The ALICE Membership system

Jomar Júnior de Souza Pereira[1,*], *Adriana* Telesca[2,**], *Jose* Manuel Seixas[1,***], and *Rodrigo* Coura Torres[1,****]

[1]Signal Processing Lab, COPPE/Poli - Federal University of Rio de Janeiro, Brazil
[2]European Organization for Nuclear Research (CERN), Switzerland

Abstract. With over 2000 active members from 174 institutes over 41 countries in the world, the ALICE experiment is one of the 4 large experiments at CERN. With such numerous interactions, the experiment management needs a way to record members' participation history and their current status, such as employment, institutes, appointments, clusters and funding agencies, as well as to automatically generate the physics paper author list considering the experiment publication policy. The ALICE Glance Membership system handles all these needs and has been active for more than 8 years, helping the collaboration to organize their data, efficiently. In this document, we describe the new ALICE Glance Membership system, which is the result of breaking the monolithic code of the old version into two well-defined environments. The first is a REST API written in Object-Oriented PHP using the Slim framework to expose the data and Doctrine to access and manage an ORACLE database together with a Domain Driven Design approach to simplify the code architecture, dividing it in smaller self-contained contexts. The second is an HTTP Client written in the Vue.js framework to create a responsive and clean user interface. We will show how these changes have minimised maintenance efforts, lessened new features' development time, reduced technical depths, facilitated newcomers onboarding and also simplified the user experience.

1 Introduction

The ALICE experiment [1], one of the four major experiments at the European Organization for Nuclear Research (CERN) plays a pivotal role in understanding the properties of quark-gluon plasma and the early universe. With a vast collaboration comprising over 2000 active members from 174 institutes spanning 41 countries worldwide, participant membership data management has become a critical aspect of the experiment's success. The ALICE Membership system is the backbone for organizing and recording members' participation history, including their current affiliation, roles and funding agencies. Furthermore, this system automates the generation of author lists for ALICE's physics papers, adhering to the experiment's rigorous publication policy.

[*] e-mail: jomar.junior@cern.ch
[**] e-mail: adriana.telesca@cern.ch
[***] e-mail: seixas@lps.ufrj.br
[****] e-mail: torres@lps.ufrj.br

The Glance team [2], which emerged from the origins of a singular system created for a different large-scale experiment conducted at CERN, initiated its journey in 2003. Undergoing a rapid evolution, it adapted its solutions to accommodate multiple stakeholders within the CERN laboratory. Over time, the team has evolved into a diverse group of skilled software engineers representing various nations worldwide. Together, they now manage the development of a wide range of applications for three distinct experiments, including the notable ALICE experiment.

Over the past eight years, the ALICE Glance Membership system has been responsible for streamlining membership data management and supporting the experiment's operations. However, as the collaboration grew and technologies evolved, the challenges associated with maintaining such a dated codebase became evident.

To address these challenges, the Glance team embarked on a significant refactoring effort, resulting in a new and improved ALICE Glance Membership system centered around two key principles: Domain Driven Design [3] and Hexagonal structure [4].

2 Background

Initially, the system was built on the concept of a common framework for the entire Glance team [5], streamlining the development of interfaces and handling back-end tasks, including data retrieval and database interactions. This approach proved methodical and facilitated the implementation of new features across various projects for nearly a decade.

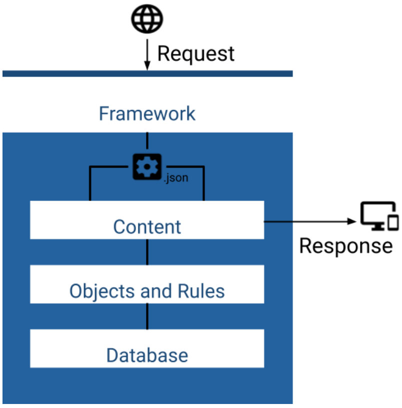

Figure 1: Original membership structure.

However, the initial structure depicted in Figure 1 adopted a monolithic design. Moreover, the adoption of an internal framework, detached from the PHP community, restricted access to emerging solutions and hindered technological advancements. The absence of comprehensive documentation further compounded the challenge, making the learning curve steep for newcomers. Finally, while providing methodical implementation for new features and projects, the systematic approach exhibit rigidity when confronted with the need for accommodating new designs and handling diverse requests. As the ALICE experiment continued to evolve and expand its collaboration, the inflexibility of the original structure crippled the seamless integration of innovative solutions and adaptability to evolving requirements.

