

Article

Privacy-Preserving Clinical Decision Support for Emergency Triage Using LLMs: System Architecture and Real-World Evaluation

Alper Karamanlioğlu ^{1,2,*} , Berkan Demirel ² , Onur Tural ³, Osman Tufan Doğan ⁴ and Ferda Nur Alpaslan ¹ 

¹ Department of Computer Engineering, Middle East Technical University, Ankara 06800, Türkiye; alpaslan@ceng.metu.edu.tr

² Product and R&D, TURKSAT, Ankara 06839, Türkiye; berkan.demirel@turksat.com.tr

³ Emergency Medicine, Zonguldak Bülent Ecevit University, Zonguldak 67100, Türkiye; onur.tural@beun.edu.tr

⁴ R&D and Innovation, Innova IT Solutions, Ankara 06800, Türkiye; osdogan@innova.com.tr

* Correspondence: alperk@ceng.metu.edu.tr; Tel.: +90-553-487-5233

Abstract

This study presents a next-generation clinical decision-support architecture for Clinical Decision Support Systems (CDSS) focused on emergency triage. By integrating Large Language Models (LLMs), Federated Learning (FL), and low-latency streaming analytics within a modular, privacy-preserving framework, the system addresses key deployment challenges in high-stakes clinical settings. Unlike traditional models, the architecture processes both structured (vitals, labs) and unstructured (clinical notes) data to enable context-aware reasoning with clinically acceptable latency at the point of care. It leverages big data infrastructure for large-scale EHR management and incorporates digital twin concepts for live patient monitoring. Federated training allows institutions to collaboratively improve models without sharing raw data, ensuring compliance with GDPR/HIPAA, and FAIR principles. Privacy is further protected through differential privacy, secure aggregation, and inference isolation. We evaluate the system through two studies: (1) a benchmark of 750+ USMLE-style questions validating the medical reasoning of fine-tuned LLMs; and (2) a real-world case study ($n = 132$, 75.8% first-pass agreement) using de-identified MIMIC-III data to assess triage accuracy and responsiveness. The system demonstrated clinically acceptable latency and promising alignment with expert judgment on reviewed cases. The infectious disease triage case demonstrates low-latency recognition of sepsis-like presentations in the ED. This work offers a scalable, audit-compliant, and clinician-validated blueprint for CDSS, enabling low-latency triage and extensibility across specialties.

Keywords: clinical decision support; triage; fair data principles; generative AI; large language models; federated learning; privacy preservation; healthcare architecture; infectious disease triage; sepsis alerting



Academic Editors: Enno van der Velde and Thomas Heston

Received: 24 June 2025

Revised: 22 July 2025

Accepted: 25 July 2025

Published: 29 July 2025

Citation: Karamanlioğlu, A.; Demirel, B.; Tural, O.; Doğan, O.T.; Alpaslan, F.N. Privacy-Preserving Clinical Decision Support for Emergency Triage Using LLMs: System Architecture and Real-World Evaluation. *Appl. Sci.* **2025**, *15*, 8412. <https://doi.org/10.3390/app15158412>

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Clinical environments—particularly emergency departments—demand rapid and accurate triage decisions to prioritize care for patients in critical condition. However, traditional decision-making heavily relies on clinician experience and often struggles with information overload from Electronic Health Records (EHRs; see List of Abbreviations for acronym definitions) and unstructured clinical notes. As the volume and complexity of medical data continue to grow, synthesizing this information in real time has become

increasingly difficult. This challenge has accelerated interest in Clinical Decision Support Systems (CDSS) that integrate artificial intelligence (AI) to support fast, consistent, and evidence-based decisions.

CDSS have increasingly been recognized for their potential to enhance clinical outcomes by delivering timely, evidence-based recommendations directly at the point of care. Modern CDSS implementations leverage advanced machine learning algorithms, such as deep learning, to interpret complex, multimodal healthcare data—ranging from structured EHR data to unstructured clinical narratives—thereby improving diagnostic accuracy and treatment effectiveness while minimizing clinician cognitive load [1].

Early applications of machine learning to structured EHR data showed promising results in predicting outcomes such as mortality and readmission [2]. Rajkomar et al. demonstrated that deep neural networks could model hospital outcomes with high accuracy when trained on large EHR datasets [3]. A comprehensive tutorial by Esteva et al. summarized early successes and pitfalls of deep learning in healthcare [4]. However, most of these models were limited in their ability to handle unstructured clinical narratives—notes, referrals, and patient reports—which often contain critical diagnostic information.

Simultaneously, concerns about data privacy and cross-institutional collaboration have constrained the centralization of health data. Regulations such as GDPR (in the EU) [5] and HIPAA (in the U.S.) [6] restrict sharing of identifiable patient information [7–9]. Federated Learning (FL) emerged as a viable solution, enabling collaborative model training without transferring raw data between hospitals. Sheller et al. demonstrated the feasibility of FL in training deep learning models across multiple hospitals for brain tumor segmentation, matching centralized performance while preserving privacy [7].

More recently, Large Language Models (LLMs) have shown strong capabilities in understanding and generating clinical language, even passing medical licensing exams [10,11]. These models can extract medical concepts, suggest differential diagnoses, and produce natural-language rationales for clinical decisions. However, integrating LLMs into latency-sensitive CDSS workflows introduces new technical challenges: high computational costs and validation of medical reasoning.

In parallel with AI advancements, the growing scale and complexity of clinical data have necessitated the use of big data infrastructure. Modern CDSS platforms must ingest, index, and retrieve data at scale across modalities—from live-streamed vital signs to historical lab results. Technologies such as distributed file systems, parallel processing engines (e.g., Spark [12]), and information retrieval frameworks have become integral in managing these volumes. The proposed architecture leverages such technologies for efficient data fusion and fast access to relevant patient or cohort records, which are both key to low-latency triage.

This paper introduces a modular, triage-focused CDSS architecture that integrates LLMs, FL, and big data concepts in a privacy-preserving and adaptable framework. The architecture comprises multiple layers, including structured and unstructured data ingestion, federated model training, LLM-based reasoning, Decision Support Engines (DSE), and clinician-facing interfaces. The system is designed to scale across hospital networks, adapt to local practice variations, and comply with FAIR (Findable, Accessible, Interoperable, Reusable) data principles.

We present a two-part evaluation to validate the performance of the system. First, we benchmarked our fine-tuned LLMs with a curated set of United States Medical Licensing Examination (USMLE) [13] questions to assess general clinical-reasoning capabilities. Second, we constructed an infectious disease cohort from the MIMIC-III dataset and applied the CDSS pipeline to both large-scale and expert-reviewed cases to assess real-world triage functionality. From an initial pool of 6 111 de-identified emergency department (ED) vis-

its whose diagnoses lay within ICD-9 [14] codes 001–139 or the severe-sepsis extensions 995.91/995.92, we retained 132 encounters after applying the full preprocessing pipeline (hash-based de-identification, triage-note normalization, vital-sign cleaning, and completeness checks). These 132 records—each with a time-stamped triage note—form the labeled infectious disease cohort used for downstream modeling and expert review.

The remainder of the paper is organized as follows: Section 2 (Materials and Methods) describes the proposed architecture, data sources, and core system components. Section 3 (Results) reports the dual evaluation—USMLE-style benchmarking and the Infectious Disease case study conducted on a curated MIMIC-III cohort. Section 4 (Discussion) analyzes the strengths of the system, limitations, and future directions. Finally, Section 5 (Conclusions) summarizes the main contributions and outlines real-world applicability.

1.1. Contributions

This study makes the following seven distinct contributions to the field of privacy-preserving, triage-oriented CDSS:

- **End-to-end, triage-focused architecture.** Designed a modular framework that fuses structured (vitals, labs) and unstructured (clinical notes) data, integrates LLMs, FL, big-data infrastructure, and digital-twin concepts, and achieves near-real-time latency under typical inference conditions.
- **Real-world infectious disease triage case study.** Applied the CDSS to a de-identified infectious disease ED cohort ($n \approx 132$), achieving 75.8% first-pass agreement with expert reviewers on diagnosis, justification, and next-step recommendations.
- **Large-scale language-model benchmarking.** Fine-tuned a Llama 3.1-70B model that achieves up to 74% overall accuracy (83% on USMLE Step 2) across 750 curated USMLE-style questions—substantially outperforming baseline LLMs and medical-student means.
- **Privacy-preserving model-development pipeline.** Coupled secure FL (differential privacy and secure aggregation) with SGX-isolated inference, enabling multi-institutional collaboration while remaining fully GDPR/HIPAA compliant.
- **FAIRification workflow and ontology-mapping service.** Implemented a reproducible pipeline that aligns all ingested resources, assigns persistent identifiers, enriches metadata, and harmonizes codes via an Ontology-Mapping and Terminology Service (OMTS)—ensuring data are Findable, Accessible, Interoperable, and Reusable.
- **Event-driven, low-latency data-fusion engine.** Demonstrated horizontal scalability under doubled ED loads using Apache Kafka, Spark, and Node-RED to orchestrate streaming vitals, IoT signals, and EHR updates without performance loss.
- **Four-pillar GDPR compliance framework.** Mapped key articles of Regulation (EU) 2016/679 to concrete technical and organizational controls—including automatic DPIA generation, rights-of-erasure workflows, and continuous privacy KPIs—providing verifiable legal conformity.

1.2. Background

Early CDSS relied on fixed if-then rules or simple acuity scores—transparent yet brittle when confronted with complex, heterogeneous data. Subsequent work introduced classical machine-learning models that used vitals and chief complaints to improve triage accuracy, but these approaches still struggled with free-text notes and could not share insights across institutions because of privacy barriers [15]. These limitations spurred the adoption of transformer-based LLMs for complex, mixed-format data and privacy-preserving methods—especially FL—to enable cross-institutional training without sharing raw patient data.

Attention Mechanisms in Clinical Contexts. Attention Mechanisms in Neural Architectures have shown strong performance in focusing computational resources on the most relevant input features, thereby improving efficiency and accuracy. In clinical settings, these mechanisms have been extensively applied to medical image analysis tasks—such as segmentation, detection, and classification—helping systems highlight important features in modalities like CT, MRI, and X-ray imaging [16]. Similar attention-based mechanisms have been shown to improve real-time tracking performance in other domains, such as poultry monitoring, where the YOLO-Chicken algorithm enhanced detection accuracy and efficiency through attention modules [17]. This demonstrates the potential utility of lightweight yet precise attention-based architectures in clinical triage scenarios, where rapid and accurate interpretation of medical images or signals is essential.

LLMs in Healthcare. The use of transformer-based LLMs in healthcare has rapidly expanded. Early efforts like ClinicalBERT and BioBERT, fine-tuned BERT variants on clinical notes, improving clinical concept extraction and note classification, Named Entity Recognition (NER) and relation extraction [18,19]. LLMs have been applied to tasks such as summarizing patient histories, answering clinicians' questions, and even generating draft reports. Notably, GPT-3.5 and GPT-4 demonstrated the ability to answer medical exam questions at a level approaching or surpassing human medical graduates [20]. In a 2024 study, Bicknell et al. evaluated a specialized GPT-4 (termed ChatGPT-4 Omni) on United States Medical Licensing Examination (USMLE) questions, finding it achieved 89–91% accuracy, significantly higher than the ~60% average of medical students [11].

Privacy-Preserving AI and FL. Ensuring patient privacy while benefiting from large-scale data is a central theme in current research. FL has been widely studied as an approach to train AI models on distributed data. Several systematic reviews have cataloged the uses of FL in healthcare, noting applications in medical imaging, remote patient monitoring, and predictive modeling [21]. A 2024 survey focusing on smart-healthcare deployments catalogs 70+ FL use-cases across sensors, imaging, and EHR data [22]. The work of Sheller et al. demonstrated feasibility for deep models on imaging data [7], and others like Li et al. have extended FL to hospital wearable sensor networks, showing that vital sign anomaly detectors can be trained collaboratively without sharing raw data [23]. In a follow-up study, Sheller et al. validated the practicality of FL in a multi-institutional medical imaging context, showing comparable accuracy to centralized training [24]. Nevertheless, vanilla FL has limitations: communication overhead, data heterogeneity between sites, and the risk of privacy leakage from model updates [25]. Recent frameworks specifically aim to integrate FL with LLM training. For instance, Fan et al. introduced FATE-LLM [26], an industrial-grade FL framework tailored for LLMs. In parallel, hybrid federated approaches combining meta-heuristic and classifier ensembles have been demonstrated for cardiovascular disease prediction [27]. Likewise, OpenFedLLM [28] enabled training a GPT-style model on decentralized clinical text data, employing techniques to reduce communication cost and handle non-IID (not independent and identically distributed) data across nodes [29]. To further bolster privacy, researchers have combined FL with techniques such as differential privacy—adding calibrated noise to model gradients—and secure multi-party computation or homomorphic encryption to prevent any single party from reconstructing the data of another [30–32].

