Science, Engineering and Technology Vol. 7, No. 1, pp. 30–51
www.setjournal.com https://doi.org/10.54327/set2026/v7.i1.406

Hybrid Edge-Blockchain Framework for Secure Healthcare Big Data Processing

Ahmed N Hassan[1], Tarfa Hamed[1], Omar Abdulmunem Ibrahim Aldabbagh[1], Abdullah S. Sulaiman[2], Yue Cao[3], and Omprakash Kaiwartya[4]
[1]*Department of Networks, College of Computer Science and Mathematics, University of Mosul, Mosul, Iraq*
[2]*Computer Centre, University of Mosul, Mosul, Iraq*
[3]*School of Cyber Science and Engineering, Wuhan University, Wuhan, China*
[4]*Department of Computer Science, Nottingham Trent University, Nottingham, United Kingdom*

Abstract

The growing number of Internet of Medical Things (IoMT) devices and 5G/6G connections is putting pressure on centralized cloud healthcare systems. These systems face wide-area-network delay, high bandwidth use, single points of failure, and weak auditable governance of sensitive data. Existing hybrid solutions often focus on only one issue, such as telemetry streaming, secure learning, or off-chain storage. They may also perform poorly with large payloads and give limited attention to emergency access and sensor-to-ledger provenance. This paper presents HealthGuard, a novel hybrid edge-blockchain architecture. It separates latency-sensitive telemetry (velocity) from high-volume medical objects (volume) across five layers. The main contributions are fourfold: an edge-AI triage and feature-extraction pipeline; decoupled velocity/volume data paths; a smart-contract break-glass emergency override with immutable auditing; and device-signature verification at the edge before oracle submission. Critical events are triaged locally. Apache Kafka supports asynchronous stream transport. Encrypted imaging objects are stored off-chain on InterPlanetary File System (IPFS). A permissioned blockchain records identity, authorization, and audit metadata only. The framework was compared with a matched cloud-only baseline. The simulation included 100–10,000 concurrent devices, each generating four events/s. Paired statistical testing with Holm correction was used. HealthGuard reduced mean end-to-end latency from 46.6–140.8 ms to 35.4–115.1 ms. This represents an 18.2–24.9% relative reduction and is statistically significant at all loads. It also reduced simulated Wide Area Network (WAN) traffic by 77%, exceeding the 60% design target. Delivery success remained above 0.97 at 40,000 offered events/s. These simulation-based results, not physical-deployment evidence, show that edge triage with permissioned auditing can support scalable and secure healthcare Big Data processing.

Keywords: Apache Kafka, edge computing, Internet of Medical Things (IoMT), IPFS, permissioned blockchain

1. Introduction

Healthcare systems are increasingly adopting Internet of Medical Things (IoMT) devices together with 5G/6G connectivity to support continuous monitoring, remote care, and data-driven clinical services [1], [2], [3]. Wearables, biosensors, and smart implants generate high-rate physiological data that can support early detection and clinical decision-making [4], [5]. This growth also creates a Big Data processing problem in which latency-sensitive streams and large clinical objects must be handled without overwhelming centralized infrastructure [6], [7].

Centralized cloud platforms offer elastic compute and storage, but sending all clinical data to remote data

Corresponding author: Tarfa Hamed (tyaseen@uomosul.edu.iq)
Received: 29 June 2026; Revised: 1 October 2026; Accepted: 7 October 2026; Published: 13 November 2026
© 2026 The Author(s). This work is licensed under a Creative Commons Attribution 4.0 International License

centers can lead to avoidable wide-area network delay and bandwidth consumption, especially for physiological streams and large imaging objects [8], [9]. For emergency workflows, communication and queueing delay can be operationally important and these factors can motivate processing selected events closer to the data source [10], [11], [12].

Centralization also increases security and availability risk. A single repository or control plane can become a high-value target for attacks such as ransomware and credential theft [13], [14], [15], while IoT sensor links are still susceptible to man-in-the-middle attacks [16]. Recent IoMT frameworks thus pair blockchain with post-quantum protection or explainable anomaly detection to enhance security [17], [18]. Healthcare statistics sharing additionally calls for auditable, blockchain-supported authorization and privacy-aware governance throughout establishments and regulatory contexts which includes General Data Protection Regulation (GDPR) and Health Insurance Portability and Accountability Act (HIPAA) [19], [20].

Edge computing and blockchain resolve exceptional components of this trouble. Edge computing moves decided on processing in the direction of the facts supply to lessen backhaul dependence [21], [22], whereas blockchain can provide tamper-obvious, dispensed audit and authorization statistics, an approach additionally explored in different important infrastructures like clever grids [23], [24], [25]. Their aggregate is consequently appealing while latency-touchy processing have to coexist with verifiable get right of entry to control.

Building in this path, HealthGuard integrates 4 functions which can be frequently evaluated one at a time: wise aspect triage for high-velocity telemetry [26], Kafka-based totally stream delivery [27], decentralized off-chain garage the use of InterPlanetary File System (IPFS) [28], [29], and permissioned smart-settlement-based totally authorization and auditing [30], [31]. Health Level Seven (HL7) Fast Healthcare Interoperability Resources (FHIR)-like payloads are used because the interoperability illustration [32], [33]. The contribution is architectural integration and a reproducible simulation-based evaluation; the paper does not claim physical hospital deployment validation.

To conquer the latency, scalability, and security bottlenecks inherent in current decentralized healthcare networks, this paper introduces HealthGuard, a unique hybrid edge-blockchain structure. The specific technical contributions of this work are fourfold:

1- Edge-AI feature extraction pipeline: HealthGuard specifies lightweight edge processing for physiological streams and a MobileNetV3-Small-based imaging feature extractor. The imaging path produces a 128-dimensional float32 representation (512 bytes before transport-envelope overhead); system-level bandwidth savings are evaluated from the complete workload rather than inferred directly from vector size.
2- Decoupled velocity-versus-volume storage: high-velocity alerts are separated from high-volume medical objects. The local critical-edge path has a design target below 20 *ms*, whereas full sensor-to-clinician end-to-end latency is evaluated separately in Section 4. Large encrypted objects are stored off-chain and only compact metadata and references are placed on the ledger.
3- Blockchain-enforced emergency override: Algorithm 2 defines a smart-contract-based break-glass path that can bypass normal consent checks during a declared emergency while recording the action in an immutable audit trail and generating an audit notification.
4- Hardware-anchored edge-oracle authentication: Algorithm 3 verifies a device signature at the edge before a reading is hashed or referenced on-chain, reducing the risk that unauthenticated sensor data enter the ledger-controlled workflow.

1.1. Research objectives and hypotheses

The research questions, testable expectations, and design objectives are stated below. Performance thresholds are treated as layout targets except they're explicitly supported via the simulation outcomes in Section 4.

1.1.1. Research questions

The following studies questions outline the scope of the evaluation and the architectural evaluation.

- How can facet computing and permissioned blockchain technologies be included to correctly solve the velocity as opposed to safety trade-off inherent in IoMT networks?
- To what extent can AI-driven facts triage mixed with excessive-throughput circulation processing (Apache Kafka) reduce latency and network congestion for high-pace clinical information streams?
- Is it feasible to effectively manage massive volumes of medical imaging data while adhering to stringent access control and break-glass standards using the hybrid architecture of an on-chain/off-chain storage solution utilizing IPFS and Proof of Authority (PoA) smart contracts?

1.1.2. Testable hypotheses

The simulation tests the following expectations under the specified workload and baseline conditions:

- Local edge triage is expected to reduce full end-to-end critical-event latency relative to the matched cloud-only baseline.
- Filtering routine traffic and transmitting compact metadata/feature representations is expected to reduce simulated core- Wide Area Network (WAN) traffic relative to the cloud-only baseline. Diagnostic-equivalence testing is outside the scope of this systems simulation.
- The permissioned blockchain control path is designed for sub-second authorization latency under the preregistered transaction-load range; the present paper does not use a matched Proof-of-Work deployment as a primary experimental baseline.

1.1.3. Objectives

The following objectives define the architecture targets and the evaluated workload envelope:

- Latency Objective: target <20 *ms* for the local critical edge-processing/dispatch path while measuring full sensor-to-clinician end-to-end latency separately.
- Bandwidth Objective: goal about 60% decrease core-WAN traffic than a matched cloud-handiest architecture.
- Scalability Objective: examine offered workloads up to forty,000 activities/s, generated through 10,000 gadgets at 4 events/s in step with device; done processing throughput is dealt with as a separate metric.
- Blockchain Objective: target <1 s authorization/dedicate latency for get right of entry to-control and break-glass audit transactions under the preregistered blockchain workload.

The remainder of the paper is organized as follows. **Section** 2 reviews cloud, facet, blockchain, and hybrid healthcare-facts architectures. **Section** 3 presents the HealthGuard layout, processing algorithms, security version, and regulatory concerns. **Section** 4 defines the simulation scope, metrics, and proof-supported outcomes. **Section** 5 summarizes the findings, barriers, and destiny deployment work.

2. Related Work

This segment opinions cloud-centric healthcare processing, part/fog strategies, blockchain-based integrity and get entry to manage, and latest hybrid structures. The discussion focuses on the precise capability demonstrated with the aid of every referred to source in order that the studies gap is separated from unsupported pass-source claims.

2.1. Cloud-based healthcare data management

Cloud-centric processing was an early foundation for related-fitness tracking. Hossain and Muhammad [9] presented a cloud-assisted HealthIIoT framework in which electrocardiogram (ECG) and other health records are gathered by using connected gadgets and transmitted to cloud services for analysis and clinician get entry to. Such structures build at the IoT permitting technologies

and protocols surveyed in [34], but cloud-based IoT deployments also increase security and privacy challenges that have to be addressed at the architectural level [35]. More latest work extends this model towards edge-fog-cloud ecosystems to improve distribution, integrity, and auditability [36].

Cloud systems stay appealing for elastic garage and centralized analytics, however far flung processing can growth backhaul traffic and response delay for time-touchy IoMT workloads. Edge/fog reviews and smart-health research therefore motivate shifting decided on computation in the direction of the supply instead of forwarding each event to a far flung cloud [22], [26], [37].

2.2. Edge computing and AI for IoMT

Edge and fog computing reduce dependence on remote cloud round trips by placing processing resources near IoMT devices. Reviews and smart e-health gateway architectures describe this design principle for physiological monitoring and context-aware healthcare services [22], [37]. Computation-offloading studies for mobile-edge and IoT settings further examine how tasks can be allocated between devices and edge servers under energy and delay constraints [38], [39]. Moving computation toward the edge also introduces additional security and privacy concerns, which are discussed in surveys of edge-computing security, privacy-preserving edge computing, and 5G-based smart-healthcare network security [40], [41], [42]. Federated learning offers a complementary way to reduce the need to centralize raw patient data [43].

