PROCEEDINGS OF SCIENCE

Secure Cloud-Native Infrastructure for AI-Based Tumor Tissue Analysis

Francesco Sinisi,[a,*] **Tomáš Brázdil,**[b] **Alessandro Costantini,**[a] **Jacopo Gasparetto,**[a] **Lukáš Hejtmánek,**[c] **Petr Holub,**[d] **Jiří Horák,**[d] **Barbara Martelli,**[a] **Vit Musil,**[b] **Giusy Sergi**[a] **and Andreas Tuerk**[d]

[a] *INFN CNAF, viale Carlo Berti Pichat 6/2, Bologna, Italy*
[b] *Faculty of Informatics, Masaryk University*
[c] *Institute of Computer Science, Masaryk University*
[d] *BBMRI-ERIC & Institute of Computer Science, Masaryk University*
E-mail: francesco.sinisi@cnaf.infn.it, xbrazdil@fi.muni.cz, alessandro.costantini@cnaf.infn.it, jacopo.gasparetto@cnaf.infn.it, xhejtman@ics.muni.cz, petr.holub@bbmri-eric.eu, jiri.horak@bbmri-eric.eu, barbara.martelli@cnaf.infn.it, musil@fi.muni.cz, giusy.sergi@cnaf.infn.it, andreas.tuerk@bbmri-eric.eu

[*] Speaker

© Copyright owned by the author(s) under the terms of the Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License (CC BY-NC-ND 4.0).
All rights for text and data mining, AI training, and similar technologies for commercial purposes, are reserved.
ISSN 1824-8039. Published by SISSA Medialab.

PoS(ISGC2026)014

Artificial intelligence (AI) is rapidly transforming biomedical research, particularly in the analysis of complex datasets such as high-resolution images of tumor tissues. As part of a collaboration between the Italian EOSC and the BBMRI-ERIC Node, INFN and BBMRI-ERIC have launched a joint initiative to define and deploy a secure and scalable infrastructure capable of supporting AI-driven workflows for histopathological image analysis. This effort is supported by the CERIT-SC team at Masaryk University, which successfully hosts a small-scale, secure, AI-ready computing environment known as *SensitiveCloud*.

INFN contributed by designing and deploying a cloud-native platform tailored to the specific requirements of medical data handling within an Information Security Management System (ISMS). The ISMS infrastructure is built on a Kubernetes cluster composed of virtual machines provisioned via OpenStack. Each VM runs a hardened operating system, and the cluster is orchestrated using RKE2 with Center for Internet Security (CIS) benchmarks enforced, ensuring compliance with best practices for secure configuration.

To facilitate controlled access to applications and data, the platform integrates Keycloak as an external identity and access management system. This setup enables federated authentication and fine-grained authorization policies, supporting multi-institutional collaboration while preserving data privacy and integrity. Integration with the LifeScience Authentication and Authorization Infrastructure (AAI) is currently under evaluation to further enhance interoperability and user trust. This system underpins authentication in the BBMRI-ERIC EOSC Node and is fully compatible with the EOSC AAI, including support for government-backed eID identities.

The infrastructure hosts AI pipelines designed to analyze digitized tumor tissue samples, leveraging deep learning models for feature extraction, classification, and pattern recognition. These workflows are containerized and deployed within the RKE2 cluster, benefiting from the elasticity and isolation provided by Kubernetes. The entire infrastructure was developed as part of the BioMedAI project at Masaryk University and transferred to INFN as Docker containers for the relevant components (MLflow, JupyterHub with custom images, xOpat viewer). The platform also supports reproducibility and traceability through integrated logging and monitoring tools.

These activities demonstrate the synergy between infrastructure providers and domain experts, paving the way for scalable, secure, and privacy-preserving AI applications in the health and life sciences.

PoS(ISGC2026)014

1. Introduction

Prostate cancer, one of the most common malignancies affecting men worldwide, is a crucial target for early and precise diagnosis, as patient survival is closely related to prompt detection and treatment. This work aims to establish a secure environment in which Artificial Intelligence (AI) models for prostate cancer screening can be validated using datasets from multiple hospitals. The main contribution of this study lies in the design and deployment of a secure, cloud-native infrastructure tailored to sensitive biomedical AI workloads, addressing key challenges such as regulatory compliance, data privacy, containerized processing pipelines, and collaboration across institutions.

