



Development of an On-Premises Retrieval-Augmented Generation System for Electronic Health Records Retrieval in Level 5 Hospitals in Kenya

Kevin Comba Gatimu^{1*}, James Mwikya², Stephen Mageto²

¹Department of Computer Science and Informatics, Kirinyaga University, Kerugoya, Kenya

²School of Pure and Applied Sciences, Kirinyaga University, Kerugoya, Kenya

*Email: kevin.comba.gatimu@outlook.com | Co-authors: jmwikya@kyu.ac.ke; snjakiba@kyu.ac.ke

DOI : <https://doi.org/10.55248/gengpi.07.0926.2903>

ABSTRACT

Architectural proposals for privacy-preserving clinical artificial intelligence are more numerous than working implementations, and the question of whether a full Retrieval-Augmented Generation (RAG) pipeline can in fact be built and operated within the hardware and connectivity envelope of an African county referral hospital remains largely untested. This article reports the second objective of a design science study: the development of an On-Premises RAG system for retrieving electronic health records in Level 5 hospitals in Kenya. Following the Design Science Research Methodology, a previously validated five-layer architecture was implemented as an Electron desktop application in which every component, comprising embedding generation, vector indexing and language-model inference, executes on local hardware. Embeddings were produced locally at 1,024 dimensions by qwen3-embedding-0.6b and indexed with Hierarchical Navigable Small World graphs in a MongoDB-compatible document database, while inference was served locally by Microsoft Foundry Local with automatic execution-provider selection. A synthetic corpus representing the electronic medical record database of a fictitious institution, comprising 60 patients and 120 clinical encounters across 25 referentially complete relational tables, yielded 1,603 source rows ingested as 1,603 chunks, each carrying lineage metadata to its source record. The corpus is exactly reproducible from a fixed seed and its integrity is verified before ingestion, a safeguard introduced after a defect in an earlier corpus invalidated a complete evaluation cycle. The seven requirements in the specification were implemented and verified, role-based access control mapped five cadre-aligned roles to twelve application pages, every action over patient data was written to an append-only audit log, and network monitoring during operation confirmed that no external call was made. The system operated within a 32 GB memory envelope at approximately 70 per cent resident memory, with inference executing on an integrated Intel Arc graphics processor through OpenVINO. Two boundaries are stated rather than left to inference: operation on a central processing unit alone was not demonstrated, and the implemented security controls were verified functionally but not subjected to security testing.

Keywords: Retrieval-Augmented Generation; on-premises deployment; small language models; electronic health records; system development; Kenya

1. Introduction

1.1 Background of the Study

The past three years have seen Retrieval-Augmented Generation move from a research construct to a deployment pattern across knowledge-intensive domains. Introduced by Lewis et al. (2020), RAG grounds generated text in evidence retrieved from an external corpus at inference time, and in clinical settings retrieval-augmented systems generally outperform foundational models alone on factual accuracy, although a systematic review of seventy studies cautions that source attribution is frequently incomplete and that standardised evaluation frameworks are lacking (Amugongo et al., 2025). What has changed more recently is not the concept but its tractability. Capable small language models, those below roughly ten billion parameters, now approach the medical performance of much larger models, with Meerkat-7B the first seven-billion-parameter model to exceed the United States Medical Licensing Examination passing threshold at 74.3 per cent on MedQA (Kim et al., 2025). Together with mature local inference runtimes and vector databases that run in a container, these developments have brought the entire RAG stack within reach of a single commodity machine.

This shift matters disproportionately in sub-Saharan Africa, where the assumptions underpinning cloud-hosted clinical artificial intelligence hold least well. Reviews of artificial intelligence in the African health space identify infrastructural limitation, an extensive urban-rural divide and the difficulty of policy implementation as the recurring constraints on deployment (Alaran et al., 2025), while the evidence base remains thin: only a handful of randomised trials of artificial intelligence for health, and none of generative artificial intelligence tools, have been conducted on the continent (Mateen et al., 2025). Local execution converts intermittent connectivity from a system-level dependency into an irrelevance.

In Kenya, Level 5 hospitals serve as county referral centres and have progressively adopted electronic medical record systems, most notably KenyaEMR, built on the OpenMRS platform and deployed on local servers in more than 2,300 facilities, primarily for HIV care management but increasingly extended to general outpatient and inpatient management. The regulatory frame reinforces the technical argument: the Data Protection Act, 2019 classifies health information as sensitive personal data under Section 44, and Section 50 of the Act, read with Regulation 26 of the Data Protection (General) Regulations, 2021, designates the provision of primary or secondary health care a strategic interest of the state requiring that such processing be carried out through a server and data centre located in Kenya. A companion study established the design requirements for these facilities and derived a five-layer On-Premises RAG architecture from them. The present article reports whether that architecture can be built.

1.2 Statement of the Problem

The ideal situation is one in which a validated architecture can be realised in working software using components a hospital information technology department can obtain, install, and maintain, on hardware it already owns. The current reality is that the privacy-preserving clinical artificial intelligence literature is dominated by architectural proposals, benchmark comparisons, and cost models rather than by operating systems. Where implementations exist, they are typically evaluated in high-income settings on hardware that county hospitals do not possess, and they rarely report the concrete artefacts against which feasibility must be judged: ingestion throughput, corpus and chunk counts, memory occupancy, role and permission structures, or verification that no data left the machine.

The knowledge gap is therefore one of demonstrated constructability. It is not known whether a full advanced RAG pipeline, comprising query transformation, hybrid retrieval, reranking, contextual compression, corrective grading and query routing, together with local embedding generation, vector indexing, authentication and audit logging, can be assembled and operated within a thirty-two-gigabyte memory envelope on a single commodity machine, and whether the accountability mechanisms that Kenyan data protection law expects can be embedded natively rather than bolted on. Nor is it known what the aggregate resource footprint of such a composition actually is, since the literature reports per-component figures and leaves additivity to be assumed.

Should this gap remain unaddressed, hospital management and county health departments have no empirical basis on which to judge vendor claims or to plan procurement, and the architectural literature continues to accumulate designs whose implementability at this tier is assumed rather than shown. This study addresses the gap by developing the system in full and reporting the implementation evidence.

1.3 Objective of the Study

To develop an On-Premises Retrieval-Augmented Generation system for retrieving electronic health records in Level 5 hospitals in Kenya.

The corresponding research question is: how can an On-Premises RAG system be developed for retrieving electronic health records in Level 5 hospitals in Kenya?

1.4 Justification and Scope

Development, rather than simulation or component benchmarking, is the appropriate method because the claim at stake is one of feasibility under composition. Individual components — an embedding model, a vector index, a small language model — are each known to run on commodity hardware; what is not known is whether they can be composed, together with orchestration, authentication, and audit machinery, while remaining within the envelope. Only construction of the complete artefact can settle this, which is precisely the warrant for the design and development activity in design science research (Peffer et al., 2007).

