

AgentCare-Guard: A Capability-Constrained Reference Monitor for Agentic FHIR Transactions

Adilakshmi Kataru

`adilakshmi.kataru@ragsec.ai`

Independent Researcher

Research Article

Keywords: Agentic artificial intelligence, FHIR, capability-based authorization, clinical cybersecurity, human-in-the-loop, zero trust

Posted Date: September 2nd, 2026

DOI: <https://doi.org/10.21203/rs.3.rs-10728193/v1>

License:   This work is licensed under a Creative Commons Attribution 4.0 International License.
[Read Full License](#)

Additional Declarations: No competing interests reported.

AgentCare-Guard: A Capability-Constrained Reference Monitor for Agentic FHIR Transactions

Adilakshmi Kataru

Independent Researcher, New Jersey City, New Jersey, United States

ORCID: 0009-0004-1335-0808

Email: adilakshmi.kataru@ragsec.ai

Acknowledgements

The author thanks the maintainers of HL7 FHIR, SMART Health IT, SPIFFE, scikit-learn, and the researchers whose standards, software, and publications supported this work.

Abstract Agentic artificial intelligence is moving healthcare automation beyond summarization toward systems that can create or modify Fast Healthcare Interoperability Resources (FHIR). Existing authorization mechanisms constrain access by user, patient, resource, and operation, but they do not define a transaction lifecycle that binds autonomous writes to an approved payload, clinical risk, current record state, and one-time execution context. AgentCare-Guard addresses this gap with an external reference monitor that treats each model-generated tool call as an untrusted proposal. The monitor verifies workload identity and a short-lived capability, checks patient and resource scope, derives an action risk tier, binds approval to the exact payload and expected version, consumes replay-prevention records atomically, and records a reason-coded AuditEvent-style trace. A fixed-seed Python evaluation produced 2,550 decisions over 150 synthetic patients, comprising 600 legitimate workflows, 1,500 authorization-boundary attacks, and 450 residual attacks. AgentCare-Guard accepted all encoded legitimate workflows and denied every boundary attack; role-only, SMART-like, and ABAC-like comparators allowed 100%, 70%, and 60% of those attacks, respectively. Six targeted ablations reopened the attack families controlled by the removed mechanism. Across 100 concurrent replay rounds with 20 workers, the atomic registry permitted exactly one execution per round, whereas a deliberately non-atomic variant produced duplicates in 73% of rounds. The results support the feasibility of transaction-level mediation for agentic FHIR workflows under controlled conditions while leaving semantic prompt injection, clinical correctness, regulatory compliance, and production FHIR conformance outside the claim.

Keywords Agentic artificial intelligence; FHIR; capability-based authorization; clinical cybersecurity; human-in-the-loop; zero trust

1 Introduction

Healthcare AI systems are beginning to do more than summarize records or recommend next steps. Emerging agents can plan multi-step workflows, retrieve context, call external tools, and persist changes to electronic health records. FHIR provides the resource and API semantics for those interactions, while SMART-on-FHIR and OAuth provide delegated application access (HL7 International 2019a; HL7 International 2024a; Hardt 2012; Campbell et al. 2020; Fett et al. 2023). Those mechanisms are essential, but possession of a valid token does not establish that an autonomous workload should carry out a particular write at a particular moment. The workload may be over-privileged, influenced by untrusted content, associated with the wrong patient, acting on an outdated approval, or replaying a request that was valid only once.

The distinction matters most for resources such as MedicationRequest, ServiceRequest, Condition, and AllergyIntolerance, where a write can affect downstream care. Resource and operation scopes narrow what an application can access, but an autonomous transaction also needs authority tied to the workload, delegating subject, patient, purpose, exact payload, risk level, record version, validity period, and execution instance. Sender-constrained OAuth reduces exposure to token theft and replay (Campbell et al. 2020; Fett et al. 2023); the remaining decision—whether this specific action is authorized now—belongs to an application-level policy boundary.

Work on healthcare agents has proposed attribute-based access control, PHI protection, encryption, tamper-evident logging, governance registries, runtime isolation, and emergency revocation (Tripathi et al. 2026; Neupane et al. 2025; Shehab 2025; Prakash et al. 2026; Maiti 2026). Research on general-purpose agent security has separately developed complete mediation, pre-action authorization, information-flow control, controlled-write protocols, idempotency, and state-witness checks (Uchibeke 2026; Costa et al. 2025; Christodorescu et al. 2025; Xiang et al. 2026; Zhu et al. 2026; Chhabra et al. 2026; Ghosh et al. 2025). The unresolved issue is therefore

narrower than basic FHIR access control: clinical agent systems still lack an integrated execution lifecycle that binds authorization to the payload, risk, state, and one-time transaction context.

From a health-informatics perspective, the problem is not limited to access control. An agentic transaction links a clinical workflow, a patient-specific resource state, a delegated identity, a model-generated action, and a server-side commit. A useful control therefore has to model the transition from proposal to approved state-changing transaction, not only whether a token permits access to a resource. AgentCare-Guard treats that transition as an explicit and auditable state path so that identity, scope, clinical risk, approval, freshness, and replay status can be evaluated at the transaction boundary.

AgentCare-Guard places a deterministic reference monitor between the reasoning agent and the FHIR execution endpoint. The agent may propose an action, but it cannot authorize that action. Patient-scoped retrieval supplies supporting evidence and policy context, yet remains outside the execution decision. The work makes five contributions:

- An action-capability model that binds workload, delegator, patient, resource, operation, purpose, audience, policy version, risk ceiling, action digest, lifetime, and nonce;
- Safe-write lifecycle that combines read/write separation, policy-derived risk, payload-bound approval, state revalidation, atomic replay control, emergency disablement, and reason-coded audit;
- Standards mapping that separates the responsibilities already covered by SMART and FHIR from the additional transaction guarantees introduced by AgentCare-Guard;
- Fixed-seed comparison with role-only, SMART-like, and ABAC-like policy models, together with six control ablations and a concurrent replay experiment; and
- Residual-risk analysis that distinguishes authorization failures from obfuscated prompt attacks and clinically unsafe actions that remain inside the granted capability.

The contribution is limited to a security mechanism and its controlled evaluation. It does not establish clinical effectiveness, medical-device safety, HIPAA compliance, formal verification, or production FHIR conformance. These limits are stated explicitly in the experimental design and interpretation.

2 Background and Related Work

2.1 FHIR Security and Transaction Semantics

FHIR defines healthcare resources, REST interactions, version history, transactions, Provenance, and AuditEvent (HL7 International 2019a; HL7 International 2019b; HL7 International 2019c; HL7 International 2019d; HL7 International 2019e). SMART App Launch and the backend-services profile provide interoperable authorization for user-facing and autonomous clients (HL7 International 2024a; HL7 International 2024b). FHIR R4 Consent can express patient choices (HL7 International 2019f), while the R5 Permission resource provides a trial-use structure for access-policy rules (HL7 International n.d.-a). Together, these standards cover identity, delegation, consent, transport, and audit, but they do not prescribe a complete write protocol for an autonomous agent. FHIR does, however, provide an important building block: version-aware updates can use entity tags and If-Match preconditions to reject writes against stale state (HL7 International 2019e). AgentCare-Guard relies on that server-side mechanism in its production mapping. FHIR adoption remains heterogeneous (Gazzarata et al. 2024), and platforms such as WiraChain focus primarily on interoperability and record integrity rather than transaction-specific authorization for AI agents (Grau et al. 2026).

2.2 Healthcare Agent Governance

Recent healthcare-agent systems have concentrated on policy orchestration and privacy. Tripathi et al. (2026) evaluated a HIPAA-oriented architecture that combines ABAC, PHI sanitization, and audit controls, building on an earlier work-in-progress design by Neupane et al. (2025). Shehab (2025) described a multilingual MCP-based prototype with RBAC, encryption, and tamper-evident logging. Prakash et al. (2026) proposed lifecycle registries, runtime policy, and kill-switch governance, while Maiti (2026) reported a zero-trust deployment pattern based on workload isolation, credential proxies, egress restrictions, and prompt-integrity labels. These studies strengthen governance and platform security, but they do not fully specify the lifecycle of an individual FHIR write, including payload binding, approval freshness, and replay handling.

