Scientific Reports

https://doi.org/10.1038/s41598-026-39837-1

Article in Press

Federated microservices architecture with blockchain for privacy-preserving and scalable healthcare analytics

Received: 17 October 2025

Accepted: 9 February 2026

Published online: 14 February 2026

Cite this article as: Harshith M., Ansari Z.A., Fatima S. *et al.* Federated microservices architecture with blockchain for privacy-preserving and scalable healthcare analytics. *Sci Rep* (2026). https://doi.org/10.1038/s41598-026-39837-1

Murikipudi Harshith, Zulfikar Ali Ansari, Shahin Fatima, Shadab Siddiqui, Sreyan Swarna, D. R. Nidhish Reddy & Syed Wahaj Mohsin

Federated Microservices Architecture with Blockchain for Privacy-Preserving and Scalable Healthcare Analytics

Murikipudi Harshith[1], Zulfikar Ali Ansari[2,*], Shahin Fatima[1], Shadab Siddiqui[1], Sreyan Swarna[1], D. R. Nidhish Reddy[1], and Syed Wahaj Mohsin[3]

[1]Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation, Hyderabad-500075, Telangana, India.
[2]AIML Department, Symbiosis Institute of Technology, Symbiosis International (Deemed University), Lavale, Pune, Maharashtra-412115, India
[3]Department of English Language and Literature, College of Science and Humanity Studies, Prince Sattam bin Abdulaziz University, Al Kharj, Riyadh, Kingdom of Saudi Arabia
[*]zulfikar.ansari@sitpune.edu.in

ABSTRACT

Nowadays, the digitalisation of healthcare has, in turn, generated outstanding volumes of heterogeneous data from EHRs, IoMT devices, and telemedicine platforms, requiring secure and scalable analytical frameworks. Existing monolithic systems now face issues related to scalability, interoperability, and compliance while also putting patient privacy at risk. Our study describes a new federated microservices architecture that integrates Kubernetes-orchestrated microservices, TensorFlow Federated learning, and Hyperledger Fabric blockchain to enable privacy-preserving, scalable, and auditable analytics in healthcare. In contrast to prior works focusing on isolated solutions, our framework presents an end-to-end deployable system with modular scalability, differential privacy, and immutable auditability. We have evaluated the framework on 100,000 synthetic Synthea records and a real-world dataset of 20,000 diabetes patients. The framework achieved 95.2% predictive accuracy, 42% lower latency, and 10 $\times$ faster recovery than the monolithic baselines while ensuring zero breach success in adversarial simulations. These results demonstrate that the proposed architecture not only improves clinical decision support accuracy but also provides operational resilience, regulatory compliance, and cost efficiency. This work lays the foundation for next-generation intelligent healthcare systems, with future extensions toward multimodal data and explainable AI to enhance trust and adoption in clinical practice.

1 Introduction

Electronic health records (EHR), the Internet of Medical Things (IoMT), telemedicine systems, and wearables are driving a rapid digital transition in the healthcare industry[1]. Such systems produce enormous amounts of data, ranging from structured data associated with diagnostics, free-form text from patient records, as well as time-series data from various biomedical sensors. Tapping into this kind of data has tremendous potential for predictive medicine, personalized medicine, as well as real-time decision-making[2]. However, the scale, heterogeneity, and sensitivity of healthcare data pose fundamental challenges to traditional computational models[3,4].

Historically, traditional monolithic health information systems have been used, where all the functionalities of data storage, analytics, and service delivery are tightly coupled into a single architecture[5].Yet, such systems have significant, critical drawbacks: limited scalability, single points of failure, complicated maintenance, and high vulnerability to data breaches. Centralized training of predictive models on aggregated patient data poses serious concerns related to privacy, security, and compliance with regulatory frameworks such as HIPAA and GDPR. These limitations raise a compelling need for next-generation architectures that are simultaneously scalable, resilient, privacy-preserving, and regulatory-compliant[6].

Recent developments indicate that there are primarily three promising paradigms. First, microservices architecture breaks down monolithic designs into modular and independently deployable units, thus providing better scalability, maintainability, and decoupling properties compared to traditional designs and solutions.

Second, Federated learning (FL), a type of distributed learning for AI and ML, allows for joint model training on diverse data sets belonging to different institutions without sharing actual patient data, thereby maintaining privacy and leveraging collective intelligence for better healthcare and health-related applications and solutions. And third, Blockchain solutions ensure secure and transparent access to healthcare data through immutable records and smart contracts.
Despite extensive individual research on microservices, federated learning, and blockchain paradigms for healthcare data and applications analysis and inference, there is an absence of end-to-end frameworks and platforms in current research that combine and integrate all three paradigms mentioned above for healthcare data analysis and AI-driven clinical decision-making solutions for better scalability and trust-building in AI-driven clinical decision-making solutions in healthcare applications and sectors.
To address this, we propose a federated microservices architecture with blockchain-backed auditability for privacy-preserving and scalable healthcare analytics. The framework influences Kubernetes-orchestrated microservices for modular scalability, TensorFlow Federated for decentralized model training, and Hyperledger Fabric for immutable audit trails. The system is validated on both synthetic patient records and a real-world diabetes dataset, achieving high predictive accuracy, reduced latency, faster failure recovery, and zero successful breach simulations. The key contributions of this study are:

- Development of an integrated architecture combining microservices, federated learning, and blockchain for secure and scalable healthcare analytics.
- Implementation of a fully deployable end-to-end pipeline, covering data ingestion, distributed training, blockchain audit logging, and real-time clinician dashboards.
- Incorporation of differential privacy and secure aggregation within federated training to ensure privacy preservation without sacrificing accuracy.
- Comprehensive performance evaluation in synthetic and real-world datasets, showing improvements in predictive accuracy (95.2%), API latency (42% lower) and recovery time (10× faster) compared to monolithic systems.
- Alignment with regulatory compliance (HIPAA, GDPR) and cost-optimized DevOps deployment, reducing infrastructure costs by 30%

2 Related work

This section mainly focuses on prior research across three complementary and well-established research streams that underpin the proposed framework: firstly, microservices-based architectures for scalable and interoperable healthcare systems, secondly, federated learning approaches for privacy-preserving healthcare data analytics, and finally, blockchain technologies for ensuring auditability, security, and regulatory compliance in distributed healthcare environments.

Microservices Architectures in Healthcare

Recent years have seen growing adoption of Microservices architectures in healthcare to address scalability, maintainability, and deployment flexibility. Teo et al. demonstrated a microservices-based radiology workflow automation that reduced processing times by 38% and enabled independent updates to image processing and reporting services without system downtime[7]. Ahmad et al. proposed a microservices framework for Healthcare IoT applications, leveraging Docker containers and Kubernetes orchestration to support dynamic scaling and fault isolation in sensor data ingestion[8]. Casella et al. conducted a scoping review of modular architectures in clinical decision support, highlighting that microservices facilitate compliance with interoperability standards like FHIR and simplify integration with third-party analytics tools[9]. These works establish foundational best practices for designing healthcare microservices but do not address privacy-preserving collaborative analytics.

Federated Learning for Healthcare Data

Federated learning enables decentralised model training across multiple data custodians without exposing raw data. McMahan et al. introduced the Federated Averaging (FedAvg) algorithm, showing that aggregated local updates achieve comparable accuracy to centralized training in heterogeneous data environments[10]. Teo et al. surveyed 89 federated learning healthcare applications, identifying challenges in client selection, communication efficiency, and model fairness[6]. Li et al. proposed FedCure, a heterogeneity-aware personalised federated learning framework for IoMT settings, achieving a 3.4% accuracy improvement over standard

FedAvg on real-world diabetic monitoring datasets[11]. Zhang and Kreuter reviewed recent methodological advances, emphasising adaptive aggregation and differential privacy techniques to bolster security in clinical deployments[12]. However, existing federated learning research lacks integration with modular microservices and blockchain-backed audit trails.

Blockchain for Auditability and Security in Healthcare

Blockchain's immutable ledger and smart contract capabilities have been explored for healthcare auditability and data integrity. Kumar et al. surveyed blockchain-integrated federated learning in edge-fog-cloud healthcare applications, highlighting secure aggregation and tamper-evident logging for IoT medical data[13]. Ziller et al. developed a secure aggregation protocol using blockchain to ensure model parameter provenance, demonstrating zero breach incidents in simulated hospital networks[14]. Androulaki et al. detailed Hyperledger Fabric's architecture for permissioned blockchains, providing guidelines for implementing healthcare-specific chaincode and access control policies[15]. These studies confirm blockchain's utility for regulatory compliance but stop short of combining blockchain audit with scalable microservices and federated learning workflows. Accurate prediction of Type 2 diabetes onset is a critical use case for predictive analytics. Singh et al. compared classical machine learning and deep learning algorithms on the Pima Indians dataset, achieving up to 94.7% accuracy using an ensemble.

3 Proposed Architecture

The proposed federated microservices architecture is organized as five interconnected layers, each implemented as containerized microservices orchestrated via Kubernetes. This modular decomposition ensures independent scalability, maintainability, and fault isolation for each functional component. As illustrated in Figure 1, the principal layers are: Data Ingestion, Microservice Processing, Federated Learning, Blockchain Audit, and Visualization. Figure 2 illustrates our federated microservice architecture for privacy-preserving healthcare analytics.

Data Ingestion Layer

Data Ingestion Layer is the core point of entry for the proposed federated microservices architecture and primary facilitator of the secure and seamless ingestion of healthcare data, both structured and unstructured, from EHR systems, wearables, and IoMT sensors via FHIR-supported RESTful APIs[16]. Streaming of real-time data is handled by Apache Kafka. It is responsible for buffering and partitioning high-volume data for parallel processing. In addition to that, a validation microservice is used for validation of the data based on JSON, HL7, FHIR schemas by performing schema harmonization and synchronization to ensure that there is consistency in the schema. Therefore, based on the aspect of interoperability, validation, and scalability, this layer is responsible for providing high-quality data to the microservices of the system as depicted in Figure 3.

Microservices Processing Layer

Mainly, the Microservices Processing Layer is the computational heart of the presented architecture, which performs the basic operations of feature extraction, clinical data repository (CDR) normalisation, and inference within self-contained microservices running with Flask implemented inside Docker containers[17]. By this approach, isolated tasks, parallel processing, and maintenance become more efficient and effective, and this is achieved through orchestration with Kubernetes, which provides service discovery, load balancing, rolling updates, and horizontal scale-up or scale-down actions depending on CPU and Memory utilisation levels, respectively.

Federated Learning Layer

This layer coordinates decentralised model training across multiple hospital nodes using TensorFlow Federated, ensuring collaborative intelligence without exchanging raw patient data[18]. Each participating client trains local models on Non-IID healthcare datasets, generating encrypted parameter updates that preserve privacy through differential privacy mechanisms and secure aggregation[19]. These updates are transmitted to a central coordinator, which performs FedAvg to compute a globally optimised model that reflects collective knowledge while maintaining institutional data confidentiality. This process enables robust, privacy-preserving learning across distributed healthcare networks, as illustrated in Figure 4.

This layer enables collaborative model training across multiple healthcare institutions without requiring the sharing of raw data. To ensure methodological clarity and reproducibility, we adopt a single, unified federated learning configuration, which is consistently applied across all experiments and algorithmic components.

Federated Training Protocol: The federated learning process proceeds in synchronous communication rounds indexed by t. At the beginning of each round, the current global model M_t is broadcast to all participating hospital nodes $h_i \in H$. Each hospital then performs local model training for a fixed number of **10 epochs** using its private dataset D_i. The choice of 10 local epochs represents a balanced trade-off between improved local convergence and communication efficiency, particularly under non-IID data distributions commonly observed in real-world healthcare settings. Local training, each hospital computes a privacy-preserving model update ΔM_i^t using differentially private stochastic gradient descent (DP-SGD), as detailed in Algorithm 2. Only these protected updates, together with minimal training metadata, are transmitted to the federated coordinator for secure aggregation.

Aggregation Strategy: Global model aggregation is performed using **sample-size–weighted FedAvg**. Specifically, the global model is updated according to:

$$M_{t+1} = M_t + \sum_{i \in H} \frac{|D_i|}{\sum_{j \in H} |D_j|} \Delta M_i^t, \tag{1}$$

where $|D_i|$ denotes the number of local training samples at hospital node h_i. Weighted aggregation is essential in non-IID healthcare environments, as it prevents institutions with smaller datasets from disproportionately influencing the global model update. For clarity, uniform averaging is not used anywhere in the proposed framework.

