The Glance project common infrastructure dependencies upgrade from the ATLAS Glance perspective

Pedro Henrique Goes Afonso[1,*], *Ana Clara* Loureiro Cruz[1], *Carolina* Niklaus Moreira da Rocha Rodrigues[1], *Gabriela* Lemos Lúcidi Pinhão[2], *Leonardo* Mira Marins[1], *Luis Guilherme* Neri Ferreira[1], *Natanael* Nunes de Moura Junior[1], *Rafaella Lenzi* Romano[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. The Glance project provides software solutions for managing high energy physics collaborations' management data and workflow. It was started in 2003 and operates in the ALICE, AMBER, ATLAS, CMS, and LHCb CERN experiments on top of CERN common infrastructure. The project develops Web applications using PHP and Vue.js, running on CENTOS virtual machines hosted on the CERN OpenStack private cloud. These virtual machines are built via Puppet for installing and configuring core software while tailoring them to meet each experiment's requirements in a collaborative approach under the Glance Project. This approach minimizes redundant work across experiments while allowing cooperation when responding to operations incidents. In the scenario of the CENTOS 7 end-of-life, the Glance project has chosen to migrate to RHEL9 while undergoing a major upgrade of PHP from (7.3 or 7.4 to 8.2) across the experiments. This presentation will expose the technical and organizational challenges the Glance project faces on common dependencies upgrades from the perspective of the ATLAS Glance team.

ATL-SOFT-PROC-2025-009
14 January 2025

1 Introduction

CERN, the European Organization for Nuclear Research, is a leading institution in particle physics, founded in 1954. One of the projects hosted at CERN is ATLAS [1], at the Large Hadron Collider (LHC) which focuses on exploring the fundamental components of the universe using high-energy protons and heavy ions collisions. With over 6,000 active members, including scientists, engineers, and technicians who work both onsite and globally, the ATLAS collaboration has made significant discoveries, one of them being the Higgs boson in 2012 [2]. Given its extensive membership and the broad scope of activities, the ATLAS collaboration faces the challenge of managing member information, enforcing internal policies, and following workflows, which can be addressed through the adoption of computer information systems.

*e-mail: pedro.afonso@cern.ch

2 The ATLAS Glance Project

The Glance project [3] started in ATLAS as an interface to the collaboration data, later evolving into an integrated ecosystem that manages data for the four LHC experiments: ALICE, ATLAS, CMS, and LHCb, and even other CERN experiments such as AMBER and SND@LHC. The ATLAS Glance project is responsible for a variety of applications, embracing a wide range of systems. For example, as shown in Figure 1, the ATLAS Publication Tracking system is designed to improve the user experience by automating and centralizing all the steps needed for the approval and submission of ATLAS publications, while ensuring adherence to its regulations and procedures. The Membership system manages the administration of ATLAS members, institutes and author lists. Finally, the Speakers system was built to manage the selection of ATLAS members to present results at conferences, and to streamline and consolidate this process. These systems are the main examples of the complex applications the ATLAS Glance project manages.

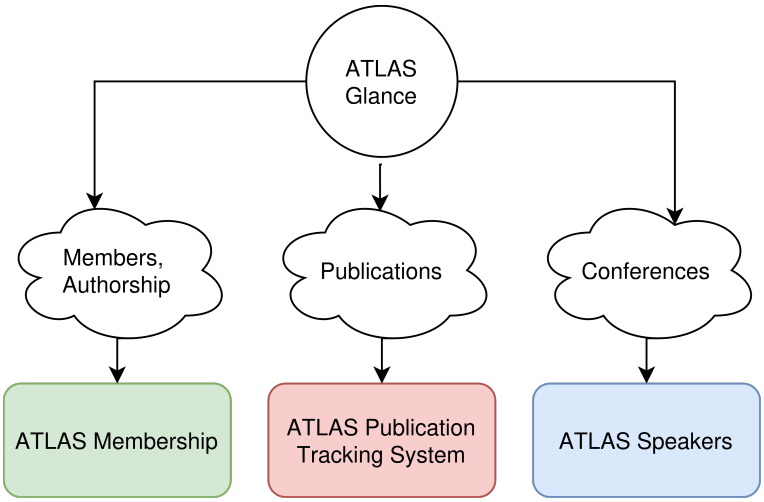

Figure 1: Atlas Glance core domains and associated systems.

