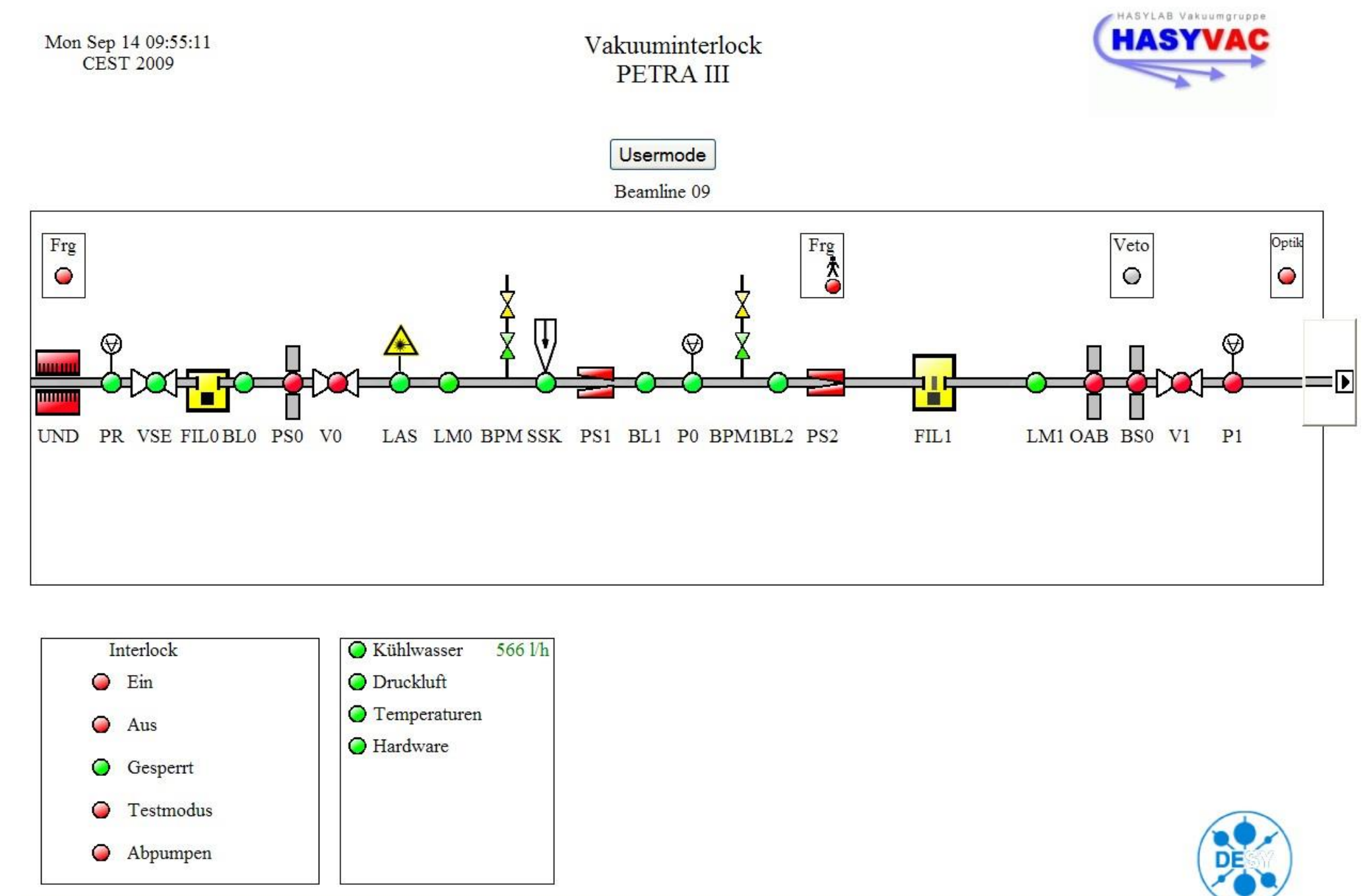
