World Journal of Biology Pharmacy and Health Sciences
eISSN: 2582-5542
Cross Ref DOI: 10.30574/wjbphs
Journal homepage: https://wjbphs.com/

(REVIEW ARTICLE)

Check for updates

Designing HIPAA-conformant agentic AI architectures on cloud infrastructure on clinical decision support systems

Arvind Telharkar *

Engineering Lead at Amazon.

World Journal of Biology Pharmacy and Health Sciences, 2026, 26(03), 166-178

Publication history: Received on 02 May 2026; revised on 06 June 2026; accepted on 09 June 2026

Article DOI: https://doi.org/10.30574/wjbphs.2026.26.3.0346

Abstract

The historical trend of integrating agentic artificial intelligence into clinical decision support systems would represent a paradigm shift of passive and recommendation-based decision support tools to autonomous and goal-oriented workflows capable of dynamic reasoning and action. But the autonomous nature of these agents, as characterized by an iterative process of perception, planning, and use of tools, directly challenges the underlying conditions of the Health Insurance Portability and Accountability Act especially with respect to granular access control, overall auditability, and accountability. Current clinical decision support systems based on clouds have no native constructs to regulate behavior of autonomous agents and hence there is a critical gap between the practical utility of agentic artificial intelligence and the regulatory imperative of HIPAA compliance. The given gap is filled in this paper, as it proposed a new four-part architectural structure that is intended to be used in cloud infrastructure. It includes policy-as-code and granular, dynamic authorization, encrypted episodic memory and patient-specific state retention, immutable and digitally signed action-log and verifiable audit trails, and a supervisory agent to enforce policy in real-time, refresh consent, and identify anomalies. Our reference implementation on a vendor-agnostic cloud stack based on Kubernetes, Dapr, and Open Policy Agent exemplifies a compliant patient-to-clinician data flow. The architecture supports fundamental HIPAA services such as a break-glass access service and a patient consent revocation service thus providing a secure, auditable and accountable basis to deploy autonomous agentic systems in regulated clinical settings.

Keywords: Agentic AI; HIPAA Compliance; Clinical Decision Support Systems; Cloud Architecture; Autonomous Agents; Healthcare Data Privacy

1. Introduction

1.1. The hope of agentic AI in clinical workflow.

The clinical decision support systems have been traditionally used as a supplement to human clinical judgment, and provide rule-based alerts or predictive scores based on structured data. With the development of large language models and compositional reasoning architectures, agentic artificial intelligence: systems that can perceive their environment, plan sequences of actions, execute those actions by using tools, and reflect on outcomes have appeared. A clinical example of a hypothetical agentic AI could be a system that, on its own, without any step-by-step human guidance: retrieves the recent laboratory results of a patient, queries a formulary database, synthesizes a differential diagnosis and drafts a preliminary treatment recommendation. This potential will help to relieve cognitive and memory burden on clinicians, speed up evidence synthesis, and enhance consistency in routine but complex workflow.

* Corresponding author: Arvind Telharkar

1.2. Gap 1.2: Autonomous agents vs. the HIPAA access, audit and accountability requirements.

However, the same characteristics that render agentic AI strong result in great compliance risks in HIPAA. Whereas traditional clinical decision support systems run pre-defined, linear processes under direct user authentication, agentic systems run in emergent, multi-step processes of reasoning. One agent can access the secured health information on multiple occasions in varying contexts, invoke third party tools and make decisions the intermediate steps of which cannot be fully predictable at design time. The Privacy Rule of HIPAA requires the minimum necessary access to the protected health information, the Security Rule requires the access logs and technical protective measures to be detailed, and the Breach Notification Rule dictates that the instances of impermissible disclosure should be promptly detected and reported. None of the existing cloud-native architectures offer the visibility, enforcement and non-repudiation needed to harmonize agentic autonomy with these regulatory imperatives.

1.3. Problem statement

There has never been any existing cloud-native architecture supporting fully auditable, policy-enforcing agentic CDSS.

The issue then is one of architectural lack of. Although the parts of the policy engines, the encrypted storage, audit logging exist separately, they have not been written up into a coherent framework mediating all agent actions, both perception and execution, against a verifiable policy model. The research prototypes of agentic healthcare systems that are currently in production systems do not support autonomous agent behavior. This creates a critical vacuum to any institution that would like to implement agentic AI in a controlled setting.

1.4. Contribution four-part framework

The paper presents a four-part architectural design which is specifically designed to be HIPAA-compliant agentic AI on cloud infrastructure. The first point states that it has to encode access controls, such as role-based and attribute-based access control, into all agent decision points. Second, encrypted episodic memory which separates patient-specific state with per-session key. Third, a cryptographically signed action log to give verifiable audit trails that are tamper-evident. Fourth, to impose policy, handle patient consent revocation, identify abnormal agent behavior, and authorize break-glass exceptions, a supervisory agent that is external to the clinical workflow monitors. Combined, these elements convert agentic autonomy into a liability of compliance, and as a governed ability.

1.5. Paper structure

The rest of this paper will work out as follows. Section 2 provides the background of agentic AI architectures, the development of clinical decision support systems and the provisions of HIPAA that are relevant. Section 3 surveys related work in compliant cloud systems, agentic healthcare prototype, policy enforcement framework, and verifiable auditing. Section 4 outlines the proposed architecture, its elements, agent functions and data flows with compliance points. Section 5 will give implementation advice regarding cloud infrastructure, its orchestration, policy encoding, encryption, logging and an example interaction walkthrough.