The Glance project applications require a robust infrastructure to perform efficiently, including computing resources provisioned by virtual machines (VMs) and a network that allows users to access the applications from the CERN campus or remotely. In order to ensure the security of sensitive data and processes, access to the applications must be restricted to authenticated users. For data persistence, a reliable database system is essential, ensuring that all application data is stored and can be retrieved as needed.

3 The Glance Project Infrastructure

To develop and maintain the Glance project's infrastructure, regular meetings are held to facilitate coordination among the Glance teams of each experiment, ensuring alignment on project goals, updates, and resolution of technical challenges. One common topic in these meetings is the management of virtual machines on CERN's Openstack [4] private cloud using Puppet [5], an automation tool that helps standardize and streamline the deployment and configuration of machines. The Glance teams experiments share a Puppet module, which

includes basic configurations for a virtual machine such as the operational system configuration, network settings, software installation, user permissions management and other system administration tasks. Puppet manifests are also leveraged to apply configurations tailored to the specific requirements of each experiment.

The common Glance project infrastructure also includes CERN-provisioned Oracle Database [6] instances to store the experiments' data and support application development. The team has deployed schemas in the development, integration and production database instances to ensure reliable operation and management of complex schemas. The Glance project common infrastructure and its core technologies are shown in Figure 2 .

In the context of quality assurance, the Glance team employs Docker [7] images within Continuous Integration (CI) pipelines to create isolated environments for building and testing applications. In addition, Sentry[8] instances monitor and track application errors in real-time, facilitating quick detection of issues.

This combination of a common infrastructure and a collaborative approach allows the Glance project to deliver reliable software solutions to high-energy physics collaborations by minimizing redundant efforts and promoting cooperation. This approach aims to achieve high standards of performance and reliability, in order to support the demanding needs of these collaborations.

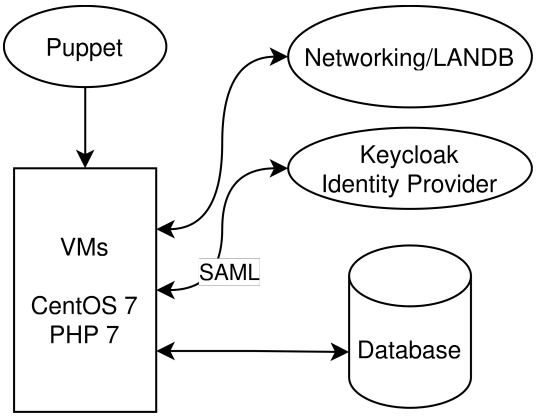

Figure 2: Glance team common infrastructure.

4 Context

The ATLAS Glance infrastructure is divided into four environments, each with its own characteristics and purpose. The development environment relies on Docker images which give each developer full control over their local setup without needing centralized management. The quality assurance environment uses a continuous integration (CI) pipeline in GitLab with Docker containers that run automated tests in isolated contexts. The integration environment is set up with a Puppet-managed CERN-provisioned VM that integrates code changes and allows testing in a similar setting to production. Finally, the production environment operates on a Puppet-managed VM that serves the applications to end-users. These VMs and Docker containers must support multiple technologies, including PHP applications, the legacy framework FENCE [9] built in PHP and jQuery by the Glance team, Vue.js applications, and Python scripts.

On June 30th, 2024, the CentOS 7 operating system installed on the Puppet-managed CERN-provisioned VMs stopped receiving software updates [10], which made the operating system progressively more vulnerable to security flaws and obsolete. In this context, the Glance team opted to migrate to Red Hat Enterprise Linux 9 due to its similarity to CentOS 7 in configuration, offered tools, and secured integration with CERN services. In order to take advantage of the necessary VM configuration and software re-installation associated with this change, the Glance team also decided to upgrade PHP from version 7.3 to 8.2.

The migration was planned to be completed in two phases as shown in Figure 3: The first phase tackled the Web servers operating system migration. This needed to be completed before the CentOS 7 end-of-life on June 30th, 2024 since the underlying VMs rely on services maintained by CERN IT, Single Signing-On and Networking integration, which would no longer support CentOS 7 virtual machines after this date. The second phase addressed the upgrade of major versions of PHP (from 7.3 to 8.2) which had a high probability of affecting application behavior and causing software malfunction.

