Web Run Control for ATLAS Trigger and Data Acquisition

Andrei Kazarov[1] and Aimilianos Koulouris[2]

[1]University of Johannesburg, SA
[2]CERN, Geneva, Switzerland

E-mail: Andrei.Kazarov@cern.ch

Abstract. The ATLAS experiment at the Large Hadron Collider (LHC) operated very successfully from 2008 to 2023. The ATLAS Control and Configuration (CC) software is the core component of the ATLAS Trigger and Data Acquisition (DAQ) system. It encompasses all the software required to configure and control the ATLAS data-taking processes, essentially acting as the glue that holds the various ATLAS sub-systems together.

In recent years, more and more CC software applications have become available as web applications, facilitating remote operations for experts. However, the main data-taking control and monitoring application, known as 'IGUI' (Integrated Graphical User Interface), was missing. IGUI is the front-end GUI tool used by operators in the ATLAS Control Room to manage data-taking sessions.

This paper introduces a new web application called 'WebRC' (Web Run Control), which provides IGUI-like functionality for monitoring and controlling data taking from a web browser. WebRC features include presenting the Run Control tree of all applications, dynamically updating their states, browsing log files and message streams, and monitoring various system parameters. This allows experts to promptly assess the state of data taking and investigate potential issues.

WebRC is built using the Apache Wicket framework. Recent developments have integrated WebRC with CERN's authentication infrastructure, allowing users full control over TDAQ data-taking sessions.

WebRC is widely used for monitoring data-taking sessions during the ongoing Run 3 period, with dozens of users connecting daily.

1 Introduction

The ATLAS experiment at the Large Hadron Collider (LHC) operated very successfully from 2008 to 2024, in periods known as Run 1, Run 2, and Run 3. The ATLAS Control and Configuration (CC) software [1] is the core component of the ATLAS Trigger and Data Acquisition (TDAQ) system, encompassing all the software required to configure and control ATLAS data-taking processes. It essentially provides the components that integrate the various ATLAS sub-systems.

In recent years, more CC software applications have been made available as web applications, facilitating remote operations for experts. The significance of web-based solutions has increased substantially, especially as teleworking has become a standard practice, even in physics experiments. However, the main data-taking control and monitoring application, known as "IGUI" (Integrated Graphical User Interface [2]), which is the front-end GUI tool used by operators in the ATLAS Control Room, was notably missing.

We present a web application called "WebRC" (Web Run Control), which offers data-taking monitoring and control functionality from a web browser. WebRC includes features such as presenting an expandable tree of all DAQ applications, reflecting changes in their states, browsing log files, subscribing to application messages, and monitoring various DAQ parameters, trigger rates, and ATLAS detectors' busy information. In short, WebRC enables experts to promptly assess the state of data taking and investigate potential issues.

2 Requirements and technology choice

The following factors were identified as important for the technology choice and for the design of WebRC:

- Integration with TDAQ Services: The back-end must be easily integrated with the main TDAQ services (Run Control, Information Service, and Error Reporting Service - ERS), all of which are accessible via CORBA middleware.
- Rich Front-End Widgets: The front-end should offer a rich set of widgets, enabling the implementation of GUI features similar to those available in IGUI.
- Scalability and Resource Efficiency: The system should be well-scalable and conservative in resource usage, supporting many user sessions and connecting to multiple TDAQ sessions in parallel.
- Dynamic and Interactive Web Features: Support for dynamic and interactive web features like Ajax or WebSockets is essential, allowing the application to visually reflect fast changes in data-taking conditions without needing to reload web pages.
- Minimal Technology Stack: The development stack should include a minimal number of technologies and should not depend on other frameworks and back-end libraries. This facilitates maintenance and allows for a fast learning curve for newcomers.

After conducting a survey and developing a prototype, Apache Wicket was identified as a solid candidate. Apache Wicket is an open-source, component-oriented, pure-Java web application framework [3]. Developed since 2004, it remains one of the mainstream Java server-side frameworks, enabling the development of powerful Java server applications that integrate naturally with TDAQ software. A significant advantage of Wicket is that it allows the development of a Java-only back-end application, without involving any JavaScript front-end, simplifying both development and long-term code maintenance.

3 Implementation

The server side of WebRC is a single multi-threaded Java application that provides HTTP access to render Wicket components, modeling different elements of a running DAQ data-taking session. These elements include a hierarchical tree of DAQ applications and streams of application messages. The front end is a simple HTML page (styled with CSS) containing stubs for visualizing the Wicket Java components, following a typical Model-View design pattern. Visualization is managed internally by Wicket, leaving only the back-end business logic to the programmer.