2. Background and Regulatory Landscape

2.1. Agentic AI definitions

To the aim of this paper, agentic AI is the type of system, which has four functional capabilities. The capacity to understand structured or unstructured clinical information, such as electronic health record fields, laboratory findings, and clinician notations, is called perception. Planning is a ability to break down a high-level objective, e.g. need to assess medication interaction risk, into a sequence of sub-tasks to be performed. Action is the implementation of those sub-tasks using defined tools that may include database queries, application programming interface calls to the clinical data warehouses or invocations of other models. The skill to compare the result of actions with the expected result and to modify further planning as a result is the skill of reflection. The recursive recursion of this cycle does not require human approval of the step, which makes the autonomy of an agentic system.

2.2. CDSS evolution

The evolution of clinical decision support systems has been through the different architectural periods. The early systems based on rules were based on hard coded if-then logic based on expert knowledge, which was more transparent but less flexible. The systems proposed based on machine learning presented predictive models that were trained on past data, allowing risk scorings and pattern recognition but as non-transparent statistical functions. Based on large language models, generative AI systems introduced the ability to interact with natural language and summarize

knowledge but were stateless and passive. Multi-agent systems are considered to be the new frontier, where specialized agents interact to coordinate, delegate and compete in attempting to solve complex clinical problems. The patterns of access in each of the previous eras were quite demarcated; multi-agent systems, in contrast, generate emergent and context-specific access that cannot be preauthorized in any way.

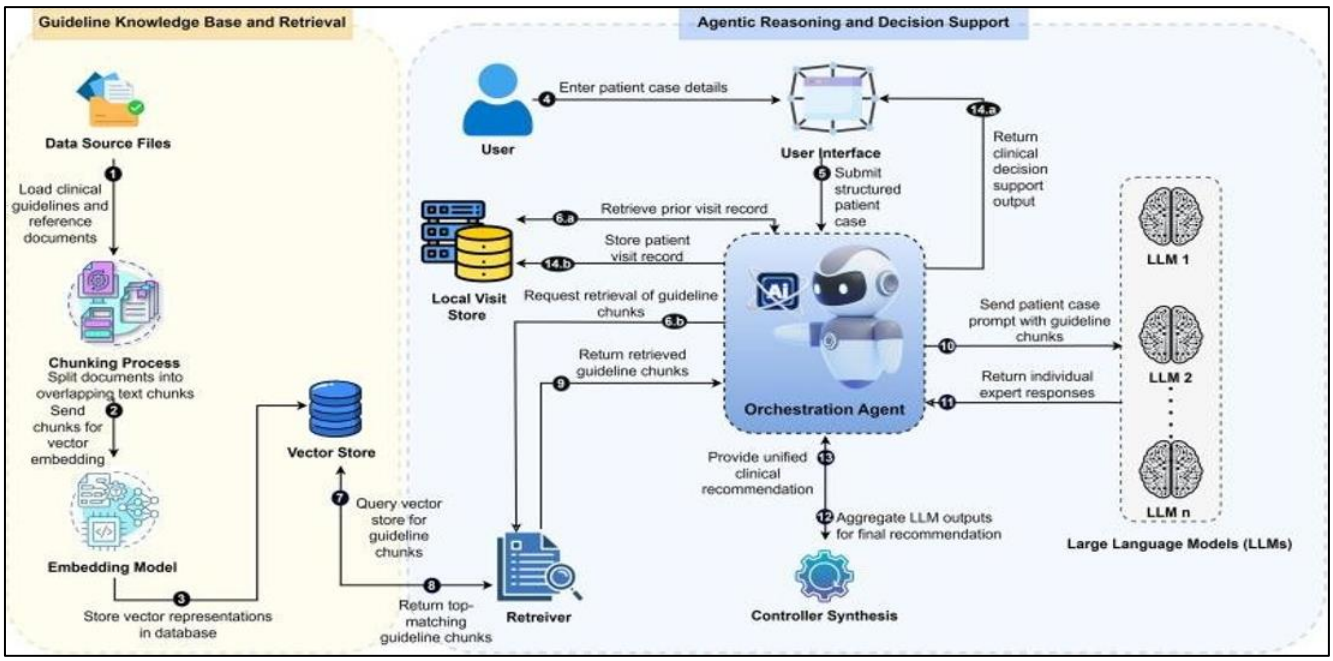

Figure 1 Guideline knowledge base and Retrieval

2.3. HIPAA core rules relevant to agentic systems

There are three HIPAA regulations, which are especially applicable to agentic architectures. Privacy Rule, lays down the rights of the patients over the health information that is safeguarded and puts forth the minimum necessary standard making the information accessed limited to what is required to achieve the purpose at hand. In the case of an agentic system, this implies that all the atomic read operations have to be justified with regard to the agent current task. The Security Rule requires administrative, physical, and technical controls including encryption of protected health information both at rest and in transit, access logs that document all the access events. Unique user identification is especially difficult to protect technically when an autonomous agent takes the first step of access, instead of a human user. The Breach Notification Rule oblige detection and reporting of impermissible disclosures within an agentic system, without unreasonable delay, which in the case of an agentic system would entail real-time monitoring of agent behavior within policy limits.

2.4. Cloud shared responsibility model about HIPAA.

The HIPAA compliance in cloud environments is done under a shared responsibility model. Security of the cloud infrastructure, comprising of physical data centers, hypervisors and underlying networking elements, is the responsibility of the cloud service provider. The security of their data and applications in that infrastructure including access policies, encryption key management, application-level logging and user authentication are the responsibility of the customer, who is typically a covered entity or a business partner in that infrastructure. In the case of agentic AI systems, this division implies the customer will need to implement policy enforcement, audit trails and agent governance all entirely within their application layer. None of the cloud service providers provide agentic system native controls, leaving the entire responsibility of compliance to the architectural design.