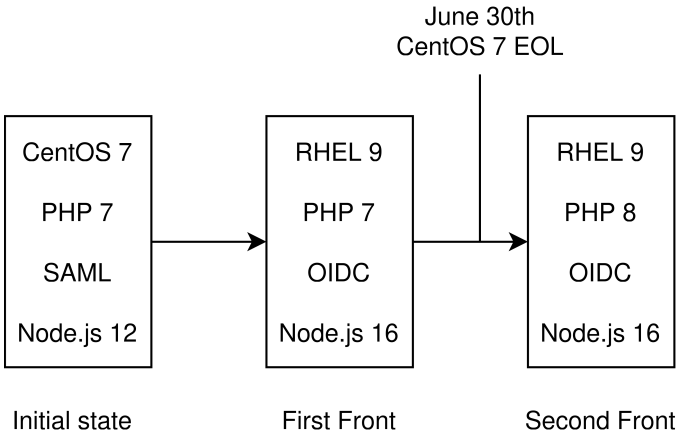

Figure 3: Migration overview.

5 Web Server Operating System Migration

The first phase began with the update of the Puppet module shared by the Glance team. This module installs and configures core dependency libraries along with previously mentioned CERN services such as the CERN networking service LANDB and the authentication protocol Open ID Connect [11] (OIDC) Single Sign-On. Moreover, this module sets up the common Glance project application stack dependencies which include PHP together with its extensions such as OCI8 and SOAP.

The migration of the integration with CERN services presented a challenge involving the transition of the authentication protocol from Security Assertion Markup Language to OIDC since Shibboleth [12] is not supported on RHEL 9 at CERN [13]. In order to address this, a development VM running CentOS 7 was built to assist in the implementation of the authentication protocol change in Apache, the configuration of a host in CERN's Keycloak-based [14] Identity Provider and the environment variables read by the application.

In this phase, the Glance team migrated Node.js [15] from version 12 to 16 due to the end of support for version 12 on RHEL 9 [16]. Node.js is used to build production Vue.js appli-

cations and run the development environment for Vue.js front-end Dockerized applications. Although this upgrade introduced challenges, particularly with the transition from *npm* [17] version 6 to 8, the overall effort was significantly smaller when compared to the PHP version migration, so the Node.js upgrade was included in this step. The web server OS migration intermediate states are illustrated in Figure 4.

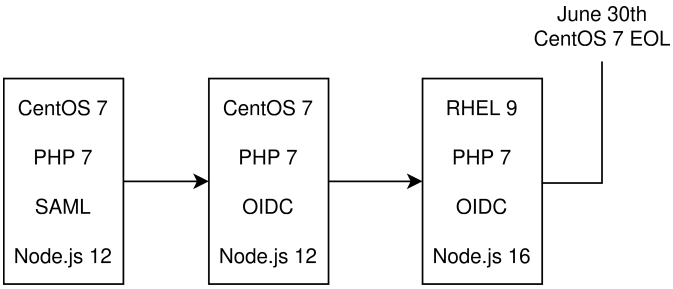

Figure 4: Web Servers OS migration overview.

To implement the first phase changes in the production environment, a new development virtual machine running RHEL 9 was built incorporating the CERN services setup from the development CENTOS 7 VM. After making necessary adjustments to configure OIDC in RHEL 9, the old CENTOS 7 production machine was shut down and a new production RHEL 9 machine was built from the development RHEL 9 machine and assigned the alias of the previous machine so that it would handle requests targeted to the application. To preserve relevant data, a snapshot of the old CENTOS 7 instance was taken using OpenStack before deleting it.

In parallel, a development RHEL 9 VM was built to install and configure PHP 8 with the FastCGI Process Manager rather than as an Apache module in order to decouple the Apache web server from the PHP runtime for more flexibility. The RHEL 9 virtual machine was used in the second phase to implement and test the changes related to the application stack migration.

6 Application Stack Migration

The second phase began with upgrading the Docker images used in the development environment and in the GitLab CI pipeline to the PHP 8 FastCGI Process Manager. This allowed the ATLAS Glance developers to work on the application upgrades independently of the first phase, and to validate the changes on the GitLab pipeline. Then, the developers addressed the upgrade of dependency packages to versions compatible with the new PHP runtime, of which the main challenge was the SLIM framework [18] upgrade. Finally, the Glance team code base was migrated to PHP 8 taking into consideration syntax changes and deprecation of features.