Big Data in Clinical AI. The effective use of big data frameworks has become essential for modern CDSS platforms, especially in emergency and high-throughput clinical settings. Frameworks such as Apache Spark and Hadoop [33] have been adopted for large-scale analytics on EHRs, enabling fast retrieval of historical cases, patient stratification, and cohort identification [34,35]. For example, Goh et al. developed the SERA algorithm, which utilizes both structured data and unstructured clinical notes to predict and diagnose

sepsis, achieving high predictive accuracy up to 12 h before onset [36]. Additionally, big data solutions underpin scalable storage and indexing of clinical narratives, with Elasticsearch and Solr frequently used to enable fast semantic search over unstructured clinical text [37].

Digital Twins and Continuous Data Integration. Digital twin frameworks have been explored for patient monitoring and personalized care [38], especially with the rise of Internet of Things (IoT) medical devices (e.g., wearable vital sign monitors). For example, Eclipse Ditto [39] is an open-source platform that manages digital twin state and synchronization with IoT data streams. In a CDSS context, digital twins can mirror the status of each patient, integrating live sensor readings, lab results, and other data, and trigger alerts or recommendations when certain criteria are met. Recent studies have integrated digital twins with event-driven architectures: Node-RED [40] workflows subscribing to events (e.g., a sudden spike in heart rate) and then invoking clinical decision logic [41,42].

Emerging Technologies for Real-Time Clinical Data. Emerging technologies such as Flying Ad Hoc Networks (FANETs), characterized by decentralized, dynamic, and rapidly deployable aerial nodes, offer promising solutions for healthcare-related applications, particularly in emergency scenarios requiring rapid data collection and dissemination. FANETs can significantly enhance the responsiveness and coverage of real-time monitoring systems by facilitating flexible, high-speed communications in remote or disaster-affected areas [43].

2. Materials and Methods

2.1. System Architecture

The proposed CDSS architecture is modular and layered, designed to handle the end-to-end flow from data acquisition to clinical decision support in a privacy-preserving, scalable, and low-latency manner. Figure 1 depicts the overall system, which we summarize as follows. The system ingests data from multiple hospital sources, processes it through successive layers including local ML/LLM analysis and federated model updates, and ultimately provides decision outputs to clinicians via a user interface. Table 1 summarizes the architectural modules and their respective roles. This overview helps contextualize the function of each layer before we examine the system architecture and evaluation. Implementation specifics—covering the underlying Technology Stack—are provided in Supplementary Section S1.

Figure 1 highlights how low-latency ingestion feeds the LLM and FL layers before recommendations surface in the User Interface (UI).

Table 1. Key System Components and Their Functions in the Proposed CDSS Architecture.

Component	Function
Data Ingestion Layer	Collect and preprocess data with clinically acceptable latency from diverse sources (EHRs, clinical notes, IoT devices, etc.)
FL	Train AI models on decentralized data at each site to preserve privacy; aggregate model updates into a global model
LLM Layer	Interpret unstructured text (e.g., clinical notes) and perform language tasks like question-answering or summarization
Big Data Management	Store and manage large volumes of structured and unstructured data; enable efficient retrieval and parallel processing
DSE	Combine observations from data (structured and unstructured) to generate clinical recommendations (e.g., triage level, diagnostic suggestions)

Table 1. Cont.

Component	Function
MLOps/CI-CD Pipeline	Automate continuous integration and deployment of models; monitor model versions and facilitate updates/improvements
UI	Provide clinicians with an interactive interface to view alerts and recommendations; allow data input and feedback
Compliance and Monitoring	Ensure all operations comply with privacy regulations (GDPR/HIPAA); log data lineage and monitor system performance and security

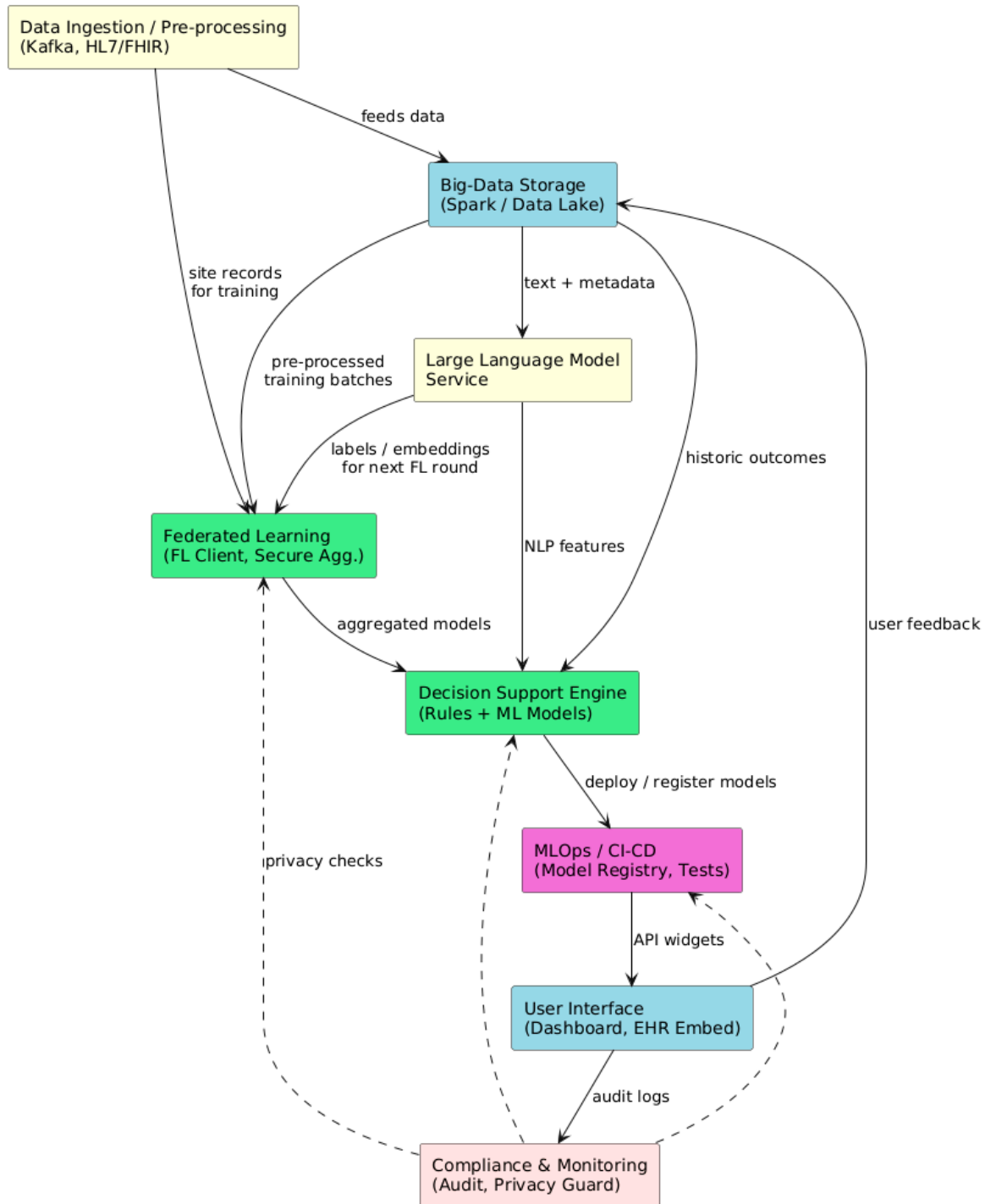


Figure 1. High-level data flow from ingestion through federated training to clinician UI.

2.1.1. Modular Layered Design

To handle the complexity of healthcare data and decision processes, the system is organized into distinct layers, each responsible for specific tasks. This layered approach follows principles of separation of concerns and makes the system more maintainable and extensible. The main layers include the following:

1. **Data Ingestion and Pre-processing Layer.** This layer continuously captures structured and unstructured data from EHR tables, HL7/FHIR messages [44], and device telemetry. Apache Kafka [45] handles high-throughput streaming, while validation and normalization routines standardize vitals, labs, and other structured fields. Clinical free-text passes through NLP pipelines, and medical images follow standard pre-processing before archival. All events have persisted in the big-data store and forwarded in a unified schema to downstream analytic services, shielding models from source heterogeneity and supporting interactive-latency inference.
2. **Big Data Management and Storage Layer.** All data (both raw and processed) as well as intermediate results need to be stored and managed efficiently. We incorporate a big data framework (e.g., Hadoop/Spark ecosystem) to handle this. Key aspects of this layer:
 - **Data Lake and Warehousing.** A distributed file store holds raw ingested data (Bronze layer in medallion architecture terms) and refined data (Silver/Gold layers) in parquet/orc formats. Structured data may also be stored in a relational warehouse for easy query (SQL) access, while text embeddings or time-series might reside in a NoSQL or specialized time-series database for fast retrieval.
 - **Parallel Processing.** Apache Spark v3.5.0 is used for heavy analytics [46,47] that can be parallelized, such as scanning through millions of records to find similar past cases (for retrieving relevant precedents for a current patient). This ensures the system can be scaled to handle enterprise-level EHR datasets and not just small samples.
 - **Indexing and Search.** For unstructured text, we maintain indexes (possibly leveraging Elasticsearch or Solr) to Support Information retrieval. This is particularly useful if the CDSS includes a knowledge base of medical guidelines or past case summaries: when the Decision Engine needs to gather supporting evidence for a recommendation, it can query these indexes (with the help of the LLM to formulate the query) to fetch relevant documents or data points.
 - **Data Lifecycle and Security.** This layer also enforces data retention policies and encryption at rest. It tags data with metadata for provenance (which is important for auditing—knowing which source a particular piece of information came from). Access controls ensure that only authorized components or users can retrieve sensitive data, aligning with hospital IT security policies.
3. **FL Layer.** Once data is preprocessed and ready for model training or updating, the FL layer comes into play for model development in a collaborative but privacy-preserving manner. In a multi-center deployment, each hospital (or clinic) runs an instance of the learning algorithm on its local data. Rather than sending raw patient data to a central server, each site computes updates to the model (e.g., gradient updates) using its local data. A central coordinator (which could be on-premises at a lead institution or in a secure cloud) collects encrypted model updates from the sites and aggregates them (for example, by averaging weights—akin to federated averaging). Only the aggregated model—which now has learned from all participating sites—is sent back to each hospital. This way, the global CDSS model benefits from a wide range of clinical cases (improving its generalization), yet no sensitive patient data ever leaves

the source institutions' control. In our architecture, this layer is crucial for training both the triage decision models and any language models on distributed text corpora (whether labeled or unlabeled, which dictates the specific learning objective but does not alter the underlying privacy guarantees). For example, if several hospitals use the system, they can collaboratively fine-tune an embedded LLM on their combined clinical note data without sharing actual notes with each other, thereby adapting the LLM to their medical domain jargon in a compliant manner. We also incorporate periodic evaluation rounds in FL—e.g., a set of common validation cases—to assess the performance of the global model after each round and ensure it is improving consistently. Figure 2 details the message exchange between local FL clients, the secure-aggregation service, and the privacy monitor.

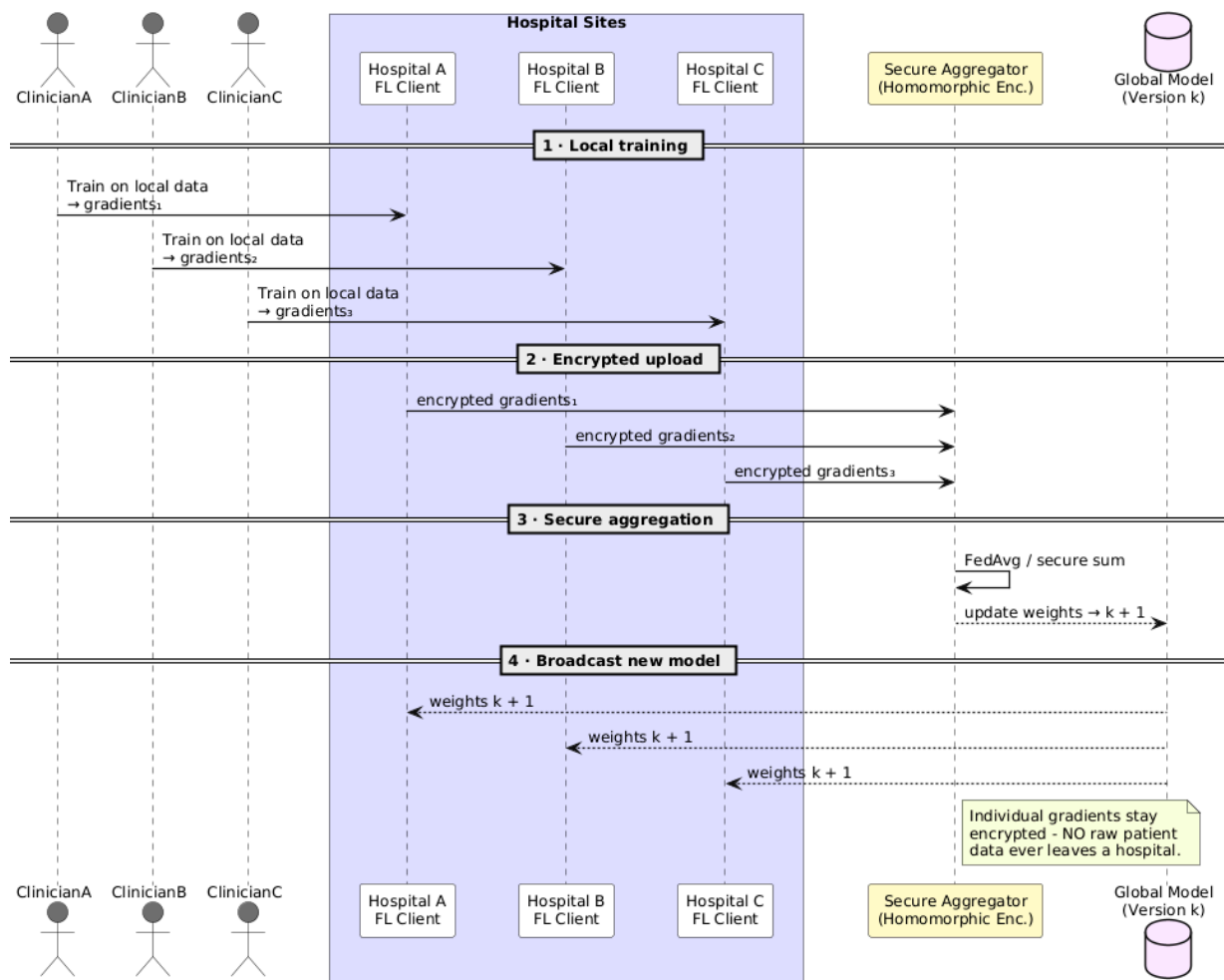


Figure 2. FL and Privacy Workflow. Note that secure aggregation prevents the coordinator from viewing individual gradients, addressing GDPR concerns.