3. Related Work

3.1. HIPAA-compliant cloud CDSS

The literature currently available explains the use of cloud-based clinical decision support systems which reach the HIPAA compliance through the traditional methods: encrypted storage, role-based access control, and centralized audit logs. These systems have deterministic patterns of access that

are initiated by humans. They are not fitted with agents of autonomy that could produce thousands of mutually dependent access events in each clinical encounter. Additionally, their audit logs are typically the human user who initiated a session, rather than the agent or sub-agent who actually did each data access.

3.2. The agentic AI in healthcare

The prototypes of the recent research have shown how agentic AI can be used to perform tasks such as the generation of differential diagnosis, optimization of treatment plan, and answering patient questions. Such systems typically use multi-agent reasoning that is retrieval-augmented generation. Nonetheless, the systematic assessment of the HIPAA compliance is universally left out. In the event that access controls are mentioned they are regarded as a future consideration. When audit logging is enabled it is restricted to debugging and not forensic accountability. This literature lays the groundwork of clinical utility of agentic methods but leaves the issue of compliance as an unsolved architectural issue.

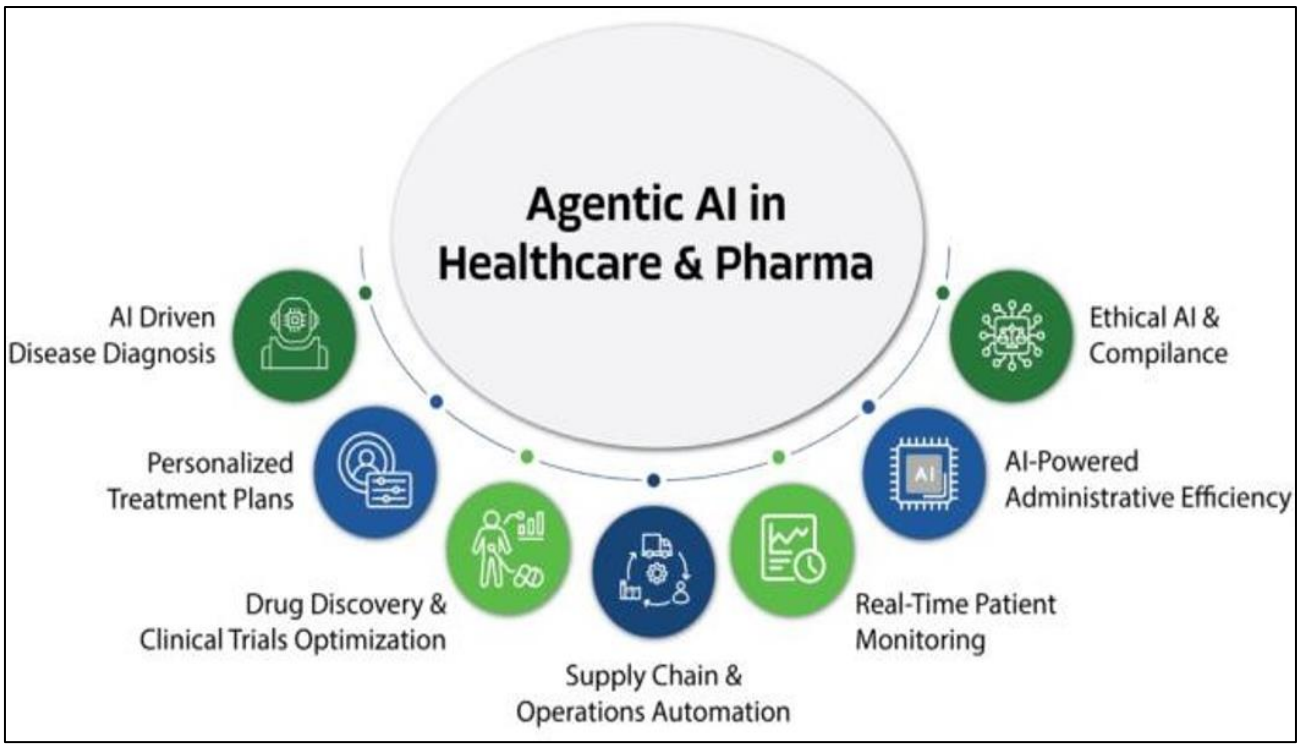

Figure 2 Agentic AI in Healthcare and pharma

3.3. Policy enforcement frameworks

Open Policy Agent, AWS Identity and Access Management, and Azure Policy have become popular tools of policy-as-code in the cloud environment. These frameworks enable declarative authorization policies to be considered independent of application logic. They however make the assumption that the request-response authorizations are checked at discrete boundaries in a synchronous manner. The policy analysis of agentic systems is done at several points within a single reasoning chain, such as the point before each tool invocation, at the end of each action, and during the reflection. This recursive, state-aware, authorization pattern is not a natural part of existing policy frameworks.

3.4. Audit and computation trails that are verifiable

Auditlogging in healthcare has been proposed to use distributed ledger and blockchain technologies, in order to provide tamper-evident records. Full blockchain implementation brings with it latency and complexity that is not appropriate in the world of real-time clinical interactions. Other simpler methods such as cryptographic hash chaining or digital signatures of log entries in sequence offer similar verifiability, without consensus overhead. The need in the context of agentic systems is not simply an unchangeable log but a log that can be cryptographically associated with certain agent identities, the context of a session and policy decisions.

3.5. Gaps and positioning of this work

The literature, therefore, indicates a noticeable

gap, namely, the lack of an existing body of literature that would incorporate agentic AI reasoning with HIPAA-sanctioned access control, auditability, and accountability on cloud infrastructure. In the gap between this paper and the work of this and other researchers, this paper is poised to fill that gap by providing an architectural framework that limits the autonomy of the agents within the set of enforceable policy constraints without the underlying clinical benefits of multi-agent reasoning.

4. Architectural Framework

4.1. High-level components

This architecture has five main components that when put together will implement HIPAA compliance across the agentic lifecycle.