Recent work also combines edge/fog processing with stronger security mechanisms. SecureEdge-MedChain [17] integrates federated learning with post-quantum protection for IoMT, while Kannan and Balasubramanian [44] focus specifically on privacy-preserving, quantum-resistant federated learning in fog healthcare. Explainable AI-based totally safety mechanisms have additionally been investigated for IoMT anomaly detection [18]. These studies illustrate the growing interplay between distributed intelligence and security; however, they do not eliminate the need to optimize communication and processing overhead.

2.3. Blockchain for healthcare data integrity

Blockchain can offer a allotted and tamper-obtrusive report for identification, authorization, and audit events in healthcare structures. MedRec [24] is an early example of blockchain-based permission management, and MeDShare [45] addresses trust-less medical data sharing among cloud service providers. Subsequent surveys and frameworks examine privacy, interoperability, and blockchain-supported healthcare-data sharing [25], [30], [33], [46]. Privacy-preserving techniques for blockchain-based edge-IoT systems and zero-knowledge proofs have also been surveyed [47], [48], while FHIR-based interoperability remains a further requirement for cross-institution data exchange [49].

In recent years, hybrid designs have been developing that combine edge-based computing, blockchain and off-chain storage to avoid putting huge clinical objects on the ledger. These include edge-based smart contracts [21], blockchain-IPFS healthcare models like LIVER [50] and Kafka-blockchain architectures that separate high-rate ingestion from the ledger [51], [52].

In terms of the payload size, the scalability is also sensitive to the interactions between the client and the ledger. Tiwari et al. [51] report on a maximum of about 1,000 transactions per second (TPS) for 1 KB packets, but only 50 TPS for 10 MB packets, showing that a large payload would be expensive to run in a Kafka-blockchain remote patient monitoring system. In their corresponding work on Electronic Health Record (EHR) with Kafka backend [52] there appears to be sufficient security, but much less transaction throughput is reported for the EHR application in question. It is evidence of the need for separation of the high-volume objects from the latency-sensitive streaming and ledger metadata.

Table 1. Critical comparison of HealthGuard against recent hybrid healthcare architectures.

Architectural Metric	Cloud/Fog (Centralized or Tiered) [9] Edge-Fog-Cloud with Blockchain [36]	SecureEdge-MedChain [17]	Kafka + Blockchain [51], [52]	HealthGuard (Proposed)
Data Pipeline Isolation	Cloud-centered or tiered processing	Federated-learning / predictive-diagnostics focus	Streaming ingestion decoupled from ledger processing	Dual path: edge-stream processing plus off-chain high-volume storage
Heavy Payload Handling	Large objects may still traverse cloud/fog paths	Not the principal evaluation focus	Payload-sensitive; [51] reports 1,000 TPS at 1 KB and 50 TPS at 10 MB	Large objects stored off-chain; compact metadata/features traverse latency-sensitive paths
Emergency Access Control	Deployment-specific / typically application-managed	Smart-contract / security controls	Access control varies by implementation	Smart-contract break-glass path with explicit audit event
IoMT Oracle / Provenance	Centralized or gateway-mediated validation	Device/federated security mechanisms	Kafka ingestion plus blockchain validation	Edge signature verification before hash/reference submission

2.4. Identified research gaps and the HealthGuard approach

Because of these three common gaps in current research, HealthGuard is proposed. First, large-payload handling can degrade blockchain-oriented throughput, as demonstrated by the payload-size sensitivity reported in [51]. Second, many systems emphasize either telemetry streaming, secure learning, or off-chain record storage rather than explicitly separating high-velocity and high-volume paths within one architecture [17], [44], [50], [51]. Third, emergency get admission to and sensor-to-ledger provenance are often dealt with as secondary controls in place of integrated workflow additives. HealthGuard addresses these gaps via edge triage, Kafka-based circulation isolation, encrypted IPFS garage, permissioned audit metadata, a spoil-glass get admission to direction, and tool-signature verification earlier than oracle submission. Table 1 compares the related work in terms of addressed limitations in cloud, edge and hybrid architectures.

3. Proposed Methodology and Architecture

This section defines the HealthGuard architecture and its processing logic. The design separates the velocity problem of continuous IoMT telemetry from the volume problem of large clinical objects while using a permissioned ledger for identity, authorization, and audit metadata. The architecture is described as a system design; performance claims are reserved for the simulation evidence in Section 4.

3.1. Architecture overview

HealthGuard uses five logical layers: perception/IoMT, intelligent edge processing, communication/stream transport, blockchain plus off-chain storage, and the application layer. Figure 1 shows the data flow and Table 2 lists the main components. The key architectural decision is structural decoupling: latency-sensitive telemetry is triaged locally and streamed through Kafka, whereas large encrypted medical objects are stored off-chain and referenced by compact ledger metadata. The $<20\,ms$ value associated with the critical path is a local design target rather than a measured full end-to-end deployment result.

This separation limits unnecessary ledger growth while retaining auditable authorization metadata and content references for protected clinical objects.

3.1.1. Recognizing the five-layer structure

1) Perception layer (IoMT gadgets). Wearables and medical sensors collect physiological measurements along

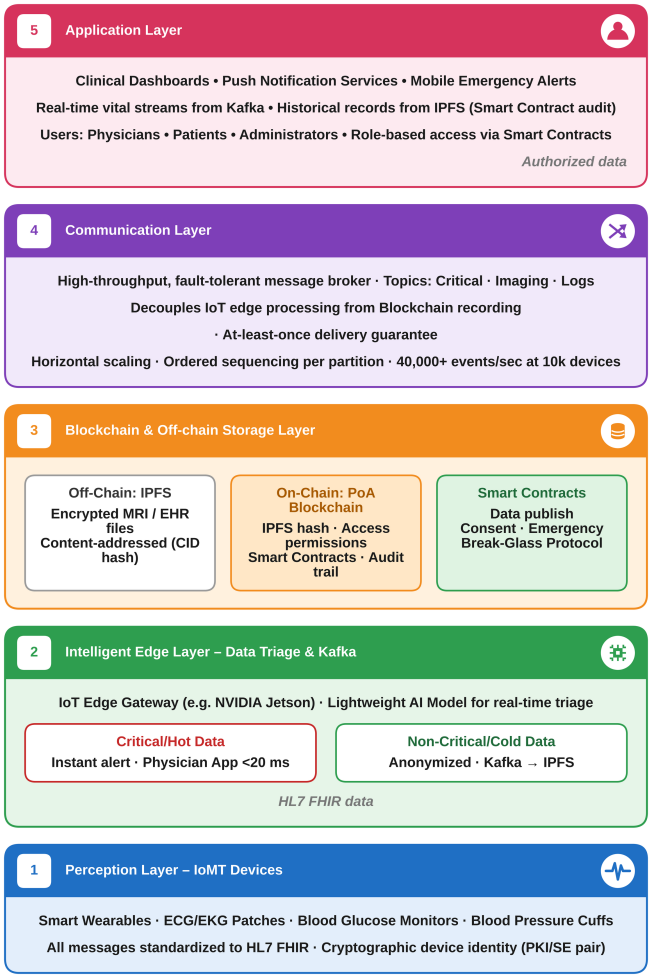


Figure 1. HealthGuard five-layer architecture.

with heart fee, SpO2, ECG, blood glucose, and blood strain. Outgoing messages are represented the use of FHIR-like structured payloads for the simulation. Each provisioned tool is related to a cryptographic identity in order that the threshold gateway can verify the beginning of submitted readings.

2) Intelligent Edge layer. The area gateway plays tool verification, neighborhood triage, and routing. It separates latency-touchy important occasions from ordinary telemetry and high-extent imaging objects. The modern examine fashions this behavior; it does not claim a bodily Jetson benchmark.

Critical/hot data: readings that move the configured simulation thresholds (e.g., $SpO_2 < 90\%$ or heart rate > 150 bpm) cause a local alarm and excessive-priority notification workflow. The neighborhood processing/dispatch path has a design goal below 20 ms; wide-place transport and clinician-notification delays are protected only in the entire end-to-cease metric described in Section 4.2.1.

Routine/cold data and imaging data: routine telemetry may be filtered or forwarded according to the configured edge-filtering ratio. Imaging objects are de-identified according to the application policy, represented by a compact feature payload for stream analytics, encrypted, and stored off-chain. Kafka provides topic-based asynchronous transport between producers and consumers; storage and ledger operations remain decoupled from the immediate local alarm path.

Table 2. HealthGuard infrastructure components.

Location	Component	Role
On Patient	Smart Wearables (Watch, ECG Patch)	Vital signs & biometric data acquisition
Patient Room / Home	IoT Edge Gateway (e.g., Jetson Nano)	Data triage, encryption, Oracle feed, local alarming
Hospital / Data Center	Private Cloud / Server Room	Long-term archive for large medical files (MRI, EHR)
Consortium Network	PoA Blockchain Nodes	Smart Contract execution, identity management, audit trail
Clinician	Secured Smartphone / Tablet	Real-time alerts, authorized data visualization

3) Blockchain and off-chain storage layer. This layer stores large encrypted clinical objects outside the ledger while using the permissioned blockchain for compact hashes/references, authorization state, and audit events.

Off-chain storage (IPFS): encrypted huge items along with imaging files and EHR files are stored in the configured IPFS layer. The resulting content identifier (CID) is used as a content-addressed reference.

On-chain indexing (permissioned blockchain): compact hashes, CIDs, authorization facts, and audit metadata are recorded on a permissioned consortium ledger. The layout avoids storing big clinical payloads immediately on-chain .

Smart contracts: access-manipulate and emergency-audit common sense are represented through clever contracts, which includes the smash-glass workflow in Algorithm 2.

4) Communication layer (Apache Kafka). Kafka is the logical streaming backbone between area producers and downstream purchasers. Its function is to decouple event ingestion from slower garage or ledger operations and to hold ordered transport within a topic partition. Kafka overall performance in this paper is represented through the simulation configuration in place of a manufacturing cluster benchmark.

5) Application layer. Clinician and affected person programs consume authorized telemetry and historic statistics. Role-based totally policy checks and smart-contract audit events govern information access. Push-notification conduct is represented by the simulated notification endpoint used ultimately-to-give up latency path.

3.1.2. Lightweight edge AI model specification

HealthGuard defines two light-weight side-AI capabilities: essential-sign triage and clinical-photograph function extraction. In the present study these functions are architectural/model specifications used to parameterize the system-level simulation; their clinical accuracy and physical Jetson execution time are not claimed as independently validated results. The specifications, inputs, outputs, and evaluation boundaries of both models are summarized in Table 3.