FL synchronization in our proposed system occurs periodically, typically at defined intervals such as daily or weekly, depending on institutional preference and clinical workflow constraints. During each synchronization round, local models are trained independently at each participating institution on their respective patient data. Only the resulting model updates—specifically encrypted gradients or weight differentials—are securely transmitted to a central coordinator. The coordinator aggregates these updates using secure aggregation methods, such as federated averaging, and returns the updated global model to the institutions. This methodology significantly minimizes communication overhead by avoiding the exchange of large datasets or frequent updates.

4. **LLM Layer.** This layer embeds one or more LLMs into the CDSS for tasks involving natural language understanding and generation. The LLMs (such as a fine-tuned version of the LLaMA-3 [48] of Meta or an open medical model like MedAlpaca [49]) can serve multiple functions:
 - **Clinical Text Understanding.** The LLM processes unstructured inputs—e.g., the free-text note of a triage nurse describing the symptoms of a patient—and extracts structured information or meaning. It can perform clinical NER (finding mentions of conditions, medications), summarization of patient history, or translation of layperson language into medical terminology.
 - **Question Answering and Reasoning.** Clinicians can query the system in natural language, and the LLM will attempt to answer. For instance, a doctor might ask, “What is the recommended triage level for a 45-year-old patient with chest pain and sweating?”. The LLM, using its medical knowledge and the data of the patient, can provide an answer or explanation. This is akin to having a knowledgeable assistant that has read vast amounts of medical literature and patient cases). Our system uses the LLM to generate differential diagnoses or triage justification texts that accompany the decision support output.
 - **Integration with Decision Logic.** Outputs from the LLM layer feed into the DSE. For example, the LLM might flag that a note of the patient contains the phrase “central crushing chest pain,” which is suggestive of a myocardial infarction. This finding can be passed as a feature into a rule-based triage algorithm or combined with structured data (like an elevated troponin lab result) to determine a high-acuity triage classification. To further improve factual grounding and reduce hallucinations, the LLM can be integrated with Retrieval-Augmented Generation (RAG) pipelines that retrieve relevant documents during inference [50]. The LLM layer thus augments traditional CDSS algorithms with knowledge extracted from text that would otherwise require manual review.

This layer also synergizes with the FL layer when fine-tuning LLMs on local data. The concept of a federated fine-tuning for LLMs is introduced: each hospital can refine the LLM on its own notes (adhering to local patient privacy) and contribute to a shared model, as described above. By doing so, the model better adapts to institutional specifics (like shorthand or dialect in notes) while remaining broadly trained.

5. **DSE Layer.** The DSE is the core analytical brain of the system that generates clinical recommendations (e.g., triage category, likely diagnoses, alerts for abnormal findings). It brings together inputs from the structured data pipelines and the LLM outputs. The engine can utilize a combination of the following:
 - **Rule-Based Logic.** Standard clinical triage protocols—such as the ESI or institution-specific guidelines—are encoded into the decision engine. For example, if systolic blood pressure is <90 mmHg and heart rate > 130 bpm, the patient is flagged for immediate attention. These encoded rules serve as safety nets to ensure the system adheres to established clinical norms and provides conservative recommendations when vital signs are critical.
 - **Machine Learning Models.** Predictive models trained on historical emergency department data are used to assess the risk of critical outcomes (e.g., ICU admission or deterioration). Models such as gradient boosting machines or neural networks process both structured features (age, vitals, lab results) and encoded clinical narratives to generate risk scores. These scores are mapped to triage levels, supporting clinician decision-making with both statistical and semantic context.

- **Guideline Retriever Module.** This module retrieves relevant clinical guidelines or medical facts from an internal knowledge base and serves as a reference component during the decision-making process. The retrieved content is then provided to the LLM, which integrates this evidence to generate context-aware recommendations. For instance, the system may identify and retrieve similar historical cases—leveraging big data infrastructure and information retrieval techniques—and instruct the LLM to compare those with the current patient presentation.
- **Digital Twin Simulation.** If applicable, the DSE also checks the digital twin of the patient for any active alerts. For example, if the twin (representing a patient with chronic conditions at home) triggers an event “oxygen saturation dropped below 85%,” the engine would generate a high-priority alert for that patient, possibly even before they arrive at the hospital.

The output of the DSE is typically a set of one or more recommended actions or decision suggestions. For triage, this could be an assigned triage level (e.g., Level 1—Immediate intervention needed, or Level 3—Urgent but not life-threatening). For diagnostic support, it might be a list of probable diagnoses or a checklist of next steps (order a CT scan, perform a certain lab test, etc.). The DSE is designed to be extensible—new rules or models can be plugged in as medical knowledge evolves. It also operates under a safety-first principle: any high-risk condition detected (like signs of sepsis or heart attack) results in an automatic high-severity recommendation, erring on the side of caution.

6. **MLOps and CI/CD Pipeline Layer.** To keep the AI models of the system up-to-date and reliable, we integrate MLOps principles. This layer handles the continuous integration and deployment (CI/CD) of model updates. When new data becomes available or when the FL rounds produce a new global model, the pipeline automatically validates the model on a hold-out set of cases (or in A/B tests within the live system) before deploying it. The pipeline also monitors model performance metrics over time—for example, if the triage recommendations of the model start to drift or degrade (perhaps due to changes in patient population or emergence of new diseases), it can alert data scientists to retrain or recalibrate. This layer encompasses the following:
 - **Version Control for Models.** Each model (and even each set of rules) is versioned. The system can roll back to a prior stable model if a newly deployed one has issues.
 - **Automated Testing.** Similarly to software testing, we maintain a suite of test scenarios (clinical vignettes with known correct outputs) to rapidly verify that any model update does not break core functionality or violate clinical constraints.
 - **Continuous Learning.** In the long run, as clinicians use the system, their feedback (e.g., whether they accepted or overrode a recommendation) can be fed back into model training. The pipeline could schedule periodic retraining (say monthly) using accumulated new data and feedback, thereby improving the models iteratively. This approach reflects the core principles of a learning health system, where continuous feedback loops refine clinical knowledge and practice [51].
7. **UI Layer.** The UI is the touchpoint between the CDSS and the clinicians (doctors, nurses, etc.). It is designed to be intuitive and fit within clinical workflows to avoid alert fatigue or disruption. Key features of the UI include the following:
 - **Dashboard for Triage Nurses.** Displaying incoming patients with their vital signs and any alerts. For each patient, if the system has a triage suggestion, it is shown (e.g., color-coded by severity). The nurse can provide input (confirm or adjust

the triage level) and add additional notes, which the system will record—and possibly send back to update the model (with clinician approval).

- **Physician View.** Where a doctor can query the system for diagnostic support. They might see a summary of the case of the patient compiled by the LLM (including relevant past history, flagged abnormal findings, etc.), and a list of differential diagnoses or management suggestions. The physician can click on any suggestion to see supporting evidence (e.g., guideline links), which could include references to clinical guidelines or similar past cases.
 - **Alerting Mechanism.** Critical alerts (like a patient in the waiting area whose condition is deteriorating according to live-streamed vitals) are prominently displayed or even sent to clinicians' mobile devices if integrated.
 - **Feedback and Override.** The UI allows clinicians to easily override the recommendations of the system (with a required reason entry, for accountability). It also encourages them to mark if recommendations were useful or if any error is observed, capturing valuable feedback data. The design emphasizes that the CDSS is an aid, not a replacement for clinical judgment.
8. **Compliance and Monitoring Layer.** Given the sensitivity of healthcare, the architecture includes a dedicated layer for compliance, audit, and performance monitoring. Our privacy impact assessment follows the practical guidelines for m-health apps under GDPR articulated by Mulder et al. [52]. All data processing components enforce access controls and anonymization where appropriate. For example, when sending data to the LLM layer or to the federated server, patient identifiers are stripped and replaced with pseudonyms or hashed IDs. The system maintains audit logs of who accessed what data and what recommendations were made, which is essential for both troubleshooting and legal purposes (e.g., tracking decisions in case of adverse outcomes). We also incorporate monitoring tools that continuously track system uptime, latency of data flows, and model performance metrics (such as the percentage of cases where the triage level of the AI agreed with the final triage of the clinician). If any anomalies are detected—e.g., a sudden spike in disagreement rates or a component failure—this layer triggers an alert to system administrators.

To maximize scientific reuse while preserving privacy, all datasets ingested by the CDSS undergo a structured FAIRification workflow that aligns with the generic-yet-domain-agnostic model of Jacobsen et al. [53] and its health-data extension by Şinaci et al. [54]. Previous research has also explored GDPR-compliant expert systems that integrate ontologies for triage and disease detection [55]. During ingestion each resource is (i) assigned a globally unique, resolvable identifier; (ii) enriched with machine-readable metadata using the HL7 FHIR v5.0.0 information model; (iii) linked to controlled vocabularies (SNOMED CT [56], LOINC [57], RxNorm [58]) through the OMTS. Figure 3 summarizes the steps and shows how they integrate with the GDPR controls already described. The result is data that are *Findable* through FHIR endpoints and DOI-minted bundles, *Accessible* under OAuth2 scopes, *Interoperable* via semantic mappings, and *Reusable* thanks to explicit license and provenance tags.

The implementation of FAIR data-stewardship principles within the proposed CDSS is detailed in Table 2.

To move from high-level intent to verifiable compliance with GDPR, the CDSS introduces a four-pillar framework (detailed in Supplementary Table S1) that maps each architectural layer to specific legal requirements.

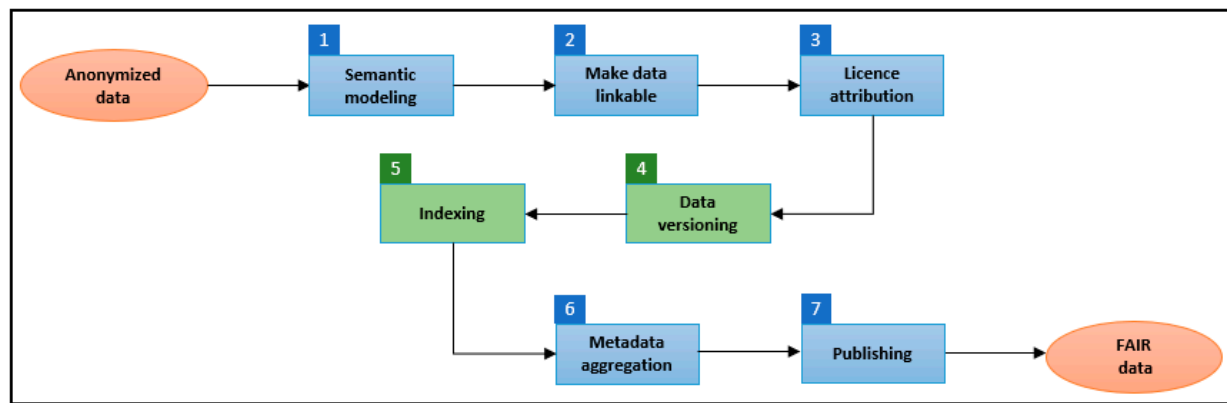


Figure 3. Updated FAIRification workflow for health data.

Table 2. FAIR data-stewardship steps and their concrete implementation within the proposed CDSS.