The study addresses the Level 5 tier in Kenya, the requirements having been framed at two Level 5 hospitals in Nyeri County, Nyeri County Referral Hospital and Consolata Hospital Mathari. Conceptually it is delimited to the development objective; the derivation of the architecture and the empirical performance evaluation are reported separately. Technically it is limited to text-based records retrieval, excluding image, audio and multimodal processing. Critically, and for the ethical reasons detailed in Section 3.4, no actual patient records were used at any stage and the developed system was at no point connected to the live electronic medical record system of either hospital: the corpus is synthetic throughout, generated programmatically to reproduce the structure, field coverage and referential integrity of Kenyan hospital record data. The regulatory scope is confined to the Kenyan framework, principally the Data Protection Act, 2019, the Data Protection (General) Regulations, 2021 (Legal Notice No. 263) and the Digital Health Act, 2023.

2. Literature Review and Conceptual Framework

2.1 Theoretical Framework

The study is anchored in the Design Science Research Methodology of Peffer et al. (2007), which specifies six activities: problem identification and motivation, definition of solution objectives, design and development, demonstration, evaluation, and communication. This article reports activities three and four. The methodological significance of that placement is that design science treats the artefact itself as the primary knowledge contribution: a working system embodies a set of claims about what is achievable under stated conditions, and building it constitutes the test of those claims.

Reporting standards for design science accordingly require that construction be documented in sufficient detail to permit both replication and independent judgement of the feasibility claim, which is the standard adopted here.

The Technology–Organisation–Environment framework of Tornatzky and Fleischer (1990) supplies the complementary lens for interpreting development decisions. Applied to artificial intelligence adoption in Finnish public social and health organisations, it identifies organisational barriers, particularly limited funding, insufficient AI competence and inadequate change management, as the most critical challenges, with legislative ambiguity and national funding shortfalls also prominent (Pulkkinen et al., 2025). In the development phase these dimensions cease to be abstract: the technology dimension is instantiated as the memory envelope and the requirement that no particular class of accelerator be mandatory, the organisational dimension as the skills a hospital information technology department can reasonably be expected to hold, and the environmental dimension as the audit and access-control mechanisms that the Kenyan accountability regime requires the software to contain. Each dimension therefore translates into a concrete implementation constraint rather than a predictor of adoption.

2.2 Review of Related Empirical Studies

Three literatures bear on the constructability question. The first concerns the local execution of language models. Small language models have improved rapidly on medical benchmarks while requiring substantially fewer resources; beyond Meerkat-7B (Kim et al., 2025), a benchmark of small models deployed on mobile and wearable devices found that they can approach the performance of much larger models on healthcare prediction tasks while offering substantial gains in efficiency and privacy, though class imbalance and few-shot settings remain difficult (X. Wang et al., 2025). Quantisation has been central to this tractability: post-training quantisation to three or four bits per weight reduces model size with negligible accuracy degradation on standard language benchmarks (Frantar et al., 2023), although systematic benchmarking of small models specifically finds that techniques optimised for large models transfer poorly to this class and that compression for it requires its own design principles (J. Wang et al., 2025). Two findings qualify the picture further. Hong et al. (2024) evaluated compressed models against the trustworthiness dimensions of fairness, privacy robustness and adversarial resilience, finding that four-bit quantisation largely preserves them while three-bit reduces them significantly; and Marcuzzi et al. (2025) show that aggressive quantisation can shift bias, with effects that vary in magnitude across demographic subgroups, which argues for disaggregated evaluation as part of validation. The system reported here does not quantise manually, its inference runtime selecting the execution provider and model format at load time, but the same cautions apply to whatever precision that runtime resolves to.

The second literature concerns pipeline composition. Gao et al. (2024) established the Modular RAG paradigm, in which the pipeline is a composition of interchangeable, independently optimisable components, an organising principle of direct implementation consequence, since modularity is what permits stages to be disabled when resources are scarce. Oche et al. (2025) identify hybrid retrieval among the emerging responses to persistent retrieval-quality and privacy problems, and Wolk (2025), comparing twelve retrieval-augmented pipelines in a secure on-premises configuration, found that hybrid fusion of dense and sparse retrieval achieved the best retrieval accuracy, at precision at five of 0.68 and normalised discounted cumulative gain at ten of 0.67, and the lowest hallucination rate among the retrieval strategies at 11.2 per cent against 18.4 per cent for dense and 20.1 per cent for sparse retrieval; a self-reflective pipeline was lowest overall at 5.8 per cent. Wampler et al. (2025) codified provenance tracking, source attribution and retrieval confidence scoring as requirements for high-stakes domains, translating in implementation terms into chunk-level lineage metadata. Singh et al. (2025) surveyed Agentic RAG, in which autonomous agents orchestrate retrieval and apply reflection before responding; the paradigm suits multi-step clinical work but its computational demands make selective, tool-bounded agent designs preferable at this tier.

The third literature concerns deployment economics and clinical precedent. Pan et al. (2025) modelled the total cost of ownership of locally deployed against commercially hosted language models and reported break-even within approximately three months for models in the twenty-four to thirty-two billion parameter range on a single consumer accelerator, though on high-income cost assumptions and without reference to healthcare specifically. Korom et al. (2025) evaluated AI Consult, a safety-net clinical decision support tool that flags documentation and decision-making errors while leaving clinical judgement with the clinician, across 39,849 patient visits at Penda Health facilities in Nairobi, and reported 16 per cent fewer diagnostic errors and 13 per cent fewer treatment errors among clinicians with access to it. This provides a Kenyan clinical precedent from a quality-improvement study, and the system is cloud-dependent, which is the deployment mode the Kenyan residency framework renders problematic. Sriram et al. (2025) built a urology assistant on domain-specific biomedical embeddings over a curated urological knowledge corpus and reported that the DeepSeek-R1 evaluator preferred its outputs to non-retrieval Llama baselines in an average of 88.5 per cent of comparisons, supporting the decision to build a purpose-configured rather than a general-purpose system.

2.3 Identification of Research Gaps

Two gaps follow. First, the component literature establishes feasibility in isolation but not under composition. X. Wang et al. (2025), Frantar et al. (2023) and J. Wang et al. (2025) investigate local execution or compression of language models, but none reports the aggregate footprint of a model together with an embedding service, a vector index, an orchestration framework, an authentication subsystem and a user interface running concurrently, which is the quantity that determines whether a hospital can deploy anything at all. Second, no published implementation in an African healthcare setting reports the artefacts required to judge on-premises feasibility: corpus and chunk counts, ingestion behaviour, memory occupancy, permission structures and positive verification of network isolation. The evidence that exists is either architectural (Gao et al., 2024; Wampler et al., 2025), economic (Pan et al., 2025), or clinical but cloud-dependent (Korom et al., 2025). This article supplies the missing implementation evidence.