2.3 Agent Tool-Use Security

General agent-security research provides the closest technical foundations. Open Agent Passport performs synchronous authorization before a tool call is executed (Uchibeke 2026). Fides applies deterministic information-flow labels to constrain agent planning (Costa et al. 2025). Systems-security analyses argue that protection

boundaries and threat models must sit outside the language model (Christodorescu et al. 2025; Xiang et al. 2026). OpenPort defines a controlled-write protocol with draft creation, human review, idempotency, impact binding, state-witness revalidation, and structured audit (Zhu et al. 2026). Surveys and operational frameworks organize the broader threat landscape and evaluation problem (Chhabra et al. 2026; Ghosh et al. 2025). AgentCare-Guard does not present these primitives as new. Its contribution lies in adapting them to patient-scoped FHIR transactions, mapping them to clinical risk, and evaluating their behavior across comparators, ablations, concurrent replay, and known residual failures.

2.4 Retrieval-Augmented Clinical Agents

RAG systems retrieve external evidence before generation (Lewis et al. 2020). BM25 and latent semantic indexing offer interpretable retrieval baselines that are suitable for a deterministic prototype (Robertson and Zaragoza 2009; Deerwester et al. 1990). In a clinical setting, retrieval is itself a security boundary because evidence from another patient can influence reasoning before the action reaches the authorization layer. AgentCare-Guard therefore filters candidates by patient and global-policy scope before ranking and retains a provenance digest for each returned document. Retrieval may improve context, but it cannot grant execution authority.

Table 1 Differentiation from representative work

Stream	Representative capability	Remaining transaction gap	AgentCare-Guard specialization
SMART/FHIR	Delegated scopes; patient, resource, and operation access; audit	No standard payload- and state-bound agent write lifecycle	Action capability; approval digest; state witness; replay registry
Healthcare governance	ABAC; PHI controls; registries; emergency disablement	Per-invocation write semantics remain incomplete	Workload-bound, single-use execution decision
Pre-action gateways	Tool interception and deterministic policy	Limited FHIR and clinical specialization	FHIR mapping; patient scope; clinical risk tiers
Controlled writes	Drafting; approval; idempotency; state witness	Limited FHIR-oriented comparative evidence	Comparators; ablations; concurrency; residual-risk tests
RAG security	Grounded retrieval and filtering	Cross-patient evidence may influence reasoning	Metadata prefiltering and non-authoritative retrieval

3 Research Questions, System Model, and Threat Model

3.1 Research Questions and Hypotheses

The evaluation was organized around five research questions:

- RQ1: How does transaction-level mediation compare with role-only, SMART-like, and ABAC-like authorization under the defined boundary attacks?
- RQ2: What happens when payload binding, approval-digest validation, replay controls, state checking, prompt screening, or the kill switch is removed?
- RQ3: Can atomic nonce and idempotency consumption preserve at-most-once execution under concurrent replay?
- RQ4: What legitimate-workflow acceptance and in-memory latency does the reference implementation achieve?
- RQ5: Which attack classes remain outside the authorization guarantee even when the reference monitor behaves as designed?

The corresponding hypotheses were: H1, full mediation would reduce boundary-attack success relative to all three comparators without reducing acceptance of the encoded legitimate workflows; H2, removing a control would reopen its targeted attack family; H3, atomic registry operations would prevent duplicate execution under concurrent replay; and H4, authorization alone would not block semantically obfuscated instructions or clinically unsafe actions that remain within scope.

3.2 Health Informatics Transaction System Model

The protected workflow is represented as a sequence of state transitions across five logical components: the clinical agent, identity and capability services, the reference monitor, approval and policy services, and the FHIR server. The agent can produce a proposal but has no direct write authority. Identity and capability services establish workload and delegated context; the reference monitor evaluates scope, payload integrity, risk, approval state,

freshness, and replay status; and the FHIR server remains authoritative for resource versioning and commit semantics. A permitted transition exists only when every required predicate is satisfied for the current patient, resource, operation, and purpose. Denied or approval-pending transitions terminate before the execution adapter. This model makes the transaction boundary explicit and allows authorization failures, stale-state failures, and clinical-safety failures to be evaluated as separate classes rather than collapsed into a single access-control outcome.

3.3 Protected Action and Capability Model

A protected action, a , is a proposed FHIR tool call. It carries the workload and delegating-subject identifiers, patient, resource type, operation, purpose, payload, evidence references, expected resource version, approval reference, idempotency key, destination audience, and request identifier. The agent cannot assert its own authority. A separate identity plane attests the workload, and a capability issuer creates a short-lived capability C :

$$C = \langle wid, sub, pat, res, op, purp, risk_max, aud, iat, exp, pv, dig, nonce \rangle (1)$$

In (1), wid identifies the workload; sub is the delegating subject; pat is the patient; res and op denote the resource and operation; $purp$ is the declared purpose; $risk_max$ is the highest permitted policy tier; aud is the destination; iat and exp define the validity window; pv identifies the policy version; dig is the canonical action digest; and $nonce$ is a single-use value. The prototype implements these semantics with signed local tokens. A production deployment should use attested workload credentials, such as SPIFFE/SPIRE, together with sender-constrained OAuth (Campbell et al. 2020; Fett et al. 2023; Rose et al. 2020; Cloud Native Computing Foundation n.d.). The design applies complete mediation and least privilege and adapts established caveated and large-scale authorization concepts to the transaction boundary (Saltzer and Schroeder 1975; Birgisson et al. 2014; Pang et al. 2019).

3.4 Adversary and Trust Boundaries

The adversary may influence prompts, retrieved text, tool arguments, payload references, repeated requests, and request timing. The evaluated attacks include cross-patient and cross-resource escalation, operation or purpose mismatch, payload changes after authorization, stale or mutated approvals, concurrent replay, and continued execution after emergency disablement. The attacker may also hide instructions through simple encoding. The MVP trusts the capability issuer, policy configuration, clinician approval decision, cryptographic root, and FHIR server. Compromise of those components, malicious but authorized clinical decisions, and vulnerabilities in the FHIR server remain outside the evaluated boundary.

Table 2 Threat families and security boundaries

Threat family	Adversarial action	Full-policy outcome	Security boundary
Cross-patient	Action patient differs from capability	Deny	Authorization
Resource/operation	Authority is reused for another FHIR interaction	Deny	Authorization
Purpose mismatch	Capability is reused for a different purpose	Deny	Authorization
Payload substitution	Patient or resource reference changes after approval	Deny	Transaction integrity
Replay	Nonce or idempotency key is reused	Deny	At-most-once execution
State drift	Resource changes after approval	Deny	Freshness
Approval mutation	Approved payload is modified	Deny	Approval integrity
Kill switch	Action is attempted after emergency disablement	Deny	Containment
Obfuscated instruction	Policy-bypass instruction is encoded	May allow	Residual content risk
Unsafe in-scope dose	Authorized tuple contains an unsafe clinical value	May allow	Clinical-safety risk

4 AgentCare-Guard Architecture

4.1 Workload Identity and Capability Issuance

Every invocation is authenticated using a verifiable workload identity rather than relying on a long-lived process session. Workload identity and user delegation are treated as separate claims: a clinician may delegate an action, but only a registered workload can exercise the agent execution path. Before issuing a capability, the issuer evaluates the delegator, patient context, intended resource and operation, purpose, destination, and active policy version.