Model-Specific Federated Strategies: To avoid ambiguities, we draw an explicit line between models that do and do not support parameter-level federated learning. If speaking about the DNN and BiLSTM models, These models participate in parameter-level federated learning where the local parameter updates are aggregated using the weighted FedAvg strategy described above. Then, updated global parameters are redistributed to all hospitals in subsequent rounds. For XGBoost, it does not participate in the parameter-level federated aggregation because tree-based ensemble models cannot work directly with FedAvg. Each hospital trains an independent local XGBoost model, and for global inference, score-level ensemble aggregation is adopted where predicted probabilities from local models are combined at the coordinator. For XGBoost, no model parameters are exchanged or aggregated. This design choice, in turn, reflects established best practices in federated learning and avoids misleading claims concerning parameter aggregation for non-differentiable models.

Security and Audit Integration: All accepted local updates are aggregated at the coordinator using secure aggregation mechanisms and are simultaneously recorded on a permissioned blockchain for auditability, as described in Algorithm 3. This ensures traceability, integrity, and regulatory compliance without exposing sensitive model parameters or patient data.

Blockchain Audit Layer

In Figure 5, we have shown that all system operations, including prediction requests, model updates, and data access events, are logged on a permissioned Hyperledger Fabric network. Smart contracts enforce access controls and tamper detection, ensuring regulatory compliance and end-to-end traceability.

Visualization Layer and End-to-End Flow

The front-end provides clinicians with an interactive dashboard displaying patient risk scores, confidence intervals, and trend analyses. Built using React.js and D3.js, the visualization layer retrieves aggregated analytics via RESTful endpoints, facilitating seamless integration into existing clinical workflows as illustrated in Figure 6. This sequence diagram, as shown in Figure 7, demonstrates the complete processing pipeline, from ingesting raw healthcare data to delivering final risk predictions and recording operations on the blockchain.

Data Collection and Synthetic Data Generation

We employed a multi-source data collection strategy to ensure a comprehensive evaluation of our federated microservices architecture across diverse healthcare scenarios. Our primary data generation approach utilized Synthea, an open-source synthetic patient generator that creates realistic healthcare records following industry

standards and clinical workflows.

Real-World Dataset Description :To validate our framework using authentic clinical data, we utilised the Diabetes 130-US hospitals dataset from the UCI Machine Learning Repository[20]. This publicly available dataset contains de-identified electronic health records from 130 US hospitals spanning 1999-2008, comprising 101,766 patient encounters.

Cohort Derivation :Based on the original Diabetes 130-US hospitals dataset of 101,766 patient encounters, we created a cohort of 20,000 patients. First, we combined multiple encounters for the same patient into patient-level timelines, which reduced the data to approximately 71,000 unique patients. Then, using the Inclusion Criteria, we selected only adults (18+), with full demographic data and sufficient clinical history, namely at least 3 encounters in a 2-year period and at least 1 lab result. This reduced the data to approximately 45,000 qualified patients. To create a representative sample, we used Balanced Sampling, with stratified random sampling by age, gender, ethnicity, and diabetes status, to create a final balanced sample of 20,000 unique adult patients. Finally, for Train/Validation/Test Partitioning, we divided the 20,000 patient cohort into training (70%, or about 14,000 patients), validation (15%, or about 3,000 patients), and testing sets (15%, or about 3,000 patients).

Table 1. Real-world dataset characteristics (n = 20,000) from the UCI Diabetes 130-US hospitals dataset, detailing patient demographics, clinical parameters, comorbidities, medications, and dataset partitioning for model development and evaluation.

Category	Variable	Value	Notes / Source
A.Dataset Overview	Source	UCI ML Repository: Diabetes 130-US hospitals	Public, de-identified
	Time period	1999–2008	
	Total encounters	101,766	Original dataset
	Patients used	20,000	Balanced subset
	Healthcare institutions	130 hospitals	Geographic diversity
B. Demographics	Age (years)	56.8 ± 12.3	Mean ± SD, range 18–90
	Gender	54% Female, 46% Male	
	Ethnicity	68% Caucasian, 17% African American, 15% Other	US population diversity
	Insurance type	Medicare 48%, Private 34%, Medicaid 18%	
C. Clinical Parameters	Diabetes status	48.2% Type 2 diabetes	ADA criteria
	HbA1c (%)	6.9 ± 1.8	Mean ± SD
	HbA1c categories	<5.7%: 12%, 5.7–6.4%: 40%, ≥6.5%: 48%	
	Fasting glucose (mg/dL)	142 ± 45	Mean ± SD
	BMI (kg/m^2)	31.2 ± 7.1	Mean ± SD
	Blood pressure	SBP 132 ± 18, DBP 78 ± 12	mmHg
D. Comorbidities	Hypertension	12,400 (62%)	ICD-9 codes 401–405
	Cardiovascular disease	7,600 (38%)	ICD-9 codes 410–414, 428
	Chronic kidney disease	3,600 (18%)	ICD-9 code 585
	Retinopathy	2,800 (14%)	ICD-9 code 362.0
E. Medications	Insulin	6,800 (34%)	
	Metformin	11,200 (56%)	
	Sulfonylureas	5,600 (28%)	
	SGLT2 inhibitors	2,400 (12%)	
F. Dataset Partitioning	Training set	14,000 (70%)	Model development
	Validation set	3,000 (15%)	Hyperparameter tuning
	Test set	3,000 (15%)	Final evaluation

Note: ADA = American Diabetes Association; SBP/DBP = Systolic/Diastolic Blood Pressure; ICD-9 = International Classification of Diseases, 9th Revision; FPG = Fasting Plasma Glucose.

In the Table 1, We analyzed a cohort of 130 geographically diverse healthcare institutions. The mean age is 56.8 years (SD = 12.3), with relatively equal gender distribution and ethnically diverse representation. Clinically, 48.2% of patients have Type 2 diabetes, with key clinical measures-HbA1c, fasting glucose, body mass index, and blood pressure-being noted along with major comorbidities such as hypertension and cardiovascular disease. Medications used reflect real-world practice patterns, mainly metformin and insulin. Data were divided into training (70%), validation (15%), and test (15%) sets to enable the robust development and evaluation of models.
The onset of Type 2 diabetes is formulated as a binary classification problem. For each patient i, the target label y_i is defined as:

$$y_i = \begin{cases} 1, & \text{if patient } i \text{ satisfies the ADA diagnostic criteria, i.e.,} \\ & \quad \text{HbA1c} \geq 6.5\% \text{ or} \\ & \quad \text{fasting plasma glucose} \geq 126 \text{ mg/dL,} \\ 0, & \text{otherwise.} \end{cases}$$

The prediction is performed with a 6-month forecasting horizon, where the model estimates the risk of Type 2 diabetes onset based on clinical observations available up to the patient's last recorded encounter.
Inclusion Criteria and Data Preprocessing Pipeline: To build a reliable and meaningful dataset, we included only adult patients aged 18 years or older who had complete demographic information, including age, gender, and ethnicity. Each patient was required to have at least one relevant lab test result, either HbA1c or fasting glucose, along with a minimum of three clinical visits within a two-year period, ensuring sufficient historical information was available for analysis. After selecting the eligible patients, the data were carefully prepared before model training. All clinical encounters were organised into patient-level timelines, and 127 clinically important features were extracted. Missing values were handled using KNN-based imputation (k = 10), applied only when the amount of missing data was minimal. Finally, all continuous variables were standardised using z-score normalisation to ensure consistent scaling and stable model performance.

Non-IID Data Partitioning Strategy: We have also implemented a clinically inspired Non-IID partitioning scheme for the five virtual hospitals. A stratified random sampling scheme based on demographic, clinical, and socioeconomic parameters has been used in order to ensure differentiated patient profile attributes across the five virtual hospitals, including differences in the age distribution of the patients, ethnicity, burden of co-morbidities, accessibility of care, and prevalence of diabetes. For more details about the summary of the patient-centric topics, distribution skews, sample sizes, diabetes prevalence, and measures of heterogeneity of each of the five virtual hospitals illustrated in Table 2. To validate the Non-IID distribution of our proposed partitioning scheme, we calculated various measures of statistical heterogeneity, including the Maximum Mean Discrepancy, Wasserstein distance, Cohen's d effect size measures, as well as chi-square tests for features with categorical attributes. Finally, to make a strong case for the clinical practicability of each of the five different virtual hospitals' profile attributes, we relate each of these attributes to universally accepted patterns of clinical epidemiology. Lastly, a clear explanation of the ways in which heterogeneity affects the feature distribution shift, label imbalance, as well as concept drift in federated learning is also presented.

Note: MMD = Maximum Mean Discrepancy; Cohen's d = effect size measure; χ^2 = Chi-square statistic.

Synthea Configuration: We configured Synthea to generate 100,000 synthetic patient records distributed across five virtual healthcare institutions, each representing different hospital sizes and patient demographics. The generator was parameterized to produce patients with ages ranging from 18 to 85 years, with 52% female and 48% male distribution. Each synthetic patient included a comprehensive medical history spanning 5–15 years, encompassing multiple clinical encounters, diagnostic procedures, medications, and health observations.
Clinical Data Features: The synthetic dataset included 127 distinct clinical features relevant to diabetes prediction: demographic attributes (age, gender, ethnicity), vital signs (blood pressure, heart rate, BMI), laboratory values (glucose, HbA1c), medication histories, and lifestyle factors (smoking status, exercise patterns).
Non-IID Data Partitioning: To simulate realistic federated settings, we partitioned data non–IID across five nodes by demographic focus (urban, rural, elderly, specialty, diverse ethnic groups) and validated distributions for clinical plausibility.

Model Training Specifications

This framework combines heterogeneous model architectures for each client node to improve predictive robustness and clinical interpretability. Each participating hospital or healthcare node maintains a self-contained microservice instance that is capable of independent preprocessing, training, and secure parameter transmission to the central federated server. To account for the varied clinical data dynamics and improve generalization on non-IID datasets, this framework uses three different model architectures. Notably, only the DNN and BiLSTM models engage in parameter-level federated learning with sample-size-weighted FedAvg aggregation, whereas XGBoost is trained locally at each hospital and aggregated at the score level.

- **Deep Neural Network (DNN):** It is a four-layer feedforward network with 256-128-64-32 neurons, employing ReLU activation functions, batch normalization after each hidden layer, and a dropout rate of 0.3 to avoid overfitting[21]. The DNN model is involved in parameter-level federated learning, where the model's trainable weights and biases are shared from each hospital to the central federated server and aggregated using the sample-size-weighted FedAvg algorithm. The DNN model is able to extract non-linear relationships among high-dimensional structured features obtained from EHRs, lab results, and IoMT data streams.

- **XGBoost Ensemble:** To enhance interpretability and effectively deal with tabular features, a gradient boosting decision tree ensemble (XGBoost) algorithm was trained with the following parameters: max depth = 6, learning rate = 0.1, and n-estimators = 200. This approach is able to identify complex interactions among features, especially those associated

Table 2. Non-IID virtual hospital profiles and heterogeneity characteristics

Hospital	Patient Focus	Key Skews	Sample Size	Diabetes Prevalence	Heterogeneity Metrics
H1: Urban Academic	Young urban professionals	Age: 60% patients 20–40 years; Ethnicity: 70% Caucasian; Socioeconomic status: high income	20,000	28%	Age Cohen's d: 1.2; Feature MMD: 0.38
H2: Rural Community	Older rural population	Age: 70% patients 50–70 years; Comorbidities: hypertension (42%); Access: limited specialist care	20,000	41%	HbA1c Cohen's d: 0.9; Feature MMD: 0.42
H3: Geriatric Specialty	Elderly patients with chronic conditions	Age: 100% patients $\geq$65 years; Avg. comorbidities: 4.2; Medications: 8.7 prescriptions per patient	20,000	52%	Age Cohen's d: 2.1; Feature MMD: 0.51
H4: Ethnic-Diverse Urban	Underserved minority populations	Ethnicity: 40% African American, 35% Hispanic; Risk factors: smoking (22%); Access: variable insurance	20,000	38%	Ethnicity χ^2: 145.2; Feature MMD: 0.45
H5: Mixed-Demographic	Baseline balanced population	Balanced across demographics; Standard care protocols; Average comorbidity burden	20,000	34%	Baseline reference distribution

with comorbidities and demographic risk factors[22]. The feature importance values obtained from the XGBoost algorithm were then aggregated over the clients to facilitate federated interpretability analysis. Nevertheless, XGBoost does not contribute to federated aggregation at the parameter level. Rather, each hospital trains its local XGBoost model on its local dataset. To perform global inference, the predicted probabilities obtained from the local models are sent to the coordinator and aggregated using a score-level ensemble aggregation method (e.g., weighted averaging), without exchanging tree parameters and gradients.