2.4 Conceptual Framework

The conceptual framework for the development phase treats the validated five-layer architecture as the input, the implementation constraints as moderating conditions, and the working system together with its implementation evidence as the output. The five layers — ingestion and indexing, vector storage, local inference, orchestration, and presentation — are shown in Figure 1. The moderating conditions are the memory envelope, the absence of mandatory graphics processing hardware, the offline operating requirement, and the accountability obligations of the Kenyan data protection regime. The output constructs against which development is judged are requirements coverage, resource conformance, data locality, and accountability instrumentation. Development thus constitutes a test of whether the architectural claims survive contact with the implementation constraints.

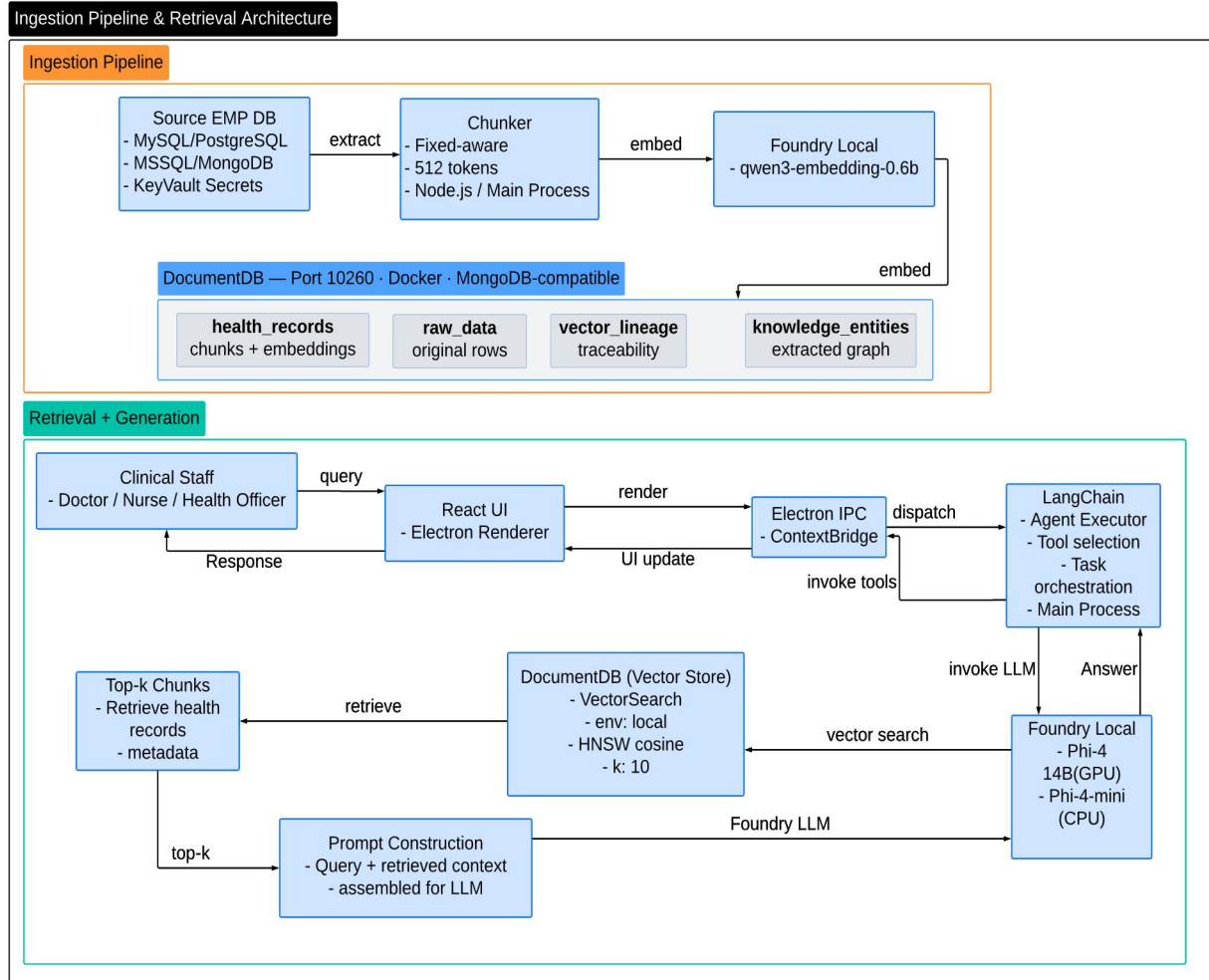


Figure 1: The Five-Layer On-Premises RAG Architecture Implemented in This Study

3. Methodology

3.1 Research Design

The study employed a design science research design corresponding to activities three and four of the Peffers et al. (2007) framework, design and development, and demonstration, under a pragmatist orientation that evaluates knowledge claims by fitness for purpose (Creswell & Plano Clark, 2018). Development proceeded iteratively, with functionality reviews every two weeks against the requirements specification, each review resolving one increment of functionality and confirming continued conformance to the resource envelope. The system was then demonstrated by ingesting a complete corpus and executing retrieval queries end to end under normal operating conditions.

3.2 Development Environment and Technology Selection

Component selection was governed by three binding constraints: every component had to execute locally, none could require a specific class of accelerator, and the aggregate footprint had to remain within thirty-two gigabytes of memory. Table 1 records the selections and the justification for each.

Table 1: Technology Stack and Selection Justification

Layer	Component selected	Selection justification
Presentation	Electron 41 desktop application (React 19, TypeScript, TanStack Router)	Runs as a local application without a web server; a context-isolated bridge separates the interface from the privileged main process
Orchestration	LangChain (Node.js) with a custom agent executor	Modular stage composition per Gao et al. (2024); permits individual stages to be disabled under resource pressure and ablated under evaluation
Inference	Microsoft Foundry Local serving Phi-4 Mini Instruct (3.8B)	Automatic execution-provider selection across CPU, OpenVINO, CUDA and NPU removes manual quantisation and model-format conversion from the workflow
Embedding	qwen3-embedding-0.6b (1,024 dimensions), served in-process through the Foundry Local SDK	Small enough to co-reside with the generation model within the memory envelope
Storage	DocumentDB, MIT-licensed and MongoDB-compatible, in a local container; HNSW cosine vector index	Native vector search and knowledge-graph collections in a single local store; no separate vector service to administer
Security	BetterAuth (scrypt hashing, eight-hour sessions); SQLite for authentication and configuration	Local authentication without an identity provider; no protected health information persisted outside the vector store

3.3 Dataset Description and Preprocessing

The knowledge base was constructed exclusively from synthetic health records representing the electronic medical record database of a fictitious institution designated Synthetic Facility A, which corresponds to no facility in the Kenya Master Health Facility Registry and is not a pseudonym for either study site. The corpus is defined by a data definition script creating a PostgreSQL 15 database of twenty-five relational tables organised into reference, patient, encounter, orders and results, decision support and billing groups, with seven enumerated types and eleven secondary indexes. It comprises 60 synthetic patients and 120 clinical encounters at two per patient, with 120 vital-sign records, 120 diagnoses, 66 laboratory orders yielding 74 results of which 16 are flagged abnormal, 72 prescriptions with 120 prescription items, 49 admissions, 60 imaging orders and 60 clinical decision support alerts, patients drawn from 20 Kenyan counties. Every field, including those that would carry directly identifying information in a live system, holds a generated value corresponding to no living person. The schema and its value pools are modelled on Kenyan hospital practice rather than on a generic electronic medical record: monetary columns are denominated in Kenya shillings, the payment method constraint admits mobile money, the insurance table admits the National Hospital Insurance Fund alongside private and corporate cover, and diagnoses are drawn from an ICD-10 pool reflecting the Kenyan burden of disease. The same dataset was materialised on MySQL, Microsoft SQL Server and MongoDB to verify that the ingestion connector operates independently of the underlying database technology; those instances hold identical data and were not evaluated. Preprocessing applied field-aware chunking, which segments records on semantically meaningful field boundaries rather than at fixed character offsets and emits one chunk per source row at approximately 512 tokens. Each chunk was embedded into a 1,024-dimensional vector and upserted with lineage metadata linking it to its source record and field.

