Enhancing data consistency in ATLAS and CERN HR databases through automated synchronization

Gabriel de Aragão Aleksandravicius[1,*], *Ana Clara* Loureiro Cruz[1,], *Carolina* Niklaus Moreira da Rocha Rodrigues[1,], *Gabriela* Lemos Lúcidi Pinhão[2,], *Pedro Henrique* Goes Afonso[1,], *Rodrigo* Coura Torres[1,], and *José Manoel* Seixas[1,**]

[1]Signal Processing Lab, COPPE/EE - UFRJ (Federal University of Rio de Janeiro)
[2]Laboratório de Instrumentação e Física Experimental de Partículas - LIP, Lisboa

Abstract. As the largest particle physics laboratory in the world, it is no surprise that CERN has a vast network of thousands of collaborators spread globally. Among CERN's experiments, ATLAS stands out with over 6,000 active members and 300 associate institutes. This extensive community must go through the standard registering and maintenance procedures within CERN's human resources database, called Foundation. Simultaneously, the Glance system, among other specific functions, also aims to fulfill the same objectives within the context of different CERN experiments, such as ATLAS, LHCb and ALICE. It achieves that by having its own database. Since members need to exist in the two databases, a lot of data was being duplicated. Manual updates by the ATLAS secretariat became necessary to maintain consistency over members and institutes data, such as names, employment information and authorship status. Today, the Glance system undergoes a transformative process to redefine its relationship with Foundation. The goal is to eliminate duplication of data by establishing a single source of truth. At the same time, the automation of a series of internal processes will be made to ensure synchronization between the two databases at all times, thus removing the need for manual intervention from the ATLAS secretariat. This requires some restructuring to Glance's database, updates to the code and overall implementation of new tools that facilitate seamless communication with Foundation.

1 Introduction

CERN, the European Organization for Nuclear Research, is a world-renowned scientific institution dedicated to pushing the boundaries of particle physics research. Since its origin in 1954, CERN has been at the forefront of groundbreaking discoveries and technological advancements in the field. Thousands of scientists, engineers, technicians and personnel engaged in various administrative functions, both onsite and distributed across the globe, collectively contribute to the dynamic workforce at CERN.

One of the major particle physics experiments conducted at CERN is ATLAS [1]. Its primary goal is to study the building blocks of the universe, which is achieved by investigating the collisions of high energy protons and heavy ions. Given its magnitude, ATLAS currently has over 6,000 active members that work together to design and operate it.

[*] e-mail: gabriel.aleks@cern.ch

1.1 Foundation

To support its extensive operations and facilitate efficient management across its collaborations, such as ATLAS, CERN employs different systems and databases. One such is Foundation, a centralized database used to store, among many other types of data, those related to personnel, such as name, nationality, institute affiliation and start and end date of employments. The access and modification to data is limited for CERN's administrative groups through a dedicated web page. So, generally, when a regular ATLAS member wants to have some sort of data updated, they don't go to Foundation and do it by themselves. Instead, they submit an update request to the ATLAS secretariat, who then proceeds with the modification. Given its centralized role within CERN, Foundation data can be accessed by other CERN systems, which are the ones actually used by members for data visualization.

1.2 ATLAS Glance

On the other hand, ATLAS Glance [2] is a system that can be used by any member within the ATLAS collaboration. It is employed by ATLAS members to visualize data and perform operations, such as registering an ATLAS member, going through a specialized workflow for the publication of a paper and managing roles within the collaboration. To achieve all of its goals, Glance has its own database where it stores ATLAS-relevant data.

1.3 Interaction between systems

From ATLAS' perspective, the utilization of an ATLAS-specific system and database to store collaboration-related data proves highly advantageous. While Foundation serves a more general purpose and does not fully meet the experimental requirements, Glance offers a multitude of benefits, ranging from presenting data to users in a standardized manner to establishing intricate workflows for various ATLAS-related processes.

Although both systems independently handle their respective processes and store unique data, overlaps concerning personnel employment information exist and encompass the following three items:

- Registration of new ATLAS members.
- Updates to member contact information, such as CERN mobile number, CERN phone number or CERN office.
- Modifications to member employment end dates.

