

A Scalable and Secured HL7 FHIR Healthcare Platform: Architecture, Load Testing, and Preliminary Findings

Valentino Šafran, Umut Arioz, Rigon Sallauka, Izidor Mlakar
Human-Centric Explorations and Research in AI, Technology, Medicine and Enhanced Data
(HUMADEx) Group, Faculty of Electrical Engineering and Computer Science,
Maribor,
Slovenia

Received: June 12, 2024. Revised: March 19, 2025. Accepted: May 23, 2025. Published: August 7, 2025.

Abstract— This paper presents a healthcare platform based on the HL7 FHIR standard that addresses two critical needs in modern healthcare information systems: secure interoperability and scalability. The platform uses a robust FHIR data model to support standardized integration with existing PROs and analytics systems. By deploying microservices as containerized services on virtual infrastructure and managing identities with Keycloak, the platform enables real-time, secure data exchange between clinicians and patients in compliance with data protection regulations. Load testing with up to 5,000 concurrent users demonstrates efficient resource utilization and highlights areas for potential optimization. These findings contribute a reference architecture and initial performance results that can guide implementation of large-scale, FHIR-compliant solutions in clinical or research settings. This approach supports secure, real-time data exchange between patients and clinicians. As a result, this study offers valuable insights into interoperability, security considerations, and system scalability in digital health environments.

Keywords— Clinical Decision Support System, Containerization, Healthcare Interoperability, HL7 FHIR, Keycloak, Load Testing, Microservices.

I. INTRODUCTION

HEALTHCARE organizations increasingly depend on digital solutions to streamline patient management and clinical workflows, [1], [2], [3]. In large hospital networks or multi-center trials, concurrency surges during telehealth sessions, shift changes, or major screening events often exceed thousands of simultaneous requests. Traditional Electronic Health Record (EHR) solutions often struggle to remain responsive in these conditions. Modern hospitals and

research institutions require robust interoperability frameworks to share patient data across disparate systems, which is why HL7 Fast Healthcare Interoperability Resources (FHIR) has emerged as a vital standard. It not only structures clinical information but also simplifies data exchange via RESTful APIs.

Equally important is data security, [4]. Unauthorized access to sensitive clinical records can lead to severe legal and ethical repercussions. Traditional username–password authentication is inadequate for large-scale deployments involving thousands of users (including patients, clinicians, and administrative staff) simultaneously access electronic health records (EHRs). To address this, the platform integrates containerized deployments with Keycloak to enable advanced identity and access management, offering token-based authentication, role-based access control, and single sign-on capabilities.

In many existing healthcare systems, data remains fragmented across incompatible platforms or locked in formats that lack a universal structure. This fragmentation not only hinders clinicians' ability to make informed decisions but also complicates data exchange for research. HL7 FHIR emerges as a powerful solution: it standardizes how medical data is represented and communicated, using well-defined resources such as Patient, Practitioner, Observation, QuestionnaireResponse, and more. However, implementing FHIR-based solutions in clinical environments continues to present challenges. Chief among them is to ensure robust security and data protection while maintaining the performance required by large-scale scenarios. Additionally, many large hospitals and clinical research institutions require flexible, fault-tolerant architectures that can scale thousands of users. Motivated by these needs, this study aims to:

- Present an HL7 FHIR-based platform designed to scale up to thousands of users, leveraging microservices and containerization.

- Ensure strong security via role-based permissions, encrypted communications, API keys, and Keycloak-based identity and access management.
- Assess load-testing results up to 5,000 concurrent users to quantify performance and guide further optimization in real-world deployments.

Ultimately, the platform demonstrates how modern containerization tools can be combined with standardized healthcare data models to meet the dual challenges of security and interoperability.

The remainder of this paper is organized as follows. Section II reviews related work in interoperability, microservice-based healthcare solutions, and advanced security mechanisms. Section III describes our methodology, detailing the platform architecture and security measures. Section IV presents load-testing methods and results. Section V discusses performance and security trade-offs, and Section VI concludes the paper by suggesting avenues for future research.

II. RELATED WORK

Existing literature underscores the significance of interoperable and secure healthcare systems for FHIR adoption and advantages. HL7 FHIR fosters data exchange among EHRs, mobile apps, and wearable sensors by relying on resource-based models, [5], [6]. Its popularity is driven by open-source tooling, extensive community support, and alignment with web technologies like REST, JSON, and XML. Due to regulations such as HIPAA and GDPR, healthcare platforms must safeguard protected health information (PHI). Solutions commonly incorporate encryption, token-based authentication, and detailed audit logs, [7], [8]. The adoption of FHIR-native microservices architecture enables healthcare organizations to create modular, independently deployable services that align with standard healthcare data exchange protocols, [9]. This approach facilitates better scalability and maintainability compared to monolithic healthcare systems, [10]. The implementation of FHIR-native microservices architecture enables healthcare organizations to create modular, independently deployable services that align with standard healthcare data exchange protocols, [11]. This approach has been successfully demonstrated in patient navigation systems, where synchronous communication between Patient and Appointment microservices maintains essential functionality while enabling data reception from multiple sources, [12]. Recent scoping reviews have identified 203 different FHIR projects applicable to clinical research, with most focused on establishing data pipelines and linking clinical systems, [11]. The current landscape of FHIR-based data models encompasses both dynamic pipeline-based and static data models, with applications spanning chronic diseases, infectious disease management, cancer research, and intensive care scenarios. The most commonly utilized FHIR resources include Observation, Condition, and Patient resources, reflecting the core data elements necessary for comprehensive healthcare informatics systems, [6].