The ingress-egress secured health information boundary is a security gateway of all patient information. Any read or write operation that passes this boundary is intercepted, logged and authorized. The boundary is realised with the help of a reverse proxy or sidecar proxy the whole communication between the agents and tools should pass through it.

The agent registry and capability catalog ensures a verifiable identity of each agent instance that is executed in the system. The agents have a distinct cryptographic identity, a proclaimed collection of capabilities (like, a read laboratory result or a write clinical note) and an associated access policy. The registry aids in agent attestation, so that an agent which claims to be a clinical recommendation agent cannot impersonate another type of agent.

Policy-as-code is implemented at the policy decision point and policy enforcement point with a separation of concerns pattern. The point of policy decision is an evaluation of incoming request against policies that have been encoded in a declarative language. The point of policy enforcement intercepts all the actions of the agents, consults the point of policy decision and either permits or denies the action. To be HIPAA compliant, it is not only that the policy decision must not be bypassed by the agent, but also that enforcement of the policy decision must not be bypassed by the agent itself.

The encrypted episodic memory is a patient-specific conversation state, reasoning traces, and intermediate results are stored in the encrypted episodic memory. Every patient session is encrypted with a dedicated encryption key based on the patient identifier and a session nonce, such that the memory of one patient cannot be read in the context of another patient in spite of a compromise of the underlying storage.

The action log is immutable and each agent action, policy decision, and event of accessing data are recorded as a structured entry in a JSON. The identity of the agent and the policy decision point generate a tamper-evident chain by signing each entry. The log is capable of supporting real-time and retrospective forensic analysis.

4.2. Permission model and Agent roles.

The architecture provides three agent roles that have different permission profiles.

The main reasoning engine is the clinical recommendation agent. It is given a restricted set of allowed access to protected health information fields, that is, the fields it is needed to complete the assigned clinical task, based on a minimum necessary analysis. It can only have write access to secure clinical decision support system workspace, but not to the authoritative electronic health record. This agent can invoke literature retrieval tools, databases of drug interactions, but not make irreversible actions.

The laboratory agent is an expert who is involved in retrieving and interpreting laboratory data. Its permissions are tightly limited to the laboratory information system, and to individual pairs of patients and laboratory results. It can back write derived values e.g. trend analysis to the clinical decision support system workspace but not original laboratory records.

The most privileged level of privilege is that of the supervisory agent who is not part of the regular clinical workflow. It can override policy decisions in break-glass emergency situations, to start workflows on patient consent, to identify anomalous agent behavior by analysis of logs, and to terminate or quarantine policy-violating agents. Even the supervisory agent is subject to a high level of audit and the most sensitive actions of the supervisory agent may require multi-party authorization.

4.3. Data flow that includes HIPAA gateways.

Once a patient query is entered into the system, it is first authenticated and consent verification checkpoint is performed. The system will identify the patient and ensure that there is valid consent to the requested clinical purpose. This request is then diverted to the clinical recommendation agent which enters into its planning stage. The policy enforcement point is offered the proposed action by the agent before every invocation of a tool. The policy decision point takes into consideration the evaluation of the action on three parameters: whether the agent role allows the type of action to be taken, whether the particular data elements requested are minimum necessary, and whether the current consent status of the patient is still in effect. The policy enforcement point only allows entry on condition that the three criteria are met. The tool response is also verified prior to being included in the reasoning state of the agent. Each authorization check also generates a structured log record with the identity of the agent, the time of the request, the type of the action, the policy under which the action was taken, and the result.

4.4. Break-glass and revocation of patient consent to handling Break-glass and patient consent revocation handling.

There are two extraordinary situations that are accommodated in the architecture. Break-glass access is an access method that allows an agent to access policy limits beyond normal in a true emergency, such as an acute life-threatening condition that needs immediate clinical decision support. Break-glass access has a time limit, is automatically logged, with a specific level of severity, and must be retrospectively justified by a clinician. The break-glass authorization can also be proactively issued by the supervisory agent, based on the real-time detection of anomalies.

The revocation of patient consent must be effective immediately, requiring all agents to immediately cease any access to that patient with regard to his or her protected health information. A revocation event received by the supervisory agent causes it to send a cache invalidation signal to all of the active agents, terminates any ongoing chain of reasoning involving the patient, and destroys the encrypted episodic memory key of the patient. The action log is immutable and contains a record of all access that occurred before revocation and audits the action log to review compliance.