2 Employment-related processes

Due to the overlapping previously mentioned, the implementation of a specific workflow to manually synchronize both databases is needed, as illustrated in Figure 1. The figure contains the three employment-related processes that require manual operations, being them from the ATLAS secretariat or from a member's team leader, who acts as their institute's representative.

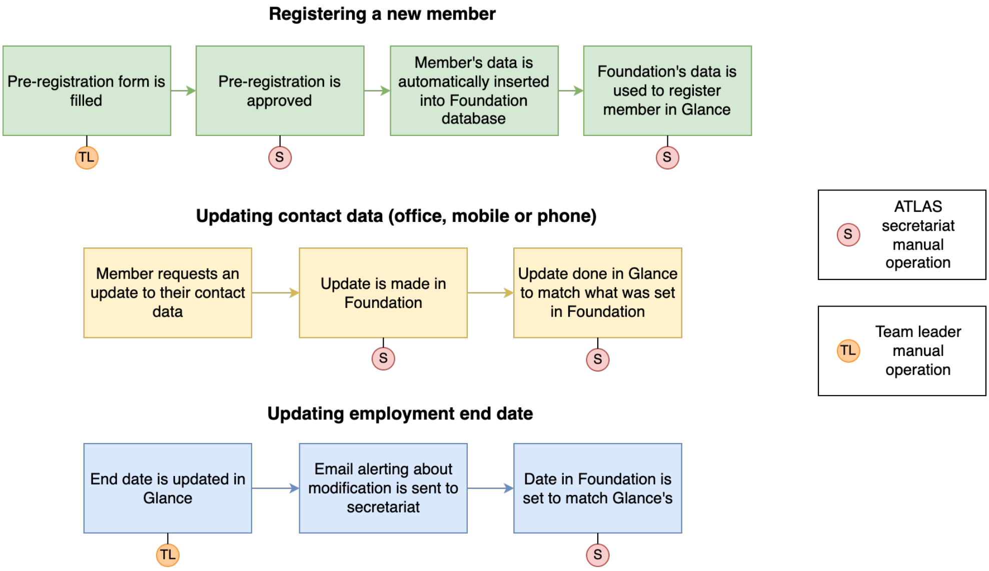
Registering a new member
Pre-registration form is filled
TL
Pre-registration is approved
S
Member's data is automatically inserted into Foundation database
Foundation's data is used to register member in Glance
S
Updating contact data (office, mobile or phone)
Member requests an update to their contact data
Update is made in Foundation
S
Update done in Glance to match what was set in Foundation
S
Updating employment end date
End date is updated in Glance
TL
Email alerting about modification is sent to secretariat
Date in Foundation is set to match Glance's
S
S ATLAS secretariat manual operation
TL Team leader manual operation

Figure 1. Workflow for employment-related processes and the manual steps involved to achieve synchronization.

The green boxes represent the process to register a new ATLAS member. To do that, the team leader fills out an online document called PREG (pre-registration). This document contains essential information about the member, such as their gender, name, country, email, institute, and employment dates. Once the PREG form is complete, it undergoes review by the ATLAS secretariat. If the provided information meets the necessary requirements, the document is approved. As a result, the approved data from PREG is synchronized with Foundation. The final step involves manually registering the member in Glance, using the same data entered into Foundation. This task is performed manually by the secretariat.

Next, the yellow boxes explain the process to update contact data of a member. In the CERN Phonebook web page[3], CERN members can access office information, CERN mobile number and CERN phone number for any member in the collaboration. The data comes from Foundation and can be propagated to other CERN systems as well. ATLAS Glance, however, does not take advantage of this and has its own version of the contact data. Whenever a member wishes to update their contact information, they must contact the ATLAS secretariat, who will manually make the changes in Foundation and subsequently synchronize the updates with Glance.

Finally, the blue boxes represent the process to update the employment end date of a member. Each CERN member has a contract with a start and end date. During the registration process, these dates are recorded in both Foundation and Glance, as mentioned earlier. If a member wishes and it is agreed by their home institute and CERN, a contract extension can be granted. Once the required administrative processes are completed, the team leader will update the member's employment end date in Glance. This action triggers a notification to the secretariat, prompting them to propagate the updated date to Foundation.