3.4 Development Procedure and Ethical Safeguards

Development followed the layer sequence of the architecture. The application shell, authentication, and configuration store were implemented first, followed by the source-database connectors, the ingestion and embedding path, the vector and graph indexes, the advanced retrieval pipeline, the agent layer, and finally the role-governed interface. On first run the application executes its database migrations, seeds a default administrator account, detects the container runtime and starts the document database, creates all required collections and vector indexes, and detects the available hardware execution providers — an installation sequence designed so that a hospital information technology officer need not assemble the stack manually.

Ethical clearance was obtained from the Kirinyaga University Institutional Research Ethics Committee and a research permit from the National Commission for Science, Technology and Innovation, with written authorisation from both hospital administrations. No actual patient health records were collected, accessed or processed at any stage, consistent with Section 44 of the Data Protection Act, 2019, and the system was at no point connected to the live electronic medical record system of either facility. A Data Protection Impact Assessment, required under Section 31 of the Act and elaborated in Regulations 49 to 53 of the Data Protection (General) Regulations, 2021, was completed before the system was operated in a hospital environment.

3.5 Analysis of Development Outcomes

Development outcomes were assessed against four operationalised constructs. Requirements coverage was assessed by tracing each requirement in the specification to an implemented feature and recording its verification method. Resource conformance was assessed by observing resident memory occupancy during ingestion and inference against the thirty-two-gigabyte envelope, together with processor and accelerator utilisation, on the single platform available for the study. Data locality was assessed by monitoring network activity throughout ingestion, querying and idle operation and recording the destination of every outbound call. Accountability instrumentation was assessed by confirming that every action over patient data produced an audit record and that every retrieved chunk resolved to its source record through its lineage metadata. Ingestion behaviour was reported descriptively as source-row and chunk counts. Two limits on this assessment are recorded at the outset: verification of the security controls was functional rather than adversarial, no penetration testing, threat modelling or security audit having been conducted; and resource conformance was established on one hardware configuration and for one corpus size, so it does not establish scaling behaviour.

4. Results

4.1 The Implemented System

The architecture was realised in full as a desktop application named Health Records Ingest, comprising a sandboxed renderer providing the clinical interface, a privileged Node.js main process, and two utility processes, one for embedding and one for ingestion. Figure 2 presents the component-level structure of the main process as implemented. Inbound bridge handlers delegate to the RAG orchestrator, which sequences the gate, semantic cache, routing, retrieval, reranking, compression, grading, and generation stages. Supporting components comprise the pipeline modules for query transformation, routing, hybrid retrieval with Reciprocal Rank Fusion, reranking, extractive compression, and corrective retrieval; an agent layer comprising the intent router, executor, and tools registry; the client responsible for chat completions against the local inference endpoint; and the embedding manager, which proxies embedding requests to the embedding utility process so that blocking native inference never occupies the interface thread.

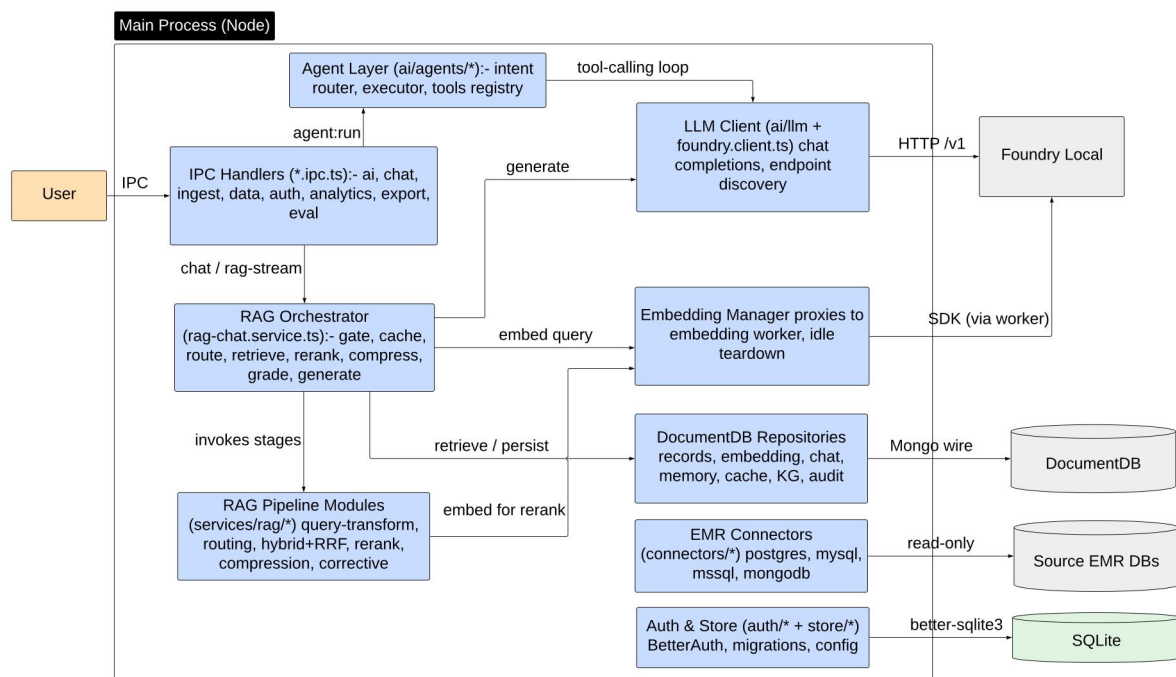


Figure 2: Component Structure of the Main Process as Implemented

The dual-path answering strategy was implemented as designed. A deterministic intent router directs conversational and narrative questions to the advanced retrieval pipeline and analytical or tool-requiring questions to the agent executor, which supports five specialised agents: Health Query, Summarize, Patient Lookup, Trends, and Ingestion. Because instruction-following varies considerably across locally deployable models, the executor

implements a dual execution strategy, using native tool-calling where the served model supports it and falling back to a deterministic pre-fetch approach otherwise — an implementation accommodation without which the agent layer would fail on the smaller model.