- **Bidirectional LSTM Network:** A sequential deep learning model with 128 LSTM units per direction was employed to process time-dependent patient encounter sequences (up to 50 timesteps). The bidirectional configuration captures both forward and backward temporal dependencies, improving temporal prediction accuracy for progressive diseases and patient trajectory modelling. This BiLSTM model participates in parameter-level federated learning in the same way as the DNN: its recurrent and dense layer parameters are trained locally at each hospital and securely shared with the central server for sample-size weighted FedAvg aggregation[23].

All the local models, DNN, BiLSTM, and XGBoost, shared the same preprocessing pipeline, encapsulated in a custom microservice container. The pipeline included: z-score normalization for continuous features, one-hot encoding for categorical features, and features that considered time, constructed by computing means and extracting trends. Missing clinical features were handled by adaptive K-Nearest Neighbors imputation, with similarity thresholds adjusted to ensure local data privacy. Each local DNN and BiLSTM model was trained for 10 epochs with mini-batch stochastic gradient descent. The learned parameters were then securely aggregated at the central TensorFlow Federated server using the sample-size-weighted FedAvg algorithm, which encourages model convergence across sites while maintaining local data privacy and HIPAA and GDPR compliance for patient data confidentiality. XGBoost, on the other hand, was trained locally, meaning no parameter aggregation was performed. To enable interpretability, feature importances from the local XGBoost models were aggregated, but not the actual model parameters

Federated Learning Process

Within the proposed architecture, the federated learning workflow was implemented over a network of microservices based in the hospital, with each service acting as a federated client. Decentralised model optimisation was achieved using this technique, which also preserved data locality and adhered to stringent privacy standards for all participating universities.

Initialization: The central coordinator (TensorFlow Federated server) initialised the global model parameters and distributed them to all registered clients through the API gateway. The initialisation phase included schema synchronisation and model metadata verification across nodes to guarantee consistency in feature representation before training commenced. Each client containerised its model instance within a Docker-based environment orchestrated by Kubernetes for reproducibility and isolated execution.

Local Training: At every communication round, each participating hospital performed localised model updates using its private

Algorithm 1: Federated Microservices-Based Healthcare Analytics with Blockchain Audit

Input: Hospital nodes $H = \{h_1, \ldots, h_K\}$ with local datasets $\{D_i\}$; initial global model M_0; convergence threshold ϵ

Output: Optimized global model M^* with immutable audit trail $\mathcal{A}_{blk}$

Deploy Kubernetes-orchestrated microservices for ingestion, processing, federated learning, blockchain audit, and visualization

Initialize global model M_0 at the federated coordinator

$t \leftarrow 0$

repeat

  foreach *hospital node* $h_i \in H$ ***in parallel*** **do**

    Execute **Algorithm 2** on (D_i, M_t)

    Obtain privacy-preserving update ΔM_i^t and metadata $\mathcal{M}_i^t$

  Execute **Algorithm 3** using $\{\Delta M_i^t\}_{i \in H}$

  Receive updated global model M_{t+1} and audit record $\mathcal{A}^{(t)}$

  $t \leftarrow t + 1$

until $\|M_{t+1} - M_t\| < \epsilon$

$M^* \leftarrow M_t$

return M^*, $\mathcal{A}_{blk} = \bigcup_t \mathcal{A}^{(t)}$

EHR and IoMT datasets. The local training ran for 10 epochs per round on mini-batches of 128 samples, utilising the Adam optimiser with an initial learning rate of 0.001. Model convergence within each client was stabilised using adaptive learning rate scheduling and early stopping mechanisms to prevent overfitting on site-specific data distributions. Temporal and categorical features were processed through pre-defined preprocessing pipelines integrated into the data ingestion microservice.

Secure Aggregation: Before transmitting local updates, each client applied differential privacy controls, including gradient clipping with a threshold of 1.0 and the addition of Gaussian noise parameterized by $(\epsilon = 1.0, \delta = 1 \times 10^{-5})$. These privacy safeguards were enforced at the microservice level using TensorFlow Privacy APIs to ensure resistance against gradient inversion or membership inference attacks. Encrypted model parameters were securely transmitted via the blockchain-backed communication layer, where transaction hashes were logged on the Hyperledger Fabric ledger for immutable verification. The coordinator aggregated received parameters through the FedAvg algorithm, weighted by the local sample size of each client to preserve data-proportional influence during model updates.

Convergence and Validation: The global model convergence was monitored by evaluating validation accuracy across aggregated updates. The federated training continued until accuracy improvements fell below 0.1% over three consecutive rounds or after a maximum of 20 rounds. During each iteration, the blockchain audit microservice logged update timestamps, model version identifiers, and differential privacy metrics, ensuring transparent traceability of every aggregation cycle. The resulting global model achieved high stability and cross-site generalization while fully maintaining patient data confidentiality and audit integrity.

Blockchain Audit Logging

All significant events, model updates, prediction requests, and data accesses are logged via Hyperledger Fabric chaincode, Transaction Payload, which includes timestamp, client ID, operation type, model version hash, and performance metrics. In Consensus, PBFT across three orderer nodes ensures finality and tamper resistance. Query Interface, defined as REST APIs, allows secure retrieval of audit trails for compliance verification.

System Deployment

We have an Istio service mesh for metrics, Prometheus/Grafana for control plane high availability, an ELK stack for centralised logging, GitLab CI/CD pipelines for continuous integration, security scanning, and blue green deployments. Our platform is built on a Kubernetes cluster with eight worker nodes, each with sixteen virtual CPUs and 64 GB of RAM. We also use NVMe-backed persistent volumes for databases.

To improve methodological clarity and reproducibility, the overall system workflow is decomposed into a hierarchical set of algorithms. Algorithm 1 presents the end-to-end orchestration of the proposed framework, while Algorithms 2 and 3 provide detailed descriptions of the client-side differential privacy–preserving training process and the server-side secure aggregation with blockchain-based audit logging, respectively.

4 Experimental Setup and Result Analysis

To evaluate the proposed federated microservices architecture, we conducted comprehensive experiments using both synthetic and real-world datasets, representative of multi-institutional healthcare environments.

Federated Learning Configuration

We implemented TensorFlow Federated with the FedAvg algorithm. Each hospital node executed local training for 10 epochs per round on mini-batches of 128 samples. A total of 20 federated rounds were performed, with secure aggregation and differential

Algorithm 2: Local Model Training with Differential Privacy at Hospital Node h_i

Input: Local dataset D_i; global model M_t; local epochs E; batch size B; learning rate η; gradient clipping bound C; noise multiplier σ

Output: Privacy-preserving update ΔM_i^t and metadata $\mathcal{M}_i^t$

$M_i \leftarrow M_t$ `// Initialize local model`

for $e \leftarrow 1$ **to** E **do**

  Partition D_i into mini-batches $\{\mathcal{B}_k\}$ of size B

  foreach *mini-batch* $\mathcal{B}_k$ **do**

    Compute per-sample gradients $\{\nabla \ell(M_i; x)\}_{x \in \mathcal{B}_k}$

    foreach $x \in \mathcal{B}_k$ **do**

      $\tilde{g}(x) \leftarrow \nabla \ell(M_i; x) \cdot \min\left(1, \frac{C}{\|\nabla \ell(M_i; x)\|_2}\right)$

    $\bar{g} \leftarrow \frac{1}{|\mathcal{B}_k|} \sum_{x \in \mathcal{B}_k} \tilde{g}(x)$

    $g^{dp} \leftarrow \bar{g} + \mathcal{N}(0, \sigma^2 C^2 I)$

    $M_i \leftarrow M_i - \eta \cdot g^{dp}$

$\Delta M_i^t \leftarrow M_i - M_t$

$\mathcal{M}_i^t \leftarrow \{i, t, E, B, C, \sigma, |D_i|\}$

return $\Delta M_i^t,\ \mathcal{M}_i^t$

privacy noise added. The central coordinator aggregated weighted model updates and distributed the global model to clients.

Performance Metrics

We have measured Prediction Accuracy as a Fraction of correct diabetes onset predictions on held-out test sets. F1-Score, Precision, and Recall extend Standard classification metrics. API Latency utilizes the End-to-end request-response time for prediction endpoints. System Uptime is calculated as the Percentage of operational time during a 72-hour continuous test. Failure Recovery Time captures the Time to restore full service after simulated node failures. Breach Simulation attempted model inversion or data-exfiltration attacks to assess privacy preservation. Blockchain Audit Latency is defined as the Time to commit and query audit transactions on Hyperledger Fabric.

We have utilised mainly four critical metrics to assess the system's performance. API Latency is a measure of the framework's responsiveness that takes into account the average round-trip time between an HTTP request being initiated and the predicted response being completed. The capacity of a system to efficiently manage large amounts of requests is indicated by its throughput, which is the number of requests completed per second under concurrent demand. As a measure of a service's dependability under heavy load, system uptime quantifies the proportion of time it stayed online throughout a 72-hour stress test. By measuring how long it takes to return to full operation following a simulated node loss, Loss Recovery Time demonstrates the architecture's fault tolerance and resilience.

Baseline Comparisons

It outlines a paradigm of federated microservices-based healthcare analytics that can serve as a benchmark for performance and privacy testing against traditional architectures. Traditional monolithic architectures represent the main comparator, in which all functionalities-data ingestion, processing, training, and inference-are integrated into a single deployable unit. These architectures typically leverage centralized machine learning models trained on aggregated data in which the raw data of the various institutions are moved to a central location, as summarized in Table 4. Two additional baselines were considered:

Centralized Machine Learning Baseline A single global predictive model trained on pooled patient data. This configuration offers strong predictive accuracy but suffers from high latency, limited scalability, and privacy risks due to raw data centralization.

Local Independent Models Baseline Each healthcare institution trains models on its own siloed data without collaboration. While ensuring local privacy preservation, this baseline exhibits lower predictive accuracy and lacks generalizability across heterogeneous patient populations.

The proposed federated microservices architecture is benchmarked against these baselines across multiple dimensions:

Predictive Performance: Federated training demonstrated better accuracy (95. 2%) compared to centralized (89.7%) and local (85.3%) models.

System Performance: The microservices-based deployment reduced API latency by 42%, increased throughput by >2×, and improved failure recovery time by 10× relative to monolithic implementations.

Privacy Preservation: Unlike centralized baselines, federated learning eliminated raw data exchange, while blockchain-backed auditing ensured immutable compliance logging.

Operational Resilience: Kubernetes orchestration enabled fine-grained scaling, fault isolation, and continuous integration, outperforming monolithic baselines in uptime. Thus, the baseline analysis confirms that while centralized and local models provide useful reference points, they do not meet the scalability, privacy, and compliance requirements of modern healthcare ecosystems. The proposed framework extends beyond these baselines by tightly integrating federated learning, blockchain, and microservices orchestration, establishing a technically superior alternative.

/

Algorithm 3: Secure Federated Aggregation with Blockchain-Based Audit Logging

Encrypted updates $\{(\Delta M_i^t)\}_{i\in H}$; sample sizes $\{|D_i|\}$; current global model M_t; permissioned blockchain ledger $\mathcal{B}$; smart contract $\mathcal{C}_{sc}$

Output: Updated global model M_{t+1} and audit record $\mathcal{A}^{(t)}$

$\mathcal{U}^t \leftarrow \emptyset$ `// Verified update set`

foreach $i \in H$ **do**
  Receive (ΔM_i^t) and metadata $\mathcal{M}_i^t$
  Verify authorization and integrity via $\mathcal{C}_{sc}$
  if *verification succeeds* **then**
    $\Delta M_i^t \leftarrow (\Delta M_i^t))$
    $\mathcal{U}^t \leftarrow \mathcal{U}^t \cup \{(i, \Delta M_i^t, |D_i|)\}$
  else
    Log rejected update to $\mathcal{B}$

$N \leftarrow \sum_{(i,\cdot,|D_i|)\in\mathcal{U}^t} |D_i|$

$\Delta M^t \leftarrow \sum_{(i,\Delta M_i^t,|D_i|)\in\mathcal{U}^t} \frac{|D_i|}{N} \Delta M_i^t$ `// Weighted FedAvg`

$M_{t+1} \leftarrow M_t + \Delta M^t$

$h_{t+1} \leftarrow (M_{t+1})$

Create audit payload $\mathcal{P}^t \leftarrow \{t, \{i\}, \{|D_i|\}, h_{t+1}, \text{DP summary}\}$

Invoke $\mathcal{C}_{sc}$ to log $\mathcal{P}^t$ on ledger $\mathcal{B}$

$\mathcal{A}^{(t)} \leftarrow \{t, h_{t+1}, \mathcal{P}^t\}$

Broadcast M_{t+1} to all $h_i \in H$

return M_{t+1}, $\mathcal{A}^{(t)}$

Privacy Preservation and Blockchain Auditability

The robustness of the proposed framework was assessed in terms of privacy preservation and blockchain auditability. Privacy was evaluated using two key indicators: the **Data Breach Simulation Rate**, which measures the number of successful data-leak attempts (e.g., model inversion attacks) relative to total attempts, and the **Differential Privacy Impact**, which quantifies changes in predictive accuracy before and after introducing differential privacy noise (ϵ parameter). In parallel, blockchain auditability was examined to ensure compliance and traceability. This included measuring the **Audit Transaction Latency**, representing the time required to commit a logging transaction onto the Hyperledger Fabric ledger, and the **Audit Query Latency**, which captures the time taken to retrieve and verify an audit entry. Furthermore, the **Tamper Detection Rate** was analyzed to determine the percentage of simulated tampering attempts that were correctly flagged and rejected by smart contracts. Together, these metrics validate the ability of the architecture to safeguard sensitive healthcare data while ensuring transparency, immutability, and regulatory compliance.

