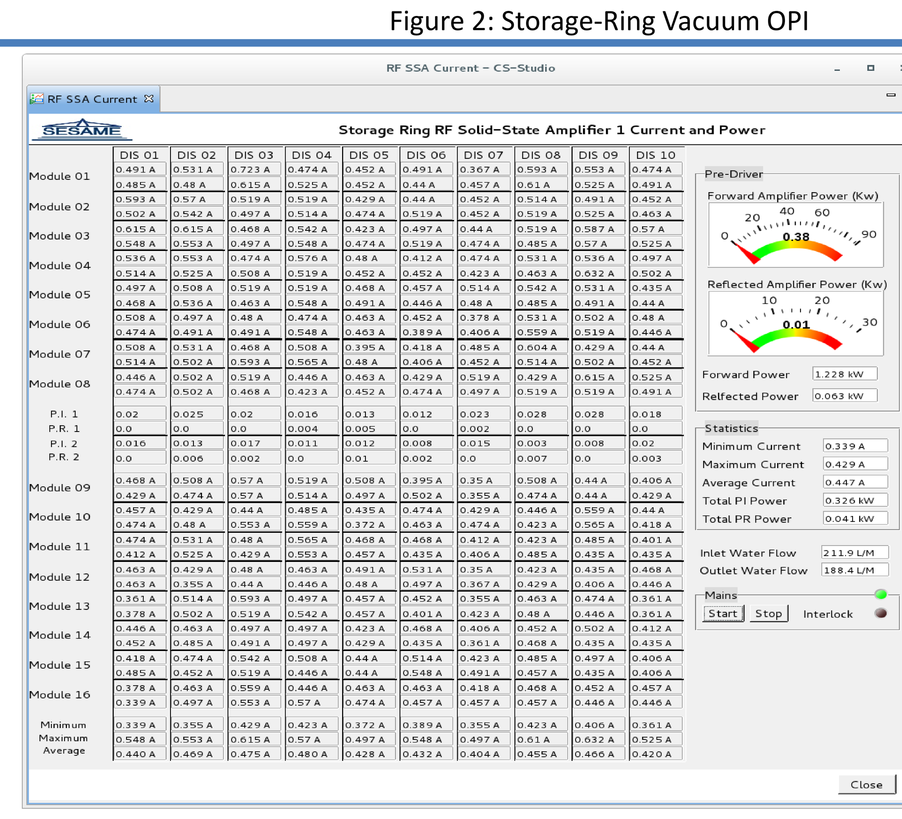

Figure 5: Storage-Ring RF Amplifiers Current OPI

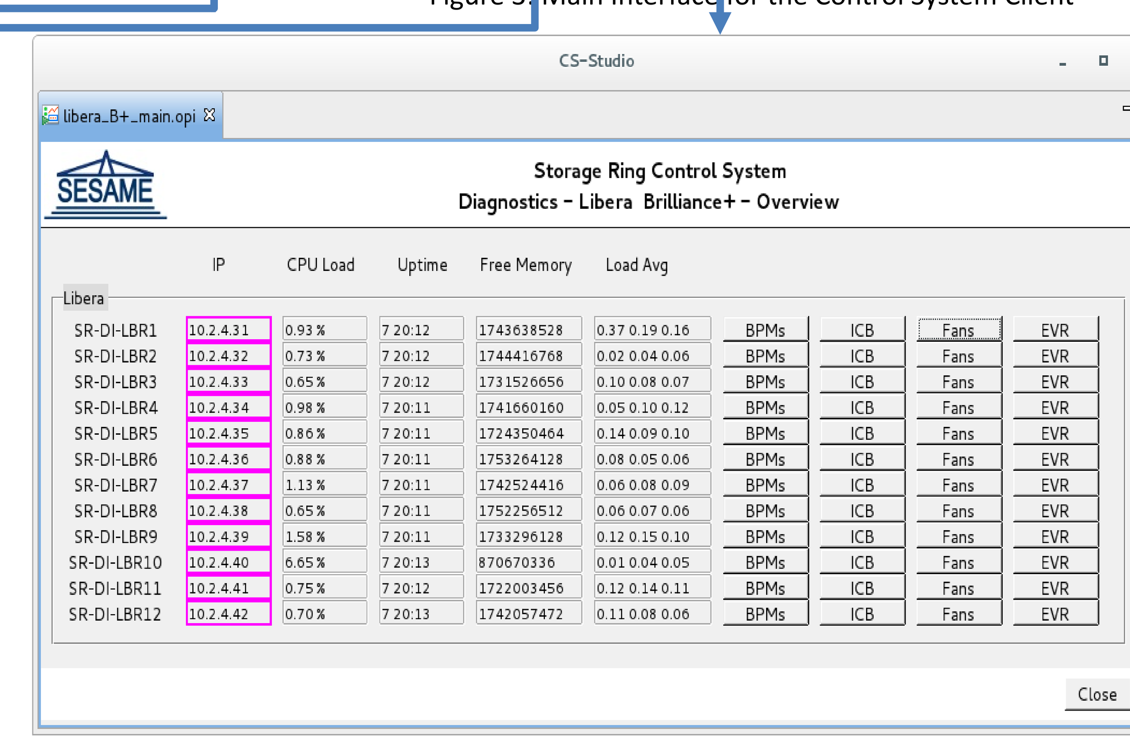

Figure 6: Storage-Ring Libera Brilliance+ OPI, detailed screens designed the same as the original EDM screens

DEPLOYING THE EPICS ARCHIVER APPLIANCE

- We are using the CSS RDB Archiver which archives data to a PostgreSQL database. The archiver is running on a virtual server with 32GB RAM and 1TB of storage
- Currently we have around 12000 input process variable for the machine with 1200 are actively archived, the currently archived process variables are consuming around 23GB/week of storage which fills up the allowed storage very quickly.
- The other problem was data retrieval within the CSS which might take hours depending on the requested time interval.
- The EPICS Archiver aims to archive millions of PVs using a clustered design approach with focus on data retrieval performance.
- The Archiver Appliance is designed in Java and deployed in terms of Tomcat webapps. It consists of 4 Tomcat instances: Engine for archiving and clustering, ETL for storage management, retrieval for client data retrieval and management for the web interface.