This paper is divided into four parts. First, we introduce the main goal of the project and the organizations involved in Section 2. In Sections 3 and 4, we present the technical tools used to achieve the objectives of the project: the computational resources, the software adopted, and how these components interact with each other. Finally, we conclude with some reflections based on the experience gained throughout the project in Section 5.

2. Context

We begin by presenting the motivation that led several institutions to initiate this research collaboration, together with an overview of the organizations involved.

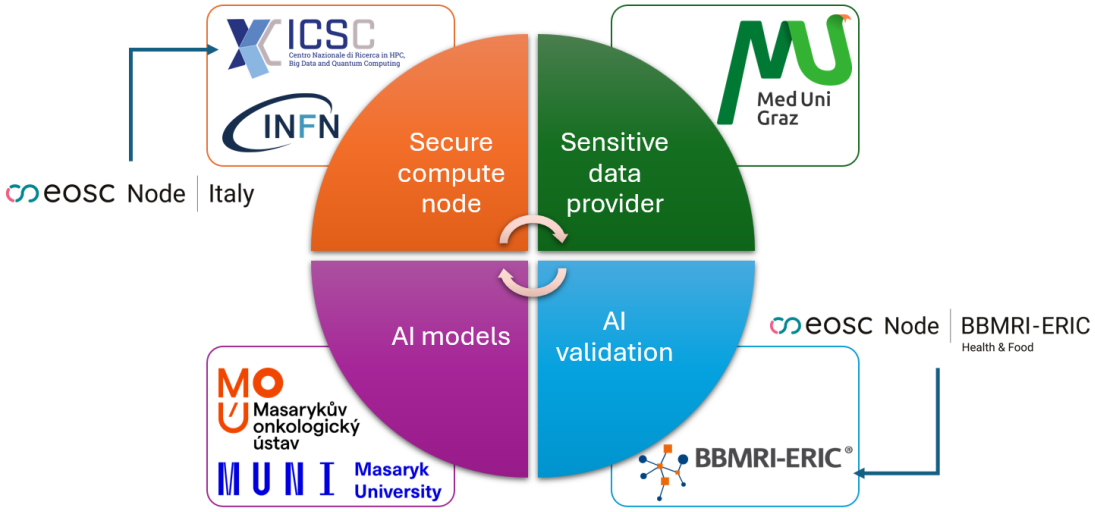

Figure 1: List of entities involved in the scientific collaboration: the Italian EOSC Node (comprising ICSC and INFN), the thematic EOSC Node BBMRI-ERIC, the universities of Graz and Masaryk, and the Masaryk Memorial Cancer Institute.

2.1 Scientific purpose: MCVAL use case

The need for this project arises from the fact that AI models are often trained using data from a single healthcare institution. As a result, these models cannot be easily transferred when applied to data from other hospitals or clinical environments. For this reason, it is essential to validate their performance using new datasets originating from different medical centers.

PoS(ISGC2026)014

The goal of the project is to develop a tool, called MCVAL (Multi-Centric VALidation), to validate AI models used in prostate cancer screening. Since this work involves sensitive healthcare data provided by multiple institutions, all activities must comply fully with the regulations governing the handling of such data. At the same time, sharing sensitive health information between hospitals for validation purposes is challenging due to regulatory, ethical, and legal constraints.

2.2 Scientific collaboration

The MCVAL use case is based on an AI model trained on data from Masaryk University and the Masaryk Memorial Cancer Institute, and validated on data from the Medical University of Graz. High-performance computing resources are provided by the Italian EOSC Node [1] and, in the future, by additional EOSC Nodes. The role of the BBMRI-ERIC EOSC Node [2] is to create a secure environment in which AI models for prostate cancer screening can be validated using data from different hospitals. Figure 1 provides an overview of the actors involved in the collaboration.

The long-term vision is to establish federated catalogs of sensitive health data and computing infrastructures accessible to AI researchers throughout Europe, allowing them to validate and train models on diverse datasets and technologies, rather than repeating the validation separately for each hospital. To address these challenges, we designed the infrastructure described in the following section.