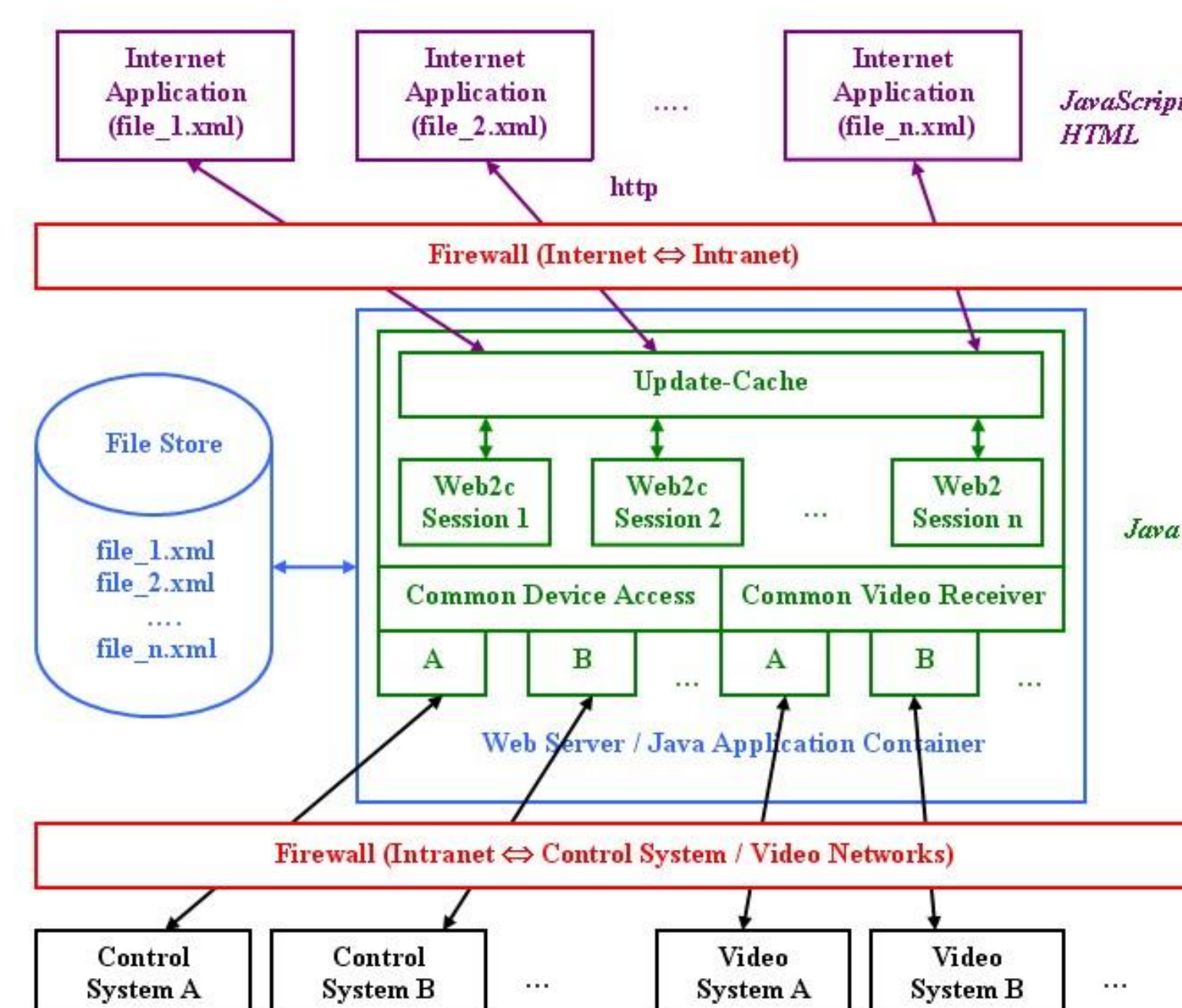