- The Archiver Appliance stores the archived samples in a custom file format in the storage, based on the software design the sample size will take around 22 KB per 1 second sample, averaging around 600 MB per year per sample.
- Taking the previous machine statistics into account, it would take less than 15 GB per week for the entire machine and that is assuming active and continuous operation.

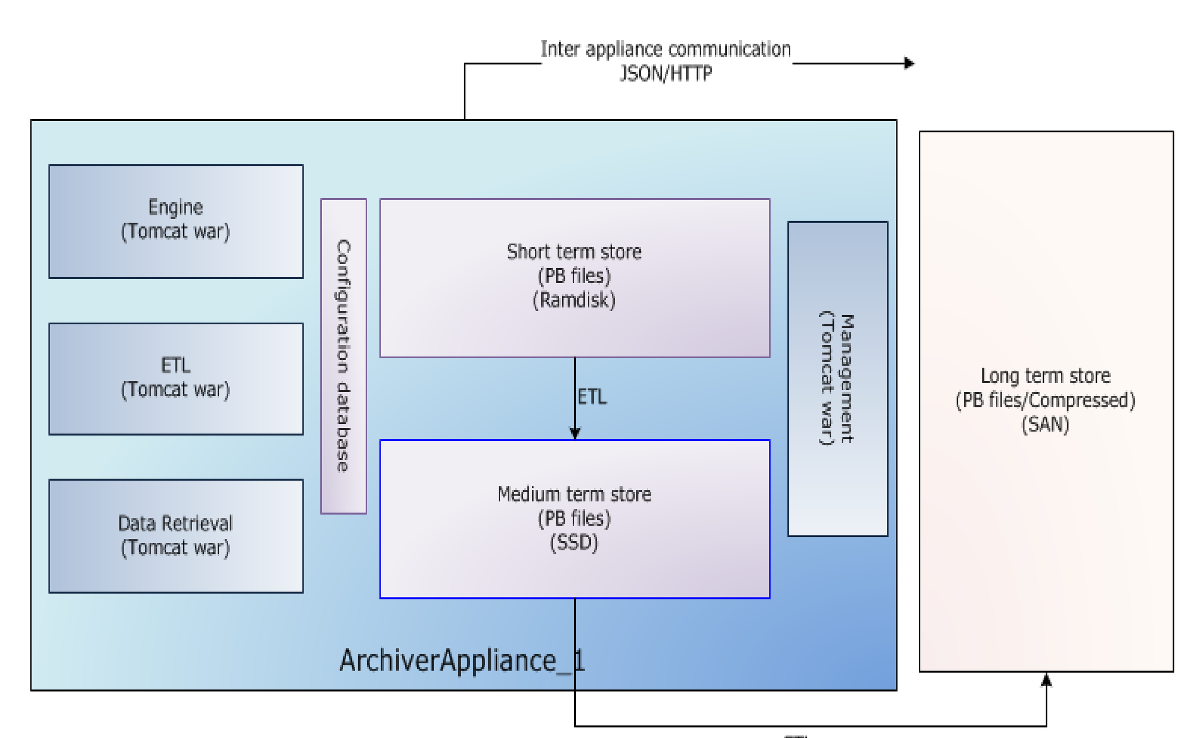

Figure 7: Architecture of a single archiver appliance

CLIENT DESIGN USING EPICS Qt

- Based on the popular Qt Creator framework and EPICS, EPICS Qt provides a high-level framework for faster prototyping and deployment of control system client.
- Being very popular and built on top of Qt and flexible UI design, EPICS Qt can be considered a good replacement for existing control system clients such as Control System Studio, EDM, etc.
- The concept is still under research before officially replacing CSS.