4.2 Ingestion and Indexing Outcomes

The ingestion agent processed the full synthetic corpus through the pipeline shown in Figure 3, reading records from the source database, applying field-aware chunking, generating embeddings locally and upserting into the indexed collections. Table 2 summarises the outcome. The corpus yielded 1,603 source rows across the twenty-five tables, ingested as 1,603 chunks at approximately 512 tokens each, field-aware chunking emitting one chunk per source row so that the chunk count equals the source-row count and every chunk carries lineage metadata resolving to its source record and field. All 1,603 chunks were successfully embedded and indexed. Corpus integrity was verified before ingestion, confirming that no encounter carried more than one vital-sign or diagnosis row, that no identical duplicate rows were present and that no orphaned foreign keys existed.

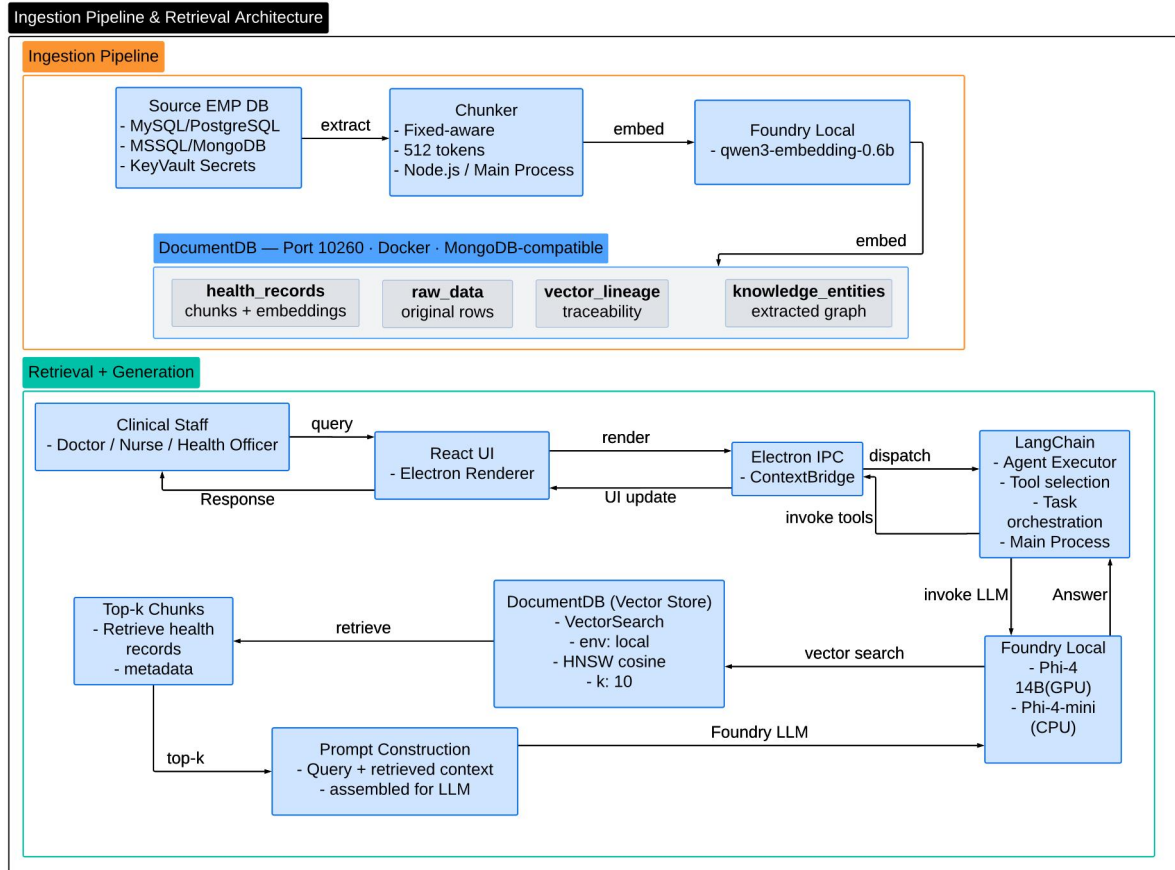


Figure 3: Implemented Ingestion and Retrieval Pipeline

Table 2: Corpus and Indexing Outcomes

Measure	Value
Synthetic patients	60
Clinical encounters	120 (2 per patient)
Relational tables ingested	25
Record types represented	Encounters, vital signs, diagnoses, prescriptions, laboratory orders and results, admissions, imaging orders, decision-support alerts, billing
Total source rows	1,603

Measure	Value
Chunks generated and indexed	1,603 (one chunk per source row)
Approximate chunk size	512 tokens
Embedding dimensionality	1,024 (qwen3-embedding-0.6b)
Vector index	HNSW graph, cosine similarity
Chunks carrying source lineage metadata	1,603 (100%)
Corpus reproducibility	Regenerable from seed 20260816; integrity verified before ingestion

4.3 Requirements Coverage, Access Control and Verification

Figure 4 presents the developed application in operation, showing the agents workspace with the five specialised agents, the active model indicator confirming Phi-4 Mini Instruct as the served generation model, the conversation history and a generated response with its accompanying chart. The application exposes twelve role-governed pages, namely Dashboard, Connections, Ingest, Data Explorer, AI Chat, AI Agents, Analytics, Outbreak Alerts, Audit Log, Settings, Admin and Profile, with access mapped to five roles through a shared permissions matrix, so that a nurse and an analyst see materially different application surfaces.

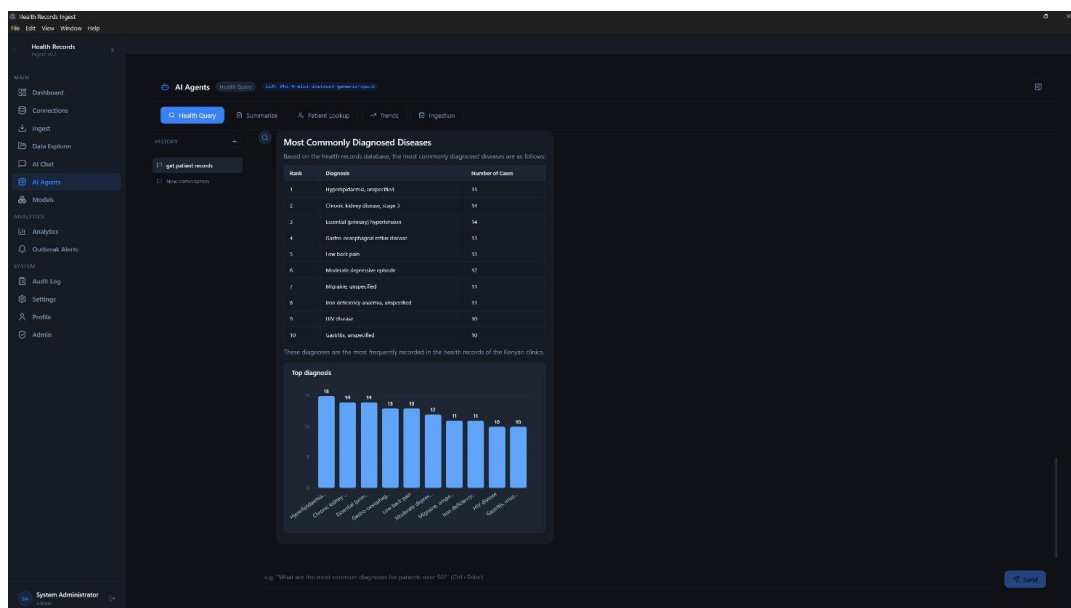


Figure 4: Agents Workspace of the Developed System (Health Records Ingest)