The single back-end application handles multiple user sessions and can connect to different independent data-taking sessions (known as "partitions"). The server application uses standard TDAQ software libraries to access all the necessary information, similar to its standalone counterpart application, IGUI.

The single web page integrates the following main elements (panels): a view on all running applications organized in a tree (known as Run Control tree), a panel providing view on all application messages passed via ERS system, and a contextual monitoring graphics dashboard.

The application can run in two modes: Display and Control. The Display mode, which is the default mode for ATLAS operations, provides monitoring-only capabilities. An example of the functionality provided by the WebRC single-web page application in Display mode can be seen in Figure 1.

Run Control Tree panel Represents all running applications in the selected partition (approximately $O(10^4)$ applications). It dynamically reflects their Run Control state, error state, and busy status, and also displays contextual monitoring information for certain elements, such as the occupancy of the trigger farm and global parameters of the Level-1 trigger.

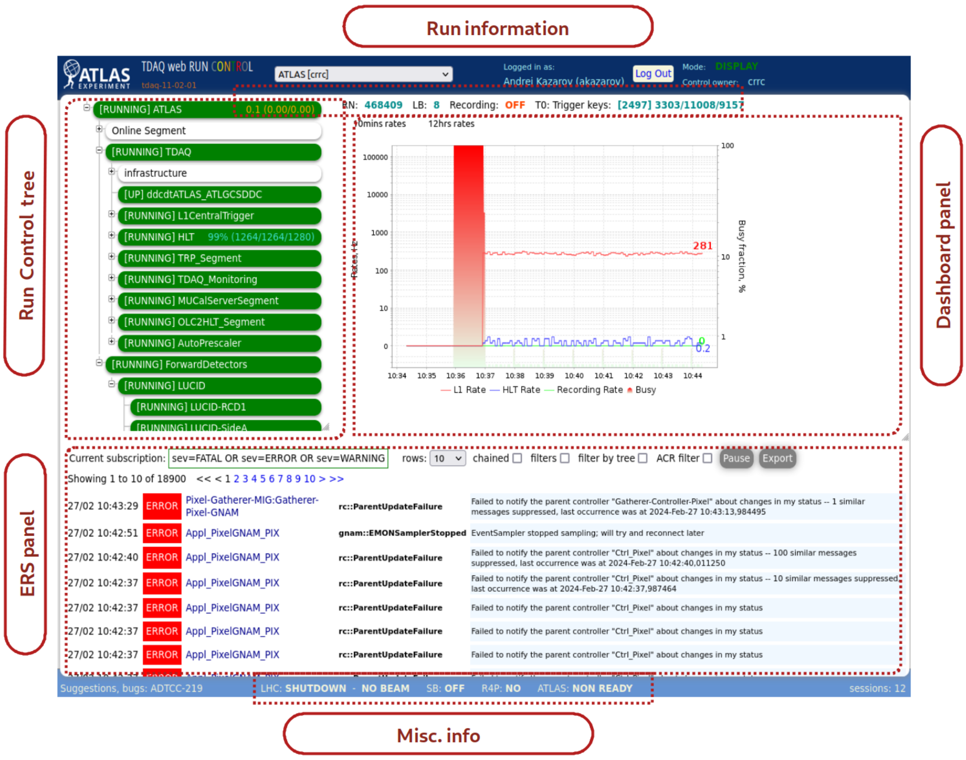

Figure 1: A screenshot of the WebRC web page with different panels.

The ERS panel Allows users to subscribe to application messages and apply advanced filtering criteria to display only relevant messages in the table. Users can also display messages from a specific group of applications selected in the Run Control Tree. This functionality is crucial, as it enables experts to focus on relevant messages.

The Dashboard panel Embeds an external monitoring dashboard associated with the selected element in the tree. Each user can customize the URL and has a personalized monitoring profile stored in the browser's cookie. The default dashboard displays the Trigger rates and Busy fraction, which are the most crucial DAQ parameters to monitor.

Misc and Run information These elements provide information about current run parameters and miscellaneous details from ATLAS and LHC monitoring. The goal is to present experts with the most relevant information on a single web page.

4 Control mode

A more advanced deployment of WebRC is implemented for the TDAQ test-bed cluster at CERN. In this setup, WebRC runs in Control mode, allowing full control over DAQ sessions (partitions) initiated on the test-bed. This includes sending global data-taking session control commands (e.g., "START"/"UNCONFIG" as shown in Figure 2), making configuration changes such as disabling or enabling specific parts of the configuration, and sending individual signals to any application in the tree, such as "Restart" or "Kill".