4.2 Risk-Derived Read/Write Separation

The policy engine derives risk instead of accepting a tier supplied by the agent. Reads are assigned tier 0. Selected writes to Observation, DocumentReference, and Task are treated as low risk; changes to Condition, AllergyIntolerance, and ServiceRequest are moderate risk; and MedicationRequest or MedicationAdministration changes are high risk because they can directly affect ordering or administration (HL7 International 2019b). Resource type is only a prototype signal. A clinical deployment would also need to consider the operation, fields changed, terminology, reversibility, patient state, consent, and institutional policy

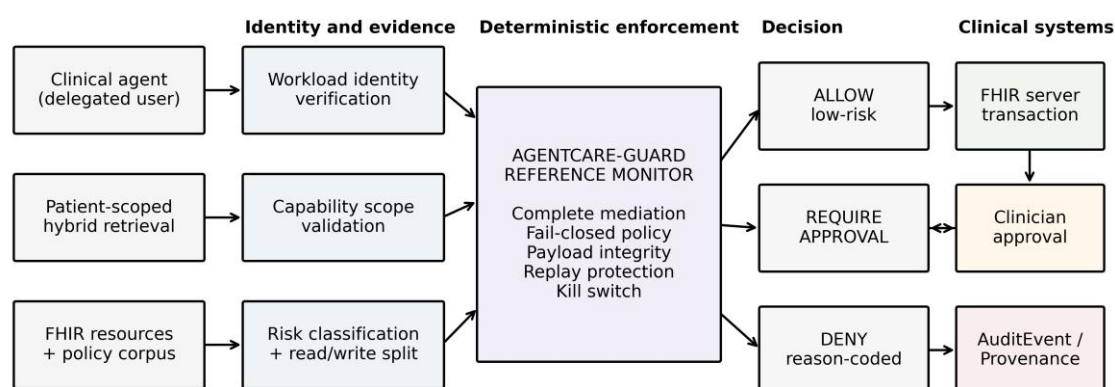


Fig. 1 AgentCare-Guard architecture and trust boundary. Retrieval and reasoning produce evidence and action proposals; only the external reference monitor can authorize the FHIR execution adapter

4.3 Payload Integrity, Approval, and State Witness

Moderate- and high-risk actions first return `REQUIRE_APPROVAL`. The approval record binds the request identifier, delegator, patient, resource, operation, purpose, destination, canonical payload, and expected version through a digest. Any change after review invalidates the approval. Immediately before execution, the monitor rechecks the policy version, kill-switch state, approval lifetime, resource version, nonce, and idempotency key. In a production FHIR path, the version witness must be coupled to a server-enforced If-Match or equivalent conditional write; otherwise, the record could change between the check and the commit (HL7 International 2019e).

4.4 Instruction and Payload Screening

Prompts, retrieved passages, and generated payloads are treated as untrusted input. The prototype screens for a limited set of explicit bypass requests, credential disclosures, prompt-extraction attempts, tool-execution instructions, and dose overrides, reflecting common LLM application risks (OWASP Foundation 2025). The screen is deliberately narrow. It provides a fail-closed signal for known patterns but is not presented as a general prompt-injection detector or a clinical validator. Passing the screen never authorizes an action; every independent policy predicate must still succeed.

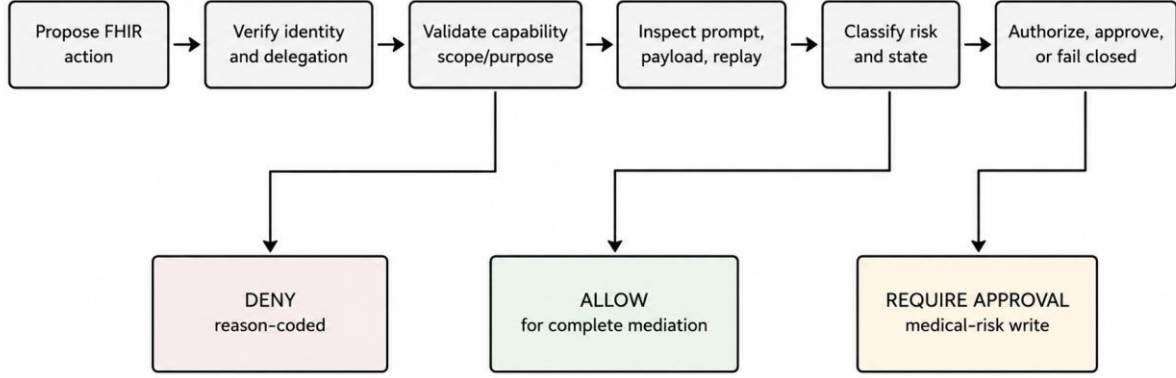
4.5 Replay, Kill Switch, and Audit

On a successful authorization, the monitor atomically consumes both the capability nonce and the action idempotency key. The kill switch is checked before identity and capability processing so that the execution fleet

can be disabled deterministically. Each ALLOW, DENY, or REQUIRE_APPROVAL outcome produces a reason-coded AuditEvent-style record. The prototype links canonical events using the SHA-256 relation in (2).

$$h_i = \text{SHA-256}(\text{canonical}(e_i) \parallel h_{i-1}), h_0 = \text{GENESIS} (2)$$

The in-memory chain can reveal modification only while its head and event store remain trusted. A production ledger would require append-only or WORM storage, protected timestamps, isolated signing keys, externally anchored checkpoints, and independent monitoring. Without those measures, a privileged attacker could truncate the log or reconstruct the chain.



Every outcome emits a reason-coded AuditEvent-style record; approved writes revalidate state before commit.

Fig. 2 Deterministic pre-action lifecycle. Consequential writes require payload-bound approval and a fresh server-side version precondition before atomic replay-record consumption and execution

Table 3 Standards responsibility mapping

Standard or control	Primary responsibility	Use in AgentCare-Guard	Outside the MVP claim
OAuth/SMART (HL7 International 2024a; Hardt 2012; Campbell et al. 2020; Fett et al. 2023; HL7 International 2024b)	Delegation, scopes, and token protection	Identity and context inputs to action policy	Replacement for SMART
FHIR Consent (HL7 International 2019f)	Patient authorization choices	Future policy input and permit/deny constraint	Implemented consent engine
FHIR Permission R5 (HL7 International n.d.-a)	Trial-use access-policy representation	Future policy serialization and mapping	R4-native implementation
FHIR If-Match (HL7 International 2019e)	Server-side version precondition	Production state-bound conditional write	Completed FHIR-server evaluation
AuditEvent/Provenance (HL7 International 2019c; HL7 International 2019d)	Security and origin evidence	Reason-coded decisions and evidence lineage	Durable compliant ledger
SPIFFE (Cloud Native Computing Foundation n.d.)	Attested workload identity	Production identity plane	Identity mechanism used by the local HMAC MVP

5 Executable Security Properties

5.1 Authorization Predicate

The reference monitor evaluates action a , capability C , approval q , current state s , policy version v , and replay registry R . Equation (3) gives the simplified authorization predicate:

$$\text{ALLOW}(a) = I \wedge C_v \wedge S \wedge P \wedge D \wedge A \wedge F \wedge K \wedge R_a (3)$$

Here, I represents valid workload identity; C_v is a valid, unexpired, audience-bound capability; S requires patient, resource, operation, and purpose scope to match; P checks payload references; D checks the action digest; A verifies any approval required by the computed risk; F confirms current resource and policy state; K requires the kill switch to be inactive; and R_a confirms that the nonce and idempotency key can be consumed atomically.

A failed predicate returns DENY, except that a valid action awaiting mandatory approval returns REQUIRE_APPROVAL.

5.2 Security Invariants

The properties in Table 4 are executable invariants enforced and tested by the prototype. They are not presented as machine-checked proofs. Evidence comes from positive and negative cases, targeted ablations, service-path tests, and concurrent replay experiments.

Table 4 Executable security invariants

Property	Executable invariant	Evaluation evidence	Remaining condition
P1 Complete mediation	Every write route invokes policy before side effects	API and service-path tests	Bypass in an untested integration
P2 Scope confinement	Action tuple matches the capability tuple	Patient, resource, operation, and purpose attacks	Unsafe values that remain within scope
P3 Approval integrity	Executed action digest matches the approved digest	Approval-mutation tests and ablation	Misleading approval interface
P4 Freshness	Approved resource and policy versions remain current	State-drift tests	Requires an atomic server precondition
P5 At-most-once	One nonce and key can authorize at most one execution	20-worker concurrent replay	Distributed-store implementation error
P6 Fail closed	Invalid or unknown conditions are denied	Signature, expiry, and kill-switch tests	Policy misconfiguration
P7 Audit completeness	Every policy outcome emits a reason-coded record	Event-count and hash-chain tests	Privileged truncation without external anchoring

6 Patient-Scoped Retrieval Component

6.1 Retrieval Design

The retrieval component indexes FHIR-style patient context together with global security policies. Candidate filtering occurs before ranking: only documents from the requested patient partition and the global-policy set enter lexical or semantic scoring. The normalized hybrid score is given in (4).

$$S(d,q) = 0.45 B_hat(d,q) + 0.45 C_hat(d,q) + 0.10 scope(d,p) \quad (4)$$

Within the filtered candidate set, B_hat and C_hat are min-max-normalized BM25-like and latent-semantic scores. The scope term equals one for an authorized patient document or global policy; documents outside that scope are removed before ranking and therefore receive no score. The semantic representation uses TF-IDF followed by fixed-seed truncated singular-value decomposition. A production system could replace this stage with a sharded approximate-nearest-neighbor index without changing the authorization boundary.