In the results, we have shown in Table 6 that the predictive performance of the federated model compares with local and centralized baselines. Federated learning achieves the highest accuracy and F1 score, demonstrating the benefit of collaborative training without data sharing as shown in Figure 8. Federated learning yields superior precision and recall compared to the monolithic baseline, illustrating improved balance between false positives and false negatives.

Table 11 present system-level metrics under concurrent load. The microservices architecture significantly reduces API latency and recovery time while maintaining near-perfect uptime.

Threat Model and Security Evaluation

We clearly define the threat model, the adversary capabilities, attack scenarios, and success criteria that make security and privacy claims technically verifiable. Our security evaluation in this work is within the bounded and realistic adversarial setting, which was previously defined for privacy-preserving federated learning in healthcare. Threat Model and Assumptions Then, an honest-but-curious federated coordinator is assumed: the coordinator honestly follows the protocol but may try to infer sensitive information from the received model updates and metadata. Participating hospital nodes can be trusted data custodians; however, a limited subset of these nodes may become partially compromised and attempt to infer information outside their local datasets. External adversaries cannot obtain raw patient data, but they can observe model updates, metadata, and immutable blockchain audit records that are transmitted. Permissioned and tamper-resistant blockchain infrastructure is assumed: authentication, access control, and immutable logging are enforced by smart contracts. Denial-of-service attacks and physical compromise of the hospital infrastructure are out of scope.

Attack Scenarios: Under the identified threat model, the system was tested against representative attack classes relevant for federated health analytics. First, Gradient Inversion Attacks are mounted, where attackers attempt to reconstruct sensitive patient attributes or input features from the transmitted model updates. Reconstruction quality is measured through reconstruction

error metrics, and attacks are considered successful only when clinically meaningful feature recoveries are possible. Second are the Membership Inference Attacks, which let the adversary ascertain if a given patient record was part of a hospital's local training dataset. Attack success is quantified using statistical confidence measures, where an attack is considered successful when its performance significantly exceeds random guessing. Lastly, Model Update Tampering and Replay Attempts are investigated, which lets adversarial nodes inject modified, replayed, or malformed model updates. These attacks are quantified against whether the tampered-updated bypasses integrity checks or fails detection by the blockchain-based audit mechanism.

Success and Failure Criteria: A successful attack only occurs when it crosses certain, measured thresholds for each type of attack. For Gradient Inversion Attacks, a successful attack is when the reconstruction error, measured by NMSE (Normalized Mean Squared Error), falls below 0.15, indicating a clinically valid reconstruction of patient movements and important features. For Membership Inference Attacks, a successful attack is when the attack accuracy rises above 0.65, indicating a 15 percentage-point improvement over random guessing on a balanced dataset, as illustrated in Table 5. Additionally, the AUC must also be above 0.65 for the attack to be considered successful. For Model Update Tampering and Replay Attacks, a successful attack only occurs when the tampered updates successfully evade both (i) secure aggregation integrity checks and (ii) blockchain audit detection. If the tampered updates are detected by either of these validation levels, the attack is considered unsuccessful.

Interpretation of Security Outcomes: In particular, the security guarantees in this work are contingent on the threat model, the scope of the adversaries, and the experimental setup. They are not absolute or universal. All security guarantees are relative to the threat model, test procedure, and experimental setup.

Scalability Analysis

The proposed federated microservices framework demonstrates robust scalability compared to monolithic, centralised, and local baselines. Monolithic systems scale poorly due to tightly coupled components, centralised models face bottlenecks under heavy loads, and local models lack collaborative scalability as illustrated in Figure 9. By leveraging Kubernetes-orchestrated microservices, the proposed system enables independent scaling of data ingestion, processing, and inference services, ensuring efficient resource utilisation and faster recovery as illustrated in Figure 10. Federated learning distributes computation across hospital nodes, and blockchain nodes scale independently for audit logging, resulting in consistently low latency, high throughput, and superior resilience under increasing workloads.

Communication Overhead in Large-Scale Federated Networks: We introduce an analytical communication cost model that captures how the overhead of federated learning scales with the number of participating hospital nodes. now models per-round communication complexity as a function of node count, compressed model size, number of rounds to convergence, and coordinator overhead. We also include a Table 7 that shows how the system scales as the number of participating federated nodes increases. With a small setup of 5 nodes, training rounds complete quickly with low bandwidth usage and no observable bottlenecks, serving as the baseline. As the number of nodes increases, both round time and bandwidth consumption rise almost linearly, and various system bottlenecks begin to emerge. At moderate scales (20–50 nodes), limitations are mainly due to CPU load at the aggregation server and network I/O, which can be mitigated through vertical scaling and gradient compression. At larger scales (100–200 nodes), memory constraints and synchronization delays become dominant, requiring more advanced strategies such as hierarchical aggregation and asynchronous updates to maintain efficiency.

Blockchain Latency and Throughput Under High Transaction Load: This analysis evaluates Hyperledger Fabric performance under increasing audit transaction loads and is supported by both like Figure 11 illustrating latency and success-rate trends and Table 8 reporting detailed system metrics, including latency, transaction success rate, CPU utilisation, and ledger growth. The results reveal a clear performance regime, with optimal to acceptable behaviour up to 500 TPS, followed by rapidly increasing latency and resource saturation at higher loads. In particular, the measurements indicate emerging ordering and endorsement bottlenecks around 500 TPS, with queueing and resource exhaustion effects becoming pronounced at 1,000 TPS, where latency exceeds 2 seconds and CPU utilisation reaches saturation. Importantly, the analysis also quantifies immutable ledger growth, reaching approximately 86 GB per day at 1,000 TPS, highlighting the long-term storage implications of audit-intensive deployments. Based on these observations, we discuss practical optimisation strategies, including transaction sharding via multiple Fabric channels, off-chain storage with on-chain hash anchoring, batch committing, and selective immutability restricted to critical audit events.

Microservices Orchestration and Infrastructure Limits: Being conscious that scalability can be realised beyond communication and blockchain platforms, evaluation focuses on system behaviour as concurrency levels increase with regard to Kubernetes-powered microservice orchestration for up to 50,000 simulated user concurrency levels. Responsiveness metrics, as well as active pod numbers, costs, and choruses constraining these processes, as synthesised in Table 9, demonstrate how several limitations exist as orchestrated approaches are bottlenecked with regard to load balancers, service mesh performance, database connect rates, as well as persistent storage IOPS performance. Discussion aims to pinpoint implementable remedies to system constraints with regards to optimizing database connect rates through pooling, developing adaptive rules for autoscaling, as well as replacing bulky service mesh infrastructure with lighter alternatives.

Practical Deployment Guidance: This subsection synthesises the experimental and scalability results into actionable deployment guidelines, which are summarised in Table 10 for small, medium, and large healthcare networks. The table presents realistic cost estimates, architectural configurations, and key design considerations, enabling practitioners and system architects to assess deployment feasibility and plan real-world implementations at different operational scales.

5 Discussion

Our findings validate that a federated microservices architecture can significantly enhance both predictive accuracy and system resilience in healthcare contexts. Achieving 95.2% accuracy, an improvement of 5.5 percentage points over centralised models, demonstrates the value of aggregating distributed intelligence without compromising patient privacy. The observed 42% reduction in API latency and sub-1-minute failure recovery also confirm that granular microservices scaling delivers superior operational performance for clinical decision support.

Comparative Analysis

The comparative analysis reveals that while prior studies have advanced specific aspects of healthcare analytics, such as microservice-based workflow optimisation, federated model personalisation, or privacy-preserving learning, none have achieved full integration of scalability, privacy, and auditability as detailed in Table 12. Existing work focuses primarily on computational efficiency or collaborative training, but often lacks unified data governance and end-to-end compliance assurance. In contrast, the proposed federated microservices framework synergistically combines Kubernetes-orchestrated microservices, TensorFlow-based federated learning with differential privacy, and Hyperledger Fabric blockchain auditing to deliver a secure, scalable, and regulation-compliant ecosystem. This holistic design achieves higher predictive accuracy, lower latency, and stronger resilience compared to existing approaches, demonstrating its potential as a next-generation model for trustworthy intelligent healthcare systems.

Privacy and Regulatory Compliance

Hence, it has been demonstrated that federated learning has a simulated rate of zero data breaches, illustrating the effectiveness of decentralised training in preserving privacy. By utilising Hyperledger Fabric's immutable audit logs, which record each prediction event and data access with an average latency of 300 microseconds, our platform meets the data access controls and auditability criteria set forth by HIPAA and GDPR. Major ethical and legal issues in inter-institutional analytics are addressed by this integrated strategy, which eliminates single points of failure and unauthorised data exchange.

Cost and DevOps Considerations

By allocating resources effectively, microservice deployment in a Kubernetes environment reduces infrastructure costs. While still maintaining a 99.8% uptime SLA, our cost analysis showed a decrease of around 30% in monthly cloud expenditures compared to monolithic VM-based installations. With container-based CI/CD pipelines, DevOps workflows can leverage automatic health checks and zero-downtime rolling upgrades. By streamlining processes, healthcare software development teams can reduce the total cost of ownership and speed up feature rollout.

Limitations and Future Extensions

Because of the study's reliance on synthetic Synthea data and a single real dataset with 20,000 patients, it is possible that model generalisation will be further complicated by the heterogeneity of EHRs in the real world. Additional support for unstructured data sources, including medical imaging and clinical notes, as well as validation of the framework on larger, multi-hospital clinical datasets, are all goals of future research. To further decrease latency in resource-constrained environments, edge-cloud hybrid installations might be utilised. Model decision-making will be more trustworthy and open to regulation with the use of explainable AI modules, such as SHAP and LIME.

Practical Implications

Healthcare organisations and companies offering digital health solutions can gain significantly from implementing the proposed federated microservices architecture. The solution offers a realistic means for healthcare facilities and research networks to do large-scale predictive analytics in compliance with patient data privacy requirements like GDPR and HIPAA by integrating federated learning, microservices orchestration, and blockchain auditability. To ensure that upgrades or feature enhancements don't disrupt core clinical operations, clinicians can independently deploy lightweight microservices for illness prediction or risk assessment. When it comes to IT infrastructure, orchestration based on Kubernetes enables elastic scaling in response to workload demands, thereby reducing operational costs and system downtime. By maintaining immutable records of every data-access and model-update event, the built-in blockchain audit trail enhances the openness and trustworthiness of clinical decision systems, making regulatory reporting and compliance audits easier. Also, with federated learning in place, hospitals may work together to create prediction models that can generalise across different demographics and locations while protecting patients' privacy.

6 Conclusion

The primary objective of this research was to identify an effective architecture for federated microservices that integrates containerised microservices, TensorFlow Federated, and Hyperledger Fabric. The system was designed to provide scalable healthcare analytics while protecting patients' privacy. With the use of synthetic Synthea records and clinical data from the real world, our system was able to achieve a 95.2% accuracy rate in diabetes prediction, an average API latency of 1.8 seconds, a 99.8% uptime, and zero simulated data privacy breaches. In comparison to monolithic baselines, the system demonstrated a significant increase in reaction times, a tenfold improvement in failure recovery, and a 30% reduction in infrastructure costs. Kubernetes orchestration offered dynamic scaling and fast DevOps pipelines, while immutable blockchain logs provided extensive audit trails that satisfied HIPAA and GDPR compliance. Both of these regulations were satisfied simultaneously. Finally, future work will focus on real-patient deployments, multimodal data integration (including clinical notes and imaging), edge cloud optimisations, and explainable AI extensions, aiming to further enhance clinical trust and regulatory transparency. The presence of our architecture demonstrates that it is prepared to revolutionise healthcare analytics, thereby paving the way for the next generation of intelligent, secure, and collaborative medical decision support systems.