WebRC switches to Control mode when the connected user name matches the owner of the partition. To achieve this, any user session with WebRC must be authenticated via the CERN Single Sign-On (SSO) portal. This is accomplished by protecting the front-end WebRC page URL behind a CERN SSO proxy application deployed in the CERN Kubernetes cluster [4].

5 Authentication and token exchange

WebRC relies on the CERN SSO infrastructure for authentication, providing not only basic user authentication but also enabling user applications registered in SSO to obtain an SSO Access Token. This token can then be exchanged for application-specific tokens using technologies such as Java Web Tokens (JWT) and OpenID Connect (OIDC). Every authenticated user session in WebRC makes a token exchange

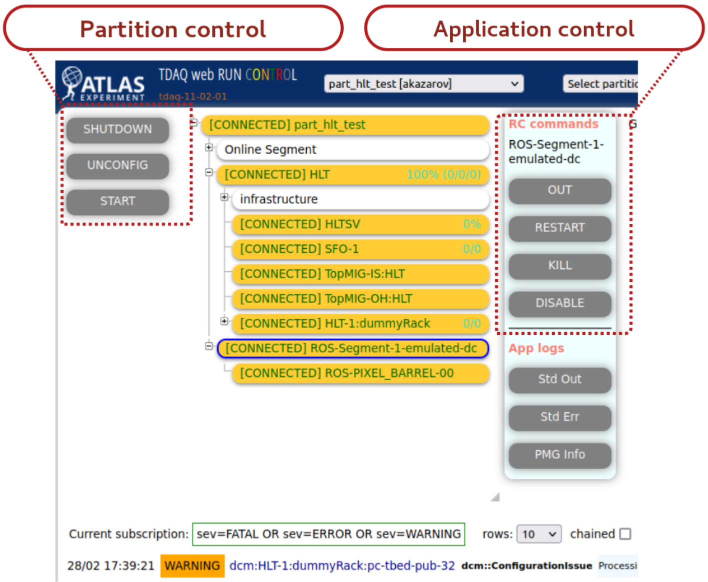

Figure 2: A screenshot of the WebRC in Control mode.

request to the SSO portal and receives its own DAQ user token [5], provided by another application deployed and registered in the OIDC infrastructure. This token allows each session to authenticate when accessing the DAQ test-bed services on behalf of the user, enabling actions as if the user were issuing DAQ commands from the command line.

Overall, this deployment, as illustrated in the Figure 3, allows full control throughout the lifetime of a user session in the test-bed, from bootstrapping the partition infrastructure to its full shutdown.

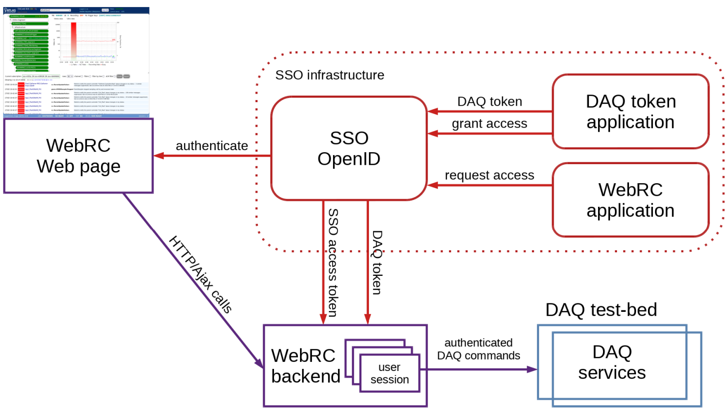

Figure 3: Integration of WebRC in SSO and token exchange.

6 Conclusions

WebRC is a single-page web application used for monitoring and controlling TDAQ data-taking sessions in the ATLAS experiment and the TDAQ test-bed. A single, powerful Java-only application serves many dozens of user sessions, facilitating the remote monitoring of ATLAS data-taking sessions and enabling experts to handle problems efficiently. A secure and robust OIDC-based authentication mechanism provides users with full access to TDAQ resources in the test-bed directly from a web browser.

References

[1] G. Lehmann Miotto et al., Nucl. Instrum. Meth. A623, 549 (2010)
[2] G Avolio et al 2011 J. Phys.: Conf. Ser. 331 022032
[3] Apache wicket, https://wicket.apache.org, accessed: 2024-06-30
[4] https://paas.docs.cern.ch/4._CERN_Authentication/1-use-cern-sso/, accessed: 2024-07-03
[5] ATLAS TDAQ software package, 2024, https://gitlab.cern.ch/atlas-tdaq-software/daq_tokens, accessed: 2024-07-03