Consequently, efforts were initiated to transition the code base to a refreshed architecture, incorporating the latest technologies and adhering to industry best practices. This transformation aimed to overcome the shortcomings of the previous system and ensure long-term sustainability and maintainability.

3 Refactoring Process

The refactoring process began with a comprehensive assessment of the existing ALICE Membership system. The team worked closely with experts to understand the key features and functionalities required for managing collaboration data effectively. Throughout this phase, constant discussions were held to identify pain points, bottlenecks and areas that needed improvement. As a result, two vital descriptions of the system were formulated: a comprehensive requirements table and an entity-relationship diagram. These valuable resources aided not only in organizing our development path but also played a significant role in defining distinct bounded contexts that the code should take into consideration.

3.1 Breaking the Monolith

To address the monolithic codebase, the team adopted a novel architecture for the refactored ALICE Glance Membership system, as illustrated in Figure 2. The new architecture was built around two key components: a Slim framework based RESTful API [6] and a Vue.js [7] based web interface. The REST API, implemented using Object-Oriented PHP [8] with Slim framework [9], interacted seamlessly with an Oracle [10] database through the Doctrine DBAL [11] library, providing a more efficient and scalable membership data management solution. Concurrently, the Vue.js based interface offered a responsive and intuitive user interface, significantly enhancing the overall user experience. During the refactoring process, the team embraced Domain-Driven Design principles. This design approach facilitated code organization into smaller, self-contained contexts identified by the previous step, further reducing technical complexities and making the codebase more maintainable.

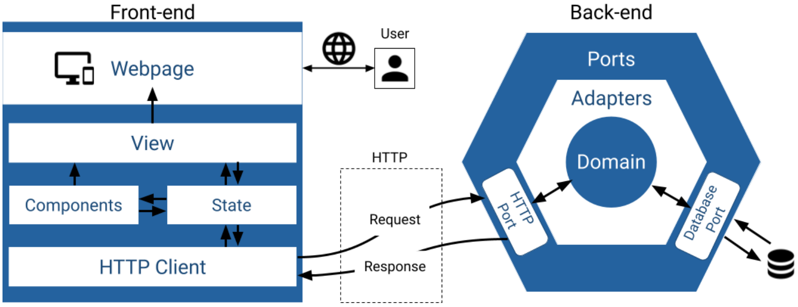

Figure 2: Refactored ALICE Glance Membership System Architecture.

3.2 Back-end refactoring

The first step in breaking the monolith involved refactoring the back-end code into a separate REST API. The team used Object-Oriented PHP to structure the API codebase and the Slim framework to handle routing and HTTP requests. The team carefully identified components that could be extracted and reused as services within the API. These components included data access modules, business logic and access control services. One noteworthy module that

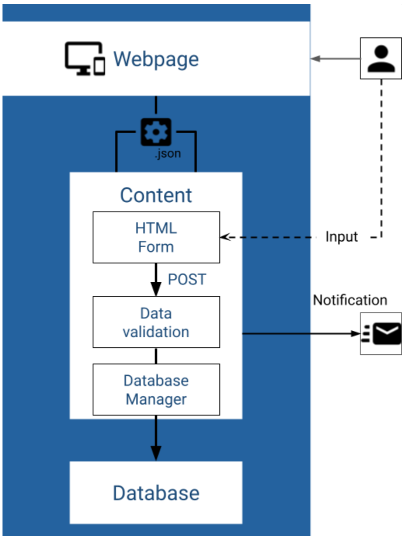

Figure 3: Original algorithm for inserting new employments.

exemplifies this approach is the Employments module, displayed in Figure 3, which has been refactored to improve its insertion functionality. The extracted components were gradually decoupled from the monolithic codebase. This required refactoring the dependencies and ensuring that the components could function independently. As a result, the employment insertion module was significantly transformed, leading to a more modular flux as depicted in Figure 4:

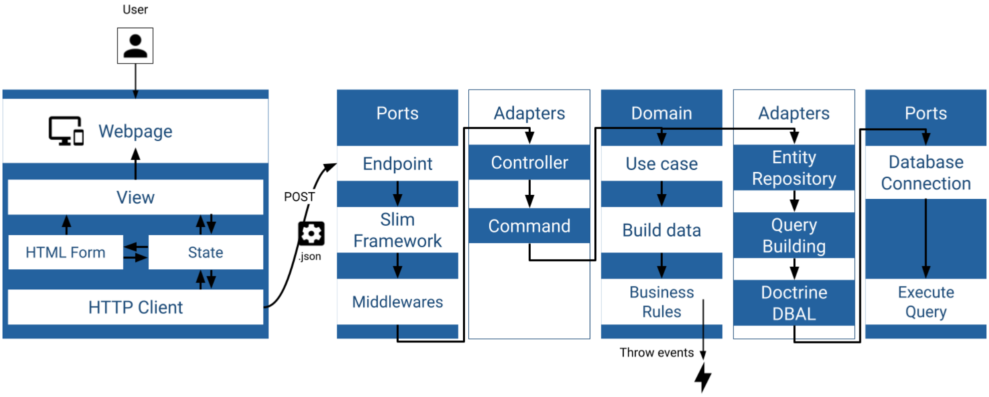

Figure 4: Refactored algorithm for inserting new employments.