3. Infrastructure and architecture

The infrastructure was designed according to three guiding principles: security by design, scalability, and interoperability. To meet these requirements, all services were deployed within a dedicated OpenStack [3] tenant[1] hosted on the EPIC secure cloud environment. The platform consists of a set of virtual machines providing compute, storage, authentication, logging, and management services, as shown in Figure 2. The core computational resources are delivered through an RKE2-based Kubernetes cluster, while complementary services such as Keycloak, network storage, centralized logging, and secure administrative access are provided by dedicated virtual machines.

Table 1 summarizes the main software components adopted in the infrastructure together with their functional role. The combination of these technologies provides a secure and reproducible environment for the deployment of AI workflows targeting sensitive biomedical datasets, enabling controlled access, auditability, and efficient processing of large-scale Whole-Slide Images (WSIs).

The following sections describe the individual components of the infrastructure and their interactions in more detail.

3.1 Secure cloud

We chose to use a cloud infrastructure in order to benefit from several advantages of this paradigm: it allows us to scale computing resources quickly, both horizontally and vertically; it simplifies the management of volumes and backups; it enables network segregation; and it allows VMs to be migrated to other hypervisors in case of failures.

[1] OpenStack tenants can be thought of as namespaces: they contain virtual machines that typically belong to the same subnet and can therefore communicate easily with one another.

PoS(ISGC2026)014

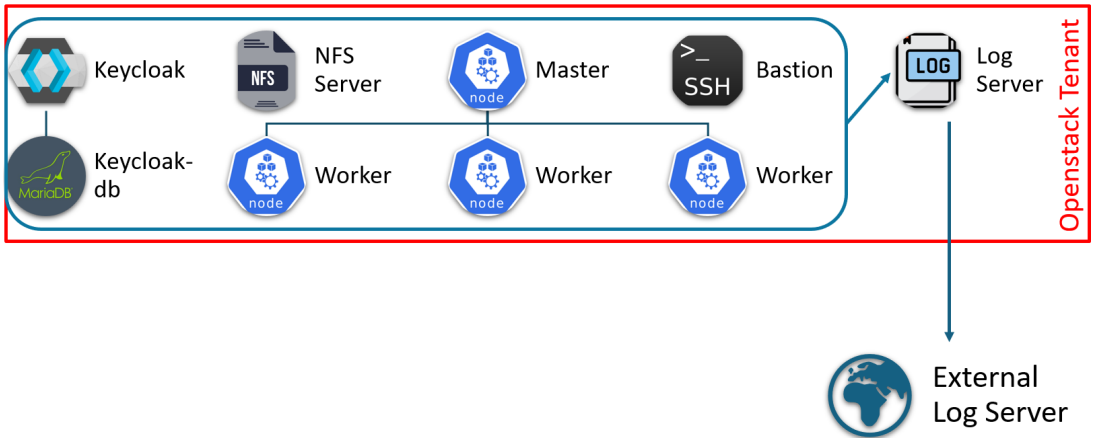

Figure 2: The image shows all the VMs present in the OpenStack tenant that are useful for the project.

Table 1: Overview of the infrastructure components, technologies, and their role within the proposed cloud-native platform.

Component	**Technology**	**Purpose**
Cloud platform	OpenStack EPIC	VM provisioning
Kubernetes	RKE2	Container orchestration
Storage	NFS server on dedicated VM	Shared data access
IAM	Keycloak + OAuth2 Proxy	Authentication
Logging	Rsyslog	Audit trail
AI environment	JupyterHub	User workspaces
MLOps	MLflow	Experiment tracking
Pathology viewer	xOpat	WSI visualization

Since data are highly sensitive, they are hosted in a secure cloud environment called EPIC (Enhanced Security and Privacy Cloud) [4], which is also based on OpenStack. This platform has obtained ISO certifications, which are reviewed annually and confirm that it meets the required security standards for handling sensitive biomedical data in a regulated environment.