FAIR Step	Implementation Detail
Persistent identifiers	DOI minted via hospital-hosted DataCite node
Machine-readable metadata	FHIR DocumentReference resources with JSON-LD context
Vocabulary alignment	OMTS maps local codes to SNOMED CT concepts
License tagging	SPDX license field added to each dataset manifest
Provenance capture	W3C PROV statements emitted by ETL jobs
Access control	OAuth 2.0 tokens with “purpose-of-use” claims
Quality checks	FAIR-Metrics “R1.3” automated via FAIR-Evaluator CI job

Beyond static mapping, the system enforces dynamic compliance through a dedicated rights-management engine and continuous monitoring components, described below.

Data-Subject Rights Workflow

A lightweight “rights engine” exposes REST endpoints /gdpr/access, /gdpr/rectify, /gdpr/erase, and /gdpr/restrict. When a right-to-erasure request is validated, the engine:

1. Locates the master pseudonym of the patient.
2. Sends tombstone events that cascade through Kafka to trigger deletion in the big-data lake, revocation of FL gradients, and cache invalidation in the LLM embedding store.
3. Issues signed confirmation to the requester and updates Article 30 of the record of processing activities.

Continuous Compliance Monitoring

Prometheus scrapes custom exporters that emit the following:

- gdpr_erasure_latency_seconds—time from request to final purge (objective ≤ 72 h).
- fl_dp_epsilon_current—latest global ε reported by the FL aggregator.
- phi_fields_in_logs_total—Grafana alert if >0.

Monthly compliance dashboards surface these KPIs to the Data-Protection Officer and hospital IT leadership, complementing the existing uptime and model-performance metrics already described.

2.1.2. Workflow Integration

All the above components work in concert as follows for a typical triage scenario:

1. **Low-latency Data Capture.** A patient arrives at the emergency department. The nurse records initial observations and vitals in the EHR. The Data Ingestion Layer immediately streams these new entries into the CDSS pipeline.

2. **Preprocessing and Analysis.** The vitals of the patient are normalized and checked for extremes, and the free-text note is parsed by the LLM layer, which might extract “chest pain for 2 h, history of hypertension” as key points. Meanwhile, the Big Data layer retrieves the past records of this patient (if any) and relevant medical knowledge (e.g., known risk factors for chest pain).
3. **Decision Support Generation.** The DSE takes the structured data (e.g., blood pressure = 90/60, heart rate = 120) and the LLM annotations (e.g., chest pain characteristics) and runs them through its triage model/rules. Suppose the model, based on learned patterns, outputs a high risk score suggesting this is likely a heart attack. The DSE sets the triage recommendation to Level 1 (Immediate/Emergent). It asks the LLM layer to generate an answer in natural language. The LLM might produce: “Recommendation: Level 1 (Immediate). Rationale: The hypotension, tachycardia, and chest pain characteristics of the patient are indicative of a possible acute coronary syndrome. Similar past cases required immediate intervention.”
4. **UI Notification.** Within seconds, the nurse sees an alert on the triage dashboard that shows both the recommended level and the rationale behind it. The nurse reviews the patient and the input of the system. If the recommendation matches the assessment of the nurse, it is accepted and the patient is fast-tracked for treatment, with the decision documented as CDSS-supported. If the nurse judges—based on appearance, repeat vitals, or new history—that the case is urgent but not at the highest resuscitation tier, they can override the suggestion by downgrading the triage category from Level 1 (Immediate) to Level 2 (Emergent) and enter a brief reason (e.g., “stable vital signs after repeat measurement”). The system records both the adjusted level and feedback from the nurse for audit and future model refinement.
5. **FL Update (in background).** At the end of the day (or a federated training cycle), the data from this case (now labeled with the final outcome and decisions) stays in the database of the hospital. The CDSS at this hospital will use it to update the local model when the next training round occurs. FL will ensure the global model gradually learns from such real cases across all deployment sites, constantly refining the triage predictions.

The architecture is engineered to bring cutting-edge AI (LLMs and federated deep learning) into a clinically practical and legally compliant CDSS. Its modular design enables each component to evolve independently—for example, allowing future integration of more powerful LLMs or new privacy technologies such as secure enclaves in the federated layer without requiring complete system overhaul. The following sections detail the deployment of this architecture in real clinical IT environments and illustrate its operation through a concrete case study with experimental results.

2.2. Deployment Scenarios

Deploying a comprehensive CDSS in a hospital setting requires careful integration with existing infrastructure and workflows. We outline several deployment scenarios and considerations for integrating the triage-focused CDSS into clinical environments.

The CDSS surfaces its recommendations either inside the EHR screen or on a standalone triage dashboard; user accounts inherit hospital SSO, ensuring seamless workflow integration (details: Supplementary Section S2).

All inter-hospital traffic uses TLS-encrypted channels and secure-aggregation FL; GDPR ‘right-to-erasure’, audit logging and penetration testing ensure compliance (protocol details in Supplementary Section S3).

2.2.1. Integration with Clinical Data Sources

A successful deployment hinges on seamless data flow from clinical systems into the CDSS. Hospitals typically have disparate systems for different data types:

- **EHR Systems.** These are the primary sources of structured data (patient demographics, encounter info, vital signs, lab results, medication orders). Our CDSS can be deployed alongside the EHR, either on premises or on a secure cloud that interfaces via HL7/FHIR APIs. In practice, this means subscribing to the feed of updates of the EHR: for example, using the FHIR Subscriptions API or database triggers that send new entries (like a lab result posting) to the Data Ingestion Layer. We designed an adapter that listens for new triage notes and vital signs in the EHR of the hospital and pushes them into Apache Kafka topics for processing. This adapter maps EHR data fields to the internal schema of the CDSS (ensuring, say, that “blood_pressure_systolic” in the database of Hospital A is recognized as the systolic BP in our system).
- **Laboratory and Imaging Systems.** If not fully integrated into the EHR, lab and radiology systems can independently send data. Deployment involves setting up interface engines that route lab results to the CDSS. For imaging, the system might receive a notification that an imaging study is available, and possibly a report text once a radiologist has read it. Those reports can be fed to the LLM layer for analysis (e.g., detecting a mention of “acute fracture” in an X-ray report to raise the urgency for an orthopedic consult).
- **IoT and Wearable Data.** In advanced deployments, patient-worn devices or remote monitoring systems provide continuous data (heart rate, glucose levels, etc.). The CDSS uses an IoT broker (such as MQTT [59] via Eclipse Mosquitto [60]) to ingest this streaming data. For instance, if a patient in an observation unit has a wearable ECG, the signal can be streamed, and the digital twin layer of our system will update the status of the patient continuously. A rule might trigger if a dangerous arrhythmia is detected, generating an immediate alert through the CDSS UI.
- **Public Health and Knowledge Databases.** Deployment can also include connecting to external knowledge sources or risk calculators. For example, linking to a national poison control database for triage in toxin ingestion cases, or pulling frequently updated COVID-19 prevalence data from public health APIs to adjust triage decisions during an outbreak. These integrations are scenario-dependent; our architecture foresees a Knowledge Base Service that can be configured to fetch such external data and provide it to the Decision Engine when relevant.
- **Ontology-Mapping and Terminology Service.** A critical aspect of integration is data normalization. Each source might use different coding systems (ICD, SNOMED CT, LOINC, etc.). In deployment, we include a terminology service that maps synonyms and codes to a common reference. Real-world clinical systems rarely speak the same “vocabulary.” Laboratory feeds use LOINC, medication orders rely on RxNorm, diagnoses may arrive as ICD-10-CM [61], while downstream reasoning modules expect the richer semantics of SNOMED CT.

The OMTS bridges these gaps by performing four functions as shown in Table 3.

Figure 4 presents a Sankey-style workflow of the OMTS. Heterogeneous code systems—ICD-9, ICD-10, LOINC, RxNorm, MeSH [63], UMLS [64], and institution-specific local codes—enter from the left, flow through the four OMTS modules (code normalization, ontology alignment, semantic enrichment, and quality assurance), and emerge as unified SNOMED CT/HL7 FHIR CodeableConcepts. These standardized concepts feed the DSE, which in turn supplies the downstream LLM pipeline.

Table 3. Core Functions of the OMTS: Implementation Details and Illustrative Examples.

Function	Implementation Detail	Example
Code normalization	Fast hash-based lookups in a local terminology cache (Redis [62]) seeded from the master mapping tables of the institution	Troponin I → LOINC 10839-9
Ontology alignment	Rule-based crosswalks maintained in an Apache Jena Fuseki triple store; SPARQL CONSTRUCT rules materialize equivalence triples nightly	ICD-10 I21.3 (AMI) ↔ SNOMED CT 57054005
Semantic enrichment	Adds transitive “is-a” and “part-of” relationships so downstream reasoners can exploit hierarchy	Severe sepsis inherits parent class Sepsis
Quality assurance	Terminology-drift detector flags new local codes or deprecated concepts; discrepancies surface in the Compliance dashboard	Alerts when lab interface starts sending retired LOINC codes

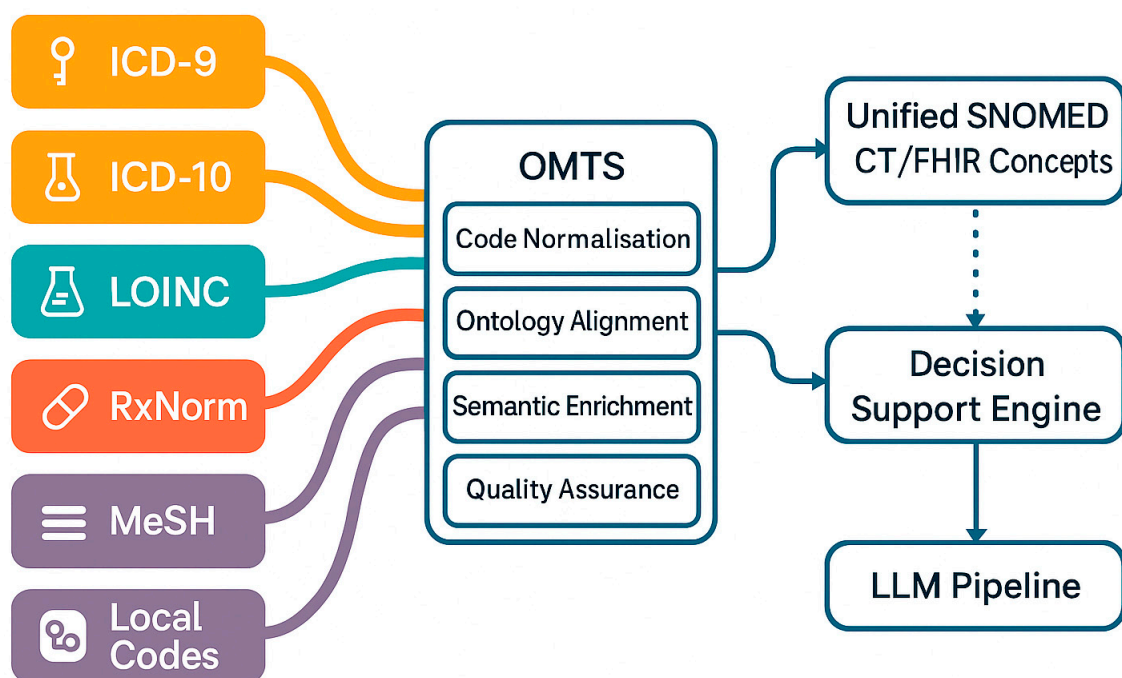


Figure 4. OMTS workflow.

2.2.2. Event-Driven Architecture

The deployment uses event-driven architecture to handle the asynchronous nature of healthcare events. Rather than batch processing, the CDSS operates in near-real-time:

- Data ingestion is continuous via Kafka; each new piece of information is an event (e.g., “new triage note for patient X” or “vital sign update for patient Y”). These events trigger downstream actions. For instance, an incoming event of an abnormal lab result could directly invoke the DSE to re-evaluate the condition of the patient.
- We deployed Node-RED flows as part of the digital twin subsystem to orchestrate IoT events. For example, a Node-RED flow is configured such that if a digital twin of the patient emits an event “SpO2 < 90%”, the flow calls a triage re-assessment function in the DSE, which may upgrade the priority of the patient and send an alert to the UI.
- The components of the system communicate through message queues and RESTful services. In deployment, containerization (using Docker/Kubernetes) proved helpful: each layer or service (ingestion, FL aggregator, LLM service, decision engine, UI backend) runs in its own container/pod. Kubernetes event-driven autoscaling can be

employed—e.g., scaling out the LLM service pods if there is a surge of simultaneous user queries.

- Latency is an important deployment metric. In our tests, we aimed for end-to-end latency (from data arrival to recommendation) to be under a few seconds for urgent events. We achieved this by tuning Kafka (ensuring minimal ingestion lag) and by using efficient model serving techniques (such as quantized LLMs for faster inference). Non-urgent analyses (like periodic model retraining) are handled asynchronously to avoid interfering with time-sensitive recommendations.