3.3 Web Interface Refactoring

The second step involved refactoring the front-end user interface, which was previously tightly integrated with the back-end. The team opted to build a separate interface using

Vue.js, which is a progressive JavaScript framework chosen for its simplicity and reactivity, making it ideal for building the modern and responsive user interface. Additionally, the Vuetify.js [12] library was integrated to provide a rich set of pre-designed UI components and ensure a consistent and polished look across the application. Finally, the state management pattern and library, Vuex [13], was introduced to handle the application's state predictably and efficiently. It allowed for easy management of shared data and ensured that components could access and update the state in a standardized way.

To encourage code reusability and maintain a structured codebase, the team adopted a component-based architecture in the Vue.js application. Each user interface element, such as a form, table, button or dialogue box, was encapsulated within a Vue component, making it easy to reuse across different parts of the application.

3.4 User Experience (UX) Enhancements

The refactoring of the ALICE Glance Membership system brought about significant improvements in user experience, liberating the interface from its rigid constraints and ushering in a modular and modern design.

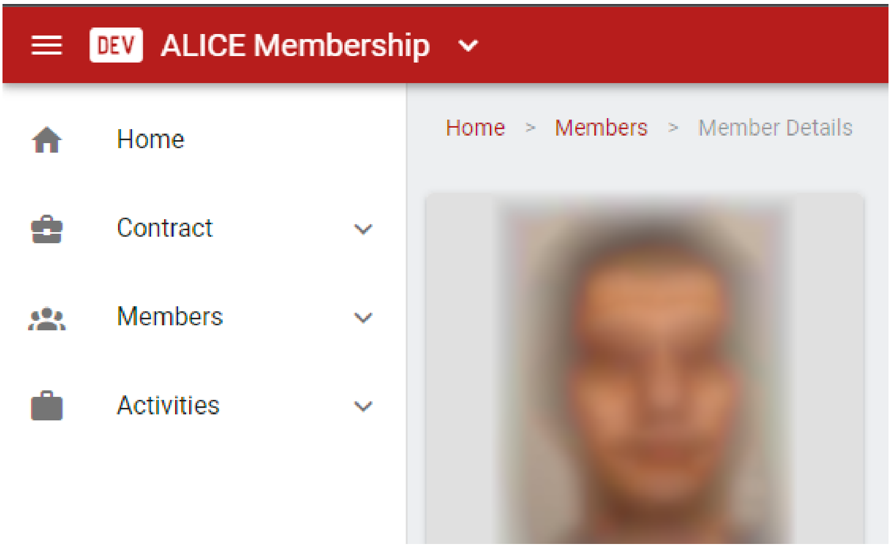

Figure 5: New and refreshed user interface.

3.4.1 *Streamlined navigation*

The old system suffered from cumbersome navigation, making it challenging for users to find their way around the application. In contrast, the refactored system implemented a streamlined navigation structure, as shown in Figure 5. A user-friendly menu, accessible from a fixed sidebar, provided quick access to different modules. Additionally, employing breadcrumb navigation improved users' ability to understand their current location within the application and facilitated seamless movement between different sections.

3.4.2 Responsive and Mobile-Friendly Design

Recognizing the importance of a responsive user interface in today's mobile-driven world, the new ALICE Membership system was developed with mobile-friendliness in mind. The implementation of the Vue.js framework for the web interface enabled a dynamic and adaptable user interface that effortlessly adjusted to various screen sizes and devices, as seen on Figure 6. This responsiveness allowed users to access and manage collaboration data conveniently, even on the go.