The cloud environment includes several security features. Access is granted only to authorized personnel through collaboration agreements, identity verification, and proof of adequate technical skills. Before accessing EPIC resources, users are required to undergo a face-to-face identity verification process (*de visu* recognition), which provides an additional layer of assurance regarding the identity of authorized users. Furthermore, OpenStack tenant networks are logically segregated from each other, ensuring stronger isolation between projects and reducing the risk of unintended access to sensitive biomedical data. Access to EPIC resources is possible only through a VPN and credentials protected by two-factor authentication. Software running in this environment must also meet specific requirements: it must be stable, properly licensed, and maintained by an active community capable of releasing security updates promptly (especially in the case of open-source software). This approach ensures that sensitive biomedical data remain confined within a controlled environment, reducing the risk of unauthorized access or data leakage.

PoS(ISGC2026)014

3.2 Kubernetes cluster

The core of the infrastructure is a Kubernetes cluster that hosts most of the software used in the project. To speed up the creation of the cluster, we used RKE2 [5], an enterprise-ready Kubernetes distribution. Although it is open-source, it is supported by a large and active community and is sponsored by SUSE. RKE2 allows us to configure a cluster with high security standards by applying the benchmarks published by the Center for Internet Security (CIS) [6], a non-profit organization that develops internationally recognized cybersecurity guidelines.

These benchmarks include recommended secure configurations, the disabling of risky features, and several best practices. In a Kubernetes environment, this means using RBAC, limiting container privileges, protecting the API server, enabling audit logging, etc.

Another important aspect of the infrastructure is the availability of two dedicated GPU worker nodes designed for AI workloads. Each worker node is equipped with an NVIDIA H100 GPU, 24 vCPUs, 128 GB of RAM, and 80 GB of local storage. The GPUs are installed on separate nodes in order to improve workload isolation and provide suitable computational resources for training, validation, and inference tasks associated with digital pathology and biomedical AI models. Some preliminary configurations were required, both at the operating system level and at the Kubernetes level, to ensure that the cluster could detect and use the GPUs.

3.3 NFS Storage

Regarding storage, we chose to adopt an NFS-based solution. A VM attached to an OpenStack volume was created within the same private network as the Kubernetes cluster and acts as an NFS server. This machine stores the research data and exports it to the clients, which are the Kubernetes worker nodes. The pods running on these nodes can read and write data in the shared folder.

A total storage capacity of 2 TB was allocated to the project for the storage and processing of biomedical datasets. In addition, a dedicated access control rule (ACL) was configured to permit incoming connections exclusively from the Medical University of Graz network, allowing project partners to securely upload datasets to the NFS server for subsequent processing within the EPIC environment.

Because we needed to process a large amount of data, we opted for an internal storage solution to minimize latency and maximize data exchange between VMs. NFS is not the most robust solution compared, for example, to an object storage cluster. However, it is relatively easy to implement within a single OpenStack tenant and provides good performance and data isolation.

Deploying complex storage systems for every collaborative project would not be sustainable. Another option would be to use a single centralized storage infrastructure, but this could introduce network latency and, more importantly, expose sensitive data to storage administrators who should not have access to it.

3.4 Authentication and logging

To complete the infrastructure, the tenant also includes two VMs providing the authentication service based on Keycloak [7], a bastion exposed externally and serving only as a jump host, and a log server.

PoS(ISGC2026)014

Keycloak was chosen as the identity and access management system for the services running on Kubernetes. To increase service resilience, it was deployed on two VMs: one hosting the stateless application and the other hosting the database.

Log collection is a mandatory requirement in this environment. Every VM in the tenant runs an agent that records user actions together with the username and timestamp. These logs are immediately sent to a central collector within the tenant, which acts as the log server. The log server then forwards the logs to an external machine to which the tenant administrators do not have access, in order to prevent any possible log tampering. The software used to enable this mechanism is Rsyslog [8].

3.5 Automation

Manually installing and configuring all the software components described so far would require a significant amount of time and in-depth knowledge of each technology involved. Automation tools help reduce deployment time, minimize human error, improve reproducibility, and simplify the installation and maintenance processes. For this reason, infrastructure provisioning and configuration management were automated through a collection of Ansible roles developed within INFN.

These roles were designed to streamline the initial deployment phase and ensure consistent configurations across the environment. In particular, the automation framework implements the deployment of the RKE2 Kubernetes cluster, including GPU support and NFS-based shared storage integration; the installation and configuration of the Keycloak-based authentication system; the setup of the centralized logging infrastructure based on Rsyslog; Linux user account creation and management; and the generation and distribution of TLS certificates used to secure inter-node communications.