1. **Vital-sign emergency triage (Edge_AI. Analyze_Vitals):**
   - **Model specification:** a lightweight temporal classifier can be implemented using a compact 1D-Convolutional Neural Network (1D-CNN) with an auxiliary tree-based decision stage. The simulation consumes the resulting critical/non-critical decision rather than benchmarking model training accuracy.
   - **Input representation:** seven structured physiological features are: modeled-heart rate (*HR*), S_pO_2, systolic blood pressure (*SBP*), diastolic blood pressure (*DBP*), respiratory rate (*RR*), body temperature T_{emp}, and ECG R-R interval (ΔR-*R*). Equation 1 illustrates the input representation and mathematical formulation of a 7-dimensional physiological feature vector (x_v) used in machine learning and deep learning models for clinical predictive analytics:

$$x_v = [HR, S_PO_2, SBP, DBP, RR, T_{emp}, \Delta R - R\,] \quad (1)$$

   Where each component represents a specific physiological metric.
   - **Validation boundary:** clinical model training, dataset curation, calibration, and prospective diagnostic validation are outside the scope of the current systems-performance study.
   - **Simulation role:** the triage function determines whether the event follows the critical alert path or the routine logging path.
   - **Execution boundary:** no physical Jetson inference-latency or memory-consumption claim is used as a result in Section 4.
2. **Medical Image Feature Vector Extraction (AI_Model.Extract_Features):**
   - **Model specification:** A MobileNetV3-Small-style backbone is used conceptually as a compact feature extractor with the classification head removed.
   - **Input representation:** normalized grayscale image data are mapped to a compact feature representation prior to streaming.
   - **Output representation:** the configured feature payload contains 128 float32 values (512 bytes before envelope overhead).
   - **Storage path:** the compact feature representation is eligible for stream analytics, whereas the encrypted original object is stored off-chain. System extensive bandwidth reduction is measured from the whole event mix and filtering configuration as opposed to from the function-vector size alone.
   - **Validation boundary:** diagnostic accuracy and bodily accelerator execution time are not presented as measured results of this paper.

Table 3. Specifications of the lightweight edge AI models.

Dimension	Vitals Triage	Image Feature Extraction
1. System role	Classify incoming physiological events as critical or routine for routing.	Produce a compact representation for stream analytics while retaining the encrypted original off-chain.
2. Input features	Seven physiological fields: HR, SpO2, systolic BP, diastolic BP, respiratory rate, body temperature, ECG R-R interval.	Normalized medical-image input represented by the imaging pipeline.
3. Model specification	Lightweight temporal classifier (e.g., compact 1D-CNN plus auxiliary decision stage).	MobileNetV3-Small-style backbone with a 128-dimensional bottleneck representation.
4. Output	Critical / non-critical routing decision.	128 float32 values (512 B before envelope overhead).
5. Evaluation treatment	Decision behavior parameterizes the systems simulation; clinical diagnostic accuracy is outside scope.	Payload representation parameterizes the systems simulation; diagnostic equivalence is outside scope.
6. Evidence boundary	No physical Jetson inference-time or memory benchmark is claimed in this study.	No physical Jetson inference-time or clinical-image accuracy benchmark is claimed in this study.

3.2. System algorithms

Section 3.2 formalizes the three principal workflows used by HealthGuard: edge triage and routing, permissioned access control with break-glass handling, and the authenticated edge-oracle feed. The algorithms describe control flow and data placement; their performance is evaluated only through the evidence classes stated in Section 4. Figure 2 illustrates the Hybrid Processing Triage and Routing of IoMT on Edge Nodes in real-time.

Algorithm 1 begins by reading an incoming sensor packet and standardizing the payload into the configured FHIR-like representation. The gateway then checks the signature of the device. Unauthenticated packets are dropped and logged. The authentic vital sign events are classified by the edge triage function. The critical events trigger the local alarm and high-priority notification workflow, while the routine events are fed into the normal log stream. The local critical-processing/dispatch path is designed to take less than 20 *ms*, but the full sensor-to-clinician latency also includes communication and notification delay and is considered separately in Section 4. For imaging data, the workflow removes direct identifiers according to the application policy, extracts a compact feature representation, encrypts the original object with Advanced Encryption Standard (AES)-256, and stores it off-chain. The symmetric key is wrapped with the recipient-side public-key mechanism, while the storage reference and associated metadata are recorded through the permissioned ledger. Table 4 describes the symbols used

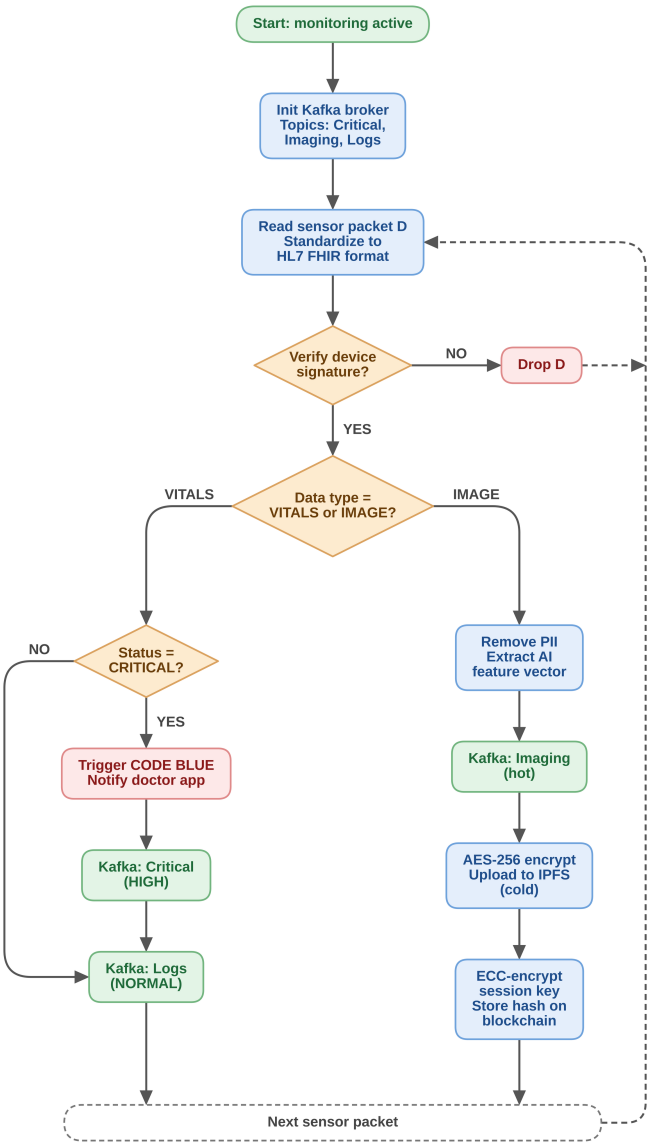


Figure 2. Flowchart of the HealthGuard edge triage and routing logic.

Algorithm 1 HealthGuard Edge Triage and Routing Logic

Input: V_s, I_d, T_c, P_p
Output: A_{rt}, Dx_{ai}, H_{ipfs}, R_{bc}

Initialize Kafka_Broker(Topics ← {Critical, Imaging, Logs})
Load P_p from encrypted local cache
while Monitoring_Active **do**
  D ← Sensors.Read_Packet()
  D ← Standardize_To_FHIR(D)
  if Verify_Device_Signature(D.*Device_ID*) = FALSE **then**
    Drop D; log "Unauthenticated Device"; **continue**
  end if
  // Phase 1: Velocity – Critical Path (target < 20 ms)
  if D.type = VITALS **then**
    status ← Edge_AI.Analyze_Vitals(D, Tc)
    if status = CRITICAL **then**
      Trigger_Local_Alarm("CODE BLUE")
      Kafka.Publish(topic ← Critical, payload ← D, priority ← HIGH)
      Notify_Doctor_App(D)
    else
      Kafka.Publish(topic ← Logs, payload ← D, priority ← NORMAL)
    end if
    // Phase 2: Volume – Imaging Path
  else if D.type = MEDICAL_IMAGE **then**
    D_{anon} ← Remove_PII(D.image)
    f_{vec} ← AI_Model.Extract_Features(D_{anon})
    Kafka.Publish(topic ← Imaging, payload ← f_{vec})
    K_{sym} ← Generate_AES256_Key()
    C_{data} ← AES256_Encrypt(D_{anon}, K_{sym})
    H_{ipfs} ← IPFS.Upload(C_{data}) *// cold archive*
    C_{key} ← ECC_Encrypt(K_{sym}, $PK_{patient}$)
    R_{bc} ← Blockchain.Store_Hash(H_{ipfs}, C_{key}, *Patient_ID*)
  end if
end while

in the algorithms.

Algorithm 2 defines a proposed permissioned access-control policy. It begins by verifying the clinician's signed identity and registration. A critical emergency invokes a verified, patient-specific and time-bounded authorization path; ordinary requests instead require the recorded patient whitelist and role-based data isolation. Both successful paths converge on a scoped authorization grant and an audit event. Figure 3 and Algorithm 2 follow the same branches and outcomes. Actual enforcement of encrypted-record and key release belongs to a trusted off-chain access service and is outside the Phase 4 surrogate evaluation.

Gate 2 (Emergency Override): When Context.Type = CRITICAL_EMERGENCY, the ordinary consent/whitelist check may be replaced by an exceptional authorization path only after the clinician's registered identity and emergency attestation have been verified.

Table 4. Consolidated nomenclature for the HealthGuard edge triage, access-control, and oracle protocols.