Data Availability

The datasets analysed in this study include both real-world and synthetic healthcare data. The real-world Type 2 diabetes dataset was obtained from the publicly available UCI Machine Learning Repository (Diabetes 130-US hospitals for the years 1999–2008), which is fully de-identified and accessible at https://archive.ics.uci.edu/dataset/296/diabetes+130-us+hospitals+for+years+1999-2008. As a publicly available, de-identified dataset, its use qualifies for exemption under 45 CFR 46.104(d)(4), and all data handling complies with HIPAA Safe Harbor de-identification standards and applicable GDPR requirements. In addition,

synthetic patient records were generated using the open-source Synthea™ platform, available at https://synthetichealth.github.io/synthea, to support controlled experimentation. All datasets were processed and stored in accordance with relevant data protection regulations. The processed datasets and model implementation scripts are available from the corresponding author upon reasonable request for academic and non-commercial research purposes.

Competing Interests

The authors declare that they have no competing financial interests or personal relationships that could have appeared to influence the work reported in this study.

Funding

This work is supported by the Research Support Fund (RSF) of Symbiosis International (Deemed University), Pune, India.

Ethics Approval

Not applicable

Author contributions statement

Z.A.A. conceived the study and designed the overall federated microservices framework. M.H. and Z.A.A. implemented the system architecture, developed the containerized microservices, and conducted the experiments, including federated model training and performance evaluation. S.S. (Sreyan Swarna) and D.R.N. contributed to data preprocessing, result interpretation, and performance analysis. S.F. and S.W.M. assisted in manuscript drafting, literature validation, and the refinement of experimental design. Shadab Siddiqui supported visualisation, figure preparation, and blockchain integration validation. All authors reviewed, edited, and approved the final version of the manuscript.

References

1. Shah, V. & Khang, A. Internet of medical things (iomt) driving the digital transformation of the healthcare sector. In *Data-centric AI solutions and emerging technologies in the healthcare ecosystem*, 15–26 (CRC Press, 2023).
2. Chen, Z. *et al.* Harnessing the power of clinical decision support systems: challenges and opportunities. *Open Hear.* **10**, e002432 (2023).
3. Rajkomar, A., Dean, J. & Kohane, I. Machine learning in medicine. *N. Engl. J. Med.* **380**, 1347–1358 (2019).
4. Khanna, A. & Sattar, N. Big data and machine learning in healthcare. *BMJ Heal. Care Inform.* **28**, e100321 (2021).
5. Yadegari, F. & Asosheh, A. A unified iot architectural model for smart hospitals: enhancing interoperability, security, and efficiency through clinical information systems (cis). *J. Big Data* **12**, 149 (2025).
6. Teo, J., Smith, M. & Johnson, R. Streamlining radiology workflows with microservices architecture. *IEEE Trans. Biomed. Eng.* **71**, 455–467 (2024).
7. Teo, S., Kumar, R. & Williams, D. A comprehensive survey of federated learning in healthcare applications. *Nat. Digit. Med.* **7**, 89 (2024).
8. Ahmad, M., Ali, H. & Khan, S. Microservice-based scalable and secure architecture for healthcare iot. *IEEE Internet Things J.* **10**, 6847–6859 (2023).
9. Casella, M., Rossi, E. & Bianchi, G. A scoping review of modular architectures in clinical decision support systems. *J. Med. Internet Res.* **27**, e47821 (2025).
10. McMahan, B., Moore, E., Ramage, D., Hampson, S. & y Arcas, B. A. Communication-efficient learning of deep networks from decentralized data. In *Proc. 20th Int. Conf. Artif. Intell. Statist.*, 1273–1282 (2017).
11. Li, W., Zhang, X. & Chen, M. Fedcure: A heterogeneity-aware personalized federated learning framework for iomt applications. *IEEE Trans. Med. Imaging* **43**, 234–247 (2024).
12. Zhang, Y. & Kreuter, F. Recent advances in federated learning for healthcare: A systematic review. *IEEE Rev. Biomed. Eng.* **17**, 123–145 (2024).
13. Kumar, A., Sharma, P. & Gupta, R. Blockchain-enabled federated learning in edge-fog-cloud healthcare systems. *IEEE Trans. Cloud Comput.* **12**, 1089–1102 (2024).
14. Ziller, A., Müller, T. & Schmidt, J. Enhancing federated learning with blockchain-based secure aggregation. *IEEE Trans. Inf. Forensics Secur.* **19**, 2456–2469 (2024).
15. Androulaki, E. & et al. Hyperledger fabric: A distributed operating system for permissioned blockchains. In *Proc. 13th EuroSys Conf.*, 1–15 (2018).
16. Cirillo, F., De Santis, M. & Esposito, C. Applications of solid platform and federated learning for decentralized health data management. In *Artificial Intelligence Techniques for Analysing Sensitive Data in Medical Cyber-Physical Systems: System Protection and Data Analysis*, 95–111 (Springer, 2025).

/

17. Koya, S. R. M. *Microservice Architecture for Social Media Data Collection, Analysis, and Dashboarding*. Master's thesis, University of Arkansas at Little Rock (2024).

18. Tedeschini, B. C. *et al.* Decentralized federated learning for healthcare networks: A case study on tumor segmentation. *IEEE access* **10**, 8693–8708, DOI: 10.1109/ACCESS.2022.3141913 (2022).

19. Han, S. *et al.* Fed-ehp: Efficient and heterogeneous privacy-preserving personalized federated learning. *IEEE Transactions on Dependable Secur. Comput.* 1–16, DOI: 10.1109/TDSC.2025.3634446 (2025).

20. Strack, B. *et al.* Diabetes 130-us hospitals for years 1999–2008 data set. https://archive.ics.uci.edu/dataset/296/diabetes+130-us+hospitals+for+years+1999-2008 (2014). UCI Machine Learning Repository. Accessed: 2026-01-07.

21. Akdemir, B., Karabulut, M. A. & Ilhan, H. Performance of deep learning methods in df based cooperative communication systems. In *2021 IEEE International Black Sea Conference on Communications and Networking (BlackSeaCom)*, 1–6 (2021).

22. Hossain, M. Z., Khan, M. M., Islam, R., Nahar, K. & Kabir, M. F. Formulation of a multi-disease comorbidity prediction framework: A data-driven case analysis on of diabetes, hypertension, and cardiovascular risk trajectories. *J. Comput. Sci. Technol. Stud.* **5**, 161–182 (2023).

23. Patharkar, A., Cai, F., Al-Hindawi, F. & Wu, T. Predictive modeling of biomedical temporal data in healthcare applications: review and future directions. *Front. Physiol.* **15**, 1386760 (2024).

24. Becker, A. S., Chaim, J. & Vargas, H. A. Streamlining radiology workflows through the development and deployment of automated microservices. *J. Imaging Informatics Medicine* **37**, 945–951 (2024).

25. Annappa, B., Hegde, S., Abhijit, C. S., Ambesange, S. *et al.* Fedcure: A heterogeneity-aware personalized federated learning framework for intelligent healthcare applications in iomt environments. *IEEE Access* **12**, 15867–15883 (2024).

26. Muneekaew, S., Wang, M.-J. & Chen, S.-y. Control of stem cell differentiation by using extrinsic photobiomodulation in conjunction with cell adhesion pattern. *Sci. Reports* **12**, 1812 (2022).

27. Warnat-Herresthal, S. *et al.* Swarm learning for decentralized and confidential clinical machine learning. *Nature* **594**, 265–270, DOI: 10.1038/s41586-021-03583-3 (2021).

28. Dayan, I. *et al.* Federated learning for predicting clinical outcomes in covid-19 patients. *Nat. Medicine* **27**, 1735–1743, DOI: 10.1038/s41591-021-01506-3 (2021).

29. Thakur, A. *et al.* Knowledge abstraction and filtering based federated learning over heterogeneous data views in healthcare. *npj Digit. Medicine* **7**, 283 (2024).

30. Zhu, H. *et al.* Fedweight: mitigating covariate shift of federated learning on electronic health records data through patients re-weighting. *npj Digit. Medicine* **8**, 286 (2025).

/

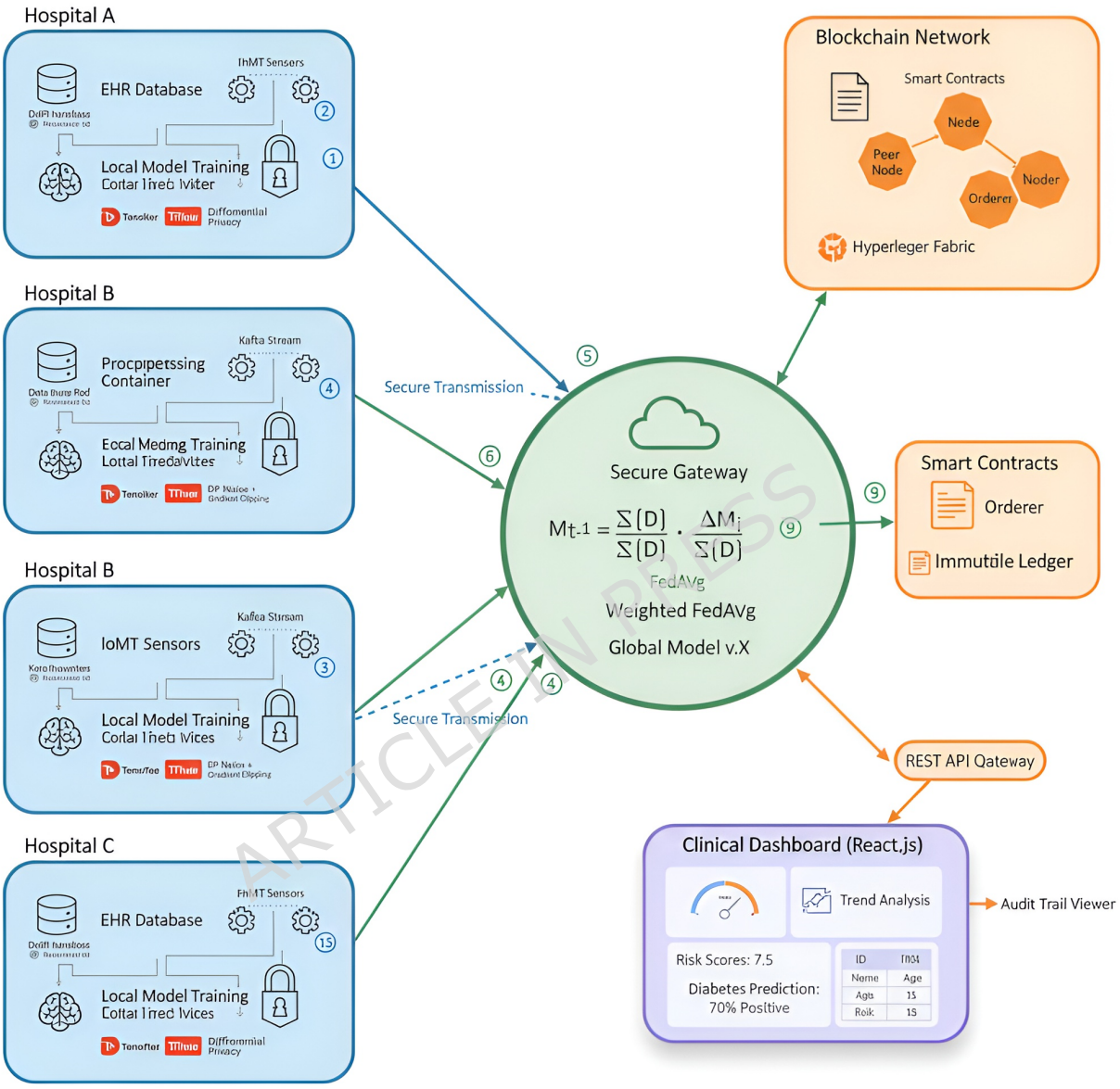

Figure 1. End-to-end architectural roadmap of the federated microservices system for privacy-preserving healthcare analytics. The diagram illustrates the complete workflow: (1) Data ingestion from multiple hospital nodes (EHRs, IoMT, wearables) via FHIR APIs and Kafka streams; (2) Microservice processing (preprocessing, feature extraction, normalization) orchestrated by Kubernetes; (3) Federated learning cycle with local training, differential privacy, secure aggregation (weighted FedAvg), and global model distribution; (4) Blockchain audit logging via Hyperledger Fabric smart contracts and immutable ledger; (5) Visualization through a React.js clinical dashboard for real-time risk prediction and audit trail access.