While prior studies have focused on HL7 FHIR implementations for secure data exchange or chronic disease management, this work distinguishes itself by addressing the combined challenge of scalability, security, and real-time bidirectional interaction in a unified platform. Unlike prior work by [13], which relies on Google Cloud Platform's built-in IAM and emphasizes blockchain-based storage for secure data management, our architecture integrates a specialized Keycloak-based token management strategy and container-level firewalls to address concurrency surges. Whereas [13] focus on throughput and file-transfer efficiency within GCP, we conduct load testing up to 5,000 simultaneous users and employ a microservices-based design that scales horizontally, ensuring robust performance under real-world, high-concurrency conditions. Moreover, this platform is among the few to integrate a multilingual Rasa chatbot and a FHIR-compliant dashboard, offering a complete user-facing ecosystem for clinical or research deployments. Recent scholarship has begun to connect clinical-simulation pedagogy with conversational AI. The study [14] propose a four-phase life-cycle which includes Conceptualization, Protocol Design, Technical Design, and Trials & Revision for building educational chatbots that faithfully emulate standardized-patient encounters, [14]. Their framework emphasizes "walled-garden" knowledge bases, explicit role definition, and iterative teacher-in-the-loop testing to minimize bias and protect privacy. Our Rasa-FHIR chatbot follows a similar human-centered philosophy: it is trained on synthetic patient bundles that mirror real FHIR resources, and its interaction scripts are version-controlled so that clinical instructors can refine prompts after each cohort exercise. By embedding [14] design guidelines inside a secured, role-based FHIR ecosystem, we move beyond standalone simulations and demonstrate how the same chatbot engine can serve both direct patient intake and structured educational scenarios within a single, scalable platform.

The inclusion of performance benchmarks, authentication overhead profiling, and detailed infrastructure metrics under peak load adds empirical rigor and practical value. As such, our work contributes a scalable, standards-compliant reference architecture, along with reproducible load-testing insights that directly address unmet needs in both clinical operations and research informatics.

Digital healthcare ecosystems increasingly leverage FHIR standards for chronic disease management, with implementations showing particular strength in cancer care (45%), cardiovascular disease (15%), and diabetes management (15%). The maturation of FHIR with each standard revision, combined with validated Implementation Guides, provides validated data sets and common shared resources that support interoperability in digital health innovation, [5].

The study [15] centers on linking fragmented chronic disease datasets via a federated HL7-FHIR architecture, while our platform emphasizes large-scale concurrency and real-time

operations. While [15] focus on record linkage and privacy in a static environment, we deploy containerized microservices and advanced token management to accommodate thousands of concurrent users and ensure robust, on-demand interoperability.

Researchers have demonstrated that adopting HL7 FHIR significantly eases the integration of disparate systems, including third-party applications such as telehealth platforms, medical device gateways, and external research registries. Its resource-based design, which encapsulates distinct healthcare domains (Patients, Observations, Conditions, etc.), supports modular and incremental growth of clinical systems. Moreover, broad community engagement has led to a rich set of implementation guides, tooling, and collaboration spaces that help developers adhere to best practices.

As recognized by multiple studies, Docker and Kubernetes are frequently used to deploy microservices-based architectures in healthcare [16], offering modularity, resource isolation, and easier maintenance. The combination of containerization and microservices ensures that updates or modifications to a single service (e.g., the authentication component) can be deployed without causing system-wide downtime. Nevertheless, security remains a top-level challenge; each microservice must guard Protected Health Information (PHI) against breaches. In that context, Keycloak and other identity management solutions add layers of protection by issuing JWT-based tokens, supporting role-based access, and allowing for custom access policies.

Another critical aspect highlighted in recent work is the impact of high concurrency on healthcare APIs, [17]. Real-world use cases, such as large research studies or telehealth sessions, can push thousands of concurrent requests. Efficient load balancing, caching, and careful resource utilization strategies are therefore essential. In many deployments, token validation and repeated lookups against a central authorization server can generate overhead, demonstrating a need for well-planned orchestration and potential vertical or horizontal scaling.

Despite progress, gaps remain in achieving seamless interoperability while ensuring high security for large, concurrent user populations. Additionally, the overhead introduced by advanced identity management systems like Keycloak requires careful design of load-balancing and caching layers. This paper contributes real-world performance data and a reference model that merges HL7 FHIR's interoperability strengths with layered security practices. By providing insights from load testing up to 5,000 users, we address a crucial need for empirical data that can guide future, large-scale clinical deployments.

III. METHODS

In this section, we describe the methodology employed in developing and deploying the FHIR-based healthcare platform. We combine microservices deployed as Docker containers or run on VMs, a robust HL7 FHIR server for data storage, and

Keycloak to handle user management. By layering security measures, such as role-based access control, TLS encryption, firewall rules, and VPN access, our approach aims to strike a balance between usability, performance, and compliance with regulatory mandates.

A. Architecture

Figure 1 illustrates the microservices-based healthcare platform, which includes a REST API (Spring Boot) to implement core business logic, orchestrating data flows between the FHIR server and other components. The FHIR server acts as a standards-compliant repository for clinical data, exposing resources such as Patient, Observation, QuestionnaireResponse, and ResearchStudy. Keycloak handles identity management and enforces secure, token-based access. Keycloak supports multi-level role definitions, allowing fine-grained access such that trainees can perform simulated updates on non-production datasets. Rasa Chatbots provide patient-facing questionnaires in multiple languages, storing responses via the FHIR server. FHIR Dashboard is a Flask-based web interface enabling clinicians and admins to view or update FHIR resources in real-time.

All major services run in Docker containers or dedicated VMs, communicating within a virtual Docker network. This design isolates failures and allows each component to scale independently, which is essential in high concurrency scenarios. The approach is particularly relevant for large healthcare systems or multicenter clinical studies where thousands of concurrent sessions are common.