6.2 Grounded but Non-Authoritative Generation

For repeatability, the evaluation uses deterministic extractive generation. The output reports retrieved identifiers and policy evidence instead of asking a generative model to compose a clinical response. Each evidence item carries the resource type, patient identifier, retrieval score, and SHA-256 provenance digest. A local or managed language model could be added later, but it would not be allowed to invoke the write endpoint directly. Retrieval error, poisoning, or hallucination can degrade advice; none of those conditions can satisfy the authorization predicate.

7 Prototype and Comparative Evaluation Harness

7.1 Implementation

The Python implementation separates domain models, token signing, data construction, retrieval, approval handling, policy evaluation, replay storage, audit, orchestration, and API contracts. The supplementary evaluation harness applies four policy models to the same generated cases, executes six single-control ablations, measures concurrent replay and latency, evaluates patient-scoped retrieval, and runs automated tests.

7.2 Public Data and FHIR-Style Transformation

The corpus is derived from the public scikit-learn diabetes regression dataset, which contains 442 samples and ten standardized variables (Pedregosa et al. 2011; Efron et al. 2004; Scikit-learn Developers n.d.). It provides a stable numerical source without real identifiers or PHI, but it retains the limitations associated with transformed or synthetic health data (Gonzales et al. 2023). Each row is represented as one Patient document, ten Observation documents, and one provisional Condition document. Five global policy documents bring the index to 5,309 entries. The target variable is used only to create a synthetic higher/lower progression label and does not support diagnostic, treatment, or prescribing claims.

7.3 Software Quality and Environment

Experiments ran with Python 3.13.5, NumPy 2.3.5, SciPy 1.17.0, scikit-learn 1.8.0, pandas 2.2.3, Matplotlib 3.10.8, and Pytest 9.0.2. All 18 automated tests passed. Combined line and branch coverage was 94% across 408 statements and 102 branches. The tests cover policy decisions, case generation, ablations, patient isolation, audit integrity, atomic and non-atomic replay behavior, and reproducible outputs.

7.4 Use of AI-Assisted Tools During Preparation

AI-assisted tools were used for language editing, manuscript organization, limited code scaffolding for the comparative evaluation, and document formatting. They were not treated as sources of evidence or as authors. The author reviewed the implementation, automated tests, numerical outputs, references, interpretations, and final manuscript and retained responsibility for all claims. Code that was generated or edited with tool assistance was subjected to the same automated tests and coverage checks described above.

Table 5 Evaluation configuration

Item	Configuration	Purpose
Random seed	20260727	Repeatable patient and case generation
Patients	150 of 442	Comparative authorization workload
Cases	2,550	600 legitimate; 1,500 boundary; 450 residual
Policy models	Role-only; SMART-like; ABAC-like; full guard	Progressive control comparison
Ablations	Six single-control removals	Control-necessity analysis
Concurrency	100 rounds $\times$ 20 workers	At-most-once replay behavior
Policy timing	4,000 decisions per model	Median, p95, and p99 latency
Retrieval	5,309 documents; 250 queries	Isolation and in-memory latency
Quality	18 tests; 94% line/branch coverage	Implementation assurance

8 Experimental Methodology

8.1 Policy Comparators and Ablations

The role-only model accepts a valid workload that carries the clinical-agent role. The SMART-like model approximates patient, resource, and operation scoping, but it is not a complete SMART server. The ABAC-like model adds purpose and policy-derived risk. AgentCare-Guard then adds payload-reference checks, approval-digest validation, state revalidation, prompt screening, atomic replay storage, and the kill switch. Each model receives the same canonical actions in the same order. The six ablations remove one control at a time from the full policy.

Table 6 Policy models and ablations

Model	Controls included	Intentionally excluded
Role-only	Signed workload role	All transaction context
SMART-like	Patient, resource, and operation	Purpose, payload, approval, freshness, and replay
ABAC-like	SMART-like controls plus purpose and risk	Payload, approval digest, freshness, replay, and kill switch

Model	Controls included	Intentionally excluded
AgentCare-Guard	All executable controls	Clinical correctness and general semantic injection
No payload binding	Full policy without payload-reference check	Payload-substitution control
No approval digest	Full policy without approval/action digest match	Post-approval mutation control
No replay controls	Full policy without nonce/idempotency consumption	At-most-once control
No state witness	Full policy without version recheck	Approval-freshness control
No prompt screen	Full policy without lexical screening	Explicit-instruction signal
No kill switch	Full policy without emergency-stop check	Fleet containment

8.2 Case Taxonomy

Each of the 150 selected patients contributes four legitimate workflows: a Patient read, a low-risk Observation create, an approved MedicationRequest create, and an approved Condition update. Ten authorization-boundary families contribute 1,500 attacks: cross-patient access, resource escalation, operation escalation, purpose mismatch, payload substitution, capability replay, idempotency replay, state drift, approval mutation, and kill-switch violation. Three residual families add 450 cases: explicit prompt injection, base64-obfuscated prompt injection, and an unsafe but scope-compliant medication dose. The final two were included specifically to test the limits of deterministic authorization.

8.3 Concurrent Replay Experiment

In each of 100 replay rounds, 20 workers submit the same nonce and idempotency key at the same time. The atomic registry performs the availability check and consumption inside one lock. The non-atomic variant separates the check from the update and introduces a scheduling window, modeling a registry that lacks compare-and-set semantics. The at-most-once property is satisfied only when exactly one request is allowed in a round.

8.4 Metrics and Statistical Treatment

Legitimate acceptance is the proportion of valid workflows allowed. Boundary-attack success and residual-attack success are the corresponding allowed proportions for their case sets. Exact Clopper-Pearson intervals are used only as descriptive bounds when all cases pass or fail; the generated cases are not random samples from a real attack population. Ablation results are interpreted by whether the removed control reopens its target family. The concurrency experiment reports the mean, range, and proportion of rounds with duplicate execution. Latency is measured with a high-resolution performance counter and summarized by the median, p95, and p99. The results do not support cross-platform or network-level performance claims.

9 Results

9.1 Comparative Authorization Outcomes

Table 7 and Fig. 3 summarize RQ1. Every model accepted all 600 encoded legitimate workflows. The role-only, SMART-like, and ABAC-like comparators allowed 100%, 70%, and 60% of the 1,500 boundary attacks, respectively. AgentCare-Guard allowed none. For descriptive context, the exact 95% upper bound for 0/1,500 is 0.246%, while the lower bound for 600/600 legitimate acceptance is 99.387%. These bounds describe the generated suite; they should not be interpreted as estimates of real-world clinical security.

Table 7 Aggregate policy outcomes

Policy	Legitimate acceptance	Boundary attack success	Residual attack success
Role-only	600/600 (100%)	1500/1500 (100%)	450/450 (100%)
SMART-like	600/600 (100%)	1050/1500 (70%)	450/450 (100%)
ABAC-like	600/600 (100%)	900/1500 (60%)	450/450 (100%)
AgentCare-Guard	600/600 (100%)	0/1500 (0%)	300/450 (66.7%)

Residual cases comprise explicit and obfuscated prompt injection together with clinically unsafe actions that still satisfy the capability. An authorization decision is not a clinical safety judgment.