Using React for Web-Based Graphical User Applications for Accelerator Controls

R. Bacher, J. Szczesny
Deutsches Elektronen-Synchrotron DESY, Germany

Legacy: Web2cToolkit

- Framework for web-based rich client control system applications
- Collection of web services (Web2c Synoptic Display Viewer, Web2c Archive Viewer, ...)
- Platform-independent
- Based on HTML, CSS, JavaScript, XML
- Asynchronous client-server communication (AJAX)
- Interfaces to TINE, DOOCS, EPICS, TANGO
- Proprietary framework
 - Hardly maintainable in terms of performance and usability
- Status
 - Outdated
 - Development discontinued
 - Only bug fixes provided



Progressive Web Apps

- Cross-platform web applications
- Based on HTML, CSS, JavaScript, WebAssembly, Service Worker, ...
- Common, platform-independent code base
- Runs in the browser engine
 - Embedded in the browser window
 - Stand-alone application added to the user's home screen
- User experience
 - Look&Feel like a native app
 - Responsive
- Discoverable by web search
- Downloadable from a web server

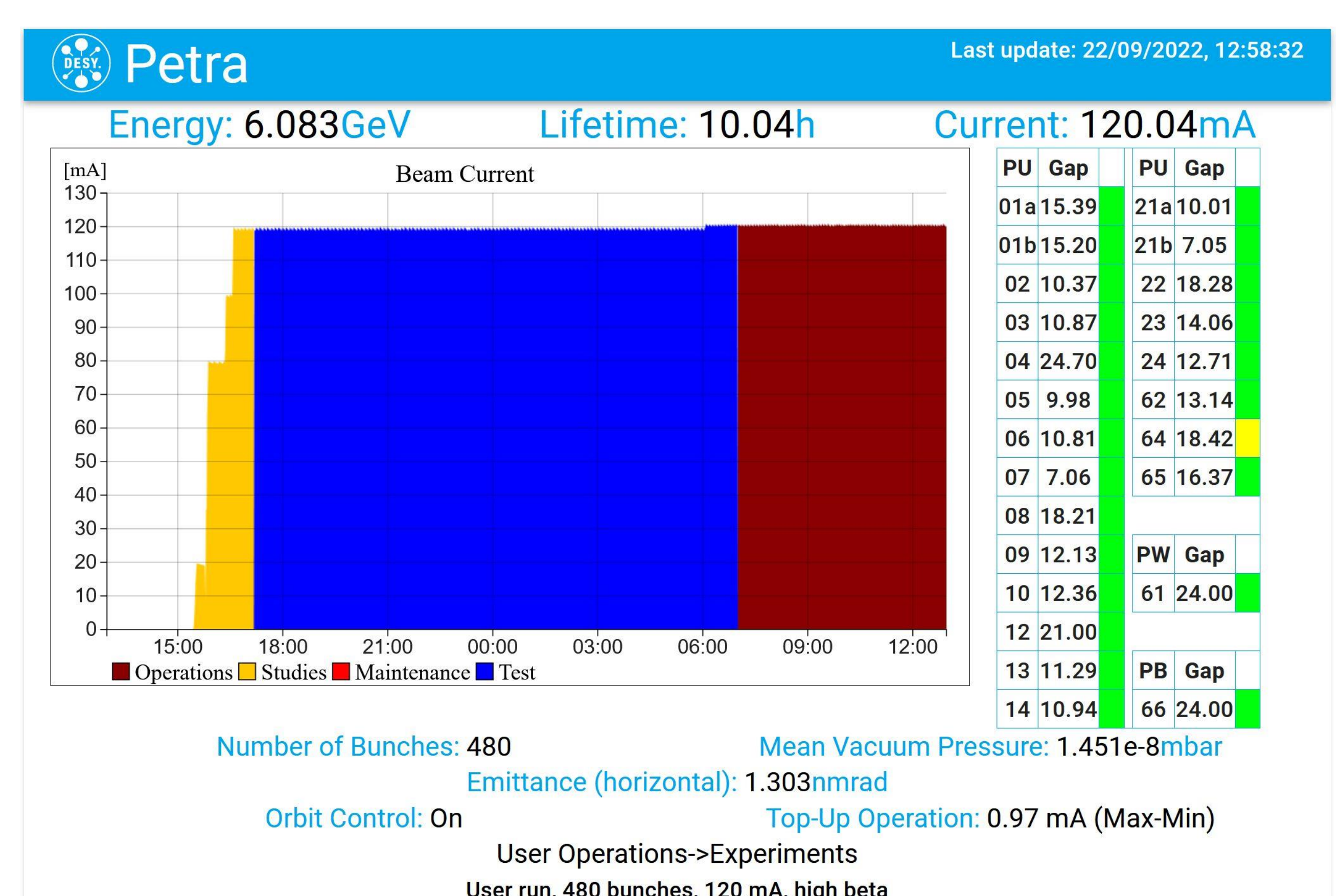
WebACOP Chart Library

- Provides a graphical component for displaying indexed data such as timeseries or histogram data
- Wraps the graphics D3.js library in a JavaScript library
 - Based on SVG, HTML5, and CSS
- Configures dynamically the D3.js-based chart
- Injects data read asynchronously from the control system gateway server into the HTML/Canvas element of the D3.js drawing area

Dashboard Apps

- Cross-platform web apps
 - Based on React, JavaScript, HTML and CSS
 - Visualize the status and the performance of the operation of the DESY accelerators
- Communication with accelerator control system
 - Asynchronous client-server communication using the promise-based axios HTTP client
 - RESTful web server acting as a gateway to the accelerator control system
- Encapsulation of control system data in JSON structure
 - Example:

```
{ "context": "PETRA", "server": "VAC.ION_PUMP", "device": "SEK.ALL", "property": "P.MEAN", "stCode": 0, "timeStamp": 1663847153800, "systemStamp": 1873725217, "fmtCode": 5, "numElements": 1, "format": "FLOAT", "status": "success", "data": "1.458399E-8" }
```



Findings & Conclusions

- React framework is a powerful tool for rapid prototyping
 - Once implemented, components can be easily reused in other projects
 - Implementation of simple components proves to be relatively easy
 - Development of complex components can be quite time-consuming, cumbersome and error-prone
- Rapid release cycle of React versions often poses challenges for the developer
 - Limited backward compatibility
 - Possible incompatibilities with other third-part libraries
- Responsive design approach
 - Works well for dashboard applications etc.
 - May not work for complex, well-designed control room applications with a variety of graphical widgets