In order to migrate the Glance code base to PHP 8, the Rector [19] tool was used to automatically adapt the code and then run automated tests to uncover issues that developers have investigated and fixed.

Version 2 of the SLIM framework does not support PHP 8 [20], while versions 3 and 4 introduce behavioral changes that break the Glance Team's customized implementation (FRAPI) on top of it. To solve this issue, the ATLAS Glance team has made a fork of the

SLIM version 2 repository into a *glance-slim* implementation that will be migrated using the same process as the Glance code base.

Thereafter, the developers performed a round of manual acceptance tests in which recently integrated members of the team had the chance to interact with a wide range of the application's feature set looking for bugs. These bugs were prioritized and tackled in a task force to which all developers of the team contributed.

To implement the second phase changes in the production environment, the team modified the RHEL virtual machine's Puppet hostgroup to install and configure PHP 8 instead of PHP 7 and changed the stored versions in the package tracking tools (*composer* and *npm*) to install the upgraded versions of the code base and dependencies.

7 Conclusion and next steps

The migration to RHEL 9 and the PHP version upgrade mark significant milestones for the Glance project. By successfully transitioning from CentOS 7, the Glance project has ensured continued security and stability for our infrastructure, while the updates to PHP 8.2 enhance compatibility, introduce new features, and improve security. These upgrades required meticulous planning, coordination among the various CERN collaborations, and execution, including comprehensive testing to ensure seamless operation across all Glance experiments.

Moving forward, the ATLAS Glance project's primary focus will be on further refining our infrastructure and software management processes. This includes migrating Vue.js to the most appropriate version for our needs, which involves selecting a version that offers the best balance of security, support, and new functionalities. Additionally, we will implement a more robust strategy for managing software dependencies and simplify the build processes for our virtual machines. By doing so, we aim to enhance efficiency, reliability, and maintainability of our systems, ensuring that the Glance project continues to support the high-energy physics collaborations effectively.

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.

References

[1] ATLAS Collaboration, JINST **3**, S08003 (2008), also published by CERN Geneva in 2010

[2] ATLAS Collaboration, Physics Letters B **716**, 1–29 (2012)

[3] C. Maidantchik, F. Grael, K. Galvao, K. Pommes, *Glance project: a database retrieval mechanism for the ATLAS detector*, in *Journal of Physics: Conference Series* (IOP Publishing, 2008), Vol. 119, p. 042020

[4] The OpenStack Project, *The OpenStack Project* (2024), `https://www.openstack.org/`

[5] Puppet by Perforce, *Puppet* (2024), `https://www.puppet.com/`

[6] Oracle, *Oracle database* (2024), `https://www.oracle.com/database/`

[7] Docker Inc., *Docker* (2024), `https://www.docker.com/`

[8] Functional Software, Inc., *Sentry* (2025), `https://sentry.io`

[9] B. Lange, C. Maidantchik, K. Pommes, V. Pavani, B. Arosa, I. Abreu, on behalf of the ATLAS Collaboration, Journal of Physics: Conference Series **664**, 062026 (2015)

[10] Red Hat, Inc., *What to know about centos linux eol* (2024), https://www.redhat.com/en/topics/linux/centos-linux-eol

[11] OpenID Foundation, *Openid connect* (2024), https://openid.net/

[12] Shibboleth Project, *Shibboleth* (2024), https://www.shibboleth.net/about-us/the-shibboleth-project/

[13] CERN Authentication and Authorization Services, *Shibboleth integration* (2024), https://auth.docs.cern.ch/user-documentation/saml/shibboleth-integration/

[14] Keycloak Authors, *Keycloak* (2024), https://www.keycloak.org/

[15] OpenJS Foundation, *Node.js* (2025), https://nodejs.org/

[16] Red Hat, Inc., *Node.js on rhel* (2024), https://access.redhat.com/articles/3376841

[17] npm, Inc., *npm* (2025), https://www.npmjs.com/

[18] J. Lockhart, A. Smith, R. Allen, P. Bérubé, the Slim Framework Team, *Slim framework* (2024), https://www.slimframework.com/

[19] Rector Team, *Rector* (2024), https://getrector.com/

[20] Mario Simão, *Support php 8 on slim 2.x* (2021), online forum post, GitHub, https://github.com/slimphp/Slim/issues/3119