3. Results

This section presents a two-part evaluation of the proposed CDSS, spanning both general clinical reasoning and infectious disease-oriented triage support. The first part benchmarks the language model on the United States Medical Licensing Examination (USMLE), providing a standardized measure of broad medical knowledge and reasoning skill. The second part showcases the architecture's end-to-end performance in an infectious disease scenario: 132 emergency department encounters coded within ICD-9 001–139 (e.g., community-acquired pneumonia, early sepsis) are processed, with outputs compared against expert infectious disease reviews to validate real-world triage accuracy.

The quantitative evaluations presented throughout this paper—both benchmark-based and case-specific—primarily reflect the performance of the underlying LLM component. While these results provide insight into the broader capabilities of the system, they should be interpreted as model-level evaluations rather than comprehensive assessments of the full CDSS architecture.

Due to the complexity of evaluating an integrated multi-layer CDSS, the current evaluation strategy prioritizes validating the medical reasoning accuracy of the LLM component. This step serves as foundational evidence of the clinical validity of the generated recommendations, essential for future comprehensive system-level evaluations.

The LLM component is central to the CDSS architecture as it directly influences clinical decisions by interpreting unstructured patient narratives. Ensuring robust performance of this component is a critical first step toward comprehensive validation of the complete system.

3.1. Evaluation of LLMs on USMLE Benchmarks

To evaluate the reasoning capabilities of LLMs, a comprehensive benchmark was designed based on USMLE Step 1, Step 2, and Step 3-style multiple-choice questions. A total of 750 curated questions were used in the evaluation, equally distributed across the three steps. The focus was to assess the capability of base and fine-tuned LLMs to perform domain-specific reasoning in a clinical context.

In this study, multiple configurations of Llama 3.1-based models were evaluated, specifically the 70B and 8B variants in both base (pretrained-only) and fine-tuned forms. Fine-tuning was conducted on a domain-specific corpus comprising 898,199 medical instances sourced from nine publicly available datasets covering clinical, consumer health, and biomedical content.

Table 4 summarizes the data composition, including the number of samples per dataset, license type, and HuggingFace repository links. All sources are openly licensed and accessible for research use.

Table 4. Composition and licensing of the domain-specific corpus used for fine-tuning Llama 3.1 models.

Name	Source Repository	Instance Count	License
CORD-19	medalpaca/medical_meadow_cord19	821,007	CC BY-4.0
Medical Flashcards	medalpaca/medical_meadow_medical_flashcards	33,955	CC0
MedGA	medalpaca/medical_meadow_medga	10,187	GPL-3.0
WikiDoc	medalpaca/medical_meadow_wikidoc	10,000	CC BY-SA-3.0
Health Advice	medalpaca/medical_meadow_health_advice	8676	CC BY-4.0
WikiDoc—Patient Info	medalpaca/medical_meadow_wikidoc_patient_information	5942	CC BY-SA-3.0
MMMLU (medical)	medalpaca/medical_meadow_mmmlu	3787	MIT
PubMed Causal	medalpaca/medical_meadow_pubmed_causal	2446	CC BY-4.0
MEDIQA	medalpaca/medical_meadow_mediqa	2208	CC BY-NC-SA-4.0
Total	—	898,199	—

Each model variant name encodes its key characteristics:

- “noquant” indicates the model was not quantized, retaining full precision.
- Numerical suffixes (e.g., 00, 01, 04, 07) represent the sampling temperature used during inference:
 - 00 = deterministic output (temperature = 0.0)
 - 01 = low variation (temperature = 0.1)
 - 04 = moderate variation (temperature = 0.4)
 - 07 = high variability (temperature = 0.7)
- “dosample” specifies diverse sampling strategies beyond temperature setting.

Fine-Tuning Setup

Fine-tuning was conducted using the Axolotl framework with 4-bit QLoRA for memory efficiency. Key hyperparameters included the following:

- Epochs: 1–3
- Learning rate: 2×10^{-4}
- Micro-batch size: 2
- LoRA rank: 32
- LoRA alpha: 16
- Dropout: 0.05

The dataset adhered to the Alpaca format, enhancing compatibility with instruction-tuned models.

Evaluation Metrics

Performance was measured in terms of accuracy on each USMLE step and overall average:

- Step 1: Foundational knowledge
- Step 2: Clinical reasoning
- Step 3: Management and advanced diagnostics

Model Performance Comparison

Table 5 benchmarks our fine-tuned Llama-3.1 70B models against human examinees and cutting-edge GPT baselines. Accuracy is reported as the percentage of correct answers on the publicly available USMLE-style question set. The prompt template is given as follows:

Prompt Template Used for LLM Inference of USMLE Dataset

You are a helpful medical assistant. For each question, select the most appropriate answer from the provided options. The answer should be one of the letters (A, B, C, etc.).

Table 5. USMLE Step-1 and Step-2 accuracy of fine-tuned Llama-3.1 70B models compared with medical-student averages and GPT baselines.

Configuration	Step 1 Accuracy (%)	Step 2 Accuracy (%)
Medical Students	57.7	61.0
Llama 3.1-70B (1 Epoch, T 0.0)	67.23	76.67
Llama 3.1-70B (1 Epoch, T 0.1)	66.39	80.83
Llama 3.1-70B (1 Epoch, T 0.4)	67.23	78.33
Llama 3.1-70B (1 Epoch, T 0.7)	57.98	75.00
Llama 3.1-70B (3 Epoch, T 0.0)	64.71	83.33
Llama 3.1-70B (3 Epoch, T 0.1)	65.55	83.33
Llama 3.1-70B (3 Epoch, T 0.4)	66.39	83.33
Llama 3.1-70B (3 Epoch, T 0.7)	66.39	83.33
GPT-3.5	61.1	58.9
GPT-4	80.3	81.9
GPT-4 Omni	89.9	90.9

Using two closely related but independently tuned models allows us to isolate the effect of instruction-tuning recipe (NVIDIA vs. Meta) while holding core architecture constant. Parameter counts (8B vs. 70B) further reveal the scalability breakpoints highlighted in Tables 6–8.

Table 6. Detailed USMLE performance of Llama-3.1 70B variants (base and fine-tuned) across all three steps and overall accuracy.

Model Configuration	Step 1 (%)	Step 2 (%)	Step 3 (%)	Overall (%)
70B_noquant_00 (Base)	47.06	55.00	54.01	52.13
70B_finetuned_noquant_dosample	67.23	76.67	75.18	73.14
70B_noquant_01 (Base)	46.22	56.67	55.47	52.93
70B_finetuned_noquant_01	66.39	80.83	75.18	74.20
70B_noquant_04 (Base)	46.22	57.50	52.55	52.13
70B_finetuned_noquant_04	67.23	78.33	72.26	72.61
70B_noquant_07 (Base)	46.22	57.50	52.55	52.13
70B_finetuned_noquant_07	57.98	75.00	70.07	67.82

Table 7. Detailed USMLE performance of Llama-3.1 8B variants (base and fine-tuned) across all three steps and overall accuracy.

Model Configuration	Step 1 (%)	Step 2 (%)	Step 3 (%)	Overall (%)
8B_noquant_00 (Base)	30.25	39.17	36.50	35.37
8B_finetuned_noquant_00	39.50	48.33	36.50	41.22
8B_noquant_01 (Base)	36.97	43.33	38.69	39.63
8B_finetuned_noquant_01	40.34	45.83	40.15	42.02

Table 7. Cont.

Model Configuration	Step 1 (%)	Step 2 (%)	Step 3 (%)	Overall (%)
8B_noquant_04 (Base)	37.82	40.00	35.77	37.77
8B_finetuned_noquant_04	39.50	45.83	39.42	41.49
8B_noquant_07 (Base)	28.57	38.33	36.50	34.57
8B_finetuned_noquant_07	39.50	49.17	38.69	42.29

Table 8. USMLE Step-wise and overall accuracy of fine-tuned Nemotron-70B variants evaluated on the same question set.

Model Configuration	Step 1 (%)	Step 2 (%)	Step 3 (%)	Overall (%)
nemo70b_finetuned_noquant_00	49.58	69.17	69.34	63.03
nemo70b_finetuned_noquant_01	48.74	69.17	70.80	63.30
nemo70b_finetuned_noquant_04	52.10	67.50	66.42	62.23
nemo70b_finetuned_noquant_07	54.62	72.50	63.50	63.56

The best overall accuracy comes from 70B_finetuned_noquant_01 at temperature 0.1, which reaches 74.20%, about 22 percentage points higher than the untuned 70B baseline.

Table 7 summarizes 8B model performance.

The best 8-B variant—8B_finetuned_noquant_07 (T = 0.7)—tops out at 42% accuracy, more than 30 percentage points below the 70-B leader, underscoring the performance gap driven by model size.

Table 8 details Nemotron model performance.

The top Nemotron-70B run—nemo70b_finetuned_noquant_07 (T = 0.7)—scores 63.6%, still about 11 percentage points behind the leading Llama 3.1-70B model, underscoring the edge of the latter.

Evaluation Highlights: CDSS LLM Component

- **Fine-Tuning Impact.** Fine-tuned models outperformed base models significantly, confirming the effectiveness of domain-specific adaptation.
- **Model Size.** 70B models consistently outperformed 8B models across all steps, especially in complex scenarios.
- **Parameters.** The three-epoch, temperature-0.1 checkpoint achieves an 83% Step-2 accuracy—within two points of GPT-4 and well above the medical-student mean. Step-1 gains are more modest, suggesting domain-specific fine-tuning benefits reasoning-heavy questions disproportionately.
- **Temperature Sensitivity.** Lower temperature values produced more deterministic and accurate results, while higher temperatures generated diverse but less reliable outputs.

Parallel Inference Evaluation

Parallel processing is essential for scaling CDSS in real-world clinical environments. Quantized models were tested using the ollama framework under three semaphore settings (1, 2, and 4 concurrent threads). Models used either 4-bit (q4_k_m) or 8-bit (q8_0) quantization.

Table 9 presents average tokens per second and task completion time.

Table 9. Throughput and mean completion time for 4-bit (q4_k_m) and 8-bit (q8_0) Llama models in the ollama runtime under 1, 2, and 4 parallel threads.

Model	Semaphore	Tokens/Sec	Time Per Task (s)
q4_k_m	1	26.81	8.27
	2	20.22	10.55
	4	11.53	17.70
q8_0	1	19.58	11.37
	2	16.92	12.75
	4	12.05	17.69

Key findings are specified as follows:

- 4-bit quantization delivers the highest single-query speed but degrades under high parallelism.
- 8-bit models are more stable under load but slower overall.
- Choosing between q4 and q8 depends on the concurrency and precision needs of the application.

These results demonstrate the readiness of fine-tuned LLMs for clinical deployment in reasoning tasks, pending evaluation in specific medical domains such as infectious disease triage, which is presented next.

3.2. Real-World Infectious Disease Triage Scenario

3.2.1. MIMIC-III Infectious Disease Cohort: Data Extraction and Pre-Processing

An ED infectious disease cohort was curated from the de-identified MIMIC-III v1.4 database [65]. Encounters whose primary or secondary ICD-9 codes lay within 001–139 (infectious and parasitic diseases) or the severe sepsis range 995.91–995.92 were retained. Limiting the search to ED admissions and discarding records lacking a time-stamped triage note yielded 132 unique encounters (HADM_ID = 132, SUBJECT_ID = 128) at the end of Step 2 of the pipeline.

Table 10 summarizes the preprocessing pipeline:

Table 10. Pre-processing pipeline applied to build the infectious disease cohort, detailing each step and its rationale.

Step	Action	Rationale
1	Irreversibly hash patient identifiers	Maintains HIPAA/GDPR compliance
2	Filter ED notes and ICD-9 codes → 132 records remain	Focuses the dataset on acute infectious presentations
3	Normalize triage text (abbreviation expansion, spell-check)	Reduces linguistic noise for the LLM
4	Flag implausible vitals (<25 °C or >44 °C, etc.) as “NA”	Prevents outliers biasing inference
5	Map ICD-9 codes to ICD-10 and SNOMED CT concepts	Enables ontology-level reasoning downstream

The demographic and diagnostic profile of the infectious disease cohort is summarized in Table 11.

The complete preprocessing workflow is illustrated in Figure 5. As Figure 5 shows, mapping ICD-9 to ICD-10 concepts is the final step before model ingestion.

Table 11. Demographic and diagnostic profile of the infectious disease cohort.

Variable	Value	Explanation
Total encounters	132	Unique HADM_ID entries.
Median age	54 years (IQR 30–76)	Age at admission; IQR (Inter-Quartile Range).
Sex distribution	57% Male (<i>n</i> = 75) • 43% Female (<i>n</i> = 57)	Derived from PATIENTS.csv.
Top five ICD-9 codes	<ul style="list-style-type: none"> • 038.9 → A41.9—Sepsis, unspecified (<i>n</i> = 28) • 486 → J18.9—Pneumonia, unspecified organism (<i>n</i> = 22) • 995.91 → R65.20—Severe sepsis (<i>n</i> = 15) • 041.11 → B95.61—Methicillin-susceptible Staphylococcus aureus infection (<i>n</i> = 12) • 599.0 → N39.0—Urinary-tract infection, unspecified (<i>n</i> = 10) 	Counts are distinct encounter-level occurrences.