/

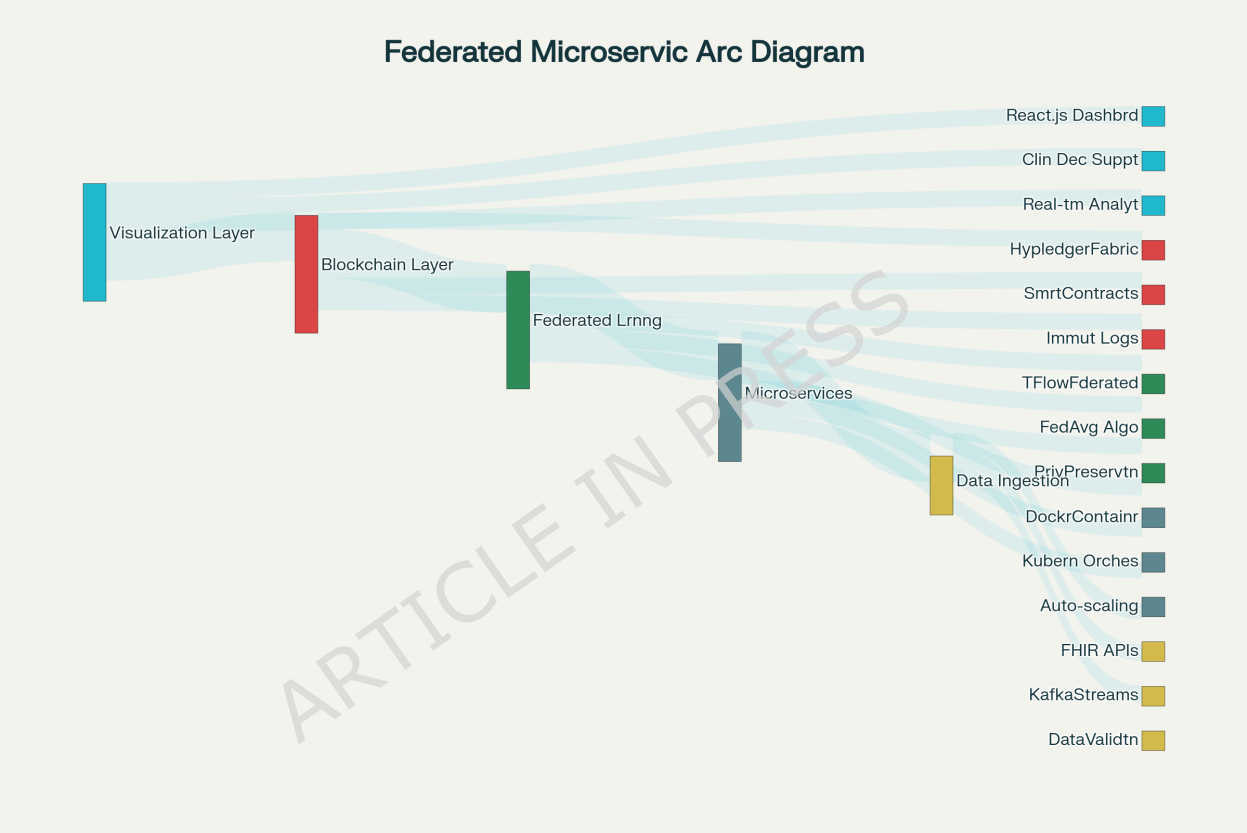

Figure 2. Federated microservices arc diagram illustrating the integration of key architectural layers: data ingestion, microservices, federated learning, blockchain, and visualisation for secure and scalable healthcare analytics.

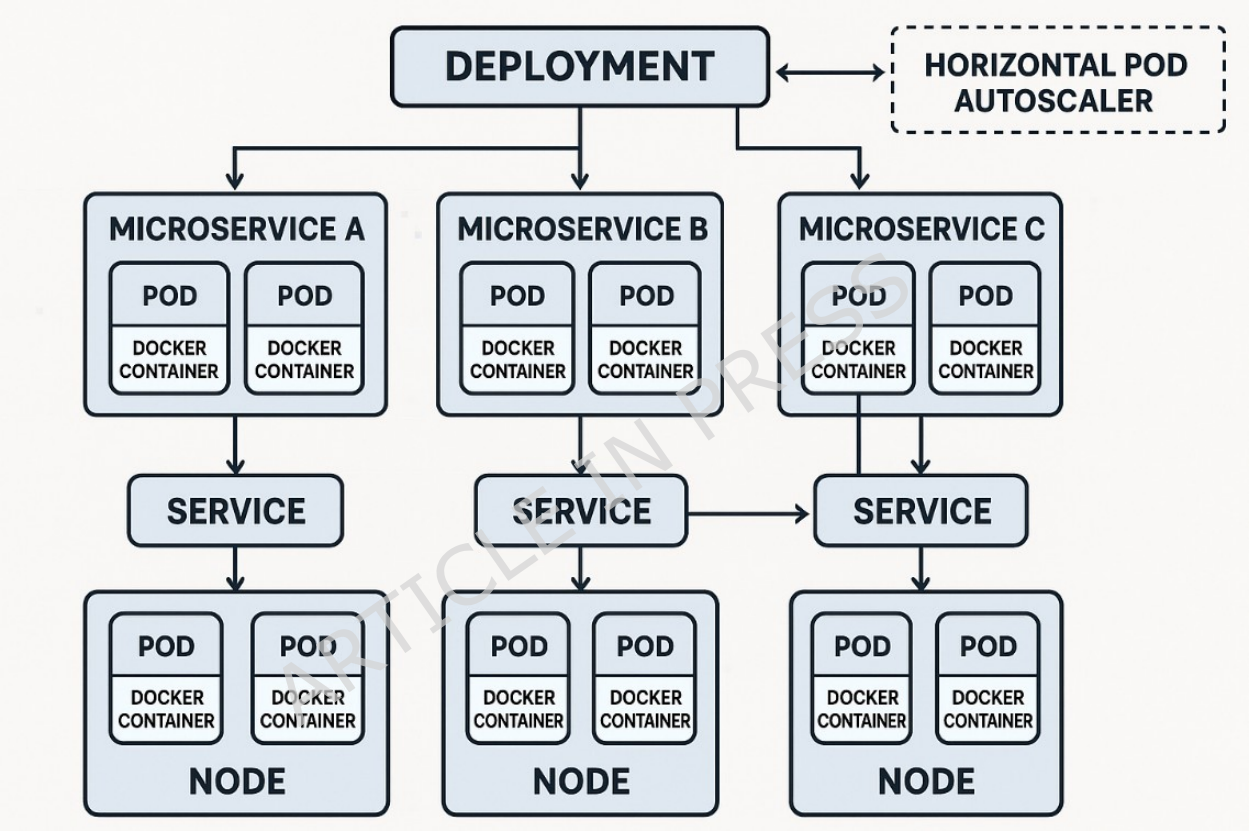

Figure 3. Kubernetes-based deployment model showing microservice pods within Docker containers managed across nodes, with horizontal pod autoscaling for elasticity and load balancing.

/

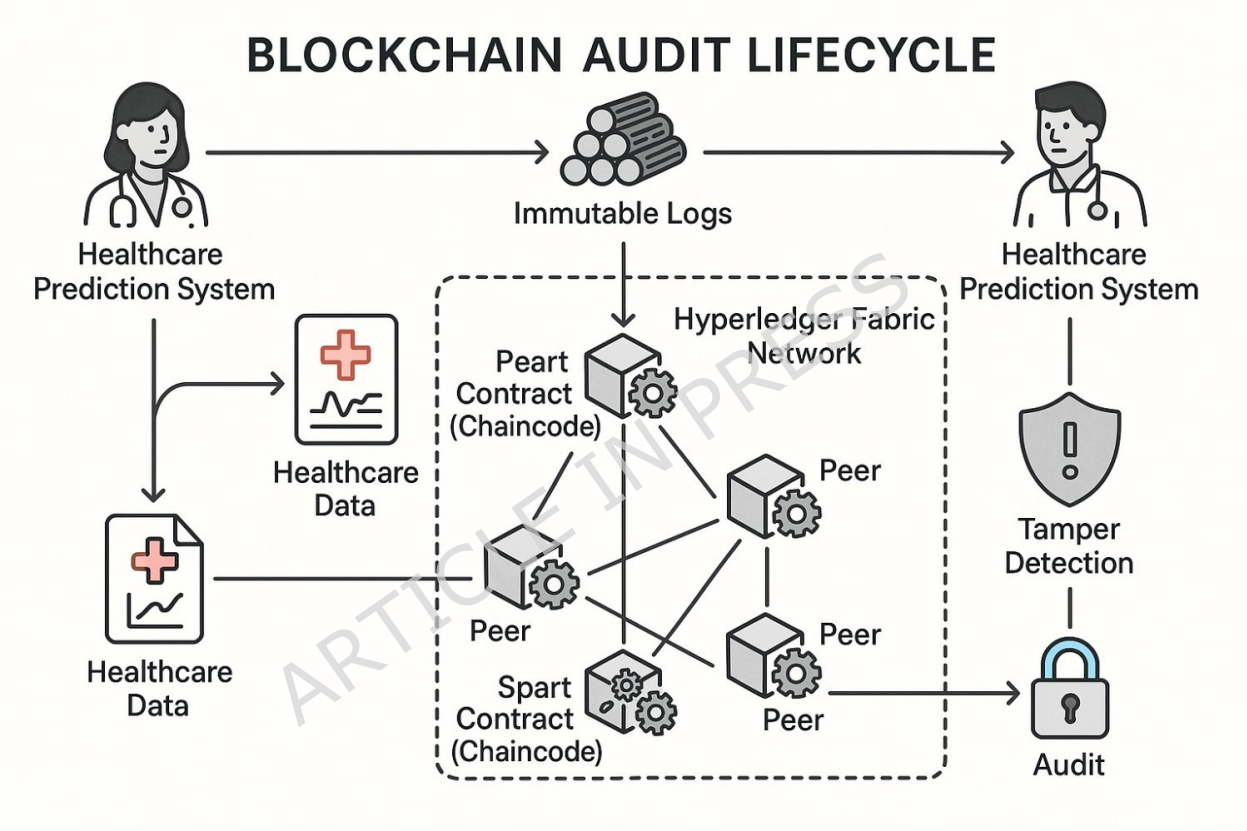

Figure 4. Blockchain audit lifecycle demonstrating how healthcare data transactions are recorded through Hyperledger Fabric smart contracts, ensuring immutable logs, tamper detection, and verifiable audits.

/

Figure 5. Clinical visualization dashboard displaying patient risk scores, real-time analytics, and disease prediction outputs to support clinician decision-making.

/

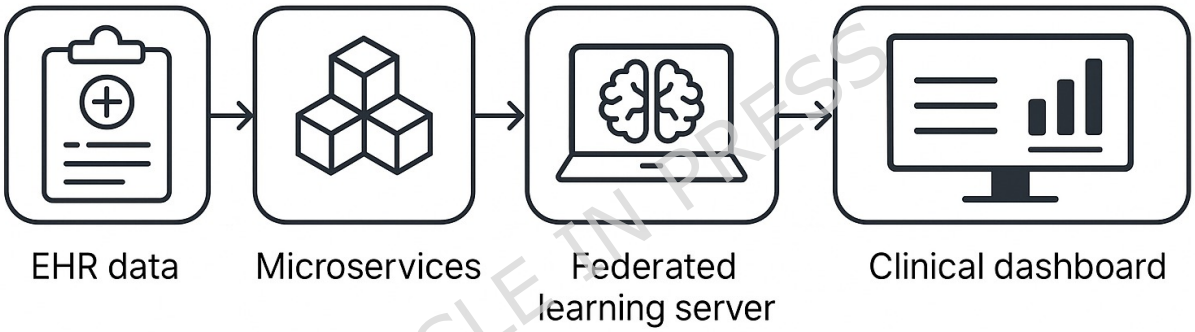

Figure 6. End-to-end system workflow connecting EHR data ingestion, microservices processing, federated learning server, and clinical dashboard for intelligent healthcare analytics.