The adoption of infrastructure automation ensured consistent deployments across environments, reduced manual configuration effort, and simplified the replication of the platform for future collaborations and research projects.

4. Software involved

Once the infrastructure is in place, we can move on to the software used in the research project. In this work, the Software as a Service (SaaS) paradigm has been adopted, which consists in delivering software applications over the network through web-based interfaces without requiring local installation or maintenance. In addition, it provides numerous advantages, such as simplified deployment, centralized management, scalability, and improved accessibility for users.

4.1 Data processing and visualization

The software used for data processing and visualization runs on a Kubernetes cluster. The core component for data processing is JupyterHub [9], a well-known open-source platform, that was selected due to its ability to provide isolated, reproducible environments for multiple users in a web-based interface. One of the project requirements was to upload custom images to JupyterHub, allowing users to launch notebooks that they could use as the basis for their work environment.

JupyterHub was used together with MLflow [10], an open-source platform designed to manage and organize the lifecycle of machine learning models. In practice, within a Jupyter notebook

PoS(ISGC2026)014

used to develop and execute code, the MLflow Python libraries were imported in order to: track experiments by saving input parameters and results; store and version trained models; reproduce experiments with the same configurations; and easily compare different machine learning outcomes. MLflow stores this information in a tracking server, where it can be viewed through a web interface.

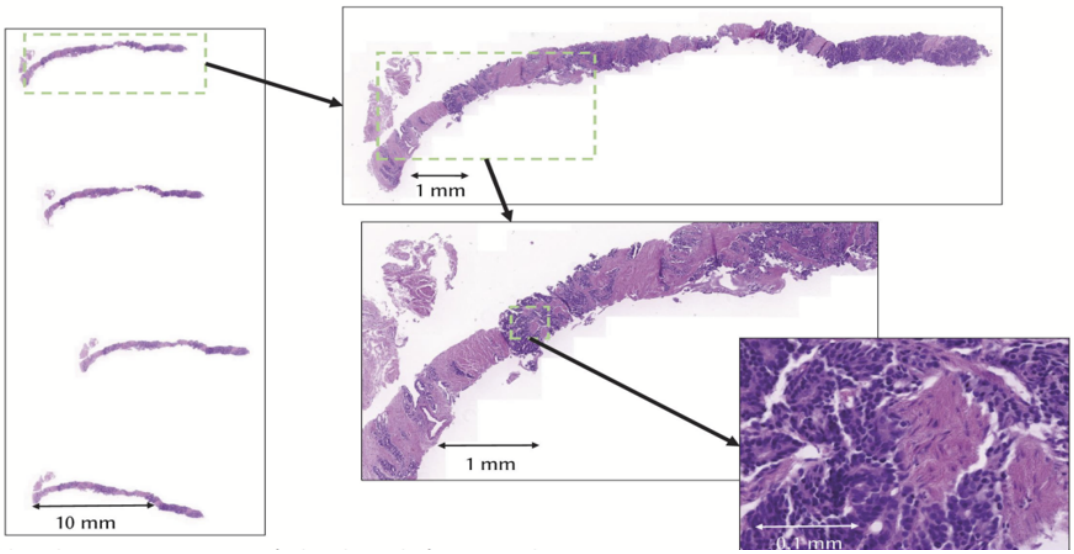

Figure 3: Example of a WSI used in the MCVAL workflow. The image illustrates the multi-resolution nature of digital pathology datasets. WSIs can contain several gigapixels and typically require storage capacities ranging from hundreds of megabytes to several gigabytes, depending on the acquisition resolution, image size, and compression method used.

For image visualization, we used two software components that together can load, manage, and display very large medical images called Whole-Slide Images (WSIs), commonly used in digital pathology. These are extremely high-resolution scans of biological tissue slides (Figure 3). Specifically, we used a backend called WSI Service, which includes a server designed to handle medical images that can reach several gigabytes in size, and a frontend called xOpat, a web viewer for digital pathology images.

4.2 Authentication

As mentioned earlier, Keycloak was chosen as the identity and access management system. It is open-source, supported by a large community, and provides two-factor authentication. For these reasons, it meets the minimum requirements for use in a secure environment. Kubernetes and Jupyter provide native integration with OIDC and OAuth2. For these tools, we were therefore able to manage both authentication and authorization based on user groups defined in Keycloak.