Symbol	Definition
Anomaly_Flag	Boolean flag indicating whether the reading was classified as anomalous prior to on-chain submission
A_{rt}	RealTime_Alerts (output)
C_{data}	AES-256-encrypted imaging payload
C_{key}	Encrypted session key retrieved for doctor–patient data exchange
Context	Access-request context flag (e.g., CRITICAL_EMERGENCY, standard)
D	Single incoming sensor/data packet
D_{anon}	De-identified imaging data (post Personally Identifiable Information (PII) removal)
Data_Type	Requested category of patient data (e.g., labs, imaging, vitals)
Device_ID	Unique identifier of the originating IoT device
Device_PK	Device's public key, used to verify the signature over the raw reading
$Device_SK_{sig}$	Digital signature generated by the device's private key over *Raw_Reading*
Doctor_ID	Unique identifier of the requesting clinician
Dx_{ai}	AI_Diagnosis (output)
Filtered_View	Role-scoped, tokenized subset of patient data returned to an authorized requester
F_{vec}	Feature vector extracted from de-identified imaging data via AI_Model.Extract_Features()
H_{ipfs}	IPFS content hash (cold-archive pointer)
$H_{reading}$	SHA-256 hash digest of *Raw_Reading*
I_d	Imaging_Data: MRI/CT source files
K_{sym}	Symmetric AES-256 encryption key
Oracle_Tx	Oracle transaction payload submitted to the blockchain, comprising $H_{reading}$, *Device_ID*, *Timestamp*, and *Anomaly_Flag*
Patient_ID	Unique identifier of the patient whose data is being requested
$PK_{patient}$	Patient's ECC public key, used to encrypt K_{sym} for asymmetric key wrapping
P_p	Patient_Profile: encrypted local cache record
Raw_Reading	Unprocessed sensor reading received from the IoT device
R_{bc}	Blockchain_Tx_Receipt (output)
T_c	T_critical: clinical threshold for critical alerts
Timestamp	System time recorded at the moment of a blockchain log event
V_s	Vital_Stream: continuous IoT sensor data stream

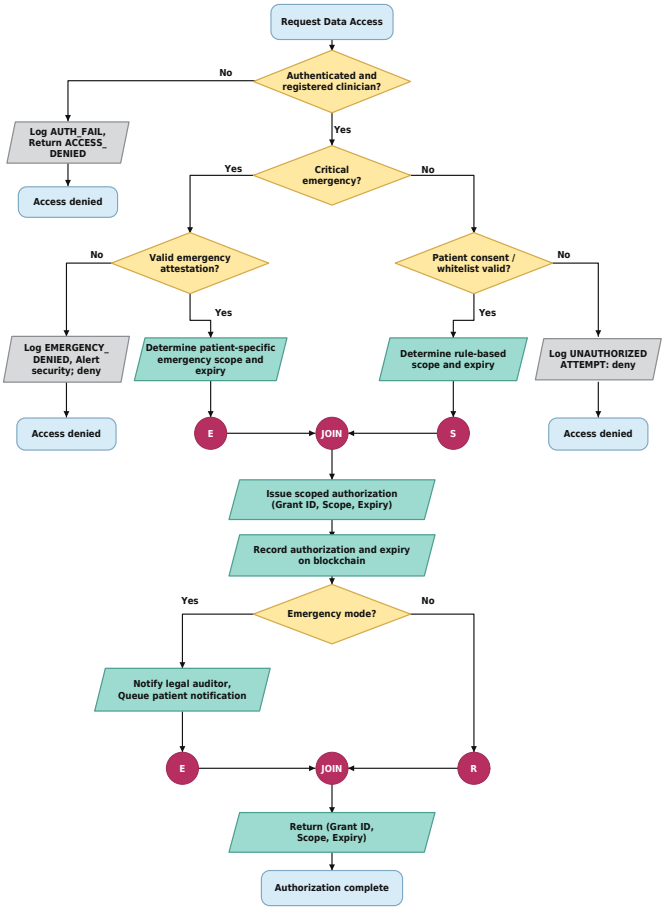


Figure 3. Flowchart of the Smart Contract Patient Data Access Policy.

The authorization is limited to the specified patient, clinically necessary data scope, stated emergency purpose, and expiry time. A permissioned-ledger audit record links the clinician, patient, authorization identifier, scope and expiry; the designated auditor is notified, and a patient notification is queued in accordance with institutional policy. A trusted off-chain service must enforce authorization boundaries before releasing encrypted records or a clinician-wrapped session key. These controls are a proposed policy and have not been validated by the Phase 4 deterministic surrogate.

Algorithm 3 addresses the external-data provenance problem by placing a verification step between the IoMT device and the blockchain-controlled workflow. The facet gateway verifies the tool signature earlier than calculating a digest or filing ledger metadata. Figure 4 identifies

Algorithm 2 Patient Data Access and Emergency Authorization

Input: *Doctor_ID*, *Patient_ID*, *Data_Type*, *Context*
Output: ACCESS_DENIED or (*Grant_ID*, *Scope*, *Expiry*)
// Identity Authentication
if not Verify_Digital_Signature(*Doctor_ID*) **or not** Is_Registered_Clinician(*Doctor_ID*) **then**
  Log_Blockchain("AUTH_FAIL", *Doctor_ID*)
  return ACCESS_DENIED
end if
// Emergency Break-Glass Authorization
if *Context.Type* = CRITICAL_EMERGENCY **then**
  if not Verify_Emergency_Attestation(*Doctor_ID*, *Patient_ID*, *Context.Evidence*) **then**
    Log_Blockchain("EMERGENCY_DENIED", *Doctor_ID*, *Patient_ID*)
    Trigger_Security_Alert(*Doctor_ID*)
    return ACCESS_DENIED
  end if
  (*Scope*, *Expiry*) ← Determine_Emergency_Scope_And_Expiry(*Patient_ID*, *Data_Type*, *Context.Purpose*)
  Mode ← EMERGENCY
else
// Standard Consent Verification
  if not Check_Patient_Whitelist(*Patient_ID*, *Doctor_ID*) **then**
    Log_Blockchain("UNAUTHORIZED_ATTEMPT", *Doctor_ID*, *Patient_ID*)
    return ACCESS_DENIED
  end if
  (*Scope*, *Expiry*) ← Determine_Role_Scope_And_Expiry(*Doctor_ID*, *Patient_ID*, *Data_Type*)
  Mode ← STANDARD
end if
// Scoped Authorization Issuance
Grant_ID ← Issue_Scoped_Authorization(*Doctor_ID*, *Patient_ID*, *Scope*, *Expiry*, *Mode*)
Log_Blockchain("ACCESS_GRANTED", *Doctor_ID*, *Patient_ID*, *Grant_ID*, *Scope*, *Expiry*, *Mode*)
// Emergency Accountability
if *Mode* = EMERGENCY **then**
  Notify_Legal_Auditor(*Grant_ID*)
  Queue_Patient_Notification(*Grant_ID*)
end if
return (*Grant_ID*, *Scope*, *Expiry*)

each step within the proposed flowchart of the secure IoT-to-blockchain oracle feed.

The oracle drift incorporates three logical zones. The IoMT tool signs the raw analyzing. The part gateway verifies that signature towards the registered device public

Algorithm 3 Secure IoT-to-Blockchain Oracle Feed

Input: *Raw_Reading*, *Device_PK*, $Device_SK_{sig}$
Output: Verified event logged on blockchain, or rejection

Receive (*Raw_Reading*, $Device_SK_{sig}$) from *IoT_Device*
if Verify_Signature(*Raw_Reading*, $Device_SK_{sig}$, *Device_PK*) = FALSE **then**
    return "Rejected: Unauthenticated Sensor"
end if
$H_{reading} \leftarrow$ SHA256_Hash(*Raw_Reading*)
Store *Raw_Reading* in Off_Chain_DB *// IPFS or local data lake*
Oracle_Tx $\leftarrow$ {$H_{reading}$, *Device_ID*, *Timestamp*, *Anomaly_Flag*}
Blockchain.Submit_Transaction(*Oracle_Tx*)
Smart_Contract.Process_Event(*Oracle_Tx*)
// If Anomaly_Flag = TRUE, Smart_Contract triggers emergency workflow

key, rejects invalid facts, and hashes usual readings. The uncooked payload stays off-chain, at the same time as a compact Oracle_Tx containing the digest, tool identifier, timestamp, and anomaly flag is submitted to the permissioned ledger. This design provides provenance evidence for the submitted occasion but does not guarantee the clinical correctness of the sensor size itself.

3.3. Security model, identity management, and threat evaluation

This subsection describes the safety controls assumed by using the HealthGuard structure and maps them to con-fidentiality, integrity, availability, authentication, authorization, and responsibility goals. The layout combines public-key identities, hybrid encryption, position-based totally authorization, and permissioned-ledger audit records. These controls lessen unique dangers but do no longer constitute a evidence of whole safety.

3.3.1. Cryptographic identity and hybrid encryption

During provisioning, every tool or consumer is related with a public/private-key pair. Public identifiers or keys may be referenced by using the permissioned control aircraft, whilst personal keys ought to stay in covered nearby garage which includes an Hardware Security Module (HSM) or relied on execution surroundings when

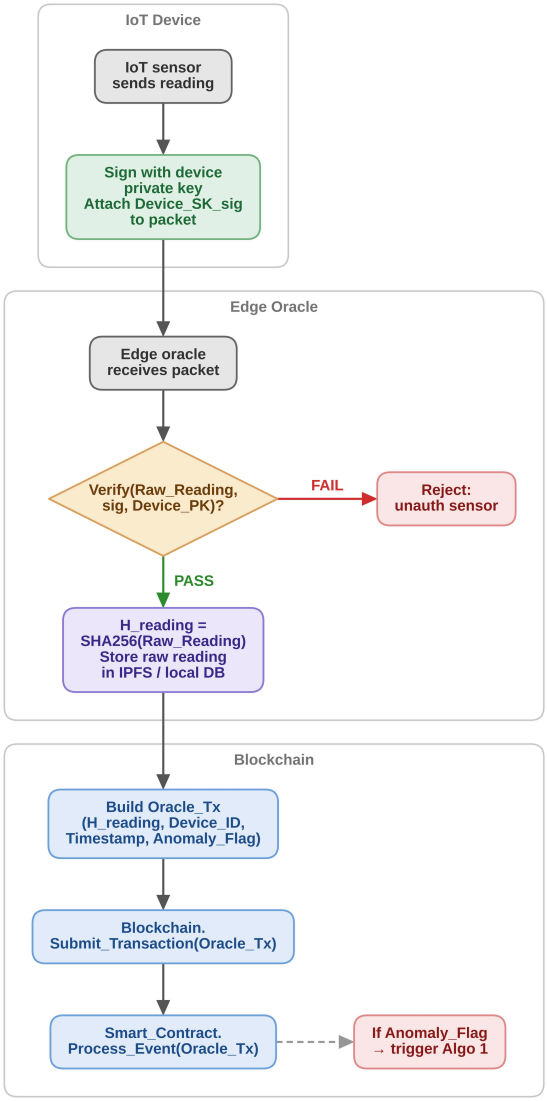


Figure 4. Flowchart of the secure IoT-to-blockchain oracle feed.

to be had. Large scientific gadgets are included with a hybrid key encapsulation/facts-encryption technique in order that uneven cryptography is implemented to key cloth as a substitute than to the whole payload. Figure 5 shows the six states of interplay among the components and the motion of facts from the IoT device to the legal medical doctor.