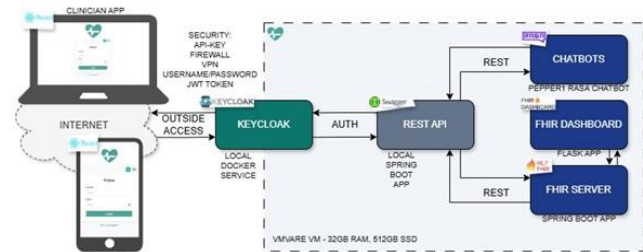


Fig. 1. The architecture of the HL7 FHIR based healthcare platform.

B. Data Flow

Figure 2 illustrates the bidirectional data flow architecture between the Patient App and Clinician App within the HL7 FHIR-based healthcare platform. At the top of the diagram, patient-generated data, such as Patient-Reported Outcomes (PROs)/Ecological Momentary Assessments (EMAs), video diaries, and other health metrics, are first processed. These inputs are then transformed into structured JSON formats and stored as FHIR resources (e.g., QuestionnaireResponse, Observation, Composition, DocumentReference) on a central FHIR server. These structured datasets are subsequently accessible to clinicians through the Clinician App, enabling real-time monitoring, clinical assessment, and evidence-based decision-making. This upper half of the diagram reflects the modular microservice architecture presented in this paper, where patient interactions, often driven by chatbot are

efficiently captured, transformed, and persisted using standardized HL7 FHIR format.

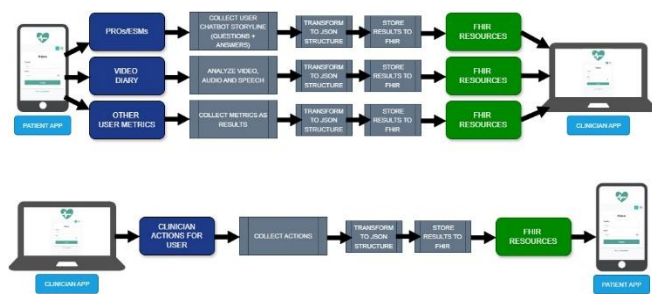


Fig. 2. FHIR-driven Ecosystem for Patient-Clinician Interaction.

The lower half of the diagram illustrates the clinician-to-patient communication loop. Here, clinicians define personalized care actions (e.g., follow-up tasks, medication changes, or activity prompts) via the Clinician App. These actions are similarly collected, structured in JSON, stored as FHIR resources (such as Task, CarePlan, or PlanDefinition), and transmitted back to the Patient App. This supports dynamic care planning and ensures patients receive timely interventions. This workflow closely aligns with the care planning and task orchestration, where resources like PlanDefinition, ActivityDefinition, and Task allow the platform to implement and track personalized, protocol-driven clinical pathways. Overall, Figure 2 visualizes the paper’s core contribution, a secure, scalable, and interoperable FHIR-driven ecosystem enabling seamless clinical collaboration through structured data flows and role-based access.

C. HL7 FHIR Data Model

Although our focus in this paper is on the platform’s architecture and security, we also leverage a robust HL7 FHIR data model that includes Patient, Practitioner, Observation, QuestionnaireResponse as Core resources as represented in Figure 3. Then ResearchStudy, Evidence, ResearchSubject as Research Resources and CarePlan, PlanDefinition, Task, ActivityDefinition representing Care planning.

These resources facilitate not just direct patient care, but also higher-level tasks such as clinical research, dynamic care pathways, and integration with external devices or hospital systems. By storing data in a standardized format, the system can scale more readily and maintain interoperability with existing EHR systems or analytics platforms.

D. Security Layers

Keycloak serves as the centralized authentication provider, generating JSON Web Tokens (JWTs). Each request to the REST API or the FHIR server is validated against the user’s token and assigned role (e.g., patient, clinician, administrator). Table I summarizes the platform’s multi-layered security framework.

For example, only a clinician with the correct role can modify patient records. Administrators maintain key rotation policies, monitor session durations, and manage forced logouts.

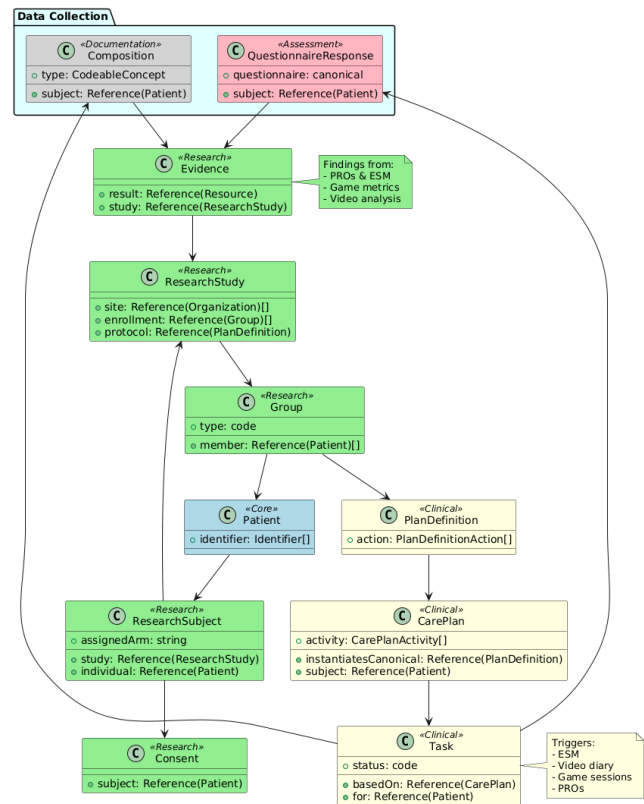


Fig. 3. UML of FHIR Data Model for Research Studies on Digital Interventions.