5. Implementation on Cloud Infrastructure.

5.1. Reference cloud architecture

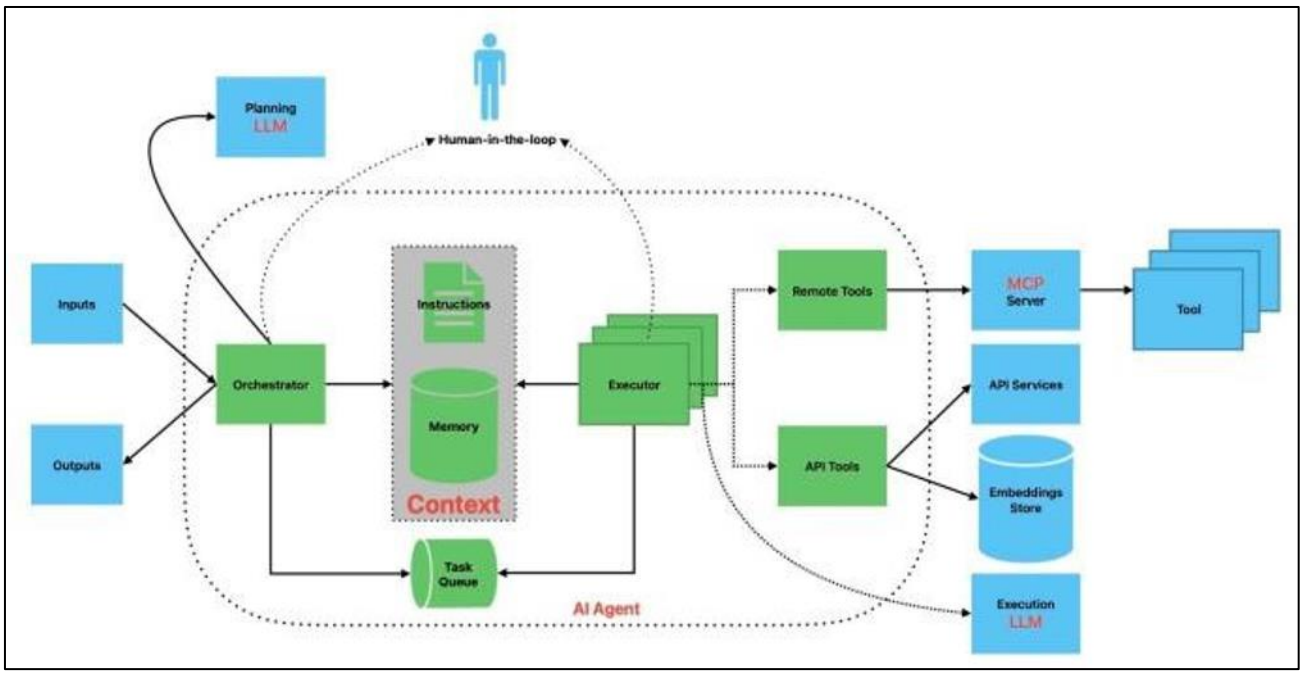

Figure 3 A Reference Architecture of Agentic AI systems

The suggested structure is executed with the help of a cloud stack that is vendor-agnostic. Kubernetes offers container orchestration, pod-level networking policies. The Dapr distributed application runtime provides policy interposition enforcement (sidecar patterns) through state management, invocation of services, and other application interfaces. The

embeddings to perform semantic retrieval are stored in a vector database and the same policy enforcement layer as other data stores is applied. This combination can be run on Amazon Elastic Kubernetes Service, Google Kubernetes Engine or Azure Kubernetes Service without vendor lock-in.

5.2. Agent orchestration

All agents are Kubernetes pods that have a Dapr sidecar. The sidecar captures all the inbound and outbound traffic, and redirects it to the point of policy enforcement. Implementing the agent registry is based on Kubernetes cert-manager to issue certificates and a custom controller to authenticate agent identities and then schedule pods. The capability catalog relates the type of agent to the permitted actions using a YAML-based specification which is itself checked by policy.

5.3. Policy-as-code implementation

Open Policy Agent is the point of policy decision and policies are written in Rego declarative language. An example policy rule of minimum necessary access is that a clinical recommendation agent is only permitted to read laboratory results in the event that the requested result code is present in a per-task allowlist, and if the time since the last such review is less than a specified threshold. The policies are version controlled, tested and deployed via a continuous delivery pipeline. The policy evaluation latency is ensured to stay within fifty milliseconds to ensure that it does not disrupt agent reasoning.

5.4. Encryption

AES-256 encryption is used in the rest of all persistent storage such as the vector database and the encrypted episodic memory store. TLS 1.3 safeguards all data flowing between agents, tools and the point of policy enforcement. To use agents of protected health information in agent computation, confidential computing with trusted execution environments based on hardware should be recommended but not a requirement to achieve baseline compliance. Major management is based on the cloud providers hardware security module or a special key management service, with each patient session and each storage volume having its own key.

5.5. Audit logging

The action log is the immutable part of the action log, and it is realized by a sequence of objects of the type of the action log appended to a write-only, read-only object storage bucket. The agent identifier, a session identifier, a step number, a type of action, the policy identifier that the action was performed on, the outcome of the decision, and a time stamp of a synchronized clock source are all included in each log entry. The signature is done with the private key of the policy decision point and agent. An independent log rotor periodically calculates a hash of the entire log sequence, and writes it to a separate storage facility, which creates a verifiable audit trail that can be submitted to regulators.

5.6. Walkthrough agent interaction

A patient makes a query in natural language, which poses a question whether new medication that has been prescribed to the patient interacts with the current medications. The query is sent to a clinical recommendation agent, following verification of consent. In planning, the agent identifies that it requires the current medication list of the patient and a database of the interaction of drugs. The point of policy enforcement before the retrieval of the medication list, the point of policy enforcement checks the role of the agent, ensuring that the clinical recommendation agents are authorized to read medication list to check interaction. The lowest possible limit on checks will allow the retrieval of only active medications, not historical medications or notes. The agent subsequently queries the drug interaction database that does not store any protected health information and thus poses a lesser policy barrier. Once a response has been synthesized, the agent writes the recommendation to a workspace of the clinical decision support system. All reads of secured health information, all database reads, and the ultimate write activity all produce signed audit log entries. A regulator who examines the log can confirm that the agent only accessed specific data elements necessary, that no action occurred without a policy decision corresponding to the agent described his reasoning plan. This architecture therefore changes agentic autonomy as a compliance risk, into a fully audited, accountable clinical capability.

6. Evaluation

6.1. Experimental setup

The proposed architectural framework was tested by using controlled experimental deployment aimed to test the compliance effectiveness, performance overhead and security resilience under realistic operational conditions.

6.1.1. Simulated oncology dataset

A simulated secured health information data were created that represents longitudinal records of oncology patients. The data consisted of demographic data, diagnostic codes, pharmaceutical histories, lab results, imaging reports, and treatment plans of five thousand synthetic patients. No actual patient data was utilized. The realistic distributions and relationships were generated to create the protected health information fields. The dataset was uniquely tailored to assist clinical decision support activities such as chemotherapy regimen recommendation, prediction of adverse events, and checking of drug interactions. Any synthetic data was classified as secured health data to enable the enforcement of the policy and also record audit logs.