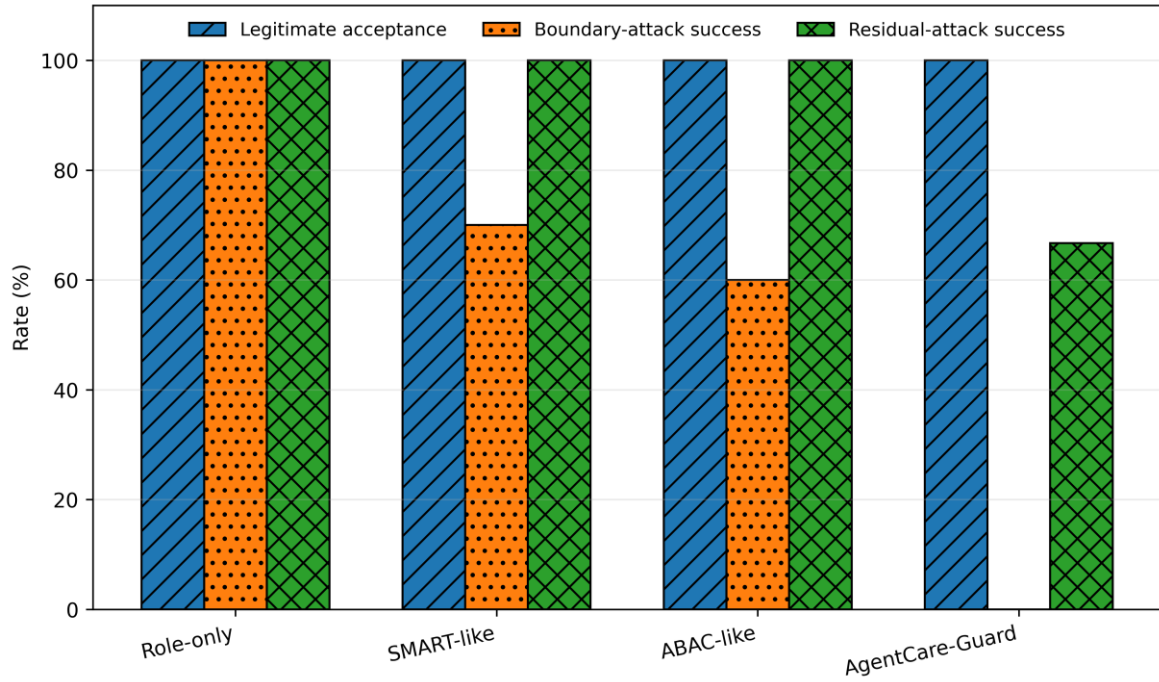


Fig. 3 Legitimate acceptance, boundary-attack success, and residual-attack success across the four policy models. Lower attack success is better

9.2 Scenario-Level Interpretation

The SMART-like comparator rejected mismatched patients, resources, and operations, but it allowed purpose mismatch, payload substitution, approval mutation, replay, stale state, and kill-switch violations. The ABAC-like model additionally rejected purpose mismatch, yet still lacked the transaction-lifecycle controls. AgentCare-Guard rejected all ten boundary families. Its lexical screen caught the explicit injection cases but not the base64-obfuscated instructions. It also allowed the unsafe dose because the patient, resource, operation, purpose, approval, state, and replay predicates were valid. That result answers RQ5: authorization can constrain who may act and under what conditions, but it cannot determine clinical appropriateness.

9.3 Ablation Outcomes

Every single-control ablation reopened the family that the removed mechanism was designed to block, while acceptance of the encoded legitimate cases remained unchanged. Removing payload binding allowed all 150 payload substitutions. Removing the approval digest allowed all 150 post-approval mutations. Removing replay controls allowed all 300 capability- and idempotency-replay cases. The state-witness ablation allowed all state-drift cases, the prompt-screen ablation allowed all explicit injections, and the kill-switch ablation allowed all actions issued after emergency disablement. Figure 4 summarizes these results.

Table 8 Targeted ablation results

Removed control	Target family	Full allow rate	Ablated allow rate
Payload binding	Payload substitution	0/150 (0%)	150/150 (100%)
Approval digest	Approval mutation	0/150 (0%)	150/150 (100%)
Replay registry	Capability/idempotency replay	0/300 (0%)	300/300 (100%)
State witness	State drift	0/150 (0%)	150/150 (100%)
Prompt screen	Explicit injection	0/150 (0%)	150/150 (100%)
Kill switch	Emergency-stop action	0/150 (0%)	150/150 (100%)

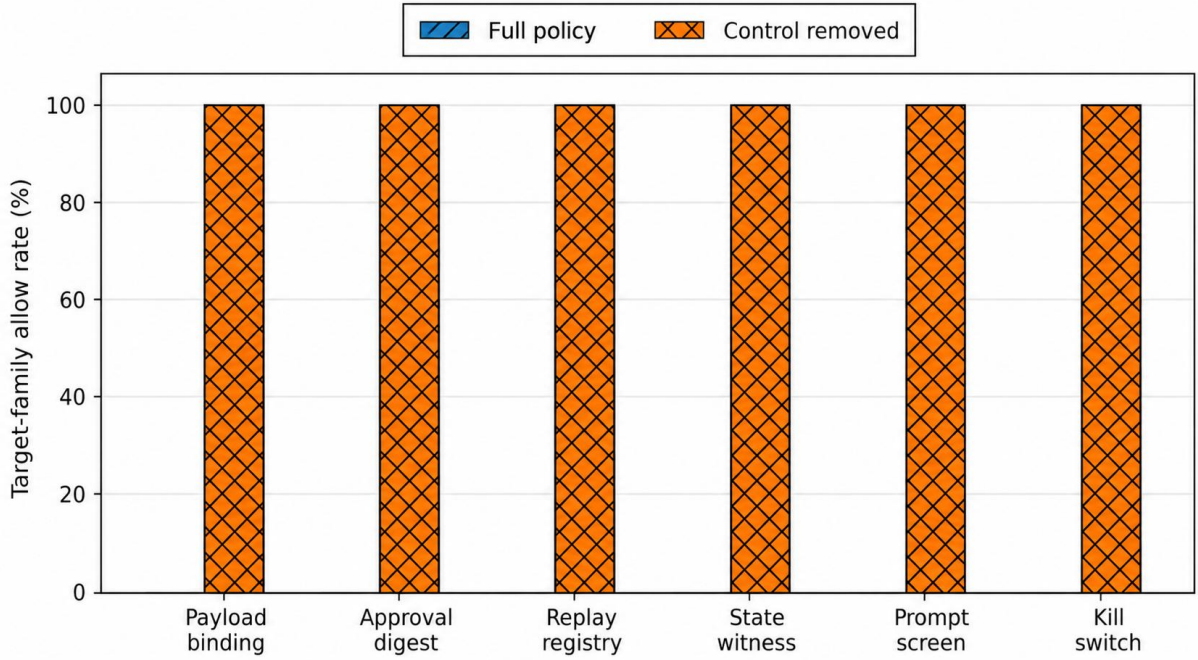


Fig. 4 Target-family allow rate for AgentCare-Guard and six single-control ablations. Removing a control reopens the corresponding attack family

9.4 Concurrent Replay

The atomic registry allowed exactly one execution in every round; the minimum, mean, and maximum ALLOW counts were all 1, and no round contained a duplicate. The non-atomic variant averaged 2.01 executions, reached a maximum of 5, and produced duplicates in 73 of 100 rounds. The result supports H3 and shows that carrying a nonce is not sufficient: the persistence layer must consume replay records atomically.

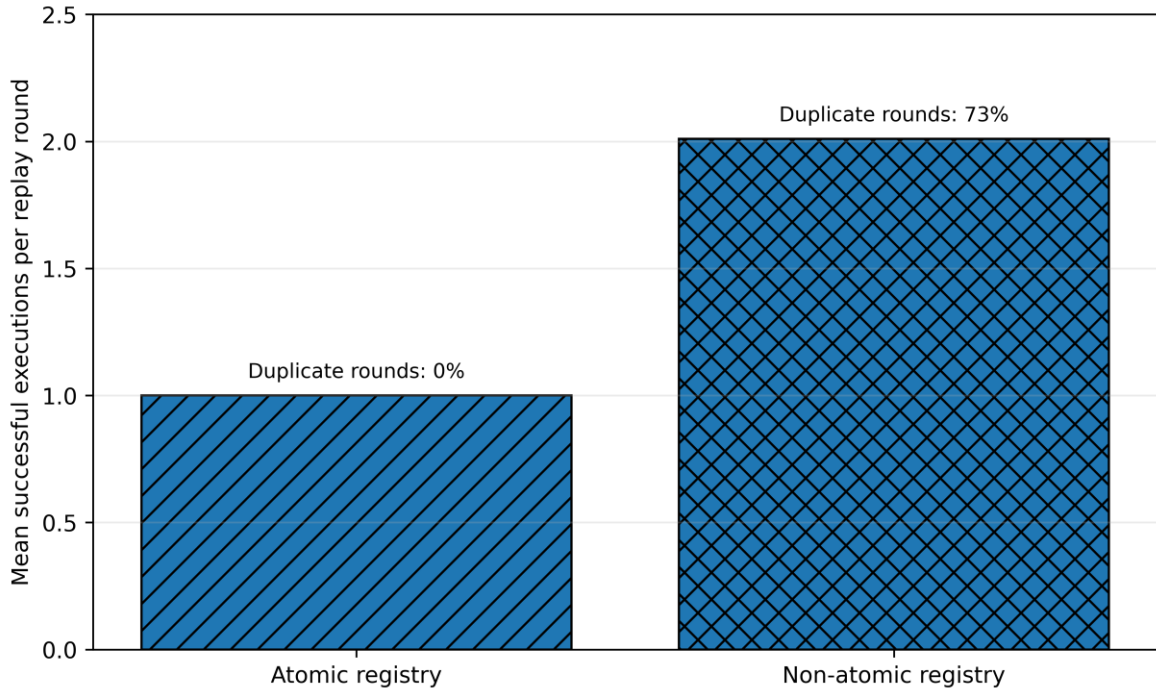


Fig. 5 Concurrent replay over 100 rounds with 20 workers per round. Atomic consumption preserves one execution; the non-atomic variant produces duplicates in 73% of rounds