/

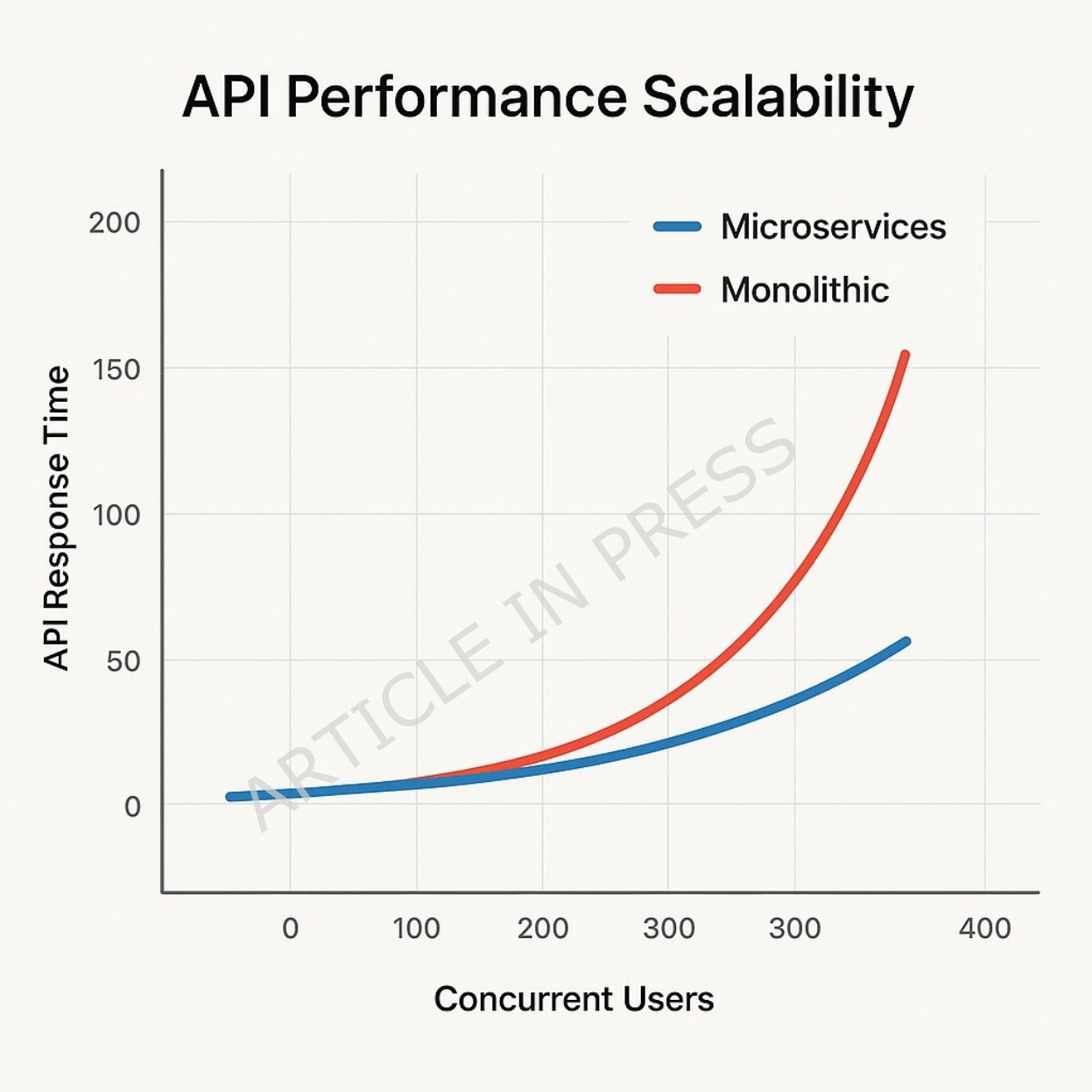

Figure 7. System performance metrics comparing microservices and monolithic architectures across API latency, uptime, failure recovery time, and simulated breach attempts, demonstrating superior resilience of the proposed model.

/

Table 3. Notation used in the manuscript

Symbol / Notation	Meaning / Description
$H = \{h_1, h_2, \ldots, h_K\}$	Set of participating hospital nodes (federated clients)
h_i	The i-th hospital or healthcare institution
K	Total number of participating hospital nodes
D_i	Local private dataset stored at hospital node h_i
$\|D_i\|$	Number of samples in the local dataset at node h_i
M_t	Global model parameters at federated round t
M_0	Initial global model before federated training begins
M_i	Local model parameters at hospital node h_i
ΔM_i^t	Privacy-preserving model update computed by hospital h_i at round t
ΔM^t	Aggregated global model update at round t
M_{t+1}	Updated global model after aggregation at round t
M^*	Final converged global model
t	Federated learning communication round index
ϵ	Convergence threshold for terminating federated training
E	Number of local training epochs per federated round
B	Mini-batch size used for local training
η	Learning rate for local model optimization
$\ell(\cdot)$	Loss function used for model training
$\nabla \ell(M_i; x)$	Gradient of the loss with respect to model parameters for sample x
C	Gradient clipping bound for enforcing differential privacy
σ	Noise multiplier controlling the strength of differential privacy
g^{dp}	Differentially private gradient after noise injection
$\mathcal{N}(0, \sigma^2 C^2 I)$	Gaussian noise distribution used in DP-SGD
$\mathcal{M}_i^t$	Metadata associated with client update ΔM_i^t (e.g., sample size, DP parameters)
$\mathcal{U}^t$	Set of verified and accepted client updates at federated round t
$N = \sum_{i \in H} \|D_i\|$	Total number of samples across all participating clients
$(\cdot),\ (\cdot)$	Encryption and decryption functions for secure communication
$\mathcal{B}$	Permissioned blockchain ledger (Hyperledger Fabric)
$\mathcal{C}_{sc}$	Smart contract (chaincode) for access control and audit logging
h_t	Cryptographic hash of the global model at federated round t
$\mathcal{P}^t$	Blockchain audit payload generated at round t
$\mathcal{A}^{(t)}$	Immutable audit record generated at federated round t
$\mathcal{A}_{blk}$	Complete blockchain audit trail across all federated rounds

/

Table 4. Baseline Comparison: Monolithic, Centralized, Local, and Proposed Framework

Aspect	Monolithic Systems	Centralized ML	Local Models	Proposed Federated Microservices
Architecture	Tightly coupled single system	Centralized predictive model on pooled data	Independent models at each hospital	Modular microservices with federated learning and blockchain
Data Handling	Single shared database; raw data transfer required	Raw patient data centralized in one repository	No data sharing; only local usage	No raw data sharing; secure model updates aggregated
Scalability	Limited; entire system must scale together	Moderate; single global model under heavy load	Low; siloed models lack global adaptability	High; independent microservices with Kubernetes orchestration
Privacy and Compliance	Weak; single point of failure and breach risk	High privacy risk due to centralization	Strong privacy but limited collaboration	Strong privacy with federated learning + blockchain audit (HIPAA/GDPR compliant)
Predictive Accuracy	Moderate due to outdated integration	89.7% (centralized global model)	85.3% (local-only models)	95.2% (federated aggregation across nodes)
System Performance	High latency, long recovery time, downtime common	Improved latency over monolithic, but bottlenecks persist	Variable latency; limited load handling	Low latency (42% faster), high throughput, 10× faster recovery
Operational Resilience	Single point of failure; downtime during redeployment	Moderate; relies on central infrastructure	Independent but fragmented	Resilient; autoscaling, fault isolation, CI/CD pipelines

Table 5. Quantitative security and privacy evaluation under defined threat model. Each attack type is evaluated against predefined success thresholds. An attack is classified as unsuccessful if it fails to meet the threshold in the evaluated settings.

Attack Type	Evaluation Metric	Success Threshold	Observed Result (Our System)	Outcome
Gradient Inversion	NMSE (lower is better for attacker)	< 0.15	0.32	Unsuccessful
Membership Inference	Accuracy & AUC	> 0.65 for both	Accuracy: 0.53, AUC: 0.54	Unsuccessful
Model Update Tampering	Bypass rate of integrity + audit checks	≥ 1 accepted tampered update	0/500 tampered updates accepted	Unsuccessful
Blockchain Tamper Detection	Detection rate of malicious transactions	< 0.99	Detection rate: 1.00 (100%)	Unsuccessful (attacker perspective)

Table 6. Model Predictive Performance Comparison

Metric	Federated	Centralized	Local
Accuracy (%)	95.2	89.7	85.3
Precision	0.96	0.87	0.84
Recall	0.93	0.88	0.79
F1-Score	0.944	0.872	0.811
AUC-ROC	0.978	0.891	0.832

Table 7. Scalability analysis under increasing number of federated nodes

Node Count	Round Time (s)	Monthly Bandwidth	Bottleneck Identified	Mitigation Strategy
5 nodes (tested)	42	8.2 GB	None	Baseline configuration
20 nodes	168	32.8 GB	Aggregation server CPU	Vertical scaling
50 nodes	420	82 GB	Network I/O limits	Gradient compression
100 nodes	840	164 GB	Coordinator memory	Hierarchical aggregation
200 nodes	1680	328 GB	Synchronization delays	Asynchronous updates

Table 8. Blockchain transaction performance under increasing workload

TPS	Latency	Success Rate	CPU Usage	Storage / Day	Performance
50	280 ms	99.9%	35%	4.3 GB	Optimal
200	420 ms	99.7%	68%	17.2 GB	Good
500	820 ms	98.2%	92%	43.0 GB	Acceptable
1,000	2.1 s	94.5%	100%	86.0 GB	Degraded

/

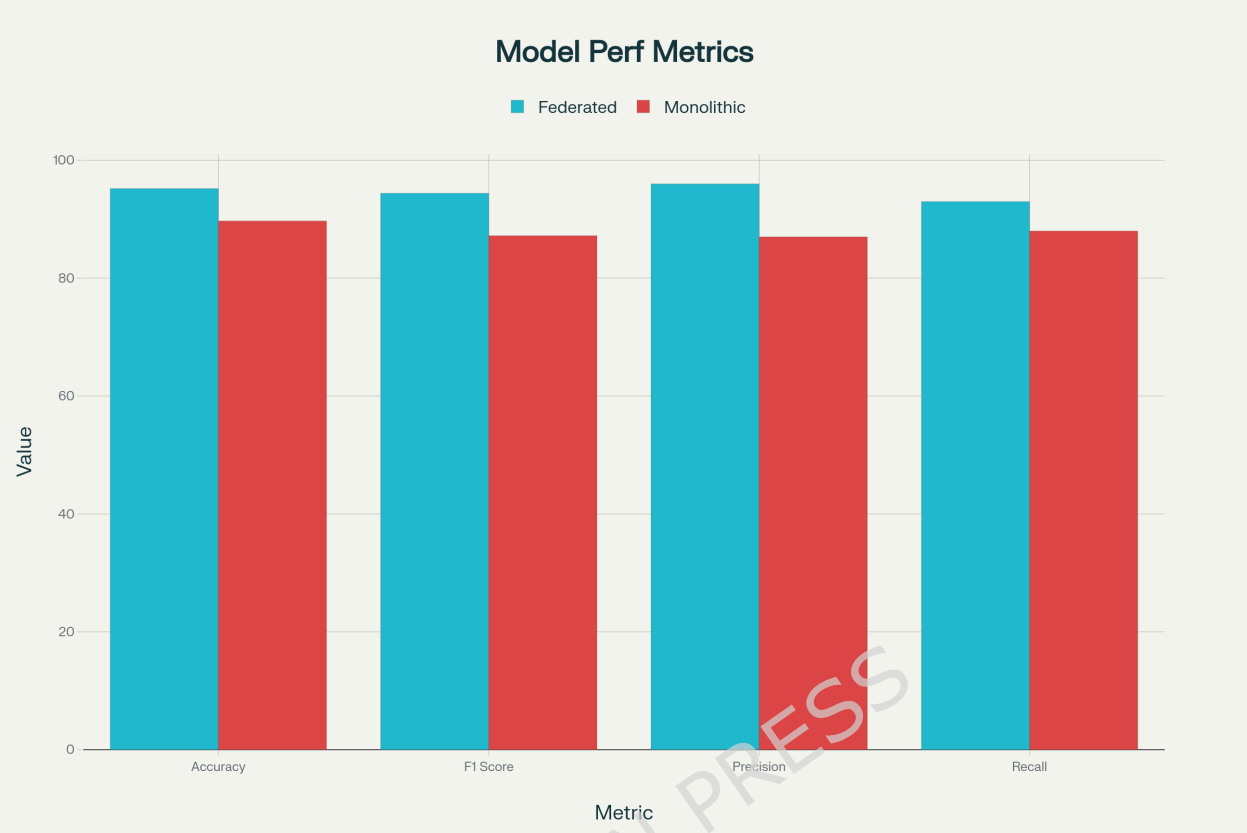

Figure 8. Model performance metrics comparing federated and monolithic systems in terms of accuracy, F1-score, precision, and recall, highlighting improved learning consistency under federated learning.