Table 3 records the verification of each requirement against the implemented system. All seven requirements in the specification were satisfied. The three of greatest consequence were verified by direct observation rather than by inspection: network activity was monitored throughout ingestion, querying and idle operation, and every outbound call was found to terminate at the local host or the local area network, with no external destination observed at any point; resident memory occupancy during ingestion and inference remained within the thirty-two-gigabyte envelope at approximately 70 per cent, with processor utilisation of about 10 per cent and the integrated Intel Arc graphics processor at 80 to 100 per cent; and every action over patient data produced a corresponding entry in the append-only audit log. The status recorded in the table is met as implemented and functionally verified; it is not a statement that the control has been tested adversarially, and Section 5.1 states that boundary explicitly.

Table 3: Verification of Requirements Against the Implemented System

Requirement	Implementation	Verification method	Status
Natural-language search with the supporting sources displayed	Advanced RAG pipeline with query transformation and routing; retrieved chunks surfaced with the answer	Functional testing; inspection of retrieved chunks	Met
Retrieval of patient profile, medication and laboratory information	Field-aware ingestion of patient, prescription and laboratory record types; Patient Lookup and Summarize agents	Functional testing	Met
Offline operation	All inference, embedding and search executed locally	Network activity monitoring throughout ingestion, querying and idle operation	Met
Local storage of embeddings and local execution of inference	Container-hosted document database and Foundry Local inference endpoint, both on the local host	Configuration and network verification	Met
A 32 GB design envelope	Phi-4 Mini Instruct (3.8B) with in-process embedding; ingestion and embedding in separate utility processes	Resident memory observation during ingestion and inference (approximately 70 per cent of 32 GB)	Met on the platform tested
Mean response within thirty seconds	Semantic answer cache; bounded retrieval at eight final context chunks	Mean end-to-end response time reported in the companion evaluation study	Met by the naive configuration
Authentication, access control and auditability	BetterAuth with script hashing and eight-hour sessions; five roles mapped to twelve pages; append-only audit log written on every patient-data action	Authentication testing; role-by-role interface inspection; log inspection after operation	Met as implemented; not security-tested

5. Discussion

5.1 Interpretation of Key Results

The central result is that composition did not defeat the resource envelope. The concern motivating this objective was not whether any single component would run locally, which the component literature had established, but whether a generation model, an embedding service, a vector index, an orchestration framework, an authentication subsystem and a desktop interface could run concurrently within the memory a county hospital server actually possesses. They could, at approximately 70 per cent of a thirty-two-gigabyte envelope, and the architectural decisions that made this possible are identifiable. Isolating embedding and ingestion into separate utility processes kept the largest transient allocations off the process that serves the interface; serving a 3.8-billion-parameter model for conversational work kept the generation footprint small; and co-locating vector search and the knowledge-graph collections in a single document store removed an entire additional service from the footprint. Two qualifications attach to this result and are stated rather than implied. The working margin is under ten gigabytes, which is sufficient for a corpus of 1,603 chunks and should not be assumed sufficient for the very much larger record set of a Level 5 hospital, since the vector index grows with the chunk count and memory scaling with corpus size was not measured. And inference executed throughout on an integrated Intel Arc graphics processor through OpenVINO, so that while the runtime provides for central-processor execution, operation without an accelerator was not demonstrated and remains an architectural claim rather than a measured result.

A second result of practical weight is the dual execution strategy in the agent layer. The finding that instruction-following varies enough across locally deployable models to break native tool-calling, and that a deterministic pre-fetch fallback restores function, is an implementation-level insight with direct consequences for anyone building agentic systems at this tier: agent designs that assume reliable tool-calling will fail intermittently on small models, and the failure will present as unpredictable behaviour rather than as a clean error. Third, the verification results bear directly on the privacy claim advanced in the companion design study. That claim was architectural, namely that locality follows from topology rather than from policy, and network monitoring converts it from a design assertion into an observed property of the running system. The claim is bounded in a way that deserves emphasis: locality relocates the trust boundary and does not secure what lies inside it. Unauthorised access by an authenticated internal user, credential compromise in the absence of a second factor, compromise of a database that is not encrypted at rest, privilege escalation, prompt injection through ingested record content and physical security of the host were all outside the scope of this study and remain unevaluated.

5.2 Integration with Existing Literature

The implementation substantiates several claims that the literature had established only in principle. The Modular RAG paradigm of Gao et al. (2024) is vindicated in an unanticipated respect: modularity proved valuable less for optimisation than for survivability, since individually toggleable stages allow the pipeline to shed reranking or compression under resource pressure rather than failing, and, as the companion evaluation study shows, it is also what makes controlled stage-level ablation possible at all. The hybrid dense-lexical retrieval identified by Oche et al. (2025) and measured by Wolk (2025) was implementable within the local store without a separate lexical service, and the provenance requirements codified by Wampler et al. (2025) were satisfiable natively through chunk-level lineage metadata rather than through an external tracking layer.

The findings also qualify parts of the literature. The Agentic RAG paradigm surveyed by Singh et al. (2025) proved implementable only in a bounded, tool-registry form with deterministic fallback; the reflective, freely orchestrating agents that the survey describes remain beyond what a 3.8-billion-parameter model reliably supports. The quantisation literature (Frantar et al., 2023; J. Wang et al., 2025) frames local deployment as a manual optimisation problem, whereas runtime-level automatic execution-provider selection removed manual quantisation and model-format conversion from the workflow entirely, a shift that materially lowers the skill threshold for hospital information technology departments and that the quantisation literature does not yet reflect. Finally, the implementation bears on the economic case of Pan et al. (2025) from a different direction: because the system required no subscription and ran on an existing workstation, the marginal software cost of deployment approaches zero. That is a narrower claim than it may appear, since the platform used carried an integrated accelerator, and the hardware question their model addresses is precisely the one this study did not settle.

5.3 Theoretical Implications

Two implications follow. First, the study suggests that feasibility in resource-constrained artificial intelligence deployment should be theorised at the level of *composition* rather than of components. The prevailing literature reports per-component resource characteristics, implicitly assuming additivity; this study indicates that aggregate footprint is instead governed by process lifecycle and co-location decisions, so that two systems with identical component inventories may differ substantially in deployability. A theory of deployment feasibility for constrained settings therefore requires constructs at the level of process architecture, not model size.