9.5 Latency, Retrieval, and Quality Gates

Across 4,000 in-memory decisions per model, median latency was 0.00046 ms for role-only, 0.00049 ms for SMART-like, 0.00050 ms for ABAC-like, and 0.00677 ms for AgentCare-Guard. The full-policy p95 and p99 latencies were 0.00777 and 0.01165 ms, respectively. These values exclude transport, external identity, a remote

policy service, durable replay storage, clinician approval, FHIR persistence, and durable audit. The 5,309-document retrieval index was built in 0.124 s. Over 250 patient-scoped queries, the median, p95, and p99 retrieval latencies were 0.629, 0.889, and 1.607 ms, respectively. No retrieved evidence crossed patient boundaries, and every query returned at least one global policy document. Figure 6 shows the local policy timing results.

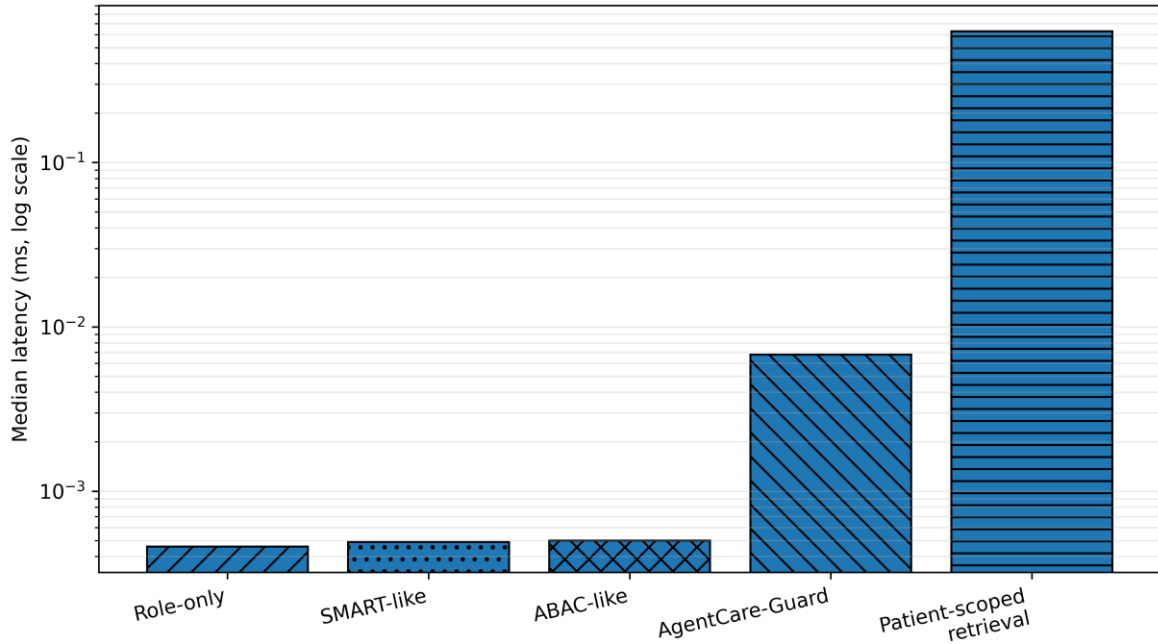


Fig. 6 Median in-memory latency for the four policy models and patient-scoped retrieval. The measurements characterize the local harness and exclude production dependencies

10 Discussion

10.1 Answers to the Research Questions

For the generated suite, RQ1 shows that transaction-level controls close gaps left by progressively richer access models without reducing the encoded legitimate acceptance. RQ2 is supported by the ablations, which show that the controls make distinct contributions rather than duplicating one another. RQ3 is supported by the replay experiment: atomic consumption is the difference between at-most-once authorization and repeatable duplicate execution. RQ4 is limited to local feasibility because realistic network and storage costs will dominate the measured policy time. RQ5 identifies the boundary of the approach: obfuscated instructions and clinically unsafe in-scope values are not solved by authorization.

10.2 Novelty and Relationship to Prior Systems

The contribution is an applied security protocol and evaluation, not a new cryptographic primitive. Open Agent Passport establishes pre-action authorization at the tool-call boundary (Uchibeke 2026), and OpenPort describes controlled-write mechanisms including approval, idempotency, impact binding, and state witnesses (Zhu et al. 2026). AgentCare-Guard specializes these ideas for FHIR transaction tuples, clinical risk gating, patient-scoped retrieval, and conditional-write semantics. Its evaluation adds a progressive policy comparison, targeted ablations, concurrent replay, and an explicit separation between authorization failures and clinical-safety failures. These results should not be read as a direct performance comparison with systems evaluated on different implementations or hardware.

10.3 Clinical Accountability

Human approval is useful but not inherently safe. Frequent prompts can produce alert fatigue, and a misleading interface can still cause an incorrect approval. A clinical review screen should show the patient, delegator, resource, operation, changed fields, supporting evidence, risk tier, policy version, and expected resource version as a clear diff. Institutions also need rules for escalation, override, break-glass access, revocation, and post-event review. Before deployment, the risk tiers must be reviewed jointly by clinical, safety, privacy, and security stakeholders.

10.4 Authorization Versus Clinical Correctness

The unsafe-dose cases are an important result because they expose the limit of the mechanism. Authorization answers whether an identified actor may perform an action under a defined policy. It does not answer whether the proposed action is medically correct. A production system therefore requires separate clinical decision support, terminology validation, dose and interaction checks, contraindication logic, uncertainty handling, and accountable clinician review. Treating authorization as a substitute for clinical validation would create a false sense of safety.

10.5 Implications for Health Informatics Systems

The results suggest that agentic clinical systems need an explicit transaction boundary between reasoning and execution. In practical health-information architectures, that boundary can sit alongside existing SMART-on-FHIR authorization rather than replace it. The health-informatics value of the approach is the separation of clinical context retrieval, delegated access, transaction authorization, and server-side commit into independently auditable stages. That separation also clarifies where further evidence is required: clinical correctness belongs to decision support and human governance, while transaction integrity belongs to the execution path. Treating these concerns separately reduces the risk of interpreting a valid access token or a grounded model response as sufficient authority for a consequential write.

11 FHIR Deployment and Scale-Out Architecture

11.1 Production Transaction Path

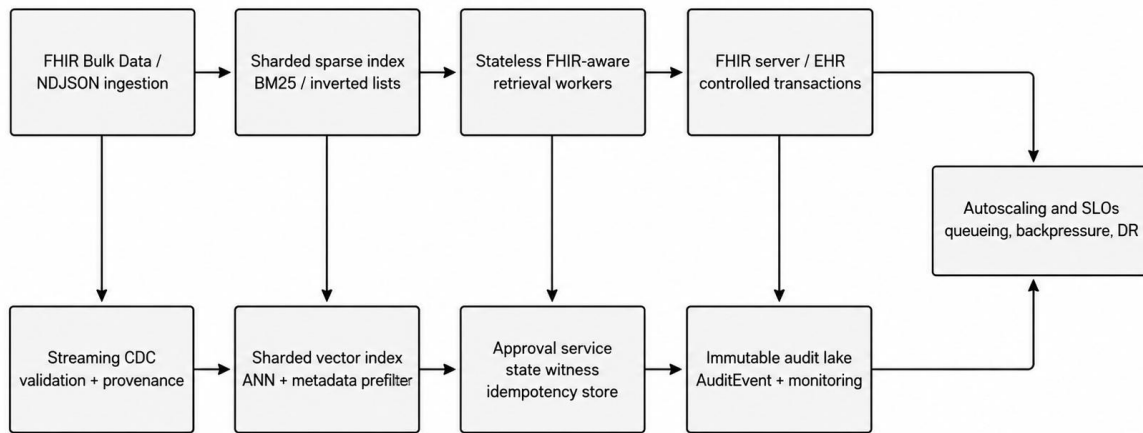
A production deployment should place the reference monitor at an API gateway or service boundary that no write route can bypass. SMART backend authorization would provide delegated client context (HL7 International 2024b), while SPIFFE/SPIRE or an equivalent system would attest workload identity (Cloud Native Computing Foundation n.d.). Institutional policy would be evaluated by an external decision point, and nonce and idempotency records would be consumed through a compare-and-set operation or unique-key transaction. Consequential writes would carry the approved resource version in an If-Match precondition so that the FHIR server can reject stale state atomically (HL7 International 2019e). AuditEvent and Provenance records would be written to independent append-only storage (HL7 International 2019c; HL7 International 2019d).