To further protect clinical data, we implemented additional security measures. Use of API keys which introduce certain microservice-to-microservice communications require custom tokens, ensuring no single compromised credential can access all endpoints. Container-level firewall ensure that inbound and outbound traffic is filtered based on known IP ranges and protocols. Then VPN Access for administrative tasks (server updates, database maintenance) are restricted to users on a secure VPN tunnel.

TABLE I. SECURITY LAYERS OVERVIEW.

Security Layer	Description	Technology Used	Scope
Authentication	User identity verification	Keycloak + JWT	All services
Authorization	Role-based access control	Keycloak roles	REST API, FHIR Server
Transport Encryption	Secure data transfer	HTTPS / TLS	API and FHIR endpoints
Network Isolation	Limit attack surface	Docker internal net.	All containers
Firewall Rules	Reject unauthorized traffic	VM + Firewall	Host and containers
API Keys	Service-to-service verification	Custom token headers	Optional services
VPN Tunnel	Secure admin access	IPsec/VPN	Admin/DB access
Audit Logging	Trace user actions	Prometheus + Keycloak logs	Admin & compliance

Standard level of TLS encryption is added to all communications among containers and virtualized microservices and external clients use HTTPS/TLS. Audit logging of activities, including authentication attempts and data reads, are centrally logged and monitored for anomalies.

E. Load Testing

To assess performance and scalability, we conducted load testing using a Python-based script that simulated up to 5,000 concurrent users. We tested up to 5,000 users to exceed the real-world requirement of supporting 2,000 users, which our paper aims to address. We attempted 10,000 concurrent users, but the host server, already running various other microservices, encountered CPU overload due to increased thermal and resource constraints. Note that the server hosts components of this platform as well as other microservices not described in this study. Our environment was configured in a secure on-premise location. Server hosting the platform has is running Ubuntu 20.04.4 LTS with AMD EPYC 7443 24-Core CPU, 256 GB of RAM, 4x NVIDIA RTX A6000 GPUs running with CUDA 11.4. Each simulated user followed realistic workflows, including:

- Submitting Questionnaires: POST requests for Chatbot conversation, reflecting patient self-reporting or chatbot-collected answers.
- Updating Clinical Data: PUT requests to update QuestionnaireResponse or Composition resources, mimicking typical questionnaire tasks in a clinical or research setting.

We used Prometheus/Grafana for runtime monitoring of CPU usage, memory consumption, and request throughput. Keycloak logs were also analyzed to check authentication overhead during high concurrency. This testing environment provided an end-to-end evaluation of how effectively the architecture manages intensive workloads without sacrificing securityh

IV. RESULTS

Here we present the results obtained from the load testing of the FHIR-based healthcare platform. The measured performance metrics include response time, CPU usage, memory usage, chatbot stability, and throughput. A Python-based load-testing script simulated up to 5,000 concurrent users. Each user performed realistic tasks: retrieving patient records, submitting questionnaires and storing data on the FHIR dashboard. The system was monitored with Prometheus/Grafana for CPU usage, memory consumption, and request throughput, as well as Keycloak’s authentication performance.

For up to ~1,000 concurrent users, both the REST API and FHIR server maintained average latencies under 120 ms, indicating that microservices can handle moderate concurrency without impacting user experience. Beyond 2,000 users, sporadic spikes up to 900 ms occurred on the FHIR server, often correlated with heavy database writes or garbage collection events. This can be seen in Figure 4 and Figure 5.

Although these spikes were temporary, they suggest areas for optimization, such as query tuning or asynchronous request processing.

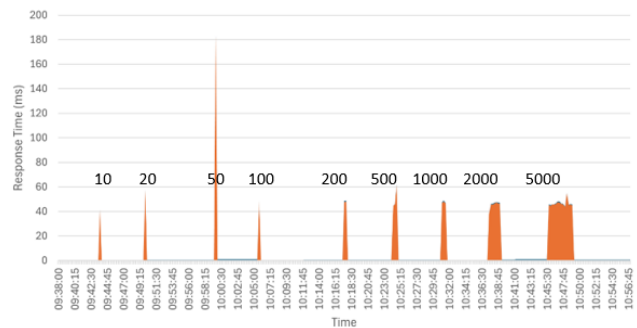


Fig. 4. Response time of the REST API during the load test, measured under increasing user loads (10-5,000).

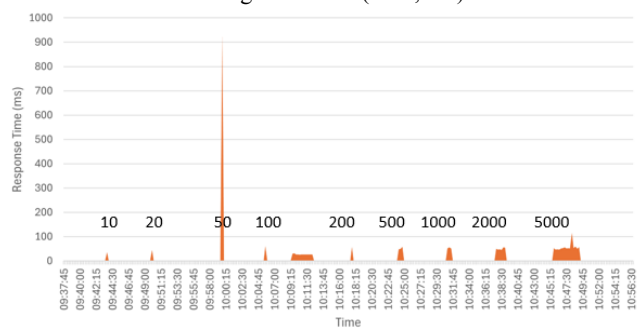


Fig. 5. Response time of the FHIR server during the load test, measured under increasing user loads (10-5,000).

Represented in Figure 6 and Figure 7, overall CPU usage remained manageable throughout. For concurrency above 2,000, CPU utilization stabilized in the 15–20% range for both the REST API and FHIR server. Occasional peaks were observed when Keycloak handled large batches of simultaneous token verifications or renewals. This suggests that, in production, horizontally scaling the authentication service or offloading certain computations (e.g., caching tokens) could maintain efficiency under high loads.

