

Modernizing Access Control and Database Reliability for Scheduling Tools

Aryan Choudhary
Part of Summer Student Programme

Uploaded on August 1, 2025

Abstract

During my 2025 internship at CERN, I contributed to strengthening the backend infrastructure supporting internal scheduling tools used during technical coordination periods like YETS (Year-End Technical Stops) and the upcoming LS3 (Long Shutdown 3). My primary focus was modernizing the access control system—introducing a flexible, role-based architecture to replace a legacy model that lacked granularity and traceability.

In addition, I contributed to the stability of database-related components within the tools' deployment lifecycle. While not directly responsible for CI/CD pipeline development, I worked to ensure that schema changes and database availability could be verified and integrated smoothly into existing automated deployment processes.

These improvements led to more secure user access, improved auditability, and increased confidence in the system's behavior during scheduled operations.

1. Introduction

CERN's research depends on meticulous coordination between hardware maintenance, experimental schedules, and large-scale system interventions. The **scheduling tools** used internally at CERN play a vital role in organizing and authorizing access during periods of limited beam time—particularly during **Year-End Technical Stops (YETS)** and the forthcoming **Long Shutdown 3 (LS3)**.

Though often behind the scenes, the digital infrastructure supporting these efforts must remain reliable and secure, given the safety-critical and multi-team nature of operations. My internship focused on strengthening this infrastructure, specifically targeting:

- Access management and authentication mechanisms for scheduling tools
- Database integrity and monitoring in the context of automated deployment

2. Authentication Redesign for Scheduling Tools

2.1 Context and Challenges

The scheduling tools previously relied on a legacy authentication system with limited role distinction and coarse-grained permissions. This created operational difficulties such as:

- Inability to clearly differentiate user roles (e.g., experiment coordinators vs. technical engineers)
- Minimal auditing capabilities for system access
- Increased administrative effort for maintaining user access as usage expanded

Given that over **2,500 CERN personnel** rely on these tools during high-stakes operational phases, addressing these limitations was critical to ensuring system integrity and ease of use.

2.2 Implementation

I implemented a new **Role-Based Access Control (RBAC)** model integrated into the existing backend (Spring Boot, Java), with supporting changes in the frontend (React, TypeScript). The model introduced structured roles such as **Admin**, **Planner**, and **Viewer**, with associated permissions scoped to relevant API endpoints and UI components.

Authentication was migrated to CERN's centralized identity provider using SSO-compatible protocols, which improved session handling and user federation across internal tools.

The system was tested in a staging environment within the production **OpenShift/Kubernetes** cluster to ensure compatibility and avoid disruptions.

2.3 Results

The new authentication and access control system enabled:

- More precise and flexible role management
- Streamlined onboarding for new users and roles
- Auditable access records and improved traceability
- Reduced manual intervention in managing permissions

3. Database Reliability and Deployment Integration

3.1 Context

The deployment process for scheduling tools had limited support for automated handling of database changes. Schema modifications were typically coordinated outside of deployment automation, increasing risk and decreasing reproducibility across environments.

Furthermore, insufficient observability of database health made it harder to detect or prevent configuration drift and runtime issues.

3.2 Contributions

To help address these gaps, I worked on database-focused enhancements that could be integrated into the CI/CD workflows. My contributions included:

- Writing **smoke tests** to validate the state of key tables and operations post-deployment
- Implementing **Kubernetes readinesss** to monitor database availability
- Standardizing **OracleDB configurations** across development, staging, and production environments

Although I did not author the pipelines themselves, these components contributed directly to making deployments more consistent and fault-tolerant.

3.3 Outcomes

The resulting improvements included:

- Better alignment between application and database lifecycle management
- Reduced risks during schema updates and releases
- Faster detection of deployment-related issues through improved health checks
- Less reliance on manual coordination between teams during rollouts

4. Technical Takeaways

Through this internship, I developed practical experience in production infrastructure, including:

- Designing and implementing RBAC in distributed systems
- Integrating database migrations and observability tooling into deployment lifecycles

- Working within Kubernetes-based cloud environments (OpenShift)
- Collaborating with cross-functional teams in a high-stakes, international setting

5. Conclusion

Although my work focused on backend infrastructure and system reliability, its impact plays out during some of CERN's most mission-critical operations. Enhancing access control and database resilience for the scheduling tools directly supports the coordination efforts required during YETS and LS3.

The experience gave me both technical growth and valuable insight into the collaborative complexity of large-scale scientific computing environments.