11.2 FHIR Conformance Validation Protocol

The next validation stage should use HAPI FHIR JPA Starter or another standards-compliant server (HAPI FHIR n.d.). Generated Patient, Observation, Condition, MedicationRequest, and transaction Bundle resources should first pass formal validation. The test plan should cover create and update operations, conditional updates, ETag and If-Match conflicts, history, transaction rollback, SMART backend scopes, Consent constraints, terminology checks, and concurrent clients. These experiments are proposed future work and are not included in the reported results.

11.3 Scale-Out Pattern

A terabyte-scale corpus would require FHIR Bulk Data or NDJSON storage, validated ingestion and change-data capture, patient or tenant partitioning, sharded lexical and vector indexes, metadata filtering before approximate-nearest-neighbor search, and stateless retrieval workers (HL7 International n.d.-b). The authorization plane should remain independent of the retrieval index. Backpressure, rate limits, circuit breakers, regional recovery, and fail-closed dependency handling would also be required. A meaningful scale evaluation should report corpus and shard size, index build time, retrieval recall, p95 and p99 latency, throughput, hot-patient skew, replay-store contention, audit durability, and recovery behavior.



The authorization plane remains independent of the retrieval index and uses fail-closed dependency handling

Fig. 7 Proposed production and scale-out architecture. High-risk writes are mapped to external identity, policy, atomic replay storage, conditional FHIR commit, and durable audit; the diagram is a deployment target rather than a completed scale benchmark

12 Security, Privacy, Safety, and Ethical Considerations

The evaluation used a public numerical dataset transformed into synthetic identifiers. It involved no real PHI, human participants, or clinical intervention. Accordingly, the results do not establish compliance with the HIPAA Security Rule (U.S. Department of Health and Human Services n.d.), medical-device requirements, privacy law, consent obligations, or institutional governance. An operational deployment would still require formal risk analysis, access-control review, integrity safeguards, audit review, incident response, key management, retention controls, and business-associate assessment where applicable.

Privacy controls must cover more than patient identifiers. Retrieved evidence, prompts, embeddings, caches, approval screens, logs, and audit records can all expose sensitive information. Purpose-of-use restrictions, minimum-necessary access, tenant isolation, consent, break-glass handling, data residency, deletion, and secondary-use governance must be applied consistently. Signed corpus manifests, provenance checks, source trust, temporal validity, and retrieval abstention can reduce poisoning and stale-evidence risk.

AgentCare-Guard must not be used autonomously for diagnosis, treatment, ordering, administration, or prescribing without independent clinical validation and accountable human governance. Security denial rates are not patient-safety metrics. A deployment study would need to measure false denial, false approval, reviewer burden, decision quality, override behavior, incident recovery, and downstream clinical outcomes.

13 Limitations and Threats to Validity

13.1 Construct and Internal Validity

The comparator models are executable emulations, not complete deployments of SMART, OPA, Cedar, or commercial policy products. The attacks were generated deterministically with knowledge of the controls, which introduces developer and branch-coverage bias. Perfect denial of the boundary cases therefore demonstrates enforcement of the stated invariants, not universal attack resistance. The fixed seed improves repeatability but does not define a real-world attack distribution.

13.2 External and Ecological Validity

The corpus is synthetic and does not include clinical notes, longitudinal encounters, medication terminology, clinician identities, institutional policies, consent directives, or the complexity of production workflows. The evaluation did not use a live FHIR server, a clinician approval interface, a workload-attestation service, a distributed policy engine, or durable audit storage. The resource-based risk tiers have not been clinically calibrated.

13.3 Security Coverage

The lexical prompt screen fails on the evaluated base64-obfuscated input and is also unlikely to detect paraphrased, multilingual, multimodal, split-context, or tool-output attacks. Additional risks include capability-compliant clinical misuse, compromised issuers, malicious approvers, deceptive approval interfaces, confused-deputy behavior, nested-reference abuse, poisoned policies, credential theft outside the token model, and vulnerabilities in the FHIR server. Each requires separate controls and evaluation.

13.4 Performance and Reproducibility

The latency figures are local, in-memory microbenchmarks. They exclude network communication, transport security, external identity and policy calls, durable atomic storage, FHIR persistence, clinician approval time, and audit retention. The supplementary package supports reproduction in the reported environment and is intended to accompany the manuscript as Online Resource 1. A separately versioned DOI archive has not yet been created, and cross-platform variation and independent replication remain open.

14 Reproducibility and Supplementary Artifacts

14.1 Supplementary Artifacts

Online Resource 1 contains the fixed-seed Python harness, automated tests, dependency list, README, per-case decisions, aggregate and family summaries, ablation outputs, concurrency measurements, latency results, retrieval metrics, coverage report, and figure-generation outputs. The diabetes dataset is loaded through scikit-learn (Scikit-learn Developers n.d.). No private data or credentials are included. The package should be uploaded with the manuscript in a standard archive format; a persistent repository release can be added separately if desired.

Table 9 Reproducibility artifacts

Artifact	Status	Verification
Evaluation source	Included	agentcare_eval.py
Automated tests	Included	18 passing tests
Coverage output	Included	94% combined line/branch
Fixed seed	Included	20260727
Dependencies	Included	requirements.txt and versions reported in the paper
Per-case decisions	Included	case_decisions.csv
Aggregates and ablations	Included	CSV and JSON summaries
Concurrency results	Included	100 rounds × 20 workers
Retrieval metrics	Included	250-query JSON output
Persistent repository archive	Not yet assigned	Recommended for long-term reproducibility; not required for the current controlled evaluation
Production FHIR server	Not included	Defined as future validation

15 Conclusion

AgentCare-Guard is a FHIR-oriented reference-monitor protocol for autonomous clinical transactions. Its enforcement path combines scoped workload authority with risk-sensitive approval, version checks, replay-safe execution, emergency shutdown, structured audit, and patient-isolated retrieval that remains outside the authorization decision. In the fixed-seed evaluation, the full policy processed 2,550 decisions, accepted all 600 encoded legitimate workflows, and denied all 1,500 authorization-boundary attacks. The progressively weaker comparators allowed between 60% and 100% of those attacks. Targeted ablations reopened their corresponding families, and the atomic replay registry prevented duplicate executions observed in the non-atomic variant.

The residual cases mark the boundary of the contribution. The monitor does not solve general semantic prompt injection, and it cannot decide whether an authorized clinical value is medically safe. Stronger evidence will require standards-compliant FHIR execution, server-enforced conditional writes, production identity and policy services, clinically governed risk rules, adaptive red teaming, durable audit, independent replication, and clinician-

centered safety studies. Within the limits of the controlled evaluation, the work provides a reproducible basis for transaction-level security in agentic FHIR systems.

Statements and Declarations

Funding: This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

Competing Interests: There are no competing interests.

Ethics Approval: Not applicable. The study used a publicly available numerical dataset transformed into synthetic FHIR-style records and involved no human participants, identifiable patient data, clinical intervention, or animals.

Consent to Participate: Not applicable.

Consent for Publication: Not applicable.

Data Availability: The public diabetes dataset used in this study is available through the scikit-learn datasets package. Synthetic FHIR-style evaluation cases, per-case decisions, aggregate results, ablation outputs, concurrency measurements, and retrieval metrics are provided in the supplementary material accompanying this manuscript (Online Resource 1).

Code Availability: The fixed-seed Python evaluation harness, automated tests, dependency manifest, and supporting output files are provided in Online Resource 1. The manuscript reports the software versions and fixed seed required to reproduce the controlled evaluation.