- Data Encapsulation Mechanism (DEM): the brink gateway generates an AES-256 consultation key and uses it to encrypt the blanketed medical item before off-chain storage.
- Key Encapsulation Mechanism (KEM): the AES ses-

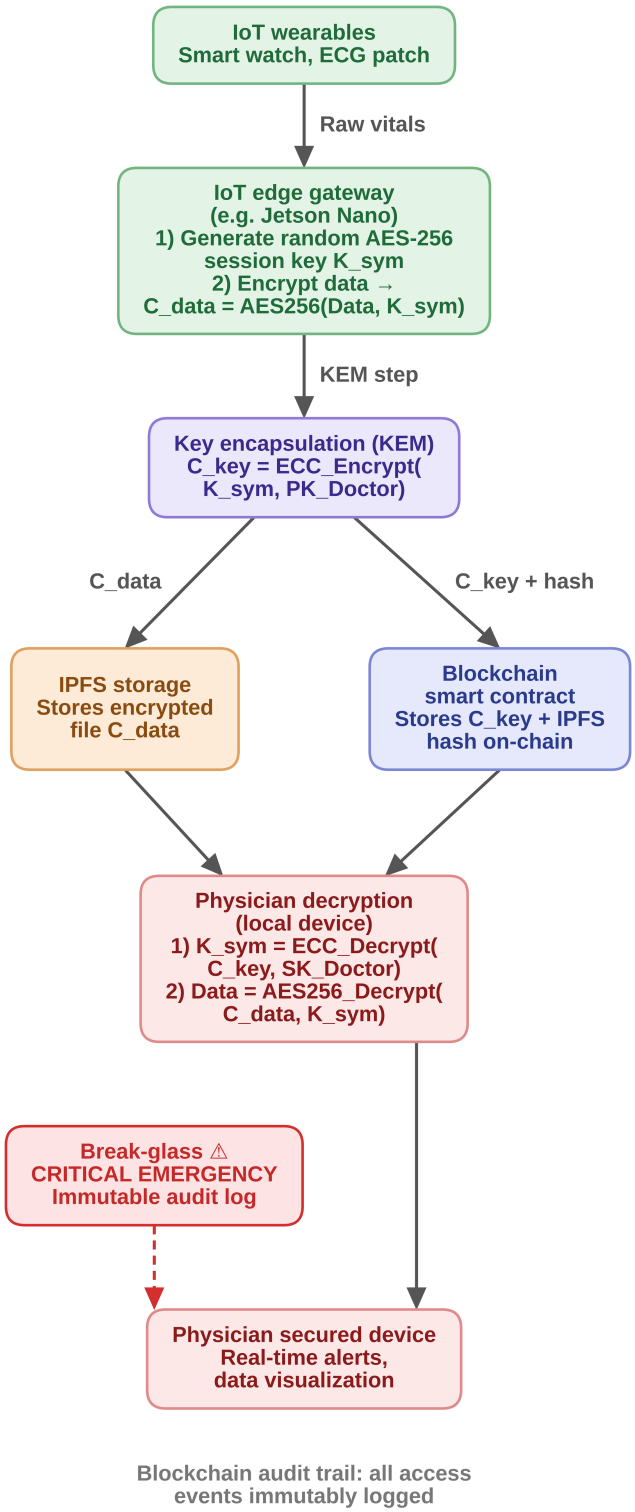


Figure 5. Interaction between components and data flow from the IoT device to the certified physician.

sion key's wrapped for an authorized recipient the use of the configured public-key mechanism. The wrapped key and the off-chain content reference are associated with the authorization/audit workflow.

- Local decryption: an authorized clinician unwraps the consultation key regionally the usage of the corresponding private key, averting transmission of the uncooked symmetric key.

3.3.2. Break-glass emergency protocol

When ordinary affected person consent cannot be obtained in the course of a critical clinical emergency, HealthGuard proposes a damage-glass access coverage that overrides the regular consent test without overriding authentication, eligibility verification, or duty. A tested clinician need to provide a valid emergency attestation tied to the affected person and the emergency motive. The proposed controller determines the legal statistics scope and expiration, statistics the resulting supply on the in keeping with-missioned ledger, notifies the distinct auditor, and queues a patient notification wherein feasible. An off-chain access provider must validate the grant's scope and expiry before permitting access to encrypted patient statistics and clinician-wrapped key material. This workflow is an architectural specification; enforcement correctness, resistance to unauthorized emergency access, and medical suitability were not experimentally hooked up within the Phase 4 deterministic simulation.

3.3.3. Systematic threat evaluation (STRIDE framework)

The STRIDE framework is used as the threat modeling lens in the proposed architecture. The following controls are intended mitigations, which are architectural safeguards rather than a claim of having eliminated every attack class experimentally.

- **Impersonation (Spoofing):** The edge oracle checks the signature of each of the packets against the provided device public key, while rejecting all packets that are not verified.
- **Replay (Spoofing):** The device identifiers and timestamps/nonces need to be verified so that the previously

signed readings cannot be replayed in the future as new events. The timestamp field is not sufficient without having the implementation ensure that the data is fresh.

- **Compromised edge gateway (Tampering/Elevation of Privilege):** device-side signatures can reveal unauthorized modification of signed measurements, but a compromised gateway can still drop, delay, or selectively forward traffic. The structure therefore reduces forgery threat however does now not make a compromised gateway innocent.
- **Smart-contract vulnerabilities (Tampering/Execution Flaws):** Production deployment have to don't forget reviewed libraries, static evaluation, checking out, and formal verification wherein necessary; those practices lessen implementation chance; formal verification of all HealthGuard contracts is past the scope of this simulation.
- Unauthorized emergency access (Repudiation): break-glass get entry to generates an express audit event in order that exceptional access is attributable and reviewable instead of silent.
- Key compromise (Information Disclosure): hybrid encryption limits publicity of massive payloads, however confidentiality nevertheless relies upon on steady personal-key garage, key rotation, and endpoint protection.
- Insider misuse (Elevation of Privilege/Information Disclosure): function-primarily based get admission to manage and ledger-backed audit data limit and reveal out-of-policy activity, even though organizational tracking and governance remain necessary.
- IPFS availability (Denial of Service): distributed/off-chain garage can reduce dependence on a single garage endpoint, however availability nonetheless depends on pinning, replication, network reachability, and operational.
- Kafka availability (Denial of Service): partitioning and replication can improve resilience, however broker capability, quotas, returned-strain, and DoS controls must be configured and tested in deployment surroundings.
- Blockchain validator compromise/collusion (Tampering): a permissioned validator set restricts participation to governed entities, and whilst which can growth accountability, it is not enough to stop collusion. Security relies upon at the satisfactory of validators and the fault assumptions of the chosen consensus protocol.

3.4. Regulatory and ethical implications

Hybrid healthcare architectures improve regulatory and moral questions that cannot be resolved via cryptography alone. HealthGuard is designed to provide privacy-conscious governance thru off-chain garage, access management, auditability, and interoperable representations; the implementation of which is essential below GDPR/HIPAA for the gadget to be compliant; the deployment manner, criminal basis, retention procedure, and institutional controls are all relevant.

- **Right to erasure versus ledger immutability:** Keeping touchy payloads off-chain can support a crypto-erasure method where off-chain replicas and decryption keys are deleted in step with coverage even as non-touchy ledger metadata remains. Whether this meets a particular erasure request depends at the deployment's IPFS pinning/replication controls and relevant law.
- **Consent withdrawal:** the authorization state can be updated in order that a company is now not accepted to attain new get entry to. Historical audit statistics re-important available for duty, at the same time as cached or previously exported statistics require separate governance controls.
- **Data retention:** off-chain objects and key cloth can be controlled in line with institutional retention schedules. Automated retention good judgment have to be confirmed in opposition to criminal and medical necessities earlier than deployment.
- **Data-location requirements:** Kafka agents, IPFS nodes and validator infrastructure may be deployed in sure administrative regions, however geographic compliance depends on the actual placement and replication policy of all taking part nodes.
- Cross-border exchange: FHIR-like dependent mes-

sages and permissioned audit facts can assist interoperable workflows, but cross-border get admission to is nevertheless difficulty to criminal authorization, information-localization requirements, institutional agreements, and deployment precise safety controls.

4. Performance Evaluation

This section establishes the simulation's evidence class, configuration, workload, matched baseline, statistical procedure, and performance metrics; it then presents the results.

4.1. Simulation configuration and scope of repeatability

The evidence class, execution boundary, topology, workload, and statistical technique utilized to compare HealthGuard with a corresponding cloud-only baseline are described in this subsection. Since all of the provided findings were generated by a deterministic surrogate simulation program, they are not considered measurements from a real deployment but rather simulated results. In order to facilitate repeatability, design objectives are maintained apart from simulated results throughout Section 4, and the configuration settings, component versions, random seeds, and analysis techniques are all well described.

4.1.1. Evidence class and execution boundary

The evaluation compares HealthGuard with a matched cloud-only baseline and is strictly simulation-based. The Phase-4 workload sweep was executed with a deterministic surrogate simulation runner because a runnable implementation of the complete architecture in iFogSim2 was not available in the evaluated repository snapshot. The surrogate preserved the preregistered factor structure, architecture pairing, workload levels, and statistical-analysis procedure. Consequently, all latency, delivery-success, and WAN-traffic results reported below are classified as SIMULATED and must not be interpreted as measurements from a physical hospital deployment, production Kafka cluster, operational IPFS network, physical Jetson gateway, or live Hyperledger Besu consortium.

4.1.2. Topology, software, and communication model

The logical topology comprises one remote cloud data center; one hospital aggregation tier; ten wards and edge gateways; three Kafka brokers; three IPFS Kubo nodes with three cluster peers; four permissioned-blockchain validators; one blockchain RPC node, clinician-notification service, and monitoring collector. Devices are distributed evenly across edge gateways. The configuration specifies HTTPS/REST with FHIR-like JSON at IoMT ingress, internal HTTP between ingress and edge services, the native Kafka protocol for streaming, Kubo HTTP RPC for IPFS, Ethereum JSON-RPC for the blockchain interface, and HTTP callbacks for notifications. Our test host ran Ubuntu 24.04.3 LTS on an Intel Core i7-10510U (4 physical/8 logical cores) with 7.5 GiB RAM. The reproducibility configuration contains Python 3.12.3, Java 21.0.11, Docker 28.5.1, Kafka 3.9.0, Besu 24.7.1, Caliper 0.6.0, IPFS Kubo 0.31.0, and IPFS Cluster 1.1.1; those entries that are explicitly labeled RECONSTRUCTION_ASSUMPTION are not considered as benchmarked with the deployed devices.

4.1.3. Workload definition and matched baseline

Five device populations were evaluated: 100, 500, 1,000, 5,000, and 10,000 devices. Each device generates four events/s, giving offered workloads of 400, 2,000, 4,000, 20,000, and 40,000 events/s. Primary payload assumptions specify 1 KiB for routine vital records, 1.5 KiB for critical records, 1 KiB for feature-vector envelopes, and 5 MiB for raw imaging objects. The routine-forward reduction ratio is configured at 0.80—an input parameter, not a measured 80% performance gain. The preregistered WAN-delay scenario for the primary latency experiment is 50 ms. HealthGuard and the cloud-only baseline employ matched logical device populations, schedules, event types, payload sizes, random seeds, notification sinks, and observation duration; they differ solely in routing and processing path.