Second, the study extends the Technology–Organisation–Environment framework by showing that the environmental dimension can be *internalised into the artefact*. Regulatory obligation is conventionally treated as an external pressure to which an organisation responds procedurally. Here, the accountability requirements of the Kenyan regime were satisfied by structures inside the software — the append-only audit log, chunk-level lineage, and cadre-aligned role mapping — such that compliance evidence is generated as a by-product of normal operation. This suggests a class of artefacts for which environmental pressure becomes a design input rather than an implementation burden, a distinction with practical consequence for institutions of limited compliance capacity.

5.4 Practical and Policy Implications

For hospital information technology departments, the finding that the workload ran within thirty-two gigabytes on a single commodity workstation implies that pilot deployments should be scoped against memory capacity, storage throughput and backup power rather than against server-class procurement. The result should not, however, be read as evidence that accelerator provision is unnecessary, because the only configuration measured ran on an integrated graphics processor. A staged ingestion trial against local record volumes should precede any hardware commitment, since memory scaling was not measured and the envelope reported here holds for a corpus of the size evaluated. Consolidation of fragmented departmental databases deserves particular emphasis: fragmentation limits ingestion coverage directly, and no retrieval system can return what was never ingested.

For developers and implementers, four implementation practices are transferable. Role-based access control, append-only audit logging and chunk-to-source lineage should be placed in the data path rather than offered as configurable features, since they underpin both clinical trust and regulatory accountability, cost nothing measurable in operation, and are markedly harder to retrofit than to build in; a control that can be switched off is not a control on which a regulator can rely. Agent layers targeting small models should implement deterministic fallbacks rather than assume reliable tool-calling. Installation should be automated end to end, covering migrations, container start-up, index creation and hardware detection, because the realistic alternative at this tier is not careful manual assembly but non-deployment. And evaluation corpora should be generated from a fixed seed and verified for integrity before ingestion: a corpus defect in this study, in which an earlier corpus was seeded twice and the clinical child tables

consequently doubled while the reference answers continued to state single-seeded values, invalidated a full evaluation cycle, and the cost of the verification step is trivial against the cost of repeating the evaluation.

For policymakers, the study demonstrates that the accountability expectations of the Data Protection Act, 2019 and its 2021 Regulations can be met by software structures rather than by institutional procedure alone. Given that the Kenya National AI Strategy 2025-2030 designates healthcare a priority sector while the localisation requirement of Section 50 of the Act, read with Regulation 26, constrains the cloud services on which most commercial health artificial intelligence depends, sector-specific guidance could usefully specify audit logging, source lineage and cadre-aligned access control as minimum technical requirements for clinical artificial intelligence procurement. Such guidance should also state what locality does not deliver, and should require an independent security assessment rather than accept an on-premises topology as sufficient in itself.

6. Conclusion, Limitations and Future Work

6.1 Summary of Main Findings

This article set out to develop an On-Premises Retrieval-Augmented Generation system for retrieving electronic health records in Level 5 hospitals in Kenya. The validated five-layer architecture was implemented in full as an Electron desktop application in which embedding generation, vector search and language model inference all execute on local hardware. A synthetic corpus of 60 patients and 120 encounters across 25 referentially complete relational tables yielded 1,603 source rows ingested as 1,603 indexed chunks, each carrying lineage metadata to its source record; the corpus is exactly reproducible from seed 20260816 and its integrity is verified before ingestion. All seven requirements in the specification were satisfied; role-based access control mapped five cadre-aligned roles to twelve application pages; every action over patient data produced an audit entry; and network monitoring confirmed that no external call was made at any point. The system operated within a thirty-two-gigabyte memory envelope at approximately 70 per cent occupancy, with inference on an integrated Intel Arc graphics processor through OpenVINO. The study concludes that the designed architecture is implementable in full using locally executable components on commodity hardware, that the computational demands of retrieval-augmented generation are no longer prohibitive at the county referral hospital tier under the conditions tested, and that accountability mechanisms can be embedded natively rather than added procedurally. The development objective was therefore achieved.

6.2 Limitations

Five limitations qualify these conclusions. First, and most importantly, the corpus was synthetic. Although modelled on Kenyan hospital record structures, synthetic records lack the abbreviation, inconsistency, incompleteness and idiosyncratic local coding of real clinical documentation, so ingestion behaviour on institutional data may differ materially. Second, the corpus was small at 60 patients and 1,603 chunks; indexing and retrieval characteristics change with scale, and behaviour at the hundreds of thousands of records a Level 5 hospital accumulates was not observed, nor was memory scaling with corpus size measured. Third, inference executed throughout on an integrated graphics processor at 80 to 100 per cent utilisation, so operation on a central processing unit alone was not demonstrated and the claim of accelerator independence remains architectural. Fourth, the security controls were implemented and verified functionally but were not tested: no penetration testing, threat modelling exercise or security audit was conducted, and internal misuse, credential compromise, encryption at rest, privilege escalation and prompt injection through ingested content remain unevaluated. Fifth, the system was developed and demonstrated in a controlled environment rather than deployed into a live clinical workflow, so integration with existing electronic medical record operations, concurrent multi-user load and clinician usability in practice were not tested; and this article reports constructibility, not performance, which the companion evaluation study addresses.

6.3 Future Research Directions

Four directions follow. First, the system should be deployed against actual de-identified institutional records at production scale, reporting ingestion throughput, index build time, memory occupancy and retrieval latency as functions of corpus size, so that the scaling envelope can be established rather than assumed. Second, multi-user concurrent operation should be characterised, since a single-user memory profile does not predict behaviour when several clinicians query simultaneously, and this is the condition under which a referral hospital would actually operate the system; benchmarking on a central processing unit alone should be conducted in the same programme, to settle whether the architecture is feasible without an accelerator. Third, because fixed top-k semantic retrieval is structurally mismatched to whole-database counting and identifier-keyed lookup, development work should implement and test the routing path for which the architecture already provides, combining deterministic database aggregation and exact-match identifier indexes with semantic retrieval under a common router. Fourth, a threat model and penetration test covering internal misuse, credential compromise, encryption at rest and prompt injection should be conducted before any deployment involving real records, and, given the demographic effects that Marcuzzi et al. (2025) associate with reduced-precision models, implementation research should incorporate disaggregated performance testing across patient subgroups as a standard element of clinical artificial intelligence validation in African settings, where training-corpus representation is weakest.