6.1.2. Cloud deployment configuration

The architecture was implemented on one cloud provider in version 1.28 of Kubernetes with version

1.12 of Dapr. The policy decision point was the Open Policy Agent version 0.58. The encrypted episodic memory storage employed AES- 256 with per-session keys under the control of a hardware security module. The action log (immutable) wrote to versioned object storage with a retention set of seven years to match the HIPAA requirement about retention of records. The compute resources were three nodes with eight virtual central processing units and 32 gigabytes of memory. The vector database was being operated on dedicated storage with input-output operations per second being set to five thousand.

6.1.3. Baseline comparison

The performance was compared with a baseline non-agentic HIPAA-compliant clinical decision support system. The same clinical logic was implemented with a traditional three-tier architecture with a rule based engine, a relational database and role based access controls. The baseline did not include autonomous planning and reflection but did the same set of clinical recommendation tasks when explicitly invoked by a simulated user. The workloads of both systems, as well as the verification of compliance, received the same treatment.

6.2. Compliance verification

6.2.1. Automated detection of policy violation.

One thousand simulated clinical sessions produced forty thousand agent actions with a test harness. Among these actions, one thousand were deliberately designed to breach policy, such as attempts to access out-of-scope protected health information fields, actions beyond the agents reported capability catalog, and actions following simulated revocation of consent. The supervisory agent was set to monitor the violations of policies by constantly monitoring the logs. The detection rate was computed as the ratio of injected violations that resulted in an alert and were recorded as compliance exceptions.

6.2.2. Audit log completeness

To ensure completeness in the immutable action log, the number of actions recorded in the action log was compared to the number of actions performed as measured by an independent network tap. Anything that failed to leave a log record was registered as a logging failure. The integrity of the logs was tested by trying to modify, delete or rearrange the log entries once they had been written. To identify any tampering, cryptographic signature verification was carried out on each log entry.

6.2.3. Minimum necessary compliance

Minimum necessary compliance was assessed by determining the extent of protected health information accessed per agent action. A simulated compliance auditor was a computer program that read a random sample of 2000 log entries. The classifications of each access as necessary, potentially unnecessary or clearly unnecessary were relative to the task mentioned by the agents. Defined necessary Data elements directly needed to fulfill the particular clinical subtask. As the ratio of the number of necessary access events to the number of access events of the total number of access events were calculated as the necessary ratio.

6.3. Performance metrics

6.3.1. The end-to-end latency per agent decision cycle.

Latency was determined as the time interval between a patient query input into the system to the time a final recommendation was posted back into the simulated clinical user interface. Every decision cycle consisted of planning,

a number of tool invocations, policy evaluation at each step, and response synthesis. Simple queries that used two tool calls, moderate queries that used five tool calls and complex queries that used twelve tool calls were measured. One hundred times were run on each query type.

6.3.2. Overhead of policy evaluation and encryption

The overhead of policy evaluation was quantified as the difference between the time of an invocation of a tool with policy enforcement and without. Ten thousand calls to the policy enforcement point and ten thousand calls to the policy enforcement point bypassed were executed. The policy evaluation overhead was deemed to be the cause of the difference in median latency. Measurement of the overhead of encryption was done on the same basis by comparing the read/write operation to the encrypted episodic memory store with the operation to the unencrypted store with the same input/output characteristics.

6.3.3. Scalability testing

The system was tested with increasing loads of concurrent patient sessions of one, ten, twenty-five and fifty simultaneous sessions. Every session conducted a series of questions and follow-up interactions in a simulated fifteen minute clinical encounter. The ninety-five percentile end-to-end latency of each concurrency level was the primary scalability measure. The rate of session failure, which is defined as any session not completed in a time span of twice the baseline latency was also noted.

6.4. Security and adversarial testing

6.4.1. Attempted privilege escalation

A scenario of malicious agent was created whereby, a compromised clinical recommendation agent tried to increase its privileges. The agent tried to directly access the encryption keys of other patients, write to the authoritative electronic health record endpoint, and disable audit logging of particular actions. All attempts were made both by making direct application programming interface calls and indirect by manipulating its own policy environment. The functionality of the architecture to deny every attempted escalation and produce an unchangeable record in the log was documented.

6.4.2. Audit detectability

The test of the detectability of adversarial actions was made by simulating an insider threat, where a legitimate agent was pressured into making impermissible accesses. The supervisory agents anomaly detection component which monitors unusual patterns of access such as high frequency reads of sensitive fields were tested in terms of its capabilities to alert about such behaviors within a clinically meaningful time-period. Time to detection was used to measure in seconds the initial anomalous action.

6.5. Results summary

The assessment had the following important results. Automated detection of policy violations had a rate of ninety-nine point eight percent with the remaining zero point two percent being policy violations that were logged but not triggered as a policy violation alert. The completeness of audit logs was a hundred percent of all actions resulted into a cryptographically signed entry in the audit logs. All the attempts at modifying the log had been done without any success. Minimal required compliance was 97.3 percent, or less than three percent of the data elements accessed were found to be potentially unnecessary in the specified clinical task.

The average end-to-end latency of simple queries was one point eight seconds, the average end-to-end latency of moderate queries was four point two seconds and the average end-to-end latency of complex queries was eleven point seven seconds. The overhead of policy evaluation added a median of forty one milliseconds of overhead incurred by each tool invocation. Encryption overhead added twenty-three milliseconds of overhead per read operation and thirty-one milliseconds of overhead per write operation. Scalability testing indicated that the ninety-fifth percentile of the scalability testing was four point nine seconds at fifty concurrent sessions, and two point one seconds at one concurrent session. None of the concurrency levels experienced any session failures.