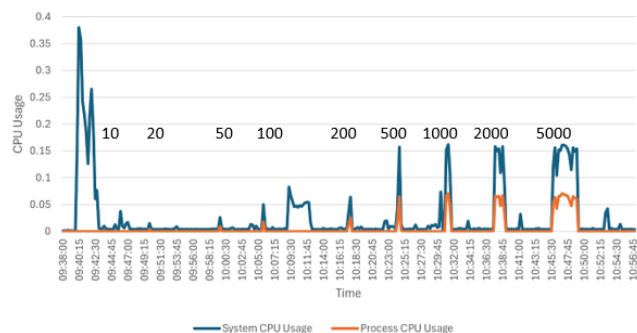


Fig. 6. CPU usage of the REST API during the load test, measured under increasing user loads (10-5,000).

As shown in Figure 8 and Figure 9, the REST API peaked at ~1.2 GB of RAM, while the FHIR server used ~1.0 GB at the highest concurrency level (5,000 users). These values remain below the typical 2-4 GB allowances available in many

production servers. The memory usage trends were generally linear with user load, punctuated by noticeable drops likely caused by garbage collection. Ensuring these memory patterns do not trigger excessive GC cycles is crucial; in extended real-world usage, further fine-tuning of JVM or Python environment parameters may be beneficial.

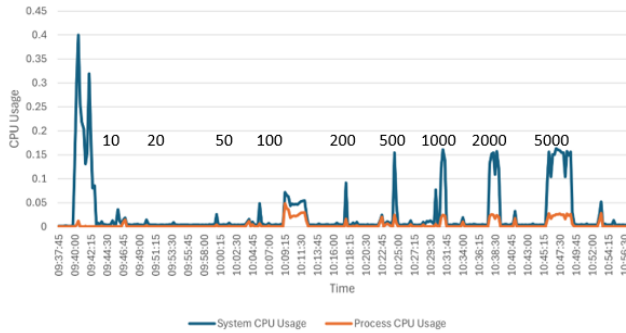


Fig. 7. CPU usage of the FHIR Server during the load test, measured under increasing user loads (10-5,000).

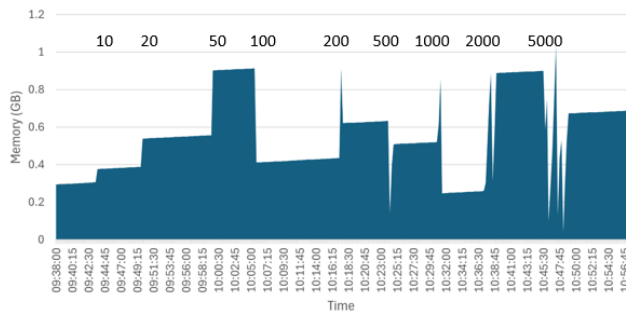


Fig. 8. Memory usage of the REST API during the load test, measured under increasing user loads (10-5,000).

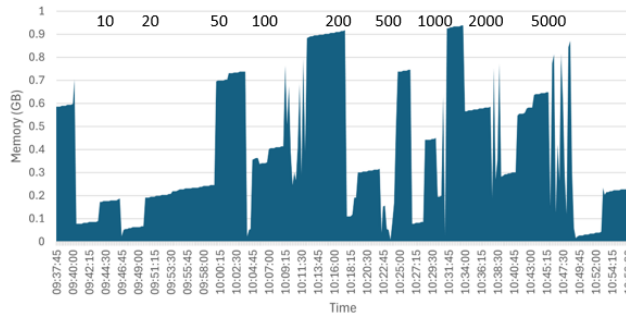


Fig. 9. Memory usage of the FHIR Server during the load test, measured under increasing user loads (10-5,000).

Over 90 Rasa containers, each handling different questionnaires or languages, scaled horizontally with minimal overhead. Chatbot interactions exhibited 90–95% of requests completing under two seconds, even under peak concurrency (Figure 10). This result underscores the effectiveness of containerized chatbots in isolating NLP tasks from the core FHIR server or REST API load. Notably, any minor delays observed often aligned with external resource calls (e.g., retrieving large or complex Questionnaire definitions).

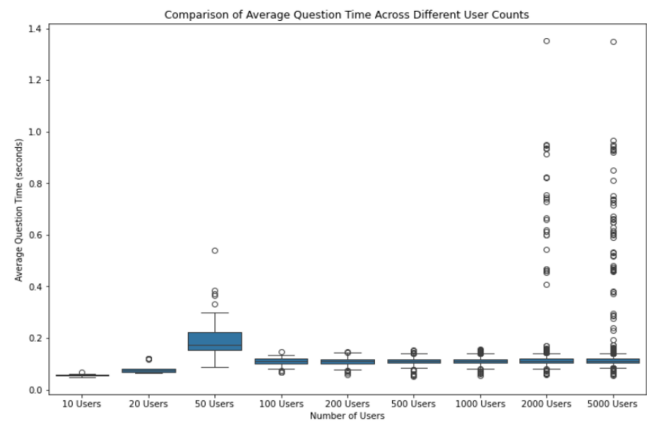


Fig. 10. Average Question Response Time for group of users (10-5,000).

As represented in Figure 11 and Figure 12 the system sustained around 330 requests/sec on the REST API and ~27 requests/sec on the FHIR server without producing significant errors. Throughout our tests, the error rate remained near zero, attesting to the system’s resilience. We did see slight throughput fluctuations at maximum loads, indicating that load-balancing or caching strategies could further smooth performance. However, no critical failure conditions or unhandled exceptions were recorded during the test runs.

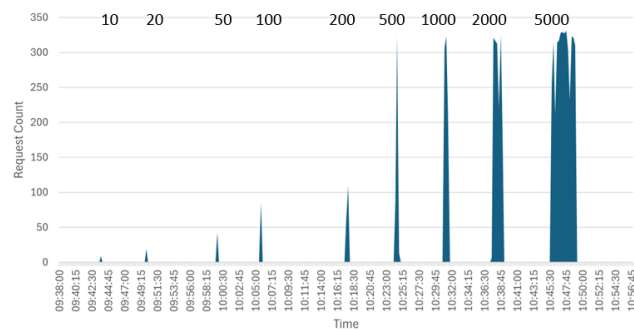


Fig. 11. Request count of the REST API during the load test, measured under increasing user loads (10-5,000).