3 Current approach issues

The three aforementioned items involve certain aspects of duplication and overlapping, either in terms of the stored data or the operations performed by the secretariat. This duplication brings significant challenges for the following reasons:

- Maintenance cost: Duplicated data in Glance and Foundation requires the ATLAS secretariat to waste a lot of their resources into manually synchronizing one database whenever there is an update in the other one.
- Inconsistency: Having the same data stored in multiple databases makes it difficult to ensure that all copies are consistently updated. Even though an effort is made to keep consistency and propagate changes, errors will inevitably occur, creating inaccuracies between the data sets and undermining its integrity over time.

Instead of duplication, the data could be managed by having a Single Source of Truth (SSOT) [4]. This can be achieved by setting it in one central place and enabling it to be updated and read from external systems. Some data will have its source of truth set as Foundation, while some other as Glance.

The next section will focus on presenting the general tools that will be used to bring to life a more efficient and reliable approach to managing data. Then, with those tools in hand, each process will be individually analyzed and optimized.

4 General implementation details

In order to address the issues discussed in the previous section, the proposed solutions will be implemented following a Domain-Driven Design (DDD) [7] approach for the codebase. The system architecture will be structured into three layers: Domain, Application and Infrastructure. Each layer encapsulates specific functionalities, ensuring flexibility and modularity within the system.

The Domain layer will contain all the entities that belong to a specific bounded context along with their business rules, as well as value objects [5], domain events [6], and abstractions of the Infrastructure. Separating domain logic from external dependencies and infrastructure matters enables a domain-first development approach and promotes effective testing strategies.

Moving on to the Application layer, its main responsibility is to coordinate the interaction between the domain and infrastructure layers and to create command objects, which represent the user's intention with the application. It contains Data Transfer Objects (DTOs) [8], also called command objects, command handlers and event subscribers that listen to domain events. It is in this layer where the application use cases are implemented, managing the flow of data and operations within the system.

Finally, the Infrastructure layer acts as a bridge between the application and external services and resources, thus enabling data persistence and retrieval. It contains the implementations of write and read model repositories, which handle the communication with external databases, such as Glance's or Foundation.

By following this layered architecture and the DDD established principles, the solutions proposed in the next sections can be effectively implemented in a way that promotes modularity, testability and maintainability [9].

5 Improving data consistency

Now that the general tools to streamline operations between Glance and Foundation have been covered, we can proceed on analysing specifically each process. The next subsections will be dedicated to presenting solutions that enhance the data integrity between the two databases and improve the efficiency of ATLAS secretariat's workflow.

5.1 Streamlining an ATLAS member registration

Completing the registration process entails two actions performed by the secretariat: approving the PREG document and manually registering the member in Glance. However, since the approval of the PREG document also means that the registration is complete regarding Foundation, it should ideally be sufficient to automatically register the member to Glance. The ideal approach involves automating the member's registration within ATLAS, eliminating the need for manual intervention by the secretariat. This could be achieved by developing a script that registers a member to ATLAS whenever they have been added to Foundation.

The basic information for a member to be registered in Glance is their institute, profession, period of employment, first and last names and email. All of that is inserted into Foundation after the PREG is approved. So, since Foundation data can be fetched from Glance with the use of a Database view, the registration could proceed in-code after it is detected that a pre-registration document has just been approved. For this, Foundation provides an API endpoint with information regarding approved PREGs.

```
[
    {
        "personId": 123456,
        "documentId": 2127421,
        "createdTime": "2023-01-15T13:34:27Z",
        "participationExperiment": "ATLAS",
        ...
    }
]
```

Figure 2. Generic response from the PREG API.

Figure 2 is what a generic response from the API looks like. The `personId` is the unique identifier of a CERN member. It can be used to access a member's record in Foundation and retrieve the necessary data for a registration in Glance.

With all of that in hand, recently accepted PREGs can now be used to register members in Glance. This can be achieved by setting up a cronjob that runs hourly, fetching PREG entries from the current day and checking if the member with the corresponding `personId` is already registered in Glance. If not, then the necessary information is retrieved from Foundation and the member will be registered. Automating this process with a cronjob ensures that new member registrations are always promptly processed and integrated to Glance in a standardized way.

To use the PREG API, a separate service was developed and installed as a dependency in the application. Then, a cronjob was created implementing this service to fetch PREG entries. A use case to register ATLAS members already exists. It uses the data provided by the ATLAS secretariat whenever a new member needs to be registered through Glance's web interface, so its command DTO and command handler could be leveraged into the cronjob. The data mapper design pattern [10] was employed to facilitate the registration: a member's basic information is fetched from Foundation and a Member entity object is constructed using that data. The responsibility of persisting this information to the database lies with the repository within the Infrastructure layer.