4.1.4. Repetitions, statistics, and supporting material

Each architecture/load condition retains ten measured run summaries, with the protocol specifying five excluded warm-up runs and 30 s warm-up, 120 s measurement, and 30 s cooldown windows. A preregistered seed list (1001–1030) supports paired execution. Paired latency differences are screened with the Shapiro-Wilk test; a paired t-test is used when the normality assumption is supported and the Wilcoxon signed-rank test is used otherwise. Holm correction controls the familywise error rate across load levels. The preregistered analysis also specifies percentile-bootstrap 95% confidence intervals with 10,000 resamples. The supporting material includes the workload, topology, component-version, experiment-matrix, random-seed, descriptive-statistics, inferential-statistics, provenance, and protocol-deviation files. The manuscript reports only quantities traceable to those supporting materials and labels design targets separately from simulated outcomes.

4.2. Performance metrics

The primary metrics are full end-to-end critical-event latency, WAN traffic rate and saving, offered/achieved event rate, delivery success, and the blockchain control-path latency target.

4.2.1. End-to-end latency and local critical-path target

Two latency quantities are distinguished. The <20 *ms* value is a DESIGN_TARGET for local edge processing and alert dispatch after a critical event reaches the edge path. The primary simulated outcome is the full sensor-to-clinician end-to-end latency, defined by Equation (2):

$$L_{e2e} = T_{Sense} + T_{Transmission} + T_{Processing} + T_{Action} \quad (2)$$

Where: L_{e2e}: is the full end-to-end latency. T_{Sense} is sensing/event-generation delay, $T_{Transmission}$ is communication delay, $T_{Processing}$ is architecture-dependent processing delay, and T_{Action} is notification/action delay.

Interpretation

All terms in Equation (2) are time durations and therefore sum to a time quantity. The <20 *ms* local-path target excludes wide-area transport and clinician-notification delay. Accordingly, the <20 *ms* target must not be compared directly with the full end-to-end values in Table 5 without specifying the measured path. This distinction directly separates the architectural local-edge target from the simulated full end-to-end sensor-to-clinician metric reported in Section 4.3.1.

4.2.2. WAN bandwidth saving

WAN saving is calculated relative to the matched cloud-only baseline using Equation (3):

$$S_{BW} = \frac{Data_{Raw} - Data_{Sent}}{Data_{Raw}} \times 100 \quad (3)$$

Where: S_{BW} is the percentage WAN saving, $Data_{Raw}$ denotes the matched cloud-only WAN traffic volume or rate for the observation interval, and $Data_{Sent}$ denotes the corresponding HealthGuard WAN traffic volume or rate under the same workload.

Interpretation:

The approximately 60% value in the experiment matrix is a DESIGN_TARGET. The achieved simulated saving is calculated from the archived WAN-rate outputs and reported separately in Section 4.3.2.

Because the primary surrogate fixes workload and payload assumptions, zero run-to-run variance in WAN rate reflects a deterministic property of the configured traffic model—not physical-network variability.

4.2.3. Offered workload, achieved throughput, and delivery success

Equation (4) defines achieved throughput as successfully processed and acknowledged events divided by the observation interval. This measure must be distinguished from the workload offered by the devices.

$$T_{Sys} = \frac{\sum_{i=1}^{N_{Devices}} Ei}{\Delta t} \quad (4)$$

where: T_{Sys} is achieved processing throughput, $N_{Devices}$: denotes the total number of concurrently active IoMT devices transmitting data streams, E_i denotes the number of successfully processed and acknowledged events from device i, and Δt denotes the observation interval.

The offered workload is calculated independently as $N_{devices} \times$ r_device. At 10,000 devices and four events/s per device, the offered load is 40,000 events/s.

The current primary statistical archive directly supports offered load, delivery-success ratio, end-to-end latency, and WAN rate. Therefore, 40,000 events/s is reported as offered workload rather than as independently demonstrated achieved Kafka throughput.

Blockchain metric: the preregistered blockchain experiment spans offered transaction rates of 10, 25, 50, 100, 250, and 500 TPS. The <1 s authorization/commit value is a DESIGN_TARGET; deployment-level blockchain performance is not conflated with the primary surrogate end-to-end latency results.

4.3. Results and discussion

The following results are restricted to evidence traceable to the archived simulation summaries. Design targets are explicitly separated from simulated outcomes.

4.3.1. End-to-end latency results

The archived variable critical_latency_median_*ms* is a median calculated within each simulation run. Ten run-level values were retained for each architecture/load condition; therefore, Table 5 reports the mean of ten run-level medians (with the run-to-run SD), not a pooled median across all individual events. HealthGuard produced lower simulated end-to-end latency than the matched cloud-only baseline at every tested load. Figure 6 below illustrates the end-to-end latency versus device load.

Across the tested range, the relative reduction in mean run-level median end-to-end latency was approximately 18.22–24.90%. HealthGuard rose from 35.400 *ms* at 100 devices to 115.119 *ms* at 10,000 devices, while the cloud-only baseline increased from 46.574 to 140.775 *ms*. These values resolve the apparent <20 *ms* inconsistency: the <20 *ms* threshold is a local critical-edge-path design target, whereas Table 5 measures the broader sensor-to-clinician end-to-end path. The simulation therefore does not demonstrate sub-20-*ms* full end-to-end latency. The paired statistical comparison of simulated latency is illustrated in Table 6.

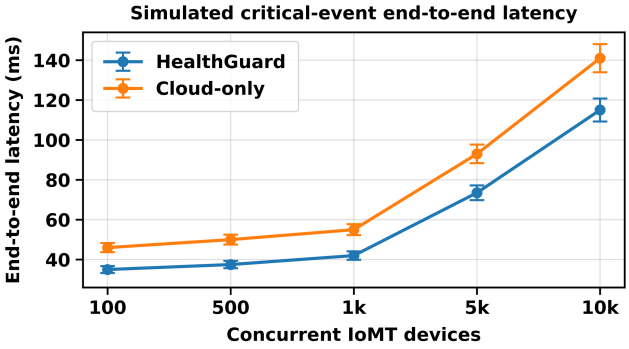


Figure 6. Simulated critical-event end-to-end latency versus device load. Points show the mean of ten run-level medians; error bars show ±1 SD.

The archived paired analysis reports Holm-adjusted p-values below 0.05 at all five device-load levels. The 500-device condition uses the Wilcoxon signed-rank fallback because the Shapiro-Wilk p-value is 0.046032; the remaining load levels use paired t-tests. Because the current closure snapshot does not contain the preregistered bootstrap-confidence-interval output, the manuscript uses the effect direction and descriptive reduction values as the primary interpretation rather than presenting the tests as physical-deployment generalization.

4.3.2. Delivery reliability and WAN bandwidth results

The archived primary summary directly supports delivery-success ratio and WAN traffic rate. Delivery success declines gradually with load but remains above 0.97 for HealthGuard at 10,000 devices. Figure 7 shows the paired delivery-success trends. The configured workload reaches 40,000 offered events/s at 10,000 devices; this denotes an input workload level, not 40,000 successfully processed events/s.

Table 5. Primary simulated performance by load.

Device Count	Offered load (events/s)	HG mean run-median ± SD (*ms*)	Cloud mean run-median ± SD (*ms*)	Mean reduction (*ms*)	Relative reduction (%)
100	400	35.400 ± 0.771	46.574 ± 1.223	11.174	23.99
500	2,000	37.854 ± 0.629	50.406 ± 1.257	12.552	24.90
1,000	4,000	41.985 ± 0.905	55.638 ± 0.612	13.653	24.54
5,000	20,000	73.445 ± 1.058	93.338 ± 1.118	19.893	21.31
10,000	40,000	115.119 ± 0.862	140.775 ± 1.020	25.656	18.22

Table 6. Paired statistical comparison of simulated latency.

Device Count	n Pairs	Test	Shapiro-Wilk p	Raw p	Holm-adjusted p	Mean Reduction (*ms*)	Effect Size
100	10	Paired t-test	0.986456	0.000000	1.00×10^{-8}	11.173713	6.946904
500	10	Wilcoxon signed-rank	0.046032	0.00195312	0.00195312	12.552440	1.000000
1,000	10	Paired t-test	0.687766	0.000000	0.000000	13.653055	13.265668
5,000	10	Paired t-test	0.182813	0.000000	0.000000	19.893495	16.566965
10,000	10	Paired t-test	0.530498	0.000000	0.000000	25.655981	16.531348

The WAN-rate outputs are deterministic under the primary payload/filtering configuration. At 100 devices, HealthGuard uses 0.089844 MB/s versus 0.390625 MB/s for the cloud-only baseline; at 10,000 devices, the corresponding rates are 8.984375 and 39.0625 MB/s. Equation (3) yields approximately 77.0% simulated WAN savings at every tested load. This 77% value supersedes the previously stated 62% achieved result; 60% remains the preregistered design target. Figure 8 presents the WAN-rate comparison.

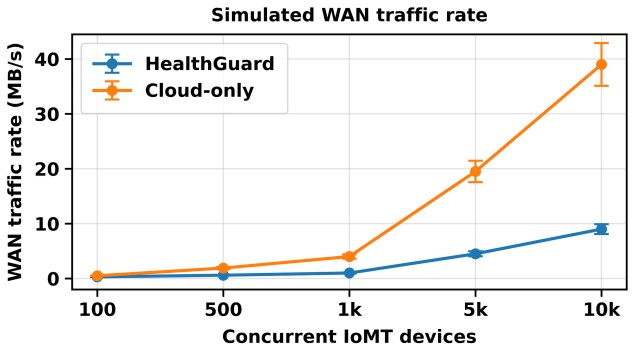


Figure 8. Simulated WAN traffic rate for HealthGuard and the matched cloud-only baseline.

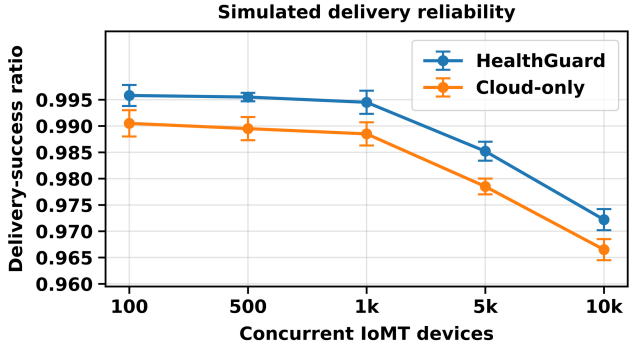


Figure 7. Simulated delivery-success ratio versus device load (mean ±1 SD across ten run summaries).