- UI design is kept the same as CSS so nothing feels different for the operators.
- Qt's flexibility will allow for more things to be implemented using the framework itself instead of relying on external scripts and commands for example.

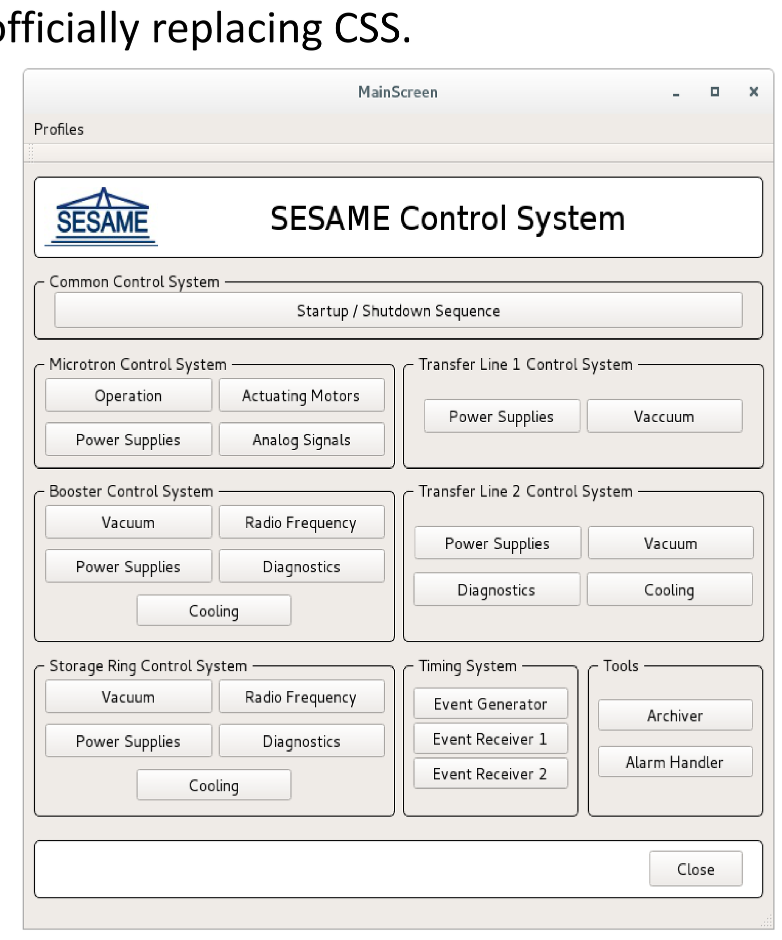

Figure 8: Qt Design for the main screen.