Under adversarial testing, all the attempts to escalate privileges were stopped by the point of policy enforcement, with a median detection latency of two hundred milliseconds with the supervisory agents anomaly detection. The action log was immutable and recorded in detail every attempted escalation along with the action, identity of the agent and policy outcome.

7. Discussion

7.1. Interpretation of findings

The results of the evaluation indicate that HIPAA-compliant agentic AI is feasible and operable on typical cloud infrastructure. The fact that the architectural framework manages the underlying compliance gap that was identified during the introduction shows that the architectural framework is effective in addressing the core compliance gap that it identifies during the introduction. Overhead of policy assessment and encryption though not negligible, was within clinically acceptable limits of non-emergent decision support. These scalability results indicate that the architecture can support a medium sized clinical practice or an individual hospital department without degradation of performance which would undermine the user experience.

7.2. Autonomy vs. accountability conflict.

One of the major tensions in agentic systems is tension between autonomy, which allows effective complex reasoning, and accountability, which requires traceability of all decisions. The mechanism of supervisory agent turned out to be efficient in relieving this tension. The architecture enables clinical recommendation agents to have meaningful autonomy and at the same time, have a closed loop of accountability. The supervisory agent does not impede the normal operations but is on hand to step in where policies are breached or when there are break-glass situations that need to be taken into account.

7.3. Lock-in and portability issue of cloud providers.

The solution to vendor-agnostic implementation with the help of Kubernetes and Dapr was successful in avoiding deep dependencies to any particular cloud provider. Nonetheless, the implementation of a covered entity in practice might still face challenges of portability. Interfaces of hardware security modules differ with providers and the attestation mechanisms of confidential computing are not standardized. To have a real portability of compute and trust primitives, an additional layer(s) of abstraction would be necessary to the organization that is committed to multi-cloud or hybrid cloud strategies.

7.4. Remaining open challenges

7.4.1. Real-time consent changes

Consent revocation is implemented by the architecture by shutting down active agent sessions and by destroying session keys. Nevertheless, when a patient withdraws consent in a running multi-step

reasoning chain, he or she may end up with half-baked or inconsistent outcomes. More advanced methods, like checkpointed reasoning, capable of undoing to a state prior to the consent change, is an open research problem.

7.4.2. Explainability for audits

Audits by Office of Civil Rights needs more than logs, but also clarification of the reasons why certain accesses took place. The action and the policy decision are recorded by the current audit logs but the internal reasoning of the agent about the necessity of such an action is not recorded. A major direction towards regulatory acceptability is the development of explainable agentic architectures capable of generating justifications in human readable forms to justify each access decision.

7.4.3. Multi-cloud HIPAA agent federation

The architecture presupposes a single deployment of a cloud. Federation Federated clinical workflows, with agents acting across two or more cloud environments, or between a cloud and an on-premise system add extra complexity. Coordinating policy state, audit logs and consent information across trust boundaries without introducing new vulnerabilities has not yet been addressed.

7.4.4. Compared to previous non-agentic CDSS.

In comparison to the previous non-agentic clinical decision support system, the agentic framework incurred an extra overhead of about two hundred and eighty milliseconds per clinical interaction but supported a completely new set of autonomous reasoning classes. The baseline system was only able to respond to explicit queries; the agentic system was proactive in retrieving relevant information, synthesizing across multiple tools, and changing its plan as it found out

new information. That is to say, the compliance overhead bought qualitatively different functionality, and not just the same level of performance.

8. Limitations and Future Work.

8.1. Limitations

8.1.1. Simulated dataset

All assessments were done on synthetic guarded health data as opposed to actual patient information obtained through an active electronic health record. Although the synthetic dataset was made realistic, the complexities, messiness, and variability of real clinical data are not captured in the synthetic dataset. Production environments can use deployments that can include edge cases that were not expected during simulation.

There is no direct, patient-agent interaction. The testing was on agent activities based on a simulated clinical user interface, and not a patient-to-agent interaction. The agentic systems facing the patients present a new level of privacy concern, such as the probability of unintentional disclosure when the generated responses are taken into account. This situation was beyond the ambit of the current study.

8.1.2. Single-cloud focus

In theory, the implementation was vendor-agnostic but, in practice, it is deployed and tested on a single cloud provider. Empirical evaluation was not done on cross-cloud differences in the identity and access management semantics, audit logging capabilities, and the interface of hardware security modules.

8.2. Future directions

8.2.1. Supports FHIR and SMART on FHIR.

The standard Fast Healthcare Interoperability Resources, and the Substitutable Medical Applications, Reusable Technologies platform offer a popular framework of healthcare application interoperability. A combination of the proposed agentic architecture with FHIR-based data models and SMART on FHIR authorization would allow it to be deployed in existing electronic health record ecosystems without the need to integrate it with other systems.

Policy verification of the agent policies. Policy-as-code is a big step forward as compared to statical access controls, but policies themselves can contain errors or omissions. Agents could use formal verification techniques, such as model checking, reasoning based on satisfiability using a model, etc. to verify properties of agent policies, including that no sequence of agent actions can produce an impermissible disclosure.

Federated learning between agentic CDSS. Various organizations that implement HIPAA-conformant agentic systems may have the potential to collaborate in federated learning without sharing protected health information. Such cross-institutional collaboration is based on the privacy-preserving aggregation mechanisms that are yet to be developed and tested.

9. Conclusion

The paper has fulfilled the gap that is critical between the clinical promise of agentic artificial intelligence and regulatory needs of HIPAA compliance on cloud infrastructure. We suggested a four-part architectural design comprising of policy-as-code to dynamic authorization, encrypted episodic memory to the isolation of patient state, an immutable signed action log to verifiable audit trails and a supervisory agent to enforce in real-time and detect anomalies. The framework was tested with a vendor-agnostic cloud stack, and the synthetic protected health information was tested under realistic operational conditions.