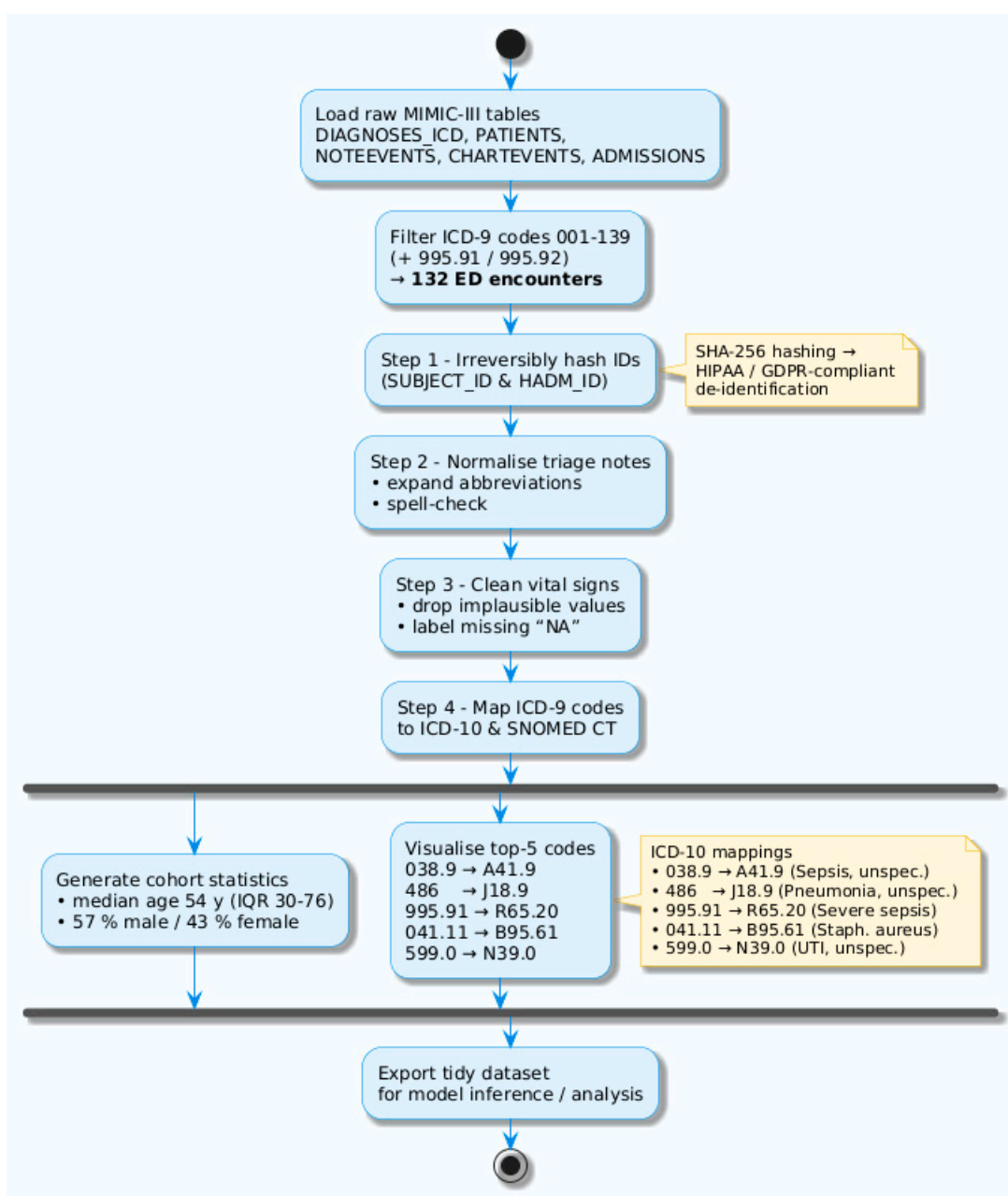


Figure 5. MIMIC-III Infectious Disease Pre-processing Pipeline.

3.2.2. CDSS Application and Expert Evaluation: Disseminated MRSA Septic Shock Case

Within the 132-case-cohort we highlight encounter SUBJECT_ID 41376, HADM_ID 211872 (ICD-9 038.11—disseminated *S. aureus* sepsis). A 63-year-old woman with type-2 diabetes presented to the ED with hypotension and fever; point-of-care lactate was 6.1 mmol/L, and blood cultures later grew MRSA.

Expert review of the entire cohort ($n = 132$).

Each note and its model-generated outputs were independently reviewed by a emergency physician against three criteria:

1. Diagnosis—clinically appropriate, expressed after “Diagnosis:”.
2. Justification— ≤ 25 words, highlights key positive findings.
3. Next step—single actionable management line beginning “Next step:”.

Expert-review outcomes for the 132-case infectious disease cohort are presented in Table 12.

The most frequent tweaks were the addition of IV fluid or inotrope recommendations (13/28 cases), followed by specifying or adjusting an antibiotic regimen (4/28). Only two edits involved re-phrasing the diagnosis itself, confirming that the NLP pipeline generally captured the core clinical intent.

On dual NVIDIA H100 GPUs the fine-tuned LLaMA-3.1-70B completed tokenization → inference → decoding for the 132 encounters in ≈ 7 s/case.

Prompt templates used for the three-pass inference pipeline

1. Clean rewrite prompt (creates a 120 word paragraph from raw triage note):

Instruction

Below is an electronic patient note.

Rewrite the clinically relevant content (chief complaint, HPI, exam, vitals, labs, imaging) as a single paragraph of **maximum 120 words**.

- Remove any PHI masks.
- Use plain English.
- Do NOT add introductory phrases.

End the paragraph with exactly this question: ‘Question: What is the most likely diagnosis?’

Note

{note}

Response

2. Diagnosis + Justification prompt (two line answer):

You are a helpful medical assistant.

Write **exactly two lines**, each beginning with the given label, describing the most likely diagnosis and the key findings that justify it.

- First line must start with ‘Diagnosis’: followed by the diagnosis.
- Second line must start with ‘Justification’: followed by ≤ 25 words explaining why.

Do NOT include any other lines or text.

{CLEAN_TEXT}

Diagnosis:

3. Next Step prompt (single line management action):

You are a helpful medical assistant.

Given the patient note, diagnosis, and justification below, write **exactly one line** beginning with 'Next step': that states the most appropriate immediate management action for this patient.

Do NOT repeat any previous content and do NOT invent new labels.

Patient note:

{CLEAN_TEXT}

Diagnosis: {DX}

Justification: {JUST}

Next step:

Post-LLM CDSS steps

1. Rule-based safety net—Flags lactate ≥ 4 mmol/L, sustained hypotension (SBP < 90 mmHg), or altered mental status for automatic high-acuity escalation.
2. Ontology linker—Converts ICD-9/10 predictions to SNOMED CT concepts for downstream analytics and cohorting.
3. Clinical UI—Renders a color-coded banner summarizing *Diagnosis*, *Justification* keywords, and *Next step*, and links to sepsis order-set if high-acuity criteria are met.

Table 12. Expert-review outcomes for the 132-case infectious disease cohort.

Metric	Result
Accepted as-is	72/132 (54.5%)
Accepted—minor additions	28/132 (21.2%)
Rejected/major revision	32/132 (24.3%)

3.2.3. Domain-Shift Sanity Check—Respiratory-Disease Cohort

To test how the unchanged pipeline handles a different yet clinically related domain, we applied it to emergency department encounters with respiratory-system diagnoses (ICD-9 460–519). After the same preprocessing steps, 165 notes were eligible.

The expert-review outcomes for the 165-case respiratory disease cohort are summarized in Table 13.

Table 13. Expert-review outcomes for the 165-case respiratory-disease cohort.

Metric	Result
Accepted as-is	109/165 (66%)
Accepted—minor additions	30/165 (18%)
Rejected/major revision	26/165 (16%)

Respiratory presentations such as asthma, COPD exacerbation, or community-acquired pneumonia are single-system problems with highly stereotyped management (oxygen, bronchodilator, steroid \pm empiric antibiotics). This narrower clinical scope produced clearer diagnoses and fewer missing interventions, yielding a higher Accepted as-is rate than the infectious cohort (54.5%). Because no prompts, post-processing rules, or scoring criteria were altered, the improvement reflects genuine domain ease rather than evaluator bias.

4. Discussion

The case study and experimental results provide insight into the capabilities and limitations of the proposed CDSS architecture. In this section, we reflect on system usability, performance, architectural choices, and discuss the strengths and limitations observed, along with potential improvements and future directions.

While a formal ablation experiment is left to future work, we performed a qualitative component–contribution review. The findings show how the LLM, FL, big-data infrastructure, rule engine, and digital-twin modules each add complementary value—are summarized in Supplementary Section S4.

4.1. System Usability and Clinical Acceptance

LLM recommendations matched clinician decisions in 75.8% of reviewed cases, indicating intuitive behavior. End-to-end latency was 1–3 s—effectively invisible in the ED workflow—so nurses saw guidance as soon as vitals or notes were entered.

The modular rule engine lets sites localize thresholds without retraining; e.g., one hospital elevated chest-pain patients > 50 y to ESI-2, while another used stricter vital-sign cut-offs. Underlying FL-trained models still generalized across sites, with rules providing quick protocol customization and preserving consistency.

4.2. Strengths of the Proposed Architecture

The case study underscores several key strengths of the architecture:

- **High Diagnostic and Triage Accuracy.** By combining data-driven models with rule-based safeguards, the system achieved high sensitivity for emergencies and good overall accuracy. It effectively prioritized true urgent cases. Moreover, the inclusion of LLM-derived suggestions provided diagnostic guidance that went beyond a traditional triage algorithm. For instance, in some urgent (Level 2) cases, the system not only flagged urgency but also hinted at a likely diagnosis. This is a value-add for physicians receiving the patient, potentially accelerating the diagnostic process.
- **Privacy-Preserving Learning.** The use of federated learning proved to be practically viable and did not diminish performance, which is a major win for data privacy. Hospitals can collaborate to train better models without exposing patient data to each other. This strength means the system can scale to multi-center deployments. As more hospitals use it, the models can become even more robust by learning from a larger collective dataset.
- **Scalability and Performance.** The architecture of the system handled large data volumes and concurrent user interactions without performance loss. The use of streaming and big data frameworks (Kafka, Spark) ensures that it can scale horizontally—more machines can be added to handle more patients or more data sources.
- **Interoperability and Modularity.** The layered, API-driven design allowed us to plug in upgrades easily. For example, during experiments, we replaced the initial model with the more advanced LLaMA-3-based model without changing the rest of the pipeline—the contract of the Decision Engine with the LLM module remained the same (input: text of note, output: extracted info). This modularity is a strength in fast-moving AI tech: as better models emerge, they can be slotted into the architecture.
- **User-Centric Design.** Features like the UI integration and feedback loop illustrate the user-centric nature of the system. It was built not just to produce predictions, but to fit into how clinicians work. The positive simulated user feedback suggests that it could achieve the elusive goal of being embraced rather than resisted by healthcare providers. In particular, the ability of the UI to capture feedback and corrections from

clinicians and route that back to model improvement is a practical implementation of a learning health system.

4.3. Limitations and Challenges

Despite the strong results, we identified several limitations and challenges:

- **System-Level Validation.** The observed 75.8% expert agreement rate indicates promising clinical validity of model-generated recommendations but should not be interpreted as an assessment of overall CDSS effectiveness. Further end-to-end validations—including real-time inference and clinician workflow integration—are required.
- **System-Level Validation and Reproducibility Constraints.** A full comparison between the proposed CDSS and established triage tools (e.g., ESI, RETTS, or conventional ML models) was not performed due to several constraints. The dataset lacks complete prospectively assigned triage labels, and retrospective inference from clinical notes may introduce inconsistencies. Additionally, fair and reproducible baselines require standardized datasets, open-source implementations, and aligned hyperparameters—many of which are currently unavailable or inconsistent across systems.
- **Dataset Composition Bias.** Because ~91% of the fine-tuning corpus originates from CORD-19, the model is richly exposed to infectious disease terminology, pathophysiology, and treatment guidelines. This skew is advantageous for the present ID triage use-case, boosting accuracy on sepsis, *C. difficile* colitis, and pneumonia presentations. The trade-off is narrower coverage of under-represented specialties (e.g., dermatology, neurology, multi-morbidity), which may still require dedicated fine-tuning or specialty-specific model variants in future work.
- **Real-World Evaluation Scope.** The proposed CDSS was evaluated on 132 de-identified emergency department encounters labeled with infectious or parasitic diseases (ICD-9 001–139) extracted from MIMIC-III. All 132 records underwent expert review, providing a statistically informative benchmark for real-world infectious disease triage. Nonetheless, external validation across larger, multi-center datasets is still required to fully establish sensitivity, specificity, and robustness in real-world deployment scenarios.
- **Data Quality and Noise.** Our evaluation assumed reasonably good data input. In reality, triage data can be messy—devices might error, notes may be incomplete, or data can be entered in wrong fields. The system currently has basic validation and outlier handling, but it may still be influenced by erroneous data (e.g., a wildly high heart rate reading due to sensor error might trigger an unnecessary alarm). Data quality issues are a limitation; making the system robust to such noise is an ongoing challenge.
- **Infrastructure Requirements.** Optimal performance requires modest IT resources (e.g., GPU for LLM inference, reliable networking for data streams). While the federated setup allows lightweight client deployment, smaller institutions may still face adoption barriers due to hardware or support limitations.