Author Contributions: The sole author is responsible for the study conception, system design, implementation, evaluation, analysis, interpretation, and preparation of the manuscript.

References

- Birgisson A, Politz JG, Erlingsson U, Taly A, Vrabie M, Lentzner M (2014) Macaroons: cookies with contextual caveats for decentralized authorization in the cloud. In: Proceedings of the 21st Annual Network and Distributed System Security Symposium (NDSS).
- Campbell B, Bradley J, Sakimura N, Lodderstedt T (2020) OAuth 2.0 mutual-TLS client authentication and certificate-bound access tokens. RFC 8705. <https://doi.org/10.17487/RFC8705>
- Chhabra A, Datta S, Nahin SK, Mohapatra P (2026) Agentic AI security: threats, defenses, evaluation, and open challenges. IEEE Access, early access. <https://doi.org/10.1109/ACCESS.2026.3675554>
- Christodorescu M et al (2025) Systems security foundations for agentic computing. arXiv:2512.01295. <https://doi.org/10.48550/arXiv.2512.01295>
- Cloud Native Computing Foundation (n.d.) SPIFFE overview: workload identity and SVIDs. <https://spiffe.io/docs/latest/spiffe-about/overview/>. Accessed 28 July 2026
- Costa M et al (2025) Securing AI agents with information-flow control. arXiv:2505.23643. <https://doi.org/10.48550/arXiv.2505.23643>
- Deerwester S, Dumais ST, Furnas GW, Landauer TK, Harshman R (1990) Indexing by latent semantic analysis. J Am Soc Inf Sci 41(6):391-407. [https://doi.org/10.1002/\(SICI\)1097-4571\(199009\)41:6<391::AID-AS11>3.0.CO;2-9](https://doi.org/10.1002/(SICI)1097-4571(199009)41:6<391::AID-AS11>3.0.CO;2-9)
- Efron B, Hastie T, Johnstone I, Tibshirani R (2004) Least angle regression. Ann Stat 32(2):407-499. <https://doi.org/10.1214/009053604000000067>
- Fett D, Campbell B, Bradley J, Lodderstedt T, Jones M, Waite D (2023) OAuth 2.0 demonstrating proof of possession (DPoP). RFC 9449. <https://doi.org/10.17487/RFC9449>
- Gazzarata R et al (2024) HL7 Fast Healthcare Interoperability Resources (HL7 FHIR) in digital healthcare ecosystems for chronic disease management: scoping review. Int J Med Inform 189:105507. <https://doi.org/10.1016/j.ijmedinf.2024.105507>
- Ghosh S et al (2025) A safety and security framework for real-world agentic systems. arXiv:2511.21990. <https://doi.org/10.48550/arXiv.2511.21990>
- Gonzales AB, Guruswamy G, Smith SR (2023) Synthetic data in health care: a narrative review. PLOS Digit Health 2(1):e0000082. <https://doi.org/10.1371/journal.pdig.0000082>
- Grau L, Urrunaga J, Santisteban J (2026) WiraChain: a blockchain and FHIR-based platform for improving clinical data interoperability in healthcare. Front Public Health 14:1746772. <https://doi.org/10.3389/fpubh.2026.1746772>
- HAPI FHIR (n.d.) JPA Server Starter: get started. https://hapifhir.io/hapi-fhir/docs/server_jpa/get_started.html. Accessed 28 July 2026

- Hardt D (2012) The OAuth 2.0 authorization framework. RFC 6749. <https://doi.org/10.17487/RFC6749>
- HL7 International (2019a) FHIR Release 4, version 4.0.1. <https://hl7.org/fhir/R4/>
- HL7 International (2019b) FHIR R4 MedicationRequest. <https://hl7.org/fhir/R4/medicationrequest.html>. Accessed 28 July 2026
- HL7 International (2019c) FHIR R4 AuditEvent. <https://hl7.org/fhir/R4/auditevent.html>. Accessed 28 July 2026
- HL7 International (2019d) FHIR R4 Provenance. <https://hl7.org/fhir/R4/provenance.html>. Accessed 28 July 2026
- HL7 International (2019e) FHIR R4 HTTP: version-aware update and If-Match. <https://hl7.org/fhir/R4/http.html>. Accessed 28 July 2026
- HL7 International (2019f) FHIR R4 Consent. <https://hl7.org/fhir/R4/consent.html>. Accessed 28 July 2026
- HL7 International (2024a) SMART App Launch Implementation Guide, version 2.2.0. <https://hl7.org/fhir/smart-app-launch/>
- HL7 International (2024b) SMART App Launch 2.2.0: Backend Services. <https://hl7.org/fhir/smart-app-launch/backend-services.html>. Accessed 28 July 2026
- HL7 International (n.d.-a) FHIR R5 Permission. <https://hl7.org/fhir/permission.html>. Accessed 28 July 2026
- HL7 International (n.d.-b) FHIR Bulk Data Access (Flat FHIR) Implementation Guide, version 3.0.0. <https://hl7.org/fhir/uv/bulkdata/>. Accessed 28 July 2026
- Lewis P et al (2020) Retrieval-augmented generation for knowledge-intensive NLP tasks. *Adv Neural Inf Process Syst* 33:9459-9474
- Maiti S (2026) Caging the agents: a zero trust security architecture for autonomous AI in healthcare. *arXiv:2603.17419*. <https://doi.org/10.48550/arXiv.2603.17419>
- Neupane S, Mitra S, Mittal S, Rahimi S (2025) Towards a HIPAA compliant agentic AI system in healthcare. *arXiv:2504.17669*. <https://doi.org/10.48550/arXiv.2504.17669>
- OWASP Foundation (2025) 2025 Top 10 risk and mitigations for LLMs and generative AI applications. <https://genai.owasp.org/llm-top-10/>. Accessed 28 July 2026
- Pang R et al (2019) Zanzibar: Google's consistent, global authorization system. In: *Proceedings of the USENIX Annual Technical Conference (USENIX ATC)*, pp 33-46
- Pedregosa F et al (2011) Scikit-learn: machine learning in Python. *J Mach Learn Res* 12:2825-2830
- Prakash C, Lind M, Sisodia A (2026) Agentic AI governance and lifecycle management in healthcare. *arXiv:2601.15630*. <https://doi.org/10.48550/arXiv.2601.15630>
- Robertson S, Zaragoza H (2009) The probabilistic relevance framework: BM25 and beyond. *Found Trends Inf Retr* 3(4):333-389. <https://doi.org/10.1561/15000000019>
- Rose S, Borchert O, Mitchell S, Connelly S (2020) Zero Trust Architecture. NIST Special Publication 800-207. <https://doi.org/10.6028/NIST.SP.800-207>
- Saltzer JH, Schroeder MD (1975) The protection of information in computer systems. *Proc IEEE* 63(9):1278-1308. <https://doi.org/10.1109/PROC.1975.9939>
- Scikit-learn Developers (n.d.) load_diabetes dataset documentation. https://scikit-learn.org/stable/modules/generated/sklearn.datasets.load_diabetes.html. Accessed 28 July 2026
- Shehab MA (2025) Agentic-AI healthcare: multilingual, privacy-first framework with MCP agents. *arXiv:2510.02325*. <https://doi.org/10.48550/arXiv.2510.02325>
- Tripathi H, Neupane S, Mittal S, Rahimi S, Gupta V (2026) A HIPAA-compliant architecture for agentic clinical AI systems. In: *Proceedings of the ACM Conference on AI and Agentic Systems (CAIS)*, pp 738-754. <https://doi.org/10.1145/3786335.3813175>
- U.S. Department of Health and Human Services (n.d.) 45 CFR Part 164—Security and Privacy. *Electronic Code of Federal Regulations*. <https://www.ecfr.gov/current/title-45/subtitle-A/subchapter-C/part-164>. Accessed 28 July 2026
- Uchibeke U (2026) Before the tool call: deterministic pre-action authorization for autonomous AI agents. *arXiv:2603.20953*. <https://doi.org/10.48550/arXiv.2603.20953>
- Xiang C et al (2026) Architecting secure AI agents: perspectives on system-level defenses against indirect prompt injection attacks. *arXiv:2603.30016*. <https://doi.org/10.48550/arXiv.2603.30016>
- Zhu G, Wang C, Wang ZY, Li Z, Li Q (2026) OpenPort Protocol: a security governance specification for AI agent tool access. *arXiv:2602.20196*. <https://doi.org/10.48550/arXiv.2602.20196>