MLflow and the image viewer, however, do not natively support these protocols. Since it is not acceptable to use software without authentication in secure environments, we partially addressed this issue using OAuth2 Proxy [11]. This open-source software, sponsored by SAP and Microsoft, protects web applications by adding authentication without requiring modifications to the application itself. It can act as either a standalone reverse proxy or a middleware component, providing a simple

PoS(ISGC2026)014

and secure way to protect your web applications with OAuth2/OIDC authentication. As a reverse proxy, it intercepts requests to your application and redirects users to an OAuth2 provider for authentication. As a middleware, such as in our case (Figure 4), it can be seamlessly integrated into your existing infrastructure to handle authentication for multiple applications.

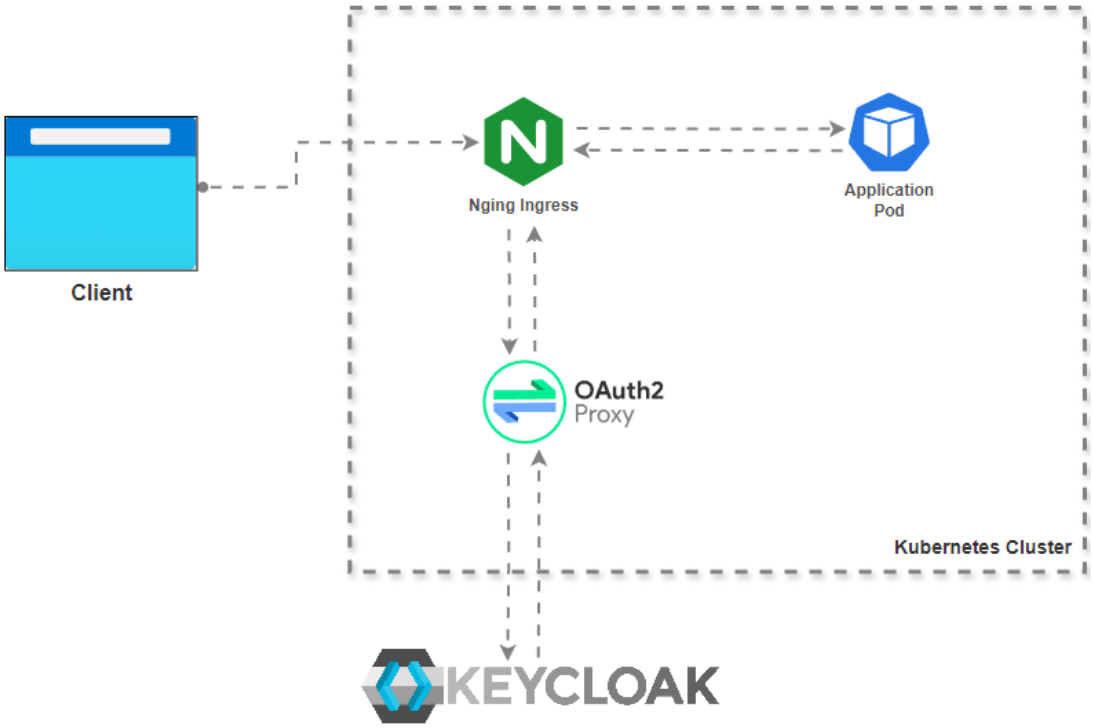

Figure 4: Authentication flow implemented in the Kubernetes environment. User requests are intercepted by OAuth2 Proxy through the NGINX Ingress Controller and authenticated against Keycloak via OIDC before access is granted to protected applications.

This solution is only a partial one, as it addresses authentication but not authorization. OAuth2 Proxy can add user information, such as email or group membership, into HTTP headers, which applications could use to manage permissions. However, MLflow and the viewer are not able to interpret these headers. To mitigate this limitation, we used some filtering features provided by OAuth2 Proxy, which can restrict access before the request reaches the application (e.g., by allowing only users with specific email addresses or domains).