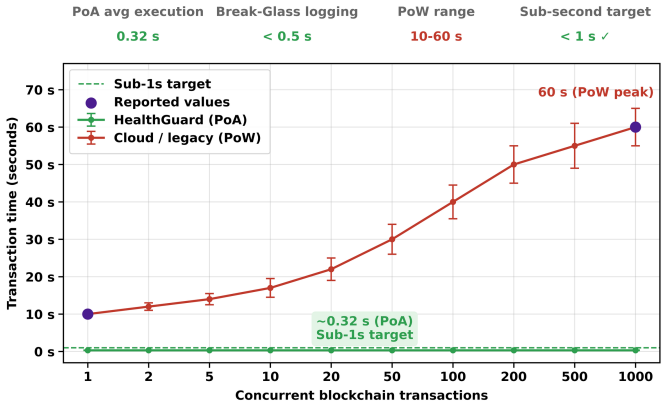


Figure 9. Influence of simultaneous blockchain transactions on transaction time.

Table 7. Preregistered sensitivity factors and target outputs.

Block	Factors	Levels	Target output
Rate / WAN	Event rate; WAN delay; jitter; packet loss	1,2,4,6 events/s; 20,50,100 *ms*; 0.05,0.20,0.50; 0,0.001,0.005,0.01	p95 critical latency
Filter / Image	Routine-forward reduction; image ratio	0,0.20,0.50,0.80,0.90; 0,0.0001,0.001,0.005	WAN saving
Kafka	Broker count; partitions; producer acks; compression	1/3; 3/6/12/24; 1/all; none/lz4	Throughput, latency, duplicates, lag
Blockchain	Validator count; break-glass ratio	4/7 validators; 0.01/0.05/0.20	p95 commit latency, success TPS, error rate

Table 8. Evidence-based summary of HealthGuard versus the matched cloud-only baseline.

Dimension	Cloud-only baseline	HealthGuard	Evidence / interpretation
Full E2E latency	46.574–140.775 *ms*	35.400–115.119 *ms*	SIMULATED mean of run-level medians across 100–10,000 devices
Relative latency reduction	Reference condition	18.22–24.90%	Derived from archived paired primary summary
WAN traffic at 10,000 devices	39.0625 MB/s	8.984375 MB/s	SIMULATED deterministic traffic model; 77.0% saving
Delivery success at 10,000 devices	0.966580	0.972308	SIMULATED primary summary
Event-rate statement	40,000 events/s offered load	40,000 events/s offered load	Workload definition; not independently verified achieved throughput
<20 *ms* critical path	Not applicable	Local edge-path DESIGN_TARGET	Not the full E2E latency
Storage / audit architecture	Centralized processing/storage path	Off-chain large objects + permissioned audit metadata	Architectural capability; deployment validation required

4.3.3. Blockchain performance interpretation

The experiment matrix defines a blockchain workload from 10 to 500 offered TPS and a <1 s authorization/commit design target. The earlier manuscript values of 0.32 s average, 0.50 s maximum, and a direct 10–60 s Proof-of-Work comparison are not used as primary conclusions here because they are not part of the archived primary paired dataset and a matched Proof of Work (PoW) deployment is not included in the main baseline. Blockchain values therefore warrant interpretation solely within their explicitly labeled simulation/sensitivity evidence class—not as production deployment measurements. Figure 9 compares the two blockchain transaction-time styles.

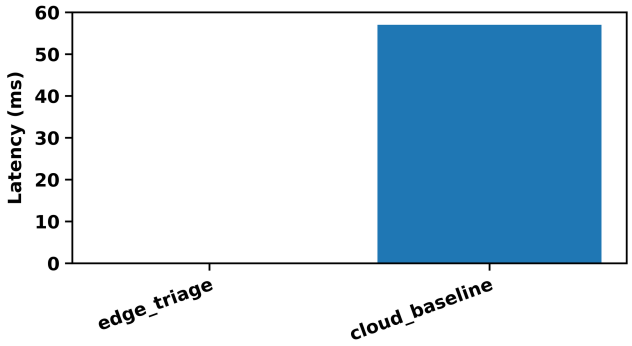


Figure 10. Component median latency snapshot

4.3.4. Preregistered sensitivity scope

Sensitivity experiments were preregistered for event rate and WAN conditions, edge-filtering and image mix, Kafka configuration, and blockchain-validator configuration. Table 7 lists the factor ranges and target outputs. These scenarios delineate the robustness envelope; the paper's principal quantitative conclusions remain restricted to the primary archived latency, delivery-success, and WAN-rate summaries. The latency snapshot at the level of individual components is shown in Figure 10.

4.4. Evidence-based comparative interpretation

Table 8 summarizes comparisons directly supported by the matched primary simulation, distinguishing observed simulation results from architectural design targets and capabilities requiring physical deployment validation. The strongest quantitative evidence indicates consistently lower simulated full end-to-end critical-event latency than the matched cloud-only baseline, alongside reduced simulated WAN traffic under the configured filtering/payload model.

The comparison does not establish clinical effectiveness, production Kafka saturation throughput, physical Jetson inference latency, or full-stack blockchain deployment performance. Architecturally, HealthGuard additionally provides off-chain large-object storage, permissioned authorization/audit metadata, a break-glass workflow, and device-signature verification. These are design capabilities whose real-world reliability, governance, and regulatory compliance must be validated in deployment-specific studies.

5. Conclusion

This paper presented HealthGuard, a five-layer hybrid edge-blockchain architecture that separates latency-sensitive IoMT telemetry from high-volume clinical data. Critical events are signature-verified and triaged at the edge and streamed through Apache Kafka, large encrypted objects are stored off-chain in IPFS, and a permissioned blockchain records compact authorization and audit metadata, including an auditable break-glass emergency path. In a matched simulation against a cloud-only baseline (100–10,000 devices), HealthGuard reduced the mean run-level median end-to-end latency from 46.574–140.775 *ms* to 35.400–115.119 *ms* (18.22–24.90%; Holm-adjusted $p < 0.05$ at every load) and, under the configured 0.80 routine-forward ratio, lowered simulated WAN traffic by 77.0%, with delivery success at 10,000 devices of 0.972 versus 0.967.

These findings remain bounded by the simulation: the $<$20 *ms* local-path, 40,000 events/s, and $<$1 s blockchain values are design targets; clinical accuracy, physical Jetson performance, and a matched Proof-of-Work comparison were not evaluated. Future work will benchmark the full stack on real edge hardware with production-grade Kafka, IPFS, and permissioned-blockchain infrastructure, measuring tail latency, achieved throughput, and fault behavior, alongside clinical validation and FHIR-profile interoperability testing.

Competing Interests Statement

The authors declare that they have no competing interests.

Data Availability Statement

The deterministic surrogate simulation code, experimental configuration files, random-seed list, synthetic run-level results, statistical analysis scripts, and selected derived outputs supporting the Phase 4 evaluation are publicly available in the HealthGuard Reproducibility Package (version 1.0.0), archived on Zenodo at https://doi.org/10.5281/zenodo.23824144.

The archive includes reproducibility instructions and methodological limitations. The released materials support independent computational reproduction of the reported surrogate simulation outputs. They do not constitute a complete deployment of the proposed healthcare edge-blockchain architecture or empirical validation on physical infrastructure.

Additional internal development materials and implementation prototypes outside the reported simulation are not included in this public release.

Statement on the Ethical Use of AI Tools

OpenAI ChatGPT was used for language editing, consistency checking, and manuscript-formatting support. It was not used to generate experimental measurements. The authors reviewed the revised manuscript and take full responsibility for the scientific content, analyses, interpretations, and conclusions.

References

[1] A. A. El-Saleh, A. M. Sheikh, M. A. Albreem, and M. S. Honnurvali, "The Internet of Medical Things (IoMT): Opportunities and challenges," *Wireless Netw.*, vol. 31, no. 1, pp. 327–344, Jan. 2025. DOI: 10.1007/s11276-024-03764-8

[2] S. Suneel et al., "Exploring the role of 6G technology in smart healthcare systems," in *Smart Hospitals: 5G, 6G and Moving Beyond Connectivity*, Wiley, 2024, pp. 287–314. DOI: 10.1002/9781394275472.ch14

[3] D. C. Nguyen et al., "6G Internet of Things: A comprehensive survey," *IEEE Internet Things J.*, vol. 9, no. 1, pp. 359–383, Jan. 2022. DOI: 10.1109/JIOT.2021.3103320

[4] A. Qayyum et al., "Secure and robust machine learning for healthcare: A survey," *IEEE Rev. Biomed. Eng.*, vol. 14, pp. 156–180, 2021. DOI: 10.1109/RBME.2020.3013489

[5] A. Adadi and M. Berrada, "Peeking inside the black-box: A survey on explainable artificial intelligence (XAI)," *IEEE Access*, vol. 6, pp. 52 138–52 160, 2018. DOI: 10.1109/ACCESS.2018.2870052

[6] M. Agiwal, A. Roy, and N. Saxena, "Next generation 5G wireless networks: A comprehensive survey," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 3, pp. 1617–1655, 3rd Quart. 2016. DOI: 10.1109/COMST.2016.2532458

[7] Y. Siriwardhana et al., "The role of 5G for digital healthcare against COVID-19 pandemic: Opportunities and challenges," *ICT Express*, vol. 7, no. 2, pp. 244–252, Jun. 2021. DOI: 10.1016/j.icte.2020.10.002

[8] A. Ahad, M. Tahir, and K. L. A. Yau, "5G-based smart healthcare network: Architecture, taxonomy, challenges and future research directions," *IEEE Access*, vol. 7, pp. 100 747–100 762, 2019. DOI: 10.1109/ACCESS.2019.2930628

[9] M. S. Hossain and G. Muhammad, "Cloud-assisted Industrial Internet of Things (IIoT)—Enabled framework for health monitoring," *Comput. Netw.*, vol. 101, pp. 192–202, Jun. 2016. DOI: 10.1016/j.comnet.2016.01.009

[10] B. Fong, A. C. M. Fong, and C. K. Li, "Internet of Things in smart ambulance and emergency medicine," in *Internet of Things A to Z: Technologies and Applications*, Wiley, 2018, pp. 475–506. DOI: 10.1002/9781119456735.ch17

[11] F. Wu, T. Wu, and M. R. Yuce, "An IoT network system for connected safety and health monitoring applications," *Sensors*, vol. 19, no. 1, Jan. 2019, Art. no. 21. DOI: 10.3390/s19010021

[12] S. Mahalakshmi et al., "Adaptive ambulance monitoring system using IoT," *Meas.: Sensors*, vol. 24, Dec. 2022, Art. no. 100555. DOI: 10.1016/j.measen.2022.100555

[13] K. Zhang, J. Ni, K. Yang, X. Liang, J. Ren, and X. S. Shen, "Security and privacy in smart city applications: Challenges and solutions," *IEEE Commun. Mag.*, vol. 55, no. 1, pp. 122–129, Jan. 2017. DOI: 10.1109/MCOM.2017.1600267CM