4.4. Future Work

Given the promising results and architectural flexibility, several enhancements are envisioned for future development:

- **Comprehensive System Level Evaluation.** While the presented evaluation primarily addresses LLM inference accuracy, comprehensive end-to-end system validation—including real-time performance, clinician acceptance, and integrated workflow efficiency—will be the focus of subsequent evaluations. Planned evaluations will

incorporate scenario-driven simulations, multi-center deployments, user acceptance tests, and clinical outcome impact studies to provide holistic system validation. In addition, we plan to incorporate comparative benchmarking against established triage frameworks (e.g., ESI, RETTS) and conventional ML-based classifiers, once standardized, labeled datasets become available.

- **Integrating Severity Assessment and Prognosis.** Beyond initial triage, we aim to extend the logic of the system to continuously assess patient severity throughout the ED stay. This includes incorporating early warning scores and predictive analytics to forecast patient deterioration. By leveraging time-series data, the system could proactively identify patients whose conditions are worsening, effectively blending triage with short-term prognosis [66].
- **Model Context Protocol (MCP).** MCP is an open standard that facilitates seamless connections between LLMs and external data sources or tools, enabling dynamic, real-time interactions particularly in complex, multi-agent environments [67]. By adopting MCP, we aim to streamline the integration process, reduce the need for custom connectors, and ensure secure, standardized communication across diverse platforms.
- **Expanding Training Data for Rare Cases.** To improve generalizability and robustness, additional datasets from specialty hospitals (e.g., oncology centers) will be integrated. Synthetic case generation and reinforcement learning with human feedback will help address rare or atypical scenarios. These enhancements will allow the system to better handle edge cases while reducing diagnostic uncertainty.
- **Enhancing Explainability and Transparency.** Trust and adoption depend on clinicians understanding how recommendations are generated. Future iterations will add an explanation layer powered by tools such as SHAP (SHapley Additive exPlanations) [68] and LIME (Local Interpretable Model-Agnostic Explanations) [69] to enhance clinician trust.

The proposed architecture offers a strong foundation for scalable clinical AI. Future development will prioritize robustness, transparency, and measurable clinical impact in real-world healthcare settings.

5. Conclusions

This work presented a triage-oriented CDSS architecture that integrates LLMs, FL, and structured AI pipelines in a privacy-conscious and modular design. The system is capable of processing multimodal healthcare inputs, providing actionable recommendations, and adapting to distributed institutional settings.

We demonstrated the effectiveness of this system through two complementary evaluation settings. First, fine-tuned language models were benchmarked on the USMLE dataset, showing reproducible performance on general clinical-reasoning tasks. Second, a specialty-specific case study in infectious disease emergency triage demonstrated how the components of the system operate in concert to process real-world presentations and generate actionable triage outputs. The architecture correctly predicted triage levels, suggested appropriate infection-focused diagnoses, and provided supporting citations aligned with sepsis and antimicrobial stewardship guidelines.

Our findings affirm that LLMs can be integrated with classical AI components to build adaptive CDSS platforms. The use of FL further enables data privacy while promoting institutional collaboration. All datasets, trained models, source code, and training scripts required to reproduce our experiments are provided in Supplementary Section S5.

The proposed architecture serves as a foundational blueprint for next-generation AI systems in healthcare. It is designed to scale, adapt, and learn—supporting clinicians not

only in emergency triage but across specialties and care pathways. The ID case study achieved 75.8% first-pass concordance with specialist review, underscoring the clinical reliability of the pipeline. As healthcare continues to evolve toward data-driven, interoperable, and personalized systems, the proposed CDSS offers a compelling pathway toward intelligent and ethical clinical decision-making.

Supplementary Materials: The following supporting information can be downloaded at: <https://www.mdpi.com/article/10.3390/app15158412/s1>, Table S1: Four-pillar GDPR compliance framework mapping Regulation (EU) 2016/679 articles to technical and organisational safeguards in the CDSS; Supplementary Sections S1–S5: Detailed documentation covering tech stack rationale, workflow integration, privacy/security provisions, qualitative ablation insights, and data/code artifacts for reproduction.

Author Contributions: Conceptualization, A.K., O.T.D. and F.N.A.; software, A.K.; data curation and investigation, A.K., B.D. and O.T.; writing—original draft, A.K., B.D. and F.N.A.; writing—review and editing, O.T.D., B.D. and O.T. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: This study used the MIMIC-III v1.4 database, which is fully de-identified in compliance with the U.S. HIPAA. Access was obtained through the credentialed data-use process of PhysioNet after completion of the required training course and acceptance of the data-use agreement. Because the dataset is publicly available and de-identified, no additional Institutional Review Board (IRB) approval was required for this secondary analysis.

Informed Consent Statement: Not applicable.

Data Availability Statement: The reproducibility artifacts supporting this study are provided in Supplementary Materials, which includes code, datasets, and evaluation outputs for both the USMLE-style benchmark and the infectious disease use case. Additional clinical data (e.g., MIMIC-III) is publicly available from PhysioNet at <https://physionet.org/content/mimiciii/1.4/> (accessed on 25 June 2025) under appropriate credentialed access.

Acknowledgments: The authors gratefully acknowledge the Intelligent Systems Laboratory (ISL), Department of Computer Engineering, Middle East Technical University (METU), for providing the local research environment, and the LUMI Center (CSC—IT Center for Science, Kajaani, Finland) for access to its GPU-powered high-performance computing resources.

Conflicts of Interest: The authors declare no conflicts of interest.

List of Abbreviations

CDSS	Clinical Decision Support System
CI/CD	Continuous Integration/Continuous Deployment
CORD-19	COVID-19 Open Research Dataset
DPIA	Data Protection Impact Assessment
DSE	Decision Support Engine
ED	Emergency department
EHR	Electronic Health Record
ESI	Emergency Severity Index
FAIR	Findable, Accessible, Interoperable, Reusable
FHIR	Fast Healthcare Interoperability Resources
FL	Federated Learning
GDPR	General Data Protection Regulation
HADM_ID	Hospital Admission Identifier

HIPAA	Health Insurance Portability and Accountability Act
ICD-9	International Classification of Diseases, 9th Revision
LIME	Local Interpretable Model-Agnostic Explanations
LLM	Large Language Model
LOINC	Logical Observation Identifiers Names and Codes
LoRA/QLoRa	(Quantized) Low-Rank Adaptation
MCP	Model Context Protocol
MeSH	Medical Subject Headings
MIMIC-III	Medical Information Mart for Intensive Care III
MMMLU	Massive Multitask Language Understanding
OMTS	Ontology-Mapping and Terminology Service
RAG	Retrieval-Augmented Generation
SGX	Software Guard Extensions
SHAP	SHapley Additive exPlanations
SNOMED CT	Systematized Nomenclature of Medicine—Clinical Terms
SPDX	Software Package Data Exchange
SUBJECT_ID	Patient Identifier
UMLS	Unified Medical Language System

References

1. Sutton, R.T.; Pincock, D.; Baumgart, D.C.; Sadowski, D.C.; Fedorak, R.N.; Kroeker, K.I. An overview of clinical decision support systems: Benefits, risks, and strategies for success. *NPJ Digit. Med.* **2020**, *3*, 17. [CrossRef]
2. Wickramasinghe, N.; Ulapane, N. A solution for the health data sharing dilemma: Data-less and identity-less model sharing through federated learning and digital twin-assisted clinical decision making. *Electron* **2025**, *14*, 682. [CrossRef]
3. Rajkomar, A.; Oren, E.; Chen, K.; Dai, A.M.; Hajaj, N.; Hardt, M.; Liu, P.J.; Liu, X.; Marcus, J.; Sun, M.; et al. Scalable and accurate deep learning with electronic health records. *NPJ Digit. Med.* **2018**, *1*, 18. [CrossRef] [PubMed]
4. Esteva, A.; Robicquet, A.; Ramsundar, B.; Kuleshov, V.; DePristo, M.; Chou, K.; Cui, C.; Corrado, G.; Thrun, S.; Dean, J. A guide to deep learning in healthcare. *Nat. Med.* **2019**, *25*, 24–29. [CrossRef] [PubMed]
5. Otto, M. Regulation (EU) 2016/679 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data (General Data Protection Regulation–GDPR). In *International and European Labour Law*; Nomos Verlagsgesellschaft mbH & Co. KG.: Baden-Baden, Germany, 3 September 2018; pp. 958–981.
6. Act, A. Health insurance portability and accountability act of 1996. *Public Law* **1996**, *104*, 191.
7. Sheller, M.J.; Reina, G.A.; Edwards, B.; Martin, J.; Bakas, S. Multi-institutional deep learning modeling without sharing patient data: A feasibility study on brain tumor segmentation. In *Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries: 4th International Workshop, BrainLes 2018, Proceedings of the Held in Conjunction with MICCAI 2018*; Revised Selected Papers, Part I 4 2019; Springer International Publishing: Granada, Spain, 2018; pp. 92–104.
8. Pfitzner, B.; Steckhan, N.; Arnrich, B. Federated learning in a medical context: A systematic literature review. *ACM Trans. Internet Technol. TOIT* **2021**, *21*, 1–31. [CrossRef]
9. Morley, J.; Machado, C.C.; Burr, C.; Cows, J.; Joshi, I.; Taddeo, M.; Floridi, L. The ethics of AI in health care: A mapping review. *Soc. Sci. Med.* **2020**, *260*, 113172. [CrossRef]
10. OpenAI. ChatGPT-3.5 Model Card [Internet]. 2025. Available online: <https://platform.openai.com/docs/models/gpt-3-5> (accessed on 1 June 2025).
11. Bicknell, B.T.; Butler, D.; Whalen, S.; Ricks, J.; Dixon, C.J.; Clark, A.B.; Spaedy, O.; Skelton, A.; Edupuganti, N.; Dzubinski, L.; et al. Chatgpt-4 omni performance in usmle disciplines and clinical skills: Comparative analysis. *JMIR Med. Educ.* **2024**, *10*, e63430. [CrossRef]
12. Salloum, S.; Dautov, R.; Chen, X.; Peng, P.X.; Huang, J.Z. Big data analytics on Apache Spark. *Int. J. Data Sci. Anal.* **2016**, *1*, 145–164. [CrossRef]
13. United States Medical Licensing Examination (USMLE). About the USMLE [Internet]. Philadelphia (PA): Federation of State Medical Boards. 2025. Available online: <https://www.usmle.org> (accessed on 1 June 2025).
14. Slee, V.N. The international classification of diseases: Ninth revision (ICD-9). *Ann. Intern. Med.* **1978**, *88*, 424–426. [CrossRef]
15. Gligorijevic, D.; Stojanovic, J.; Satz, W.; Stojkovic, I.; Schreyer, K.; Del Portal, D.; Obradovic, Z. Deep attention model for triage of emergency department patients. In *Proceedings of the 2018 SIAM International Conference on Data Mining, San Diego, CA, USA, 3–5 May 2018*; pp. 297–305.