5. Conclusions

Within the context of the Italian EOSC Node and the BBMRI-ERIC EOSC Node collaboration, this work has demonstrated the design and deployment of a secure cloud-native infrastructure for AI-based tumor tissue analysis. The platform combines OpenStack-based secure computing resources, Kubernetes orchestration, centralized authentication, shared storage, logging services, and application-level tools for biomedical AI workflows, providing an environment specifically designed for the management and processing of sensitive health data.

PoS(ISGC2026)014

Beyond the infrastructure deployment itself, the platform has been adopted within the MCVAL (Multi-Centric VALidation) use case, whose objective is the validation of AI models for prostate cancer screening across multiple institutions. The deployment supports the collaboration between the Italian EOSC Node, the BBMRI-ERIC EOSC Node, Masaryk University, the Masaryk Memorial Cancer Institute, and the Medical University of Graz. Public presentations of the initiative have described the use of the platform as part of the EOSC Federation scientific use cases, supporting collaborative validation workflows on digital pathology datasets and AI models for prostate cancer screening [12, 13].

The focus of this contribution is the infrastructure and security architecture enabling the MCVAL use case rather than the biomedical outcomes themselves. Scientific results derived from the analysis workflows are therefore outside the scope of the present work and are reported separately by the biomedical partners involved in the project.

This experience highlights how the integration of EOSC-compliant infrastructure with application level services can support secure and interoperable collaboration across institutions working with sensitive biomedical data. To continuously expand the boundaries of human knowledge, research increasingly depends on collaboration among professionals with diverse expertise. Facilitating such collaboration requires not only suitable computing resources, but also trusted authentication mechanisms and interoperable services that can be adopted across organizational boundaries.

To avoid some of the limitations described in this work, it is advisable to use software that natively integrates with authentication systems and supports authorization based on user attributes. Other possible improvements concern storage resilience and extending infrastructure automation. As mentioned earlier, we used NFS storage. Although it was sufficient for our use case, it is not the most resilient solution available, which is an important factor especially for large and long-term research projects. An excellent alternative solution could be to integrate Kubernetes with CephFS [14], which would increase the reliability of the storage component. Finally, while software deployment was automated using Ansible, virtual machine provisioning could be further improved through infrastructure-as-code tools such as Terraform [15] or OpenTofu [16].

6. Acknowledgements

The work presented in this paper has been supported co-funded by the Italian Complementary National Plan PNC-I.1 "Research initiatives for innovative technologies and pathways in the health and welfare sector" D.D. 931 of 06/06/2022, "DARE - DigitAl lifelong pRevEntion" initiative, code PNC0000002, CUP: B53C22006480001.

References

[1] *EOSC Node | Italy*
https://eosc.eu/building-the-eosc-federation/eosc-node-italy

[2] *EOSC Node | BBMRI-ERIC*
https://eosc.eu/building-the-eosc-federation/eosc-node-bbmri-eric

PoS(ISGC2026)014

[3] *OpenStack*
https://www.openstack.org

[4] *EPIC*
https://www.cnaf.infn.it/en/epic-cloud-en

[5] *RKE2*
https://docs.rke2.io

[6] *CIS*
https://www.cisecurity.org

[7] *Keycloak*
https://www.keycloak.org

[8] *Rsyslog*
https://www.rsyslog.com

[9] *Jupyter*
https://jupyter.org

[10] *MLflow*
https://mlflow.org

[11] *OAuth2 Proxy*
https://oauth2-proxy.github.io/oauth2-proxy

[12] *EOSC Federation use case: Multi-centric validation of AI Models for prostate cancer screening. YouTube presentation.*
https://www.youtube.com/watch?v=UPtEFGLruh4

[13] *Welcoming the EOSC Federation: Scientific Use Cases from the First Wave of EOSC Candidate Nodes.*
https://indico.cern.ch/event/1543880/contributions/6744855/attachments/3167355/5628512/20251103_BBMRI_MCVAL_HOLUB_TUERK_Welcoming_the_EOSC_Federation_Scientific_Use_cases_from_the_first_wave_of_EOSC_candidate_nodes_EOSC%20Symposium2025.pdf

[14] *Ceph*
https://ceph.io

[15] *Terraform*
https://developer.hashicorp.com/terraform

[16] *OpenTofu*
https://opentofu.org

PoS(ISGC2026)014