Following this approach, the application benefits from the reusability of existing components while ensuring an efficient automated member registration process. The data mapper pattern provides a seamless integration of data from Foundation into the application's domain model, ensuring that all business rules are correctly followed and providing a robust solution for the registration.

5.2 Setting a single source of truth for contact information

The contact information, which includes the CERN office, CERN mobile number and CERN phone number, is an essential component associated with every CERN member. There is no need for this type of data to be duplicated in Glance's database. Instead, Glance could retrieve and display the relevant information from Foundation in a read-only mode, eliminating the need for redundant data storage. By adopting this approach, Glance would effectively serve as a user-friendly interface for accessing the most up-to-date contact information while relying on Foundation as the original source of this data.

To establish Foundation as the sole source of information for contact data, some modifications need first to be made to the database. The CERN office, CERN mobile number and CERN phone number should be selected from Foundation and not duplicated in Glance's database. To reflect these changes, the code should be adjusted. This approach ensures consistency and accuracy by eliminating redundant data storage after reading the most up-to-date contact details.

5.3 Synchronizing employment end dates automatically

Whenever a member has their employment end date updated in Glance, the new value should be propagated to Foundation. After such an update happens, an email is sent to the secretariat containing the member's name, the old date and the new one. Subsequently, the secretariat searches for the corresponding member in Foundation and matches the employment date with the updated value from Glance.

This process could be automated to eliminate the need for manual synchronization. These are the requirements:

- Each member is assigned a specific CERN status, but only members with the PART (meaning 'participant to the experiment') status can have their end date propagated from Glance to Foundation.
- It is allowed for an end date to be set in Foundation to a maximum of 5 years in the future. Glance, on the other hand, does not impose any limit. A team leader should be able to set any date they want in Glance as long as it is after the start date and does not create overlapping employments for the member.
- Notifications should be triggered during the update process. The secretariat should be informed every time an update is made to Glance. If the member is of PART status, the notification should indicate whether the propagation to Foundation was successful or not.

By automating the update process in the two databases for employment end dates while addressing these requirements, the effort from doing manual synchronizations will be eliminated and ensure consistency. First, the user, in the web interface, updates the end date. This creates a command object that will be handled by a service in the Application layer. It incorporates an 'update employment' use case responsible for handling the modification of each attribute of the Employment entity. Following the data mapper design pattern, this process entails the retrieval of relevant data from the corresponding object in the database, making the necessary updates within the Domain layer, and subsequently persisting the modified object back to the database.

To ensure the propagation of the updated employment end date to Foundation, an event driven approach was adopted [11]. Domain events were introduced within the method responsible for updating the Employment entity's end date. Depending on the member's CERN status and the new end date value, the appropriate event will be recorded. It will determine if the date should be propagated or not to Foundation. Upon completing the update of the Employment entity and all associated database processes within Glance, the event is dispatched and handled by its respective subscriber. If the member's CERN status is not PART, the subscriber will be responsible for sending a notification to the secretariat informing about the update made in Glance. However, if the member's status is PART, the event subscriber acts as another application service. It retrieves a Foundation Employment entity from that database into its correspondent entity, updates the end date and persists the modified entity back to Foundation. Another event is published during the update of the Foundation Employment entity, this time related only to triggering a notification. If the operation runs successfully, the secretariat is warned about an update made both to Glance and Foundation. If it fails, the message will say that Glance was updated but the date could not be pushed to Foundation and will request for a manual update. This whole process is represented in Figure 3.