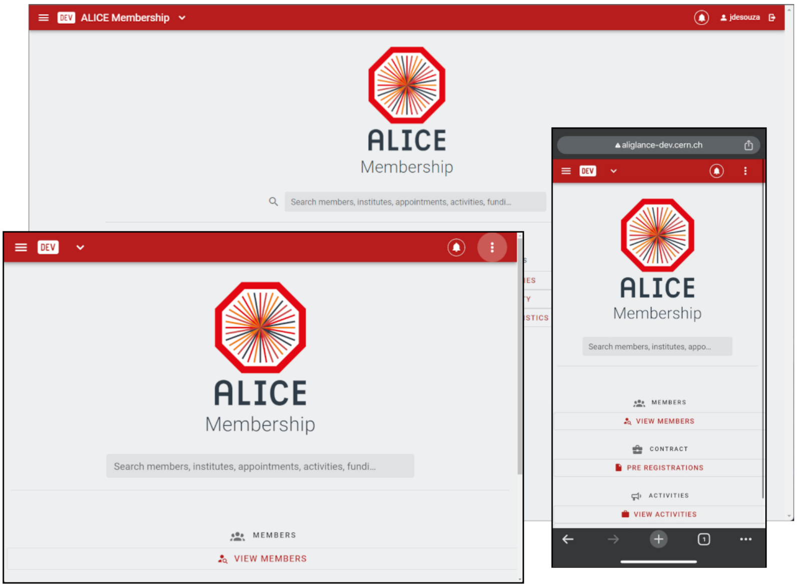

Figure 6: The app displayed in many viewports.

4 Conclusion

The refactoring of the ALICE Glance Membership system has been a transformative journey, addressing the challenges of a dated codebase and paving the way for a modern and sustainable membership data management solution. The adoption of Domain-Driven Design and Hexagonal structure principles, along with the incorporation of modern technologies, has resulted in a more flexible, maintainable, and user-friendly system.

By breaking the monolithic design into a RESTful API and a Vue.js based web interface, the team achieved greater code modularity and improved scalability. The use of Object-Oriented PHP, Slim framework, and Doctrine DBAL in the back-end, coupled with Vue.js, Vuetify.js, and Vuex in the front-end, has enhanced the overall performance and responsiveness of the application.

The refactoring process not only improves technical aspects but will also bring significant enhancements to the user experience. The streamlined navigation, responsive design and component-based architecture have made it easier for the ALICE experiment members to access and manage collaboration data efficiently.

With these advancements, the refactored ALICE Glance Membership system is now better equipped to support the collaboration's ever-growing needs. It offers a solid foundation for future developments, ensuring the continued success of the ALICE experiment in unraveling the mysteries of quark-gluon plasma and the early universe. As the experiment evolves, this refactoring effort serves as a testament to the team's commitment to continuous improvement and innovation in membership data management and scientific research.

Acknowledgments

The authors would like to thank CNPq, CAPES, FAPERJ and RENAFAE (Brazil), as well as CERN and the ALICE collaboration for providing financial support for this work. We also thank Mr. Joel Closier for supporting the development of the presentation and paper.

References

[1] The ALICE Collaboration, JINST **3**, S08002 (2008)

[2] C. Maidantchik, F.F. Grael, K.K. Galvão, K. Pommès, J. Phys.: Conf. Ser. **119**, 042020 (2008)

[3] E. Evans, *Domain-Driven Design: Tackling Complexity in the Heart of Software.* (Addison-Wesley Professional, Boston, 2004)

[4] A. Cockburn, *Hexagonal Architecture* (2005), URL https://www.alistair.cockburn.us/hexagonal-architecture/ [accessed 21-Jul-2023]

[5] B. Lange, C. Maidantchik, K. Pommes, V. Pavani, B. Arosa and I. Abreu. An object-oriented approach to deploying highly configurable Web interfaces for the ATLAS experiment **J. Phys.: Conf. Ser.** 2015

[6] L. Richardson, M. Amundsen and S. Ruby *Restful Web Apis.* (O'Reilly, 2013)

[7] *Vue.js - The Progressive JavaScript Framework*, URL https://www.vuejs.org/ [accessed 21-Jul-2023]

[8] The PHP Group. *PHP: Hypertext Preprocessor*, URL https://www.php.net/ [accessed 18-Aug-2023]

[9] The Slim Framework Team. *Slim Framework*, URL https://www.slimframework.com/ [accessed 21-Jul-2023]

[10] Oracle. *Oracle | Cloud Applications and Cloud Platform*, URL https://www.oracle.com/ [accessed 18-Aug-2023]

[11] Doctrine Project. *Database Abstraction Layer*, URL https://www.doctrine-project.org/projects/dbal.html [accessed 21-Jul-2023]

[12] *Vuetify - A Vue Component Framework*, URL https://vuetifyjs.com/en/ [accessed 21-Jul-2023]

[13] *What is Vuex?*, URL https://www.vuex.vuejs.org/ [accessed 21-Jul-2023]