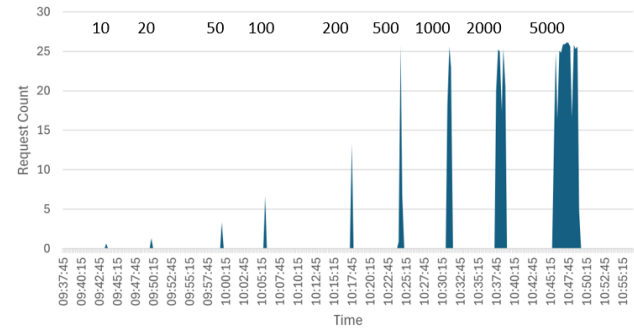


Fig. 12. Request count of the FHIR Server during the load test, measured under increasing user loads (10-5,000).

The entire system successfully served requests under heavy loads, though periodic response-time spikes appeared. These

likely stem from concurrency surges, garbage collection cycles, or Keycloak token validation overhead. Introducing advanced load-balancing or asynchronous request handling could mitigate such bottlenecks in a production environment. Summary of the results is provided in Table II.

TABLE II. THE SYSTEM PERFORMANCE SUMMARY.

Metric	10–500 Users	500–1,000 Users	1,000–2,000 Users	2,000–5,000 Users
Avg. REST API Response Time (ms)	< 50	60–80	100–120	120–180 (peaks)
Avg. FHIR Server Response Time (ms)	< 100	100–300	300–600	600–900 (peaks)
REST API CPU Usage (%)	1–18	15–20	15–20	15–20
FHIR Server CPU Usage (%)	1–17	15–20	15–20	15–20
Memory (REST API)	~0.2 GB	~0.5 GB	~0.8 GB	~1.2 GB
Memory (FHIR Server)	~0.1 GB	~0.4 GB	~0.7 GB	~1.0 GB
Request Throughput (API/sec)	~330	~330	~330	~330
Request Throughput (FHIR/sec)	~27	~27	~27	~27
Chatbot Completion Time	~1.1s	~1.3s	~1.7s	~2.1s
Observed Errors (%)	0%	0%	0%	0%

While the current load testing focused on user concurrency, future experiments will simulate educational use cases, such as class-based simulated diagnoses using synthetic patient data, to assess platform responsiveness in academic settings. Although the platform demonstrates strong technical performance, its broader potential extends beyond clinical applications into education and training, as discussed next.

V. EDUCATIONAL USE CASES OF THE PLATFORM

Beyond clinical deployment, the proposed HL7 FHIR-based platform demonstrates strong potential as an educational tool for both healthcare professionals and patients. Digital health systems increasingly serve dual purposes, not only as operational infrastructure but also as learning environments for medical trainees, researchers, and patients seeking to improve health literacy.

The platform’s modular architecture, built around microservices and standardized FHIR resources, lends itself to creating interactive educational scenarios. For instance, resources such as QuestionnaireResponse, CarePlan, and PlanDefinition can be repurposed to simulate diagnostic cases, treatment planning exercises, and decision-making pathways.

By utilizing synthetic or anonymized patient data, institutions could create structured clinical simulations to train medical students and interns in a realistic, privacy-compliant environment. In a supervised simulation mode, students could receive anonymized patient records through the FHIR Dashboard, make clinical decisions, and receive automated feedback based on predefined CarePlan responses.

The FHIR Dashboard further supports this use case. It can be extended to allow learners to view and manipulate FHIR resources under supervision, enabling real-time engagement with structured patient data. Coupled with Keycloak’s role-based access control, educational roles (e.g., student, trainee, observer) can be created with restricted permissions, ensuring that educational activities do not compromise real clinical data integrity.

Additionally, the integration of Rasa chatbots introduces a novelty for interactive learning. These bots can deliver context-aware health education, interactive case-based quizzes, and simulated patient interviews. This capability enables asynchronous and multilingual training sessions for a variety of learner levels, from early-stage students to continuing medical education participants.

In patient-centered education, the chatbot could serve as a virtual health coach, providing personalized education on treatment adherence, lifestyle changes, or condition-specific advice. This dual functionality, supporting both clinician and patient learning, makes the platform an attractive candidate for broader deployment in academic medical centers and digital health training initiatives.

By incorporating such features, the platform addresses the growing need for capacity-building tools in digital health, and it aligns with modern trends in competency-based medical education, which emphasize real-world readiness and digital tool fluency. Building on the technical results and educational applications outlined above, we now synthesize the findings and explore the platform’s broader implications.

VI. DISCUSSION

Overall, the system meets the needs of a large-scale, secure clinical environment. Key findings include strong scalability, where modular microservices architectures prevent single points of failure and enable the platform to handle high concurrency. Meanwhile, Keycloak authentication checks introduce only moderate latency, primarily during token renewal intervals for large user groups.

Despite strong performance, a few bottlenecks emerged. Response-time spikes (up to 900 ms) at the FHIR server typically aligned with concurrency surges, highlighting areas where thread pools or database indices might be refined. Although Keycloak-based authentication is essential for security, repeated token checks can add overhead at scale, especially with short token lifespans or complicated role checks. Memory usage displayed periodic drops, pointing to potential GC cycles. While not catastrophic, these cycles could exacerbate latency spikes when concurrency is already high.

In realistic hospital settings, systems must handle bursts of activity, shift changes, mass check-in times, or major system events like an influx of telehealth patients. Observed latency

spikes in these load tests mirror the concurrency surges seen in real-world hospital usage patterns. Mitigating these spikes via caching or asynchronous processing would ensure that clinicians receive data quickly, thus reducing the risk of treatment delays.