16. Li, X.; Li, M.; Yan, P.; Li, G.; Jiang, Y.; Luo, H.; Yin, S. Deep learning attention mechanism in medical image analysis: Basics and beyonds. *Int. J. Netw. Dyn. Intell.* **2023**, *2*, 93–116. [[CrossRef](#)]
17. Jiang, D.; Wang, H.; Li, T.; Gouda, M.A.; Zhou, B. Real-time tracker of chicken for poultry based on attention mechanism-enhanced YOLO-Chicken algorithm. *Comput. Electron. Agric.* **2025**, *237*, 110640. [[CrossRef](#)]
18. Alsentzer, E.; Murphy, J.R.; Boag, W.; Weng, W.H.; Jin, D.; Naumann, T.; McDermott, M. Publicly available clinical BERT embeddings. *arXiv* **2019**, arXiv:1904.03323.
19. Lee, J.; Yoon, W.; Kim, S.; Kim, D.; Kim, S.; So, C.H.; Kang, J. BioBERT: A pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics* **2020**, *36*, 1234–1240. [[CrossRef](#)] [[PubMed](#)]
20. Kung, T.H.; Cheatham, M.; Medenilla, A.; Sillos, C.; De Leon, L.; Elepaño, C.; Madriaga, M.; Aggabao, R.; Diaz-Candido, G.; Maningo, J.; et al. Performance of ChatGPT on USMLE: Potential for AI-assisted medical education using large language models. *PLoS Digit. Health* **2023**, *2*, e0000198. [[CrossRef](#)] [[PubMed](#)]
21. Teo, Z.L.; Jin, L.; Liu, N.; Li, S.; Miao, D.; Zhang, X.; Ng, W.Y.; Tan, T.F.; Lee, D.M.; Chua, K.J.; et al. Federated machine learning in healthcare: A systematic review on clinical applications and technical architecture. *Cell Rep. Med.* **2024**, *20*, 5.
22. Abbas, S.R.; Abbas, Z.; Zahir, A.; Lee, S.W. Federated Learning in Smart Healthcare: A Comprehensive Review on Privacy, Security, and Predictive Analytics with IoT Integration. *Healthcare* **2024**, *12*, 2587. [[CrossRef](#)]
23. Li, T.; Sahu, A.K.; Talwalkar, A.; Smith, V. Federated learning: Challenges, methods, and future directions. *IEEE Signal Process. Mag.* **2020**, *37*, 50–60. [[CrossRef](#)]
24. Sheller, M.J.; Edwards, B.; Reina, G.A.; Martin, J.; Pati, S.; Kotrotsou, A.; Milchenko, M.; Xu, W.; Marcus, D.; Colen, R.R.; et al. Federated learning in medicine: Facilitating multi-institutional collaborations without sharing patient data. *Sci. Rep.* **2020**, *10*, 12598. [[CrossRef](#)]
25. Taiello, R.; Cansiz, S.; Vesin, M.; Cremonesi, F.; Innocenti, L.; Önen, M.; Lorenzi, M. Enhancing Privacy in Federated Learning: Secure Aggregation for Real-World Healthcare Applications. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*; Springer Nature: Cham, Switzerland, 7 October 2024; pp. 204–214.
26. Fan, T.; Kang, Y.; Ma, G.; Chen, W.; Wei, W.; Fan, L.; Yang, Q. Fate-llm: A industrial grade federated learning framework for large language models. *arXiv* **2023**, arXiv:2310.10049.
27. Yaqoob, M.M.; Nazir, M.; Khan, M.A.; Qureshi, S.; Al-Rasheed, A. Hybrid classifier-based federated learning in health service providers for cardiovascular disease prediction. *Appl. Sci.* **2023**, *13*, 1911. [[CrossRef](#)]
28. Ye, R.; Wang, W.; Chai, J.; Li, D.; Li, Z.; Xu, Y.; Du, Y.; Wang, Y.; Chen, S. Openfedllm: Training large language models on decentralized private data via federated learning. In *Proceedings of the 30th ACM SIGKDD Conference On Knowledge Discovery and Data Mining, Barcelona, Spain, 25–29 August 2024*; pp. 6137–6147.
29. Zhang, F.; Kreuter, D.; Chen, Y.; Dittmer, S.; Tull, S.; Shadbahr, T.; Schut, M.; Asselbergs, F.; Kar, S.; Sivapalaratnam, S.; et al. Recent methodological advances in federated learning for healthcare. *Patterns* **2024**, *5*, 101006. [[CrossRef](#)] [[PubMed](#)]
30. Kaissis, G.A.; Makowski, M.R.; Rückert, D.; Braren, R.F. Secure, privacy-preserving and federated machine learning in medical imaging. *Nat. Mach. Intell.* **2020**, *2*, 305–311. [[CrossRef](#)]
31. Mugunthan, V.; Polychroniadou, A.; Byrd, D.; Balch, T.H. Smpai: Secure multi-party computation for federated learning. In *NeurIPS 2019 Workshop on Robust AI in Financial Services*; MIT Press: Cambridge, MA, USA, 2019; Volume 21.
32. Choudhury, O.; Gkoulalas-Divanis, A.; Salonidis, T.; Sylla, I.; Park, Y.; Hsu, G.; Das, A. Differential privacy-enabled federated learning for sensitive health data. *arXiv* **2019**, arXiv:1910.02578.
33. Nandimath, J.; Banerjee, E.; Patil, A.; Kakade, P.; Vaidya, S.; Chaturvedi, D. Big data analysis using Apache Hadoop. In *Proceedings of the 2013 IEEE 14th International Conference on Information Reuse & Integration (IRI), San Francisco, CA, USA, 14 August 2013*; pp. 700–703.
34. Sharma, K.; Parashar, D.; Mengshetti, O.; Ahmad, R.; Mital, R.; Singh, P.; Thawani, M. Apache spark for analysis of electronic health records: A case study of diabetes management. *Rev. D'intelligence Artif.* **2023**, *37*, 1521. [[CrossRef](#)]
35. Dash, S.; Shakyawar, S.K.; Sharma, M.; Kaushik, S. Big data in healthcare: Management, analysis and future prospects. *J. Big Data* **2019**, *6*, 54. [[CrossRef](#)]
36. Goh, K.H.; Wang, L.; Yeow, A.Y.; Poh, H.; Li, K.; Yeow, J.J.; Tan, G.Y. Artificial intelligence in sepsis early prediction and diagnosis using unstructured data in healthcare. *Nat. Commun.* **2021**, *12*, 711. [[CrossRef](#)]
37. Sivarajkumar, S.; Mohammad, H.A.; Oniani, D.; Roberts, K.; Hersh, W.; Liu, H.; He, D.; Visweswaran, S.; Wang, Y. Clinical information retrieval: A literature review. *J. Healthc. Inform. Res.* **2024**, *8*, 313–352. [[CrossRef](#)]
38. Halpern, G.A.; Nemet, M.; Gowda, D.M.; Kilickaya, O.; Lal, A. Advances and utility of digital twins in critical care and acute care medicine: A narrative review. *J. Yeungnam Med. Sci.* **2024**, *42*, 9. [[CrossRef](#)]

39. Eclipse Foundation. Eclipse Ditto: Open-Source Framework for Creating and Managing Digital Twins in the IoT [Internet]. 2025. Available online: <https://eclipse.dev/ditto> (accessed on 1 June 2025).
40. Node-RED Project. Node-RED: Low-Code Programming for Event-Driven Applications [Internet]. 2025. Available online: <https://nodered.org> (accessed on 1 June 2025).
41. Zhang, L.; Yang, G.; Li, J.; Liang, J. A novel medical cyber-physical systems based on digital twin-driven platform. *Int. J. High Speed Electron. Syst.* **2024**, *2540129*. [[CrossRef](#)]
42. Kommera, A.R. The power of event-driven architecture: Enabling real-time systems and scalable solutions. *Turk. J. Comput. Math. Educ.* **2020**, *11*, 3048–3055.
43. Bekmezci, I.; Sahingoz, O.K.; Temel, Ş. Flying ad-hoc networks (FANETs): A survey. *Ad. Hoc Netw.* **2013**, *11*, 1254–1270. [[CrossRef](#)]
44. Bender, D.; Sartipi, K. HL7 FHIR: An Agile and RESTful approach to healthcare information exchange. In Proceedings of the 26th IEEE International Symposium on Computer-Based Medical Systems, Porto, Portugal, 20–22 June 2013; pp. 326–331.
45. Kreps, J.; Narkhede, N.; Rao, J. Kafka: A distributed messaging system for log processing. *Proc. NetDB* **2011**, *11*, 1–7.
46. Zaharia, M.; Chowdhury, M.; Franklin, M.J.; Shenker, S.; Stoica, I. Spark: Cluster computing with working sets. In *2nd USENIX Workshop on Hot Topics in Cloud Computing; HotCloud 10*: Boston, MA, USA, 2010.
47. Armbrust, M.; Das, T.; Sun, L.; Yavuz, B.; Zhu, S.; Murthy, M.; Torres, J.; van Hovell, H.; Ionescu, A.; Łuszczak, A.; et al. Delta lake: High-performance ACID table storage over cloud object stores. *VLDB Endow.* **2020**, *13*, 3411–3424. [[CrossRef](#)]
48. Meta, A.I. Introducing Meta Llama 3 [Internet]. 2025. Available online: <https://ai.facebook.com/blog/meta-llama-3> (accessed on 1 June 2025).
49. Han, T.; Adams, L.C.; Papaioannou, J.M.; Grundmann, P.; Oberhauser, T.; Löser, A.; Truhn, D.; Bressemer, K.K. MedAlpaca—An open-source collection of medical conversational AI models and training data. *arXiv* **2023**, arXiv:2304.08247.
50. Lewis, P.; Perez, E.; Piktus, A.; Petroni, F.; Karpukhin, V.; Goyal, N.; Küttler, H.; Lewis, M.; Yih, W.; Rocktäschel, T.; et al. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems*; Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M.F., Lin, H., Eds.; Curran Associates: San Jose, CA, USA, 2020; pp. 9459–9474.
51. Krumholz, H.M. Big data and new knowledge in medicine: The thinking, training, and tools needed for a learning health system. *Health Aff.* **2014**, *33*, 1163–1170. [[CrossRef](#)]
52. Mulder, T. Health apps, their privacy policies and the GDPR. *Eur. J. Law. Technol.* **2019**, *10*, 3.
53. Jacobsen, A.; Kaliyaperumal, R.; da Silva Santos, L.O.; Mons, B.; Schultes, E.; Roos, M.; Thompson, M. A generic workflow for the data FAIRification process. *Data Intell.* **2020**, *2*, 56–65. [[CrossRef](#)]
54. Sinaci, A.A.; Núñez-Benjumea, F.J.; Gencturk, M.; Jauer, M.L.; Deserno, T.; Chronaki, C.; Cangioli, G.; Cavero-Barca, C.; Rodríguez-Pérez, J.M.; Pérez-Pérez, M.M.; et al. From raw data to FAIR data: The FAIRification workflow for health research. *Methods Inf. Med.* **2020**, *59*, e21–e32. [[CrossRef](#)]
55. Karamanlioglu, A.; Sunar, E.T.; Cetin, C.; Akca, G.; Merdanoglu, H.; Dogan, O.T.; Alpaslan, F.N. GDPR and FAIR compliant decision support system design for triage and disease detection. In *International Conference on Information Technology—New Generations*; Springer International Publishing: Cham, Switzerland, 2023; pp. 331–338.
56. Donnelly, K. SNOMED-CT: The advanced terminology and coding system for eHealth. *Stud. Health Technol. Inform.* **2006**, *121*, 279.
57. McDonald, C.J.; Huff, S.M.; Suico, J.G.; Hill, G.; Leavelle, D.; Aller, R.; Forrey, A.; Mercer, K.; DeMoor, G.; Hook, J.; et al. LOINC, a universal standard for identifying laboratory observations: A 5-year update. *Clin. Chem.* **2003**, *49*, 624–633. [[CrossRef](#)] [[PubMed](#)]
58. Liu, S.; Ma, W.; Moore, R.; Ganesan, V.; Nelson, S. RxNorm: Prescription for electronic drug information exchange. *IT Prof.* **2005**, *7*, 17–23. [[CrossRef](#)]
59. MQTT Org. The standard for IoT messaging [Internet]. 2025. Available online: <https://mqtt.org> (accessed on 1 June 2025).
60. Eclipse Foundation. Eclipse Mosquitto: An Open-Source MQTT Broker [Internet]. 2025. Available online: <https://mosquitto.org> (accessed on 1 June 2025).
61. DiSantostefano, J. International classification of diseases 10th revision (ICD-10). *J. Nurse Pract.* **2009**, *5*, 56–57. [[CrossRef](#)]
62. Carlson, J. *Redis in Action*; Simon and Schuster: New York, NY, USA, 2013.
63. Lipscomb, C.E. Medical subject headings (MeSH). *Bull. Med. Libr. Assoc.* **2000**, *88*, 265.
64. Bodenreider, O. The unified medical language system (UMLS): Integrating biomedical terminology. *Nucleic Acids Res.* **2004**, *32* (Suppl. 1), D267–D270. [[CrossRef](#)]
65. Johnson, A.E.; Pollard, T.J.; Shen, L.; Lehman, L.W.; Feng, M.; Ghassemi, M.; Moody, B.; Szolovits, P.; Anthony Celi, L.; Mark, R.G. MIMIC-III, a freely accessible critical care database. *Sci. Data* **2016**, *3*, 1–9. [[CrossRef](#)]
66. Baig, M.M.; Hobson, C.; GholamHosseini, H.; Ullah, E.; Afifi, S. Generative AI in improving personalized patient care plans: Opportunities and barriers towards its wider adoption. *Appl. Sci.* **2024**, *14*, 10899. [[CrossRef](#)]
67. Krishnan, N. Advancing multi-agent systems through model context protocol: Architecture, implementation, and applications. *arXiv* **2025**. [[CrossRef](#)]

68. Lundberg, S.M.; Lee, S.I. A unified approach to interpreting model predictions. *Adv. Neural Inf. Process Syst.* **2017**, *30*, 4765–4774.
69. Ribeiro, M.T.; Singh, S.; Guestrin, C. “Why should i trust you?” Explaining the predictions of any classifier. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 13 August 2016; pp. 1135–1144.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.