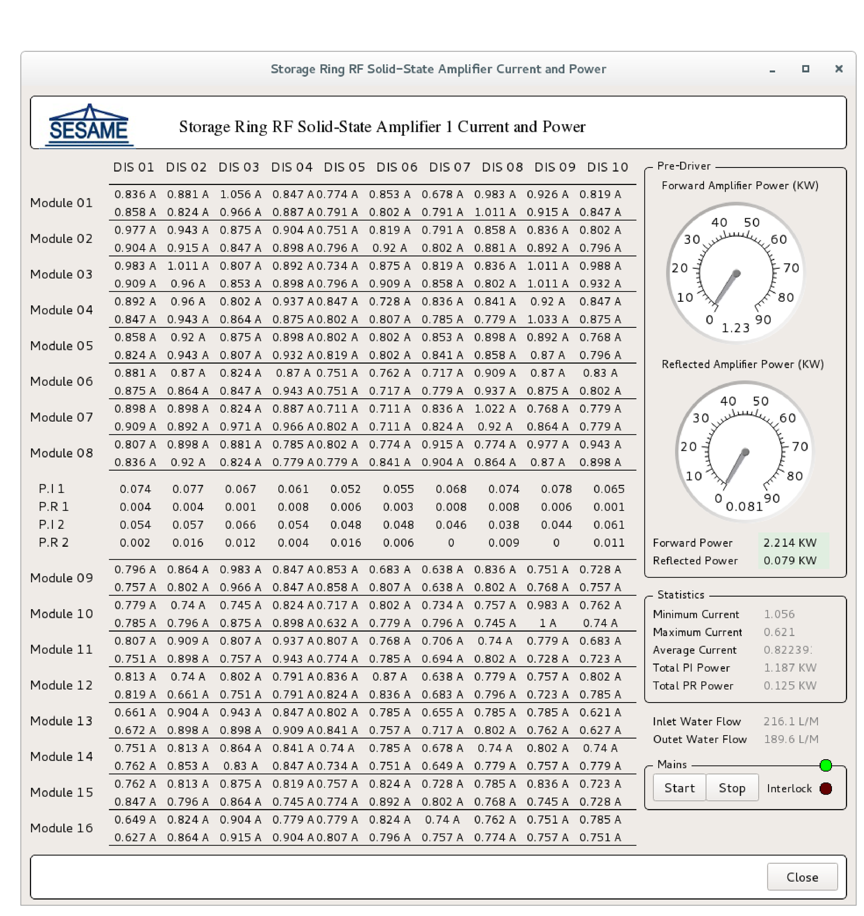

Figure 9: Qt Design for the storage-ring RF.

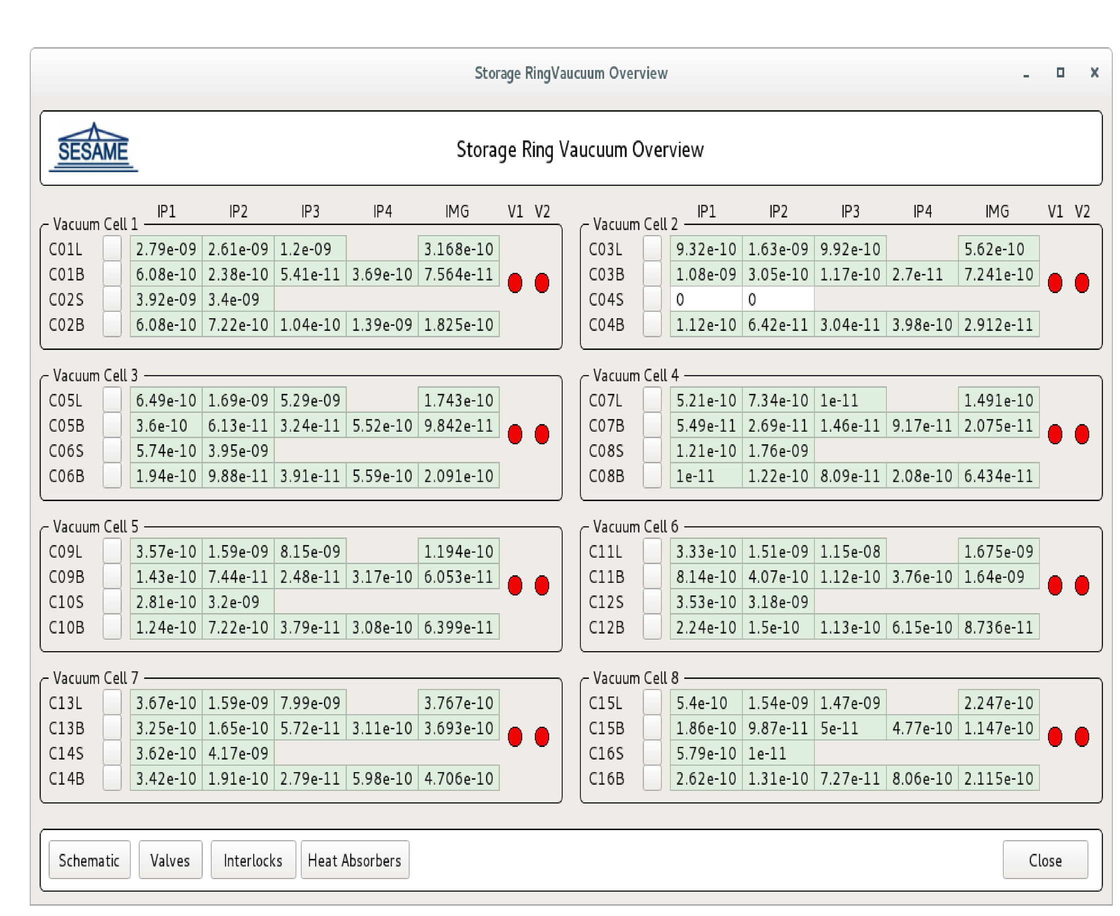

Figure 10: Qt Design for the storage-ring vacuum overview.