References

- Alaran, M. A., Lawal, S. K., Jiya, M. H., Egya, S. A., Ahmed, M. M., Abdulsalam, A., Haruna, U. A., Musa, M. K., & Lucero-Prisno, D. E., III. (2025). Challenges and opportunities of artificial intelligence in African health space. *Digital Health*, 11, 20552076241305915. <https://doi.org/10.1177/20552076241305915>
- Amugongo, L. M., Mascheroni, P., Brooks, S., Doering, S., & Seidel, J. (2025). Retrieval augmented generation for large language models in healthcare: A systematic review. *PLOS Digital Health*, 4(6), e0000877. <https://doi.org/10.1371/journal.pdig.0000877>
- Creswell, J. W., & Plano Clark, V. L. (2018). *Designing and conducting mixed methods research* (3rd ed.). SAGE Publications.
- Data Protection (General) Regulations, 2021, Legal Notice No. 263. (2021). Republic of Kenya. Kenya Gazette Supplement No. 186. <https://www.odpc.go.ke/wp-content/uploads/2023/05/Kenya-Gazette-Supplement-No.-186-Data-Protection-General-Regulations-2021.pdf>
- Data Protection Act, No. 24 of 2019. (2019). Republic of Kenya. Kenya Gazette Supplement. <https://www.odpc.go.ke/wp-content/uploads/2023/01/Kenya-Data-Protection-Act-2019.pdf>
- Digital Health Act, No. 15 of 2023. (2023). Republic of Kenya. Kenya Gazette Supplement. https://kenyalaw.org/kl/fileadmin/pdfdownloads/Acts/2023/TheDigitalHealthAct_No.15of2023.pdf
- Frantar, E., Ashkboos, S., Hoefler, T., & Alistarh, D. (2023). GPTQ: Accurate post-training quantization for generative pre-trained transformers. *arXiv*. <https://arxiv.org/abs/2210.17323v2>
- Gao, Y., Xiong, Y., Gao, X., Jia, K., Pan, J., Bi, Y., Dai, Y., Sun, J., Wang, M., & Wang, H. (2024). Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997*. <https://arxiv.org/abs/2312.10997>
- Hong, J., Duan, J., Zhang, C., Li, Z., Xie, C., Lieberman, K., Diffenderfer, J., Bartoldson, B. R., Jaiswal, A. K., Xu, K., Kailkhura, B., Hendrycks, D., Song, D., Wang, Z., & Li, B. (2024). Decoding compressed trust: Scrutinizing the trustworthiness of efficient LLMs under compression. In *Proceedings of the 41st International Conference on Machine Learning (PMLR Vol. 235, pp. 18611–18633)*. <https://proceedings.mlr.press/v235/hong24a.html>
- Kim, H., Hwang, H., Lee, J., Park, S., Kim, D., Lee, T., Yoon, C., Sohn, J., Park, J., Reykhart, O., Fetherston, T., Choi, D., Kwak, S. H., Chen, Q., & Kang, J. (2025). Small language models learn enhanced reasoning skills from medical textbooks. *npj Digital Medicine*, 8, Article 240. <https://doi.org/10.1038/s41746-025-01653-8>
- Korom, R., Kiptinness, S., Adan, N., Said, K., Ithuli, C., Rotich, O., Kimani, B., King'ori, I., Kamau, S., Atemba, E., Aden, M., Bowman, P., Sharman, M., Soskin Hicks, R., Distler, R., Heidecke, J., Arora, R. K., & Singhal, K. (2025). AI-based clinical decision support for primary care: A real-world study. *arXiv preprint arXiv:2507.16947*. <https://arxiv.org/abs/2507.16947>
- Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W., Rocktäschel, T., Riedel, S., & Kiela, D. (2020). Retrieval-augmented generation for knowledge-intensive NLP tasks. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, & H. Lin (Eds.), *Advances in neural information processing systems 33* (pp. 9459–9474). Curran Associates.
- Marcuzzi, F., Ning, X., Schwartz, R., & Gurevych, I. (2025). How quantization shapes bias in large language models. *arXiv preprint arXiv:2508.18088*. <https://arxiv.org/abs/2508.18088>
- Mateen, B. A., Menon, V., Agweyu, A., Korom, R., Omluabi, E., McAfee, D., Shimelash, N., Rutunda, S., Rugege, C., Williams, G., Emmanuel-Fabula, M., Denniston, A. K., Liu, X., & Miles, M. (2025). Trials for LLM-supported clinical decisions in African primary healthcare. *Nature Medicine*, 31(9), 2833–2835. <https://doi.org/10.1038/s41591-025-03815-3>
- Microsoft. (2025). *Foundry Local: Run AI models locally with hardware acceleration* [Computer software]. Microsoft Corporation. <https://github.com/microsoft/Foundry-Local>
- Oche, A. J., Folashade, A. G., Ghosal, T., & Biswas, A. (2025). A systematic review of key retrieval-augmented generation (RAG) systems: Progress, gaps, and future directions. *arXiv preprint arXiv:2507.18910*. <https://arxiv.org/abs/2507.18910>
- Pan, G., Chodnekar, V., Roy, A., & Wang, H. (2025). A cost-benefit analysis of on-premise large language model deployment: Breaking even with commercial LLM services. *arXiv preprint arXiv:2509.18101*. <https://arxiv.org/abs/2509.18101>
- Peppers, K., Tuunanen, T., Rothenberger, M. A., & Chatterjee, S. (2007). A design science research methodology for information systems research. *Journal of Management Information Systems*, 24(3), 45–77. <https://doi.org/10.2753/MIS0742-1222240302>
- Pulkkinen, J., Huttu, K., & Suhonen, M. (2025). Systemic challenges in AI adoption in public social and health organizations in Finland: A technology-organisation-environment perspective. *Journal of Health Organization and Management*, 39(9), 435–456. <https://doi.org/10.1108/JHOM-06-2025-0309>
- Singh, A., Ehtesham, A., Kumar, S., Khoei, T. T., & Vasilakos, A. V. (2025). Agentic retrieval-augmented generation: A survey on agentic RAG. *arXiv preprint arXiv:2501.09136*. <https://arxiv.org/abs/2501.09136>
- Sriram, A., Maheswaran, N., Sundan, B., & Krishnamoorthy, S. (2025). Context-aware retrieval-augmented generation for artificial intelligence in urology. *Cureus*, 17(7), e88167. <https://doi.org/10.7759/cureus.88167>

Tornatzky, L. G., & Fleischer, M. (1990). *The processes of technological innovation*. Lexington Books.

Wampler, D., Nielson, D., & Seddighi, A. (2025). Engineering the RAG stack: A comprehensive review of the architecture and trust frameworks for retrieval-augmented generation systems. *arXiv preprint arXiv:2601.05264*. <https://arxiv.org/abs/2601.05264>

Wang, J., Zeng, Y., Guo, J., Ma, Y., Liu, A., & Liu, X. (2025). SLMQuant: Benchmarking small language model quantization for practical deployment. *arXiv preprint arXiv:2511.13023*. <https://arxiv.org/abs/2511.13023>

Wang, X., Dang, T., Zhang, X., Kostakos, V., Witbrock, M. J., & Jia, H. (2025). HealthSLM-Bench: Benchmarking small language models for mobile and wearable healthcare monitoring. *arXiv preprint arXiv:2509.07260*. <https://arxiv.org/abs/2509.07260>

Wołk, K. (2025). Evaluating retrieval-augmented generation variants for clinical decision support: Hallucination mitigation and secure on-premises deployment. *Electronics*, 14(21), Article 4227. <https://doi.org/10.3390/electronics14214227>