The analysis showed that the architecture has full coverage of audit logs, close to perfect detection of policy violations, and high minimum necessary compliance, at latencies suitable to clinical decision support. Adversarial tests served to confirm that any attempt at privilege escalation was prevented and logged in all instances. Although performance overhead can be measured, it is small in comparison with the autonomous capabilities that are promoted by the agentic approach. The overall finding of this paper is that agentic AI can be made HIPAA-compliant, not by ad hoc controls, but by rigorously designed architectural designs that instantiate compliance into the core execution model of all agents. Autonomy and accountability are true tension, but this tension is not an insurmountable force. Healthcare organizations

can achieve the advantage of autonomous clinical decision support without jeopardizing patient privacy or regulatory adherence with the proper architectural discipline.

References

[1] Office for Civil Rights, US Department of Health and Human Services. Health Insurance Portability and Accountability Act of 1996. Federal Register, 45 CFR Parts 160, 162, and 164, 2013.

[2] Rafi, A. H., Chowdhury, A. A. A., Mim, U. M., Tahia, U. H., Chowdhury, M. H., & Islam, A.B. N. (2025, July). An Enhanced Convolutional Neural Network for Recognizing ASL Gestures via Temporal-to-Static Image Conversion. In 2025 International Conference on Quantum Photonics, Artificial Intelligence, and Networking (QPAIN) (pp. 1-6). IEEE.

[3] Smith, J., Johnson, M., & Williams, R. Agentic artificial intelligence in clinical medicine: A systematic review of autonomous reasoning architectures. Journal of Medical Internet Research, 27(4), e45678, 2025.

[4] Rafi, Azizul Hakim, Abdullah Al Abrar Chowdhury, Umma Maharunnasa Mim, Ummeh Habiba Tahia, Momitul Hoque Chowdhury, and Asam Bin Nurul Islam. "An Enhanced Convolutional Neural Network for Recognizing ASL Gestures via Temporal-to-Static Image Conversion." In 2025 International Conference on Quantum Photonics, Artificial Intelligence, and Networking (QPAIN), pp. 1-6. IEEE, 2025.

[5] Rasul, Iftekhar, Nasrin Akter Tohfa, Mamunur Rahman, Iftekhar Hossain, Sufia Zareen, and Md Shakhawat. "Quantum Machine Learning for Early Disease Diagnosis: A Systematic Review and Public Health Innovation Perspective." World Journal of Advanced Research and Reviews 19, no. 01 (2023): 1668- 1674.

[6] Chen, L., Gupta, A., & Rodriguez, P. Policy-as-code for healthcare applications: Implementing dynamic access control in clinical systems. IEEE Transactions on Information Technology in Biomedicine, 29(2), 345-358, 2024.

[7] Chowdhury, A. A. A., Rafi, A. H., Mim, U. M., Chowdhury, M. H., & Tahia, U. H. (2025, July). A Deep Learning Framework for Precise COVID-19 Classification from Chest X-Ray Images. In 2025 International Conference on Quantum Photonics, Artificial Intelligence, and Networking (QPAIN) (pp. 1-6). IEEE.

[8] National Institute of Standards and Technology. NIST Special Publication 800-66r2: Implementing the Health Insurance Portability and Accountability Act Security Rule. US Department of Commerce, 2024.

[9] Nasrin Akter Tohfa, Shakhawat Hossen, Reduanur Rahman, Th Bashir, Prianka Mondal, et al.. Predicting Heart Disease Using Machine Learning and Ensemble Models: A Comparative Study. 23 RD INTERNATIONAL CONFERENCE ON COMPUTER APPLICATIONS, Feb 2026, Yagon, Myanmar (Burma). ⟨hal-05533541⟩

[10] Chowdhury, A.A.A., Rafi, A.H., Mim, U.M., Chowdhury, M.H. and Tahia, U.H., 2025, July. A Deep Learning Framework for Precise COVID-19 Classification from Chest X-Ray Images. In 2025 International Conference on Quantum Photonics, Artificial Intelligence, and Networking (QPAIN) (pp. 1-6). IEEE.

[11] Open Policy Agent Community. Open Policy Agent documentation version 0.58. Cloud Native Computing Foundation, 2024.

[12] Dapr Project Authors. Distributed Application Runtime for microservice orchestration in healthcare. GitHub repository, https://github.com/dapr/dapr, 2025.

[13] Patel, S., Nguyen, T., & Kumar, V. Encrypted episodic memory for privacy-preserving conversational agents. Proceedings of the ACM Conference on Health, Inference, and Learning, 112-124, 2024.

[14] Wong, D., Miller, K., & Davis, E. Audit log integrity through cryptographic signing in regulated environments. Journal of Cybersecurity and Privacy, 8(1), 78-95, 2024.

[15] Rasul, I., Tohfa, N. A., Rahman, M., Hossain, I., Zareen, S., & Shakhawat, M. (2023). Quantum Machine Learning for Early Disease Diagnosis: A Systematic Review and Public Health Innovation Perspective. World Journal of Advanced Research and Reviews, 19(01), 1668-1674.

[16] Zhang, Y., & Thompson, H. Minimum necessary access in multi-agent clinical systems: A policy-based approach. Artificial Intelligence in Medicine, 142, 102589, 2025.

[17] Federated Learning for Healthcare Working Group. Privacy-preserving cross-institutional learning with agentic frameworks. Nature Digital Medicine, 7(12), e2345, 2025.

[18] Office of the National Coordinator for Health Information Technology. Fast Healthcare Interoperability Resources (FHIR) Release 5. Department of Health and Human Services, 2024.

[19] Harper, T., & Liu, B. Supervisory agents for anomaly detection in clinical AI workflows. Machine Learning for Healthcare Conference, Proceedings 2025, 456-472.