From a research perspective, the platform’s capacity to handle large user loads is beneficial for multi-center clinical trials. When multiple study sites or thousands of participants simultaneously submit questionnaires, the system can maintain reliable performance. This reliability also makes it a candidate for real-time data collection in digital health interventions, such as remote monitoring apps.

Load tests simulated synthetic user workflows, focusing on read/write operations involving common FHIR resources. More complex real-world scenarios (e.g., high volumes of binary file attachments in DocumentReference or complex queries across multiple FHIR resources) might reveal new bottlenecks. Additionally, the tests were performed in a controlled network environment; real-world deployments may face variable latency, intermittent connectivity, or external dependencies that further affect performance.

As summarized in Table III, our platform differs notably from existing HL7 FHIR implementations in both scope and design. While other studies have addressed data linkage, cloud-native deployments, or secure access management, few provide a unified, high-concurrency solution that integrates interactive front-end components and fine-grained access control. Additionally, this work is one of the first to explicitly examine the educational potential of a FHIR-based platform, a dimension often overlooked in prior research.

TABLE III. COMPARATIVE ANALYSIS OF HL7 FHIR-BASED PLATFORMS.

Platform / Study	Architecture	Scalability	Security	Educational Support
Ginavane & Prasanna (2024)	Monolithic (GCP services)	Moderate (Cloud-based)	Google IAM, Blockchain overlay	Not addressed
Ngo et al. (2024)	Federated / Centralized	Low to Moderate	Custom privacy-preserving layers	Not addressed
Opie (2024) [8]	FHIR Server + Security Audit	Not specified	OAuth 2.0, role-based auth	Not addressed
This Study (Our Platform)	Microservices + Containers	High (5,000 concurrent users)	Keycloak-based + Token Security	Strong potential (Chatbot, Dashboard, Roles)

Unlike previous HL7 FHIR-based implementations, our work uniquely integrates: (1) high-concurrency load testing, (2) Keycloak-based role enforcement, (3) educational roles for patient and clinician training, and (4) chatbot-driven multilingual patient interaction, all in one unified platform.

Potential directions for improvement include employing Kubernetes or similar platforms could dynamically scale

resources based on real-time metrics and traffic patterns, mitigating concurrency surges without manual intervention. Implementing distributed caching layers (e.g., Redis) might decrease repeated retrieval times for frequently accessed data, especially for read-heavy operations like lab results or demographics. Extending token lifespans for certain tasks, employing token introspection caches, or offloading identity checks could streamline request handling. Fine-tuning indexes, exploring partitioned databases, or using specialized FHIR servers with proven scalability may further reduce high-load latency. Deploying the platform in multiple geographic regions could enhance responsiveness for globally distributed clinical environments and research sites.

VII. CONCLUSION

This paper demonstrates that a secure and scalable HL7 FHIR platform can be achieved by integrating containerized microservices, robust identity management, and performance-aware architecture, even under high concurrency conditions. Through systematic load testing with up to 5,000 concurrent users, we validated the platform’s ability to maintain authentication integrity via Keycloak and to sustain high transaction volumes with moderate resource consumption. The key contributions of this work include:

- (1) a reference architecture combining containerized services, a flexible REST layer, and secure token-based authentication; and
- (2) empirical performance data to guide future deployments of large-scale, standards-compliant healthcare platforms.

To the best of our knowledge, this is one of the first HL7 FHIR-based platforms designed to simultaneously support clinical, research, and educational use cases within a unified and secure infrastructure.

Future improvements may include Kubernetes-based autoscaling, distributed caching (e.g., Redis), and advanced indexing strategies to reduce latency. As digital health ecosystems expand into telehealth, remote monitoring, and multi-site clinical trials, this study offers a roadmap for achieving scalable and secure interoperability without compromising user experience.

ACKNOWLEDGMENT

This work was supported by the Slovenian Research Agency (Research Core Funding) No. 0552-0796 P2-0069, Young Researcher Funding 0733/2022/P157/522-KZ and from the European Union’s research and innovation programme, project SMILE, supported under grant agreement No 101080923. The content of this paper does not reflect the official opinion of the European Union or any other institution. Responsibility for the information and views expressed therein lies entirely with the authors.

DECLARATION OF GENERATIVE AI AND AI-ASSISTED
TECHNOLOGIES IN THE WRITING PROCESS

The authors wrote, reviewed and edited the content as needed and verifies that none utilised artificial intelligence (AI) tools were used. The authors take full responsibility for the content of the publication.