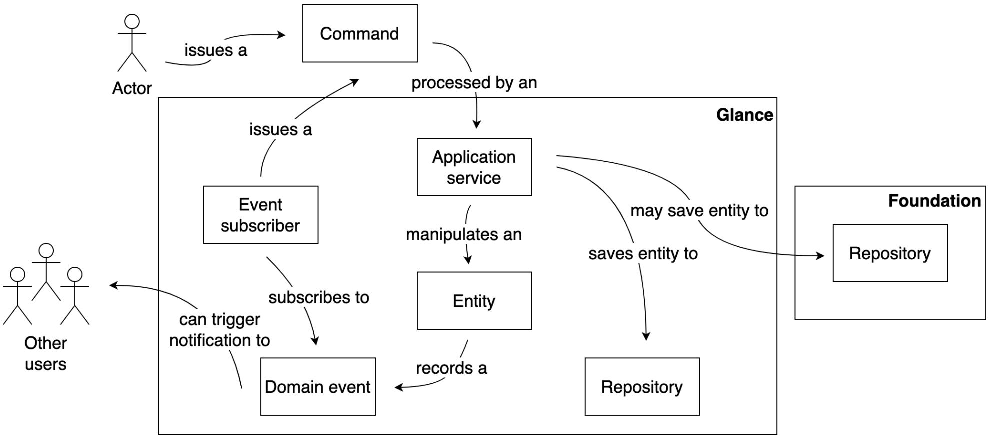

Figure 3. Diagram for actions that take place when updating a member's end date.

Since Foundation is an external database, Glance does not have access to it directly. To be able to update end dates, a SQL procedure [12] was provided. It establishes the necessary communication with Foundation and enables the synchronization of end dates.

6 Current status

The optimizations discussed on Section 5.1 and Section 5.2 have already been developed and deployed. Following the automation of the registration process, the head of the secretariat reported significant time savings, estimating that over an hour of work per day for their team has been saved. Additionally, considerable effort has been spared by designating Foundation as the definitive source for contact information.

Regarding the implementation mentioned in Section 5.3, work is currently underway. It is anticipated that upon completion, an additional 30 minutes per day of the secretariat's work will be saved by eliminating the manual data entry into Foundation.

7 Conclusion

Glance is used within the ATLAS collaboration to handle personnel, workflow for publications, roles assignment and more. Foundation, a database used by CERN's administrative bodies, also encompasses a lot of the same goals, which leads to data duplication.

Due to the high maintenance cost and the potential for data inconsistencies, since manual interventions have to be made to synchronize the two databases, a solution was developed. By adopting a Domain-Driven Design approach and leveraging tools provided by the Foundation development team, such as database views for data retrieval, an API for accessing approved PREGs and a SQL procedure for updating Foundation, Glance became programmatically able to synchronize itself with Foundation, thus setting a single source of truth for the data. This integration streamlined operations, enhanced data integrity and significantly improved the efficiency of the ATLAS secretariat's workflow.

Acknowledgments

The authors would like to thank CNPq, CAPES, FAPERJ and RENAFAE (Brazil), as well as CERN and the ATLAS 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] ATLAS Collaboration, JINST 3, S08003 (2008). DOI: 10.1088/1748-0221/3/08/S08003

[2] C Maidantchik, F F Grael, K K Galvão and K Pommès. Glance project: a database retrieval mechanism for the ATLAS detector. **J. Phys.: Conf. Ser.**, 2008. DOI: 10.1088/1742-6596/119/4/042020

[3] *CERN Phonebook* (https://phonebook.cern.ch/, Accessed 28-Jul-2023).

[4] *What is a single source of truth (SSOT)? | Dropbox* (https://experience.dropbox.com/resources/source-of-truth, Accessed 28-Jul-2023).

[5] *ValueObject | Martin Fowler* (https://martinfowler.com/bliki/ValueObject.html, Accessed 28-Jul-2023).

[6] *Domain Events | Martin Fowler* (https://www.martinfowler.com/eaaDev/DomainEvent.html, Accessed 28-Jul-2023).

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

[8] *Data Transfer Object | Martin Fowler* (https://martinfowler.com/eaaCatalog/dataTransferObject.html, Accessed 28-Jul-2023).

[9] M Noback. *Advanced Web Application Architecture*. 2020. ISBN: 978-90-821201-6-5.

[10] *Data Mapper | Martin Fowler* (https://martinfowler.com/eaaCatalog/dataMapper.html, Accessed 28-Jul-2023).

[11] A Bellemare. *Building Event-Driven Microservices: Leveraging Organizational Data at Scale* (O'Reilly Media, Inc, 2020)

[12] *Procedures and Packages | Docs Oracle* (https://docs.oracle.com/cd/A57673_01/DOC/server/doc/SCN73/ch14.htm, Accessed 28-Jul-2023).