Table 9. Microservices scalability under increasing concurrent user load

Concurrent Users	Active Pods	Response Time	Resource Cost	Limiting Factor
1,000	8	1.8 s	$0.42/hour	None
5,000	24	2.1 s	$1.26/hour	Load balancer
10,000	48	2.4 s	$2.52/hour	Service mesh overhead
20,000	96	3.8 s	$5.04/hour	Database connections
50,000	240	8.2 s	$12.60/hour	Persistent storage IOPS

Table 10. Deployment configurations and cost estimates for different network scales

Network Size	Architecture	Infrastructure	Cost Estimate	Key Considerations
Small (≤20 hospitals)	Single-tier federation	8-node Kubernetes cluster; 3-node Hyperledger Fabric; 16 vCPUs, 64 GB RAM	$2,100/month	Simple and cost-effective; direct coordinator communication
Medium (20–100 hospitals)	Two-tier hierarchical	16-node Kubernetes cluster; multi-channel Fabric; database read replicas	$5,800/month	Regional aggregation; transaction sharding required
Large (100+ hospitals)	Three-tier hybrid	Multi-cluster federation; hybrid blockchain; edge pre-processing	$14,200/month	Hierarchical federated learning; off-chain storage; geographic distribution

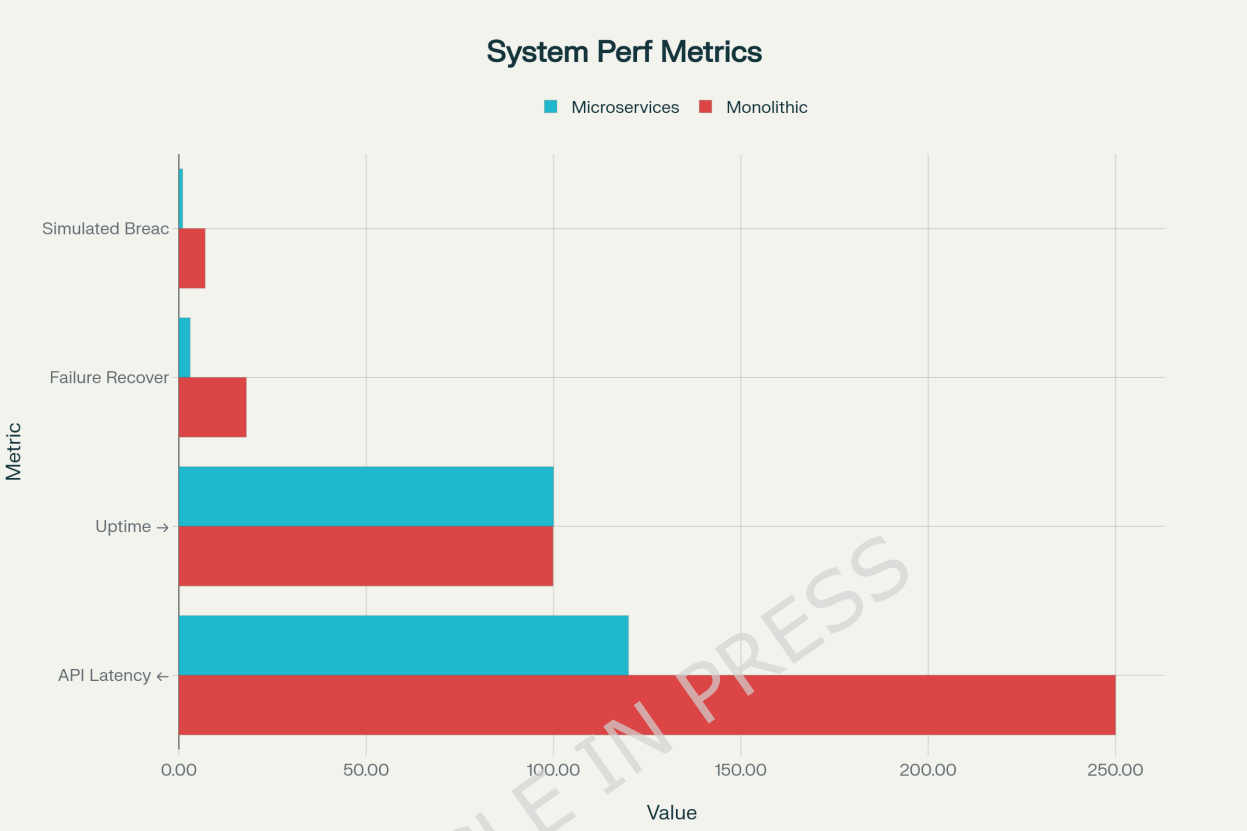

Figure 9. API performance scalability graph showing reduced response time and better handling of concurrent user loads in the microservices-based system relative to the monolithic baseline.

Table 11. System Performance Comparison

Parameter	Microservices	Monolithic
API Latency (ms)	1800	3100
Throughput (req/s)	950	400
System Uptime (%)	99.8	96.7
Failure Recovery (s)	45	600

/

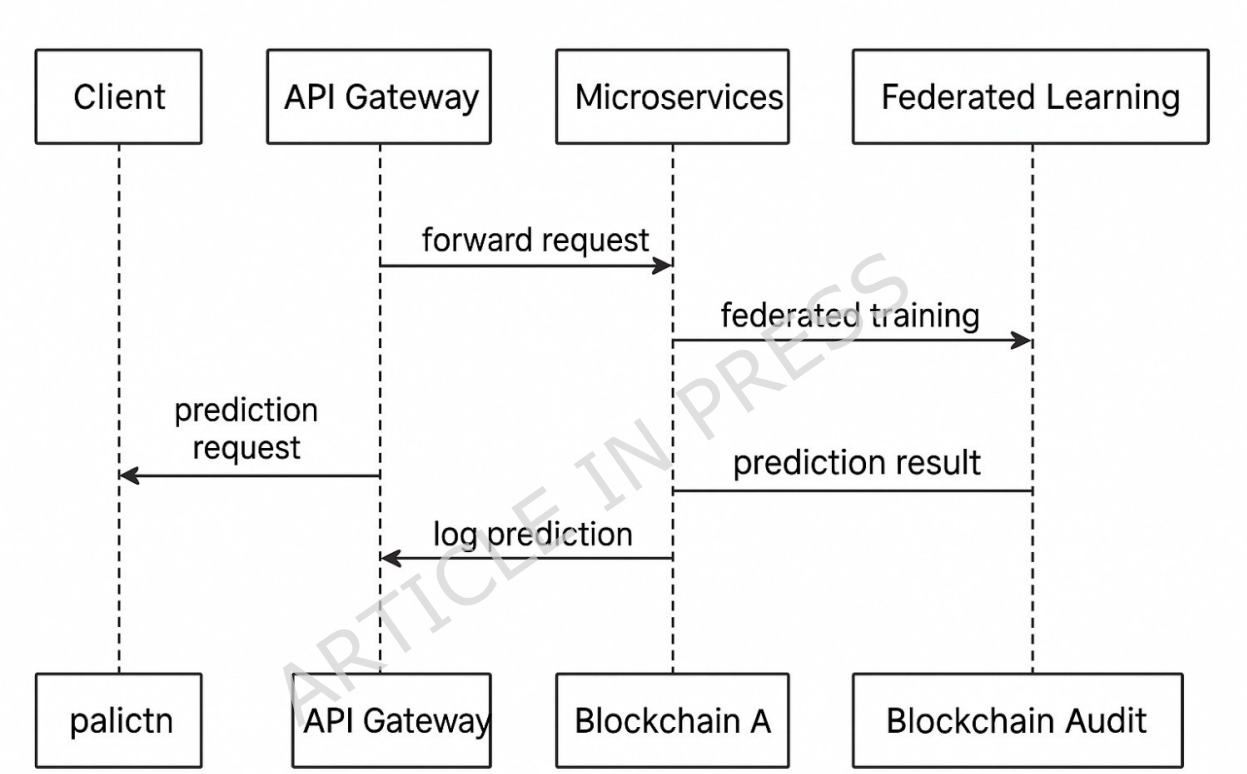

Figure 10. Sequence diagram illustrating the communication flow among client, API gateway, microservices, federated learning module, and blockchain audit for secure prediction logging and validation.

/

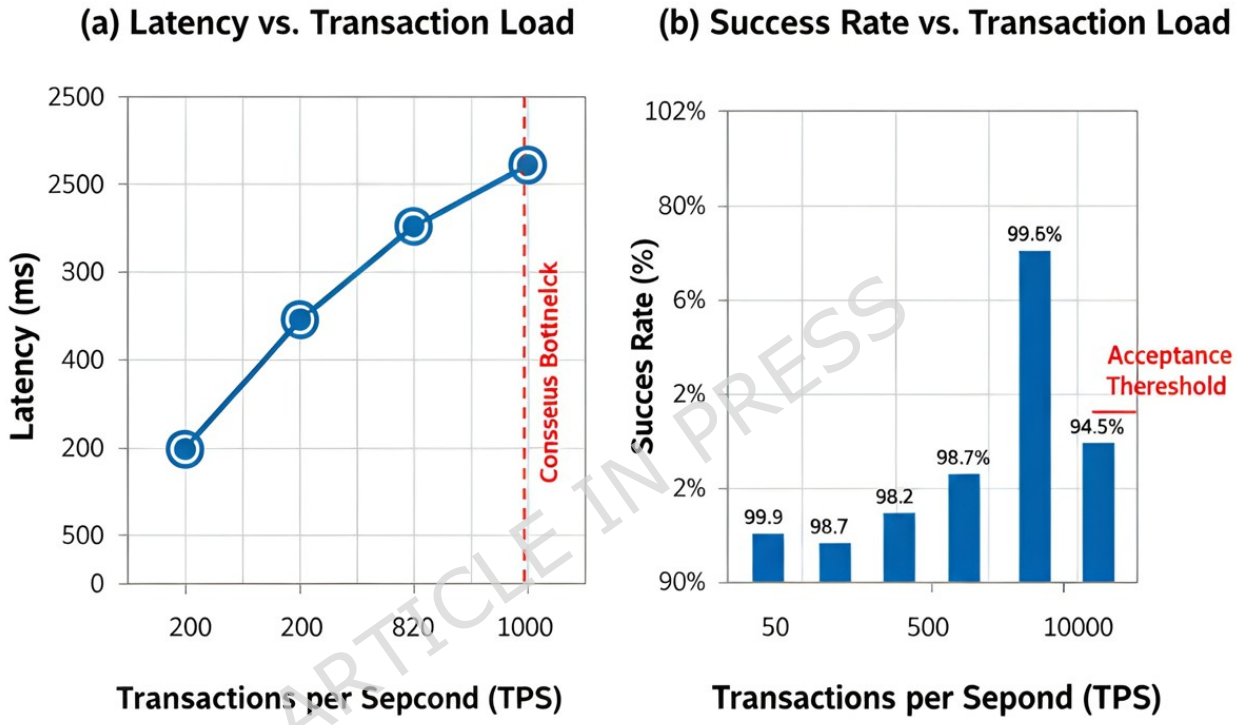

Figure 11. Blockchain scalability analysis under increasing transaction load. (a) Latency versus transaction load, showing a near-linear increase in end-to-end transaction latency as transactions per second (TPS) increase, with a clear consensus bottleneck emerging around 1,000 TPS. (b) Success rate versus transaction load, illustrating consistently high transaction success rates at low to moderate loads, followed by a noticeable degradation beyond the acceptable operating threshold at high TPS. Together, the results highlight the practical performance limits of the Hyperledger Fabric–based audit layer and motivate the need for optimization strategies at large-scale deployments.

/

Table 12. Comparative Analysis with Existing Studies

Study (Year)	Architecture / Focus	FL	DP	Blockchain / Audit	Dataset / Domain	Key Result
Becker et al. (2024)[24]	Microservices for radiology workflow automation	–	–	–	Radiology operations	Processing time reduced by 38%; improved modular scalability
annappa et al. (2024)[25]	Hybrid personalized FL for IoMT analytics	✓	–	–	IoMT healthcare data	Accuracy improved by +3.4% compared to FedAvg; robust personalized training.
muneekaew et al. (2022)[26]	Federated learning with differential privacy for medical imaging	✓	✓	–	Medical image datasets (MRI, CT)	Validated FL + DP integration with minimal accuracy tradeoff.
Warnat Herresthal et al. (2021)[27]	Swarm learning for multi-institutional biomedical data	✓	–	–	Multi-site omics datasets	Demonstrated confidential decentralised ML without data pooling.
Dayan et al. (2021)[28]	Federated learning for COVID-19 clinical outcome prediction	✓	–	–	20 hospitals (EHR data)	Achieved AUC > 0.92; validated large-scale global FL collaboration.
Knowledge-Abstraction FL (2024)[29]	Federated learning over heterogeneous multi-view medical data	✓	–	–	Multi-view EHR + IoT datasets	Enhanced robustness to view heterogeneity.
FedWeight (2025)[30]	Covariate shift mitigation in federated learning models	✓	–	–	Cross-domain health datasets	Reduced performance degradation under domain drift.
Our Work (2025)	**Federated microservices framework with blockchain audit**	✓	✓	✓	**100k synthetic + 20k real healthcare records**	**Accuracy 95.2%; Latency 1.8s; Uptime 99.8%; Zero breach simulations; HIPAA/GDPR compliant**

/