References

- [1] H. M. M. Allabbas *et al.*, “Improving Patient-Centered Care through Process Optimization in Medical Clinics: A Review,” *J. Ecohumanism*, vol. 3, no. 8, Art. no. 8, Dec. 2024, doi: 10.62754/joe.v3i8.5524.
- [2] G. M. S. Himel, Md. M. Islam, Kh. A. Al-Aff, S. I. Karim, and Md. K. U. Sikder, “Skin Cancer Segmentation and Classification Using Vision Transformer for Automatic Analysis in Dermoscopy-Based Noninvasive Digital System,” *Int. J. Biomed. Imaging*, vol. 2024, no. 1, p. 3022192, 2024, doi: 10.1155/2024/3022192.
- [3] Y. Chen, C. U. Lehmann, and B. Malin, “Digital Information Ecosystems in Modern Care Coordination and Patient Care Pathways and the Challenges and Opportunities for AI Solutions,” *J. Med. Internet Res.*, vol. 26, no. 1, p. e60258, Dec. 2024, doi: 10.2196/60258.
- [4] “Index - FHIR v5.0.0.” Accessed: Apr. 01, 2025. [Online]. Available: <https://hl7.org/fhir/>
- [5] R. Gazzarata *et al.*, “HL7 Fast Healthcare Interoperability Resources (HL7 FHIR) in digital healthcare ecosystems for chronic disease management: Scoping review,” *Int. J. Med. Inf.*, vol. 189, p. 105507, Sep. 2024, doi: 10.1016/j.ijmedinf.2024.105507.
- [6] P. Tabari, G. Costagliola, M. D. Rosa, and M. Boeker, “State-of-the-Art Fast Healthcare Interoperability Resources (FHIR)-Based Data Model and Structure Implementations: Systematic Scoping Review,” *JMIR Med. Inform.*, vol. 12, no. 1, p. e58445, Sep. 2024, doi: 10.2196/58445.
- [7] V. Šafran, S. Horvat, B. Ilijevec, I. R. Roj, V. Flis, and I. Mlakar, “Integrating HL7 FHIR into Clinical Decision Support Systems: A Real-World Application with Pepper Humanoid Robot in Hospital During Doctor Visits,” in 2024 9th International Conference on Mathematics and Computers in Sciences and Industry (MCSI), Aug. 2024, pp. 139–145. doi: 10.1109/MCSI63438.2024.00031.
- [8] C. A. Opie, “Exploring Security Vulnerabilities in FHIR Server Implementations: A Case Study on IBM’s FHIR Server in the Context of the 21st Century Cures Act,” M.S., University of Hawai’i at Manoa, United States -- Hawaii, 2024. Accessed: Mar. 31, 2025. [Online]. Available: <https://www.proquest.com/docview/3114269633/abstract/E379FD3C234B4DAFPQ/1>
- [9] Building healthcare microservices: a FHIR-native approach.” Accessed: Jul. 22, 2025. [Online]. Available: <https://www.health-samurai.io/articles/building-healthcare-microservices-a-fhir-native-approach>
- [10] J. I. Akerele, A. Uzoka, P. U. Ojukwu, and O. J. Olamijuwon, “Improving healthcare application scalability through microservices architecture in the cloud,” *Int. J. Sci. Res. Updat.*, vol. 8, no. 2, pp. 100–109, 2024, doi: 10.53430/ijrsru.2024.8.2.0064.
- [11] G. N. Bettoni, T. Camargo, B. G. T. dos Santos, C. D. Flores, and F. S. D. Silva, “Application of HL7 FHIR in a Microservice Architecture for Patient Navigation on Registration and Appointments,” Jun. 2021, pp. 44–51. doi: 10.1109/SEH52539.2021.00015.
- [12] S. N. Duda *et al.*, “HL7 FHIR-based tools and initiatives to support clinical research: a scoping review,” *J. Am. Med. Inform. Assoc.*, vol. 29, no. 9, pp. 1642–1653, Sep. 2022, doi: 10.1093/jamia/ocac105.
- [13] G. A and S. Prasanna, “Improving Healthcare Data Management in HL7-Based EHR Systems with the Secure Infrastructure of Google Cloud Platform,” in 2024 Second International Conference on Intelligent Cyber Physical Systems and Internet of Things (ICoICI), Aug. 2024, pp. 195–202. doi: 10.1109/ICoICI62503.2024.10696524.
- [14] A. M. Drelick, C. Woodfield, and J. E. Freedman, “Educational chatbot development informed by clinical simulations,” *Interact. Learn. Environ.*, vol. 33, no. 3, pp. 2044–2055, Mar. 2025, doi: 10.1080/10494820.2024.2388782.
- [15] V. M. Ngo, G. Sood, F. Donohue, P. Kearney, C. Buckley, and M. Roantree, “Using HL7-FHIR as an Integration Platform for Chronic Disease Services Management and Planning in the Irish Healthcare Sector,” presented at the The Joint Conference of ISEH ICEPH & ISEG on Environment and Health, Galway, Ireland: ISEH-ICEPH, Aug. 2024. Accessed: Apr. 02, 2025. [Online]. Available: <https://www.universityofgalway.ie/iseh-iceph/>
- [16] Z. Aboushanab, “Optimizing Containerized Spring Boot Microservices in Kubernetes: Development, Experimentation, and Performance Analysis,” 2024. doi: 10.13140/RG.2.2.35888.78081.
- [17] E. S. Rigas *et al.*, “Semantic interoperability for an AI-based applications platform for smart hospitals using HL7 FHIR,” *J. Syst. Softw.*, vol. 215, p. 112093, Sep. 2024, doi: 10.1016/j.jss.2024.112093.

Contribution of individual authors to the creation of a scientific article (ghostwriting policy)

- Valentino Šafran contributed to conceptualization, formal analysis, methodology, software development, validation – formal analysis, investigation, data curation, original draft, writing – review & editing.
- Umut Arioz contributed to software development, validation, formal analysis, investigation, data curation, writing – review & editing.
- Rigon Sallauka contributed to software development, validation, formal analysis, investigation, data curation, writing – writing – review & editing.
- Izidor Mlakar contributed to conceptualization, methodology, validation, project administration – original draft, writing – review & editing – funding acquisition, project administration, supervision.

Sources of funding for research presented in a scientific article or scientific article itself

This work was supported by the Slovenian Research Agency (Research Core Funding) No. 0552-0796 P2-0069, Young Researcher Funding 0733/2022/P157/522-KZ and from the European Union's research and innovation programme, project SMILE, supported under grant agreement No 101080923. The content of this paper does not reflect the official opinion of the European Union or any other institution. Responsibility for the information and views expressed therein lies entirely with the authors.

Conflicts of Interest

The authors have no conflicts of interest to declare that are relevant to the content of this article.

Creative Commons Attribution License 4.0 (Attribution 4.0 International, CC BY 4.0)

This article is published under the terms of the Creative Commons Attribution License 4.0
https://creativecommons.org/licenses/by/4.0/deed.en_US