[14] B. Kelly et al., "Cybersecurity in healthcare," in *Trends of Artificial Intelligence and Big Data for E-Health*, Springer, 2023, pp. 213–231. DOI: 10.1007/978-3-031-11199-0_11

[15] R. ElGawish et al., "Detecting ransomware within real healthcare medical records adopting Internet of Medical Things using machine and deep learning techniques," *Int. J. Adv. Comput. Sci. Appl.*, vol. 13, no. 2, pp. 377–384, 2022. DOI: 10.14569/IJACSA.2022.0130270

[16] T. B. Josey and D. S. Misbha, "Man-in-the-Middle attack mitigation in IoT sensors with hash-based multidimensional Lamport digital signature," in *Proc. Int. Virtual Conf. Ind. (IVCI)*, Springer, 2021, pp. 47–56. DOI: 10.1007/978-981-19-9989-5_5

[17] R. Sivasubramanian and M. Rajagopal, "SecureEdge-MedChain: A post-quantum blockchain and federated learning framework for real-time predictive diagnostics in IoMT," *Sensors*, vol. 25, no. 19, Sep. 2025, Art. no. 5988. DOI: 10.3390/s25195988

[18] A. Sharma, S. Sharma, and S. Kaur, "Enhancing IoMT security: A blockchain-based anomaly detection framework leveraging explainable AI," *Cluster Comput.*, vol. 29, no. 2, 2026, Art. no. 87. DOI: 10.1007/s10586-025-05836-6

[19] S. Rouhani and R. Deters, "Blockchain-based access control systems: State of the art and challenges," in *Proc. IEEE/WIC/ACM Int. Conf. Web Intell. (WI)*, 2019, pp. 423–428. DOI: 10.1145/3350546.3352561

[20] N. Ettaloui, S. Arezki, and T. Gadi, "An overview of blockchain-based electronic health records and compliance with GDPR and HIPAA," *Data and Metadata*, vol. 2, 2023, Art. no. 166. DOI: 10.56294/dm2023166

[21] J. Pan et al., "EdgeChain: An edge-IoT framework and prototype based on blockchain and smart contracts," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 4719–4732, Jun. 2019. DOI: 10.1109/JIOT.2018.2878154

[22] S. Dash et al., "Edge and fog computing in healthcare—A review," *Scalable Comput.: Pract. Exper.*, vol. 20, no. 2, pp. 191–206, 2019. DOI: 10.12694/scpe.v20i2.1504

[23] B. Appasani et al., "Blockchain-enabled smart grid applications: Architecture, challenges, and solutions," *Sustainability*, vol. 14, no. 14, Jul. 2022, Art. no. 8801. DOI: 10.3390/su14148801

[24] A. Azaria, A. Ekblaw, T. Vieira, and A. Lippman, "MedRec: Using blockchain for medical data access and permission management," in *Proc. 2nd Int. Conf. Open Big Data (OBD)*, 2016, pp. 25–30. DOI: 10.1109/OBD.2016.11

[25] I. Yaqoob et al., "Blockchain for healthcare data management: Opportunities, challenges, and future recommendations," *Neural Comput. Appl.*, vol. 34, no. 14, pp. 11 475–11 490, Jul. 2022. DOI: 10.1007/s00521-020-05519-w

[26] A. A. Abdellatif et al., "Edge computing for smart health: Context-aware approaches, opportunities, and challenges," *IEEE Netw.*, vol. 33, no. 3, pp. 196–203, May/Jun. 2019. DOI: 10.1109/MNET.2019.1800083

[27] E. Yıldırım, A. Çalhan, and M. Cicioğlu, "Performance analysis of disease diagnostic system using IoMT and real-time data analytics," *Concurrency Computat.: Pract. Exper.*, vol. 34, no. 13, Jun. 2022, Art. no. e6916. DOI: 10.1002/cpe.6916

[28] S. Kumar, A. K. Bharti, and R. Amin, "Decentralized secure storage of medical records using Blockchain and IPFS: A comparative analysis with future directions," *Security and Privacy*, vol. 4, no. 5, Sep./Oct. 2021, Art. no. e162. DOI: 10.1002/spy2.162

[29] M. Mukhedkar et al., "Advanced and secure data sharing scheme with blockchain and IPFS: A brief review," in *Proc. 15th Int. Conf. Comput. Commun. Netw. Technol. (ICCCNT)*, 2024, pp. 1–5. DOI: 10.1109/ICCCNT61001.2024.10724999

[30] A. Alsadhan, A. Alhogail, and H. Alsalamah, "Blockchain-based privacy preservation for the Internet of Medical Things: A literature review," *Electronics*, vol. 13, no. 19, 2024, Art. no. 3832. DOI: 10.3390/electronics13193832

[31] A. Dubovitskaya et al., "Secure and trustable electronic medical records sharing using blockchain," *arXiv:1709.06528*, 2017.

[32] M. Pedrera-Jiménez et al., "Can OpenEHR, ISO 13606, and HL7 FHIR work together?" *J. Med. Internet Res.*, vol. 25, 2023, Art. no. e48702. DOI: 10.2196/48702

[33] M. Halimuzzaman et al., "Blockchain technology for integrating electronic records of digital healthcare," *Integr. Biomed. Res.*, vol. 8, no. 7, pp. 1–11, 2024. DOI: 10.25163/angiotherapy.879740

[34] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of Things: A survey on enabling technologies, protocols, and applications," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 4, pp. 2347–2376, 4th Quart. 2015. DOI: 10.1109/COMST.2015.2444095

[35] J. Zhou, Z. Cao, X. Dong, and A. V. Vasilakos, "Security and privacy for cloud-based IoT: Challenges," *IEEE Commun. Mag.*, vol. 55, no. 1, pp. 26–33, Jan. 2017. DOI: 10.1109/MCOM.2017.1600363CM

[36] R. A. Lima, D. D. de Macedo, and C. A. Lima, "Blockchain-enabled distributed data management for smart health: Enhancing integrity and auditability in edge-fog-cloud ecosystems," in *Proc. IEEE Int. Conf. Edge Comput. Commun. (EDGE)*, Jul. 2025, pp. 1–12. DOI: 10.1109/EDGE67623.2025.00010

[37] A. M. Rahmani et al., "Exploiting smart e-Health gateways at the edge of healthcare Internet-of-Things: A fog computing approach," *Future Gener. Comput. Syst.*, vol. 78, pp. 641–658, Jan. 2018. DOI: 10.1016/j.future.2017.02.014

[38] H. Zhou et al., "Deep reinforcement learning for energy-efficient computation offloading in mobile-edge computing," *IEEE Internet Things J.*, vol. 9, no. 2, pp. 1517–1530, Jan. 2022. DOI: 10.1109/JIOT.2021.3091142

[39] Y. Chen et al., "Energy efficient dynamic offloading in mobile edge computing for Internet of Things," *IEEE Trans. Cloud Comput.*, vol. 9, no. 3, pp. 1050–1060, Jul.–Sep. 2021. DOI: 10.1109/TCC.2019.2898657

[40] Y. Xiao, Y. Jia, C. Liu, and X. Cheng, "Edge computing security: State of the art and challenges," *Proc. IEEE*, vol. 107, no. 8, pp. 1608–1631, Aug. 2019. DOI: 10.1109/JPROC.2019.2918437

[41] L. Gao, T. H. Luan, B. Gu, Y. Qu, and Y. Xiang, *Privacy-Preserving in Edge Computing*. Cham, Switzerland: Springer, 2021. DOI: 10.1007/978-981-16-2199-4

[42] A. Ahad et al., "A comprehensive review on 5G-based smart healthcare network security: Taxonomy, issues, solutions and future research directions," *Array*, vol. 18, Jul. 2023, Art. no. 100290. DOI: 10.1016/j.array.2023.100290

[43] D. C. Nguyen et al., "Federated learning for Internet of Things: A comprehensive survey," *IEEE Commun.*

Surveys Tuts., vol. 23, no. 3, pp. 1622–1658, 3rd Quart. 2021. DOI: 10.1109/COMST.2021.3075439

[44] N. Kannan and K. Balasubramanian, "Privacy-preserving and quantum-resistant federated learning framework for fog healthcare systems," *Int. J. Comput. Intell. Syst.*, vol. 19, 2026, Art. no. 345. DOI: 10.1007/s44196-026-01477-8

[45] Q. I. Xia, E. B. Sifah, K. O. Asamoah, J. Gao, X. Du, and M. Guizani, "MeDShare: Trust-less medical data sharing among cloud service providers via blockchain," *IEEE Access*, vol. 5, pp. 14 757–14 767, 2017. DOI: 10.1109/ACCESS.2017.2730843

[46] H. Rafik, A. Ettaoufik, and A. Maizate, "Reliable secure medical data sharing model based private blockchain structure," in *Proc. 14th Int. Conf. Intell. Syst.: Theor. Appl. (SITA)*, Nov. 2023, pp. 1–8. DOI: 10 . 1109 / SITA60746.2023.10373693

[47] A. K. Ranjan and P. Kumar, "A survey on blockchain-based privacy preserving techniques for edge Internet of Things," *Int. J. Comput. Appl.*, vol. 47, no. 6, pp. 497–508, 2025. DOI: 10.1080/1206212X.2025.2498687

[48] S. Gupta, "Zero-knowledge proofs for privacy-preserving systems: A survey across blockchain, identity, and beyond," *Eng. Technol. J.*, vol. 10, no. 7, pp. 5755–5761, 2025. DOI: 10.47191/etj/v10i07.23

[49] C. N. Vorisek et al., "Fast Healthcare Interoperability Resources (FHIR) for interoperability in health research: Systematic review," *JMIR Med. Inform.*, vol. 10, no. 7, Jul. 2022, Art. no. e35724. DOI: 10.2196/35724

[50] S. R. Mallick, V. Goswami, R. K. Lenka, R. K. Barik, R. Soorat, and N. K. Ray, "LIVER: A next-generation secure and lightweight blockchain-IPFS edge computing model for healthcare with queueing techniques," *Int. J. Netw. Distrib. Comput.*, vol. 13, no. 2, pp. 1–14, 2025. DOI: 10.1007/s44227-025-00070-3

[51] N. Tiwari, P. K. Biswal, and P. Ranjan, "Decoupling data ingestion and blockchain processing: A high-performance event-driven Kafka architecture for remote patient monitoring," *Cluster Comput.*, vol. 29, no. 8, 2026, Art. no. 508. DOI: 10.1007/s10586-026-06311-6

[52] N. Tiwari, P. K. Biswal, and P. Ranjan, "Blockchain-enhanced electronic health records: A Kafka-based approach for IoMT integration," *J. Netw. Syst. Manage.*, vol. 34, no. 2, 2026, Art. no. 56. DOI: 10.1007/s10922-025-10025-2