Innovation and Integrative Research Center Journal
ISSN: 2584-1491 | www.iircj.org
Volume-3 | Issue-09 | September - 2025 | Page 65-82

Generative AI and Cloud Computing: Six-Layer Reference Architecture for Security, Governance, and Operational Assurance

Dr. Praveen Kumar Shrivastava
Assistant Professor, I.T. Department
CSIT, Durg, Chhattisgarh, India.

Abstract

Generative artificial intelligence (GenAI) services increasingly depend on cloud infrastructure, managed model endpoints, retrieval systems, identity services, orchestration components, and external tools. This compositional design creates security and governance dependencies that are not fully captured by treating either the cloud platform or the language model as the sole unit of analysis. This paper proposes a six-layer reference architecture for securing GenAI-enabled cloud applications: (1) user and interaction; (2) orchestration and policy enforcement; (3) model and inference; (4) retrieval, data, and knowledge; (5) cloud runtime and infrastructure; and (6) assurance, observability, and governance. The architecture is synthesized from established cloud security guidance, generative-AI risk management guidance, and research on retrieval-augmented generation and indirect prompt injection. Its claimed contribution is not a new cryptographic primitive or a replacement for existing frameworks. Rather, it is a cross-layer design model that makes trust boundaries, control ownership, evidence generation, and failure propagation explicit across the GenAI application lifecycle. A structured, targeted evidence review is described with search sources, query strings, eligibility rules, and a screening procedure; because a complete database export and independent dual screening were not performed, the paper does not claim to be a completed PRISMA systematic review or report fabricated flow counts. Analytical validation maps six threat scenarios to preventive, detective, and corrective controls and specifies observable acceptance criteria for future implementation testing. The analysis indicates that the architecture can support traceable design reviews and test planning, but empirical effectiveness, operational overhead, and comparative superiority remain unproven. The paper therefore provides a reproducible design and validation blueprint for researchers and practitioners rather than claiming experimentally demonstrated security gains.

Innovation and Integrative Research Center Journal
ISSN: 2584-1491 | www.iircj.org
Volume-3 | Issue-09 | September - 2025 | Page 65-82

Keywords: generative artificial intelligence; cloud computing; reference architecture; large language model security; prompt injection; retrieval-augmented generation; zero trust; AI governance; threat modeling

1. Introduction

Generative AI is increasingly delivered as a cloud-hosted capability rather than as a stand-alone model. A production application may combine a user interface, application programming interfaces, orchestration logic, retrieval-augmented generation (RAG), vector and document stores, a managed or self-hosted model, cloud identity, secrets, logging, and tools that can modify external systems. The resulting security boundary is therefore distributed. A control that protects the model endpoint does not, by itself, prevent a retrieved document from carrying hostile instructions; a secure cloud network does not, by itself, ensure that an agent is authorized to invoke a payment or file-management tool; and a content filter does not replace tenant isolation, identity controls, audit evidence, or incident response.

Existing guidance provides important but differently scoped building blocks. The NIST AI Risk Management Framework (AI RMF) organizes AI risk work around Govern, Map, Measure, and Manage, while its Generative AI Profile identifies risks and suggested actions for generative systems (Autio et al., 2024; National Institute of Standards and Technology [NIST], 2023). OWASP's 2025 Top 10 for LLM Applications highlights application-specific risks such as prompt injection, sensitive information disclosure, supply-chain weaknesses, and data/model poisoning (OWASP GenAI Security Project, 2024). Cloud reference frameworks, including AWS's Security Pillar and Google Cloud's Well-Architected Framework, emphasize identity, data protection, infrastructure security, detection, and incident response (Amazon Web Services [AWS], 2024; Google Cloud, 2026). These sources are complementary; they are not interchangeable, and none should be represented as a complete GenAI application architecture.

The research problem addressed here is a synthesis problem: how can the security-relevant components and control responsibilities of a GenAI application be represented in a single architecture that supports threat analysis, design review, and test planning without claiming that a layered diagram alone guarantees security? This paper answers that question through a six-layer reference architecture and a transparent analytical validation exercise.

Innovation and Integrative Research Center Journal
ISSN: 2584-1491 | www.iircj.org
Volume-3 | Issue-09 | September - 2025 | Page 65-82

1.1 Research questions

1. RQ1: Which security and governance responsibilities must be represented when a GenAI application spans application interaction, orchestration, model inference, retrieval/data, cloud runtime, and assurance functions?
2. RQ2: How does the proposed six-layer architecture differ in purpose and unit of analysis from existing AI risk-management and cloud security reference frameworks?
3. RQ3: Can the architecture map representative GenAI/cloud threat scenarios to explicit control owners, evidence artifacts, and testable acceptance criteria?

1.2 Contributions and scope

- A six-layer architectural decomposition with explicit trust boundaries and control responsibilities.
- A comparative synthesis of AI risk guidance, cloud security frameworks, RAG research, and prompt-injection research.
- A structured search and screening protocol that can be rerun and completed by a review team.
- A threat-to-control traceability matrix and scenario-based analytical validation, with observable criteria for later prototype evaluation.

The paper does not claim that the six-layer arrangement is uniquely novel in the broad sense of being the first layered architecture for AI or cloud security. Its narrower novelty claim is that it explicitly connects (a) application-level instruction and tool authorization, (b) model-specific risks, (c) retrieval and data provenance, (d) cloud workload controls, and (e) assurance evidence in one cross-layer model. The distinction is an architectural synthesis and traceability contribution, not proof of improved security outcomes.

2. Background and Related Work

2.1 Foundation models and application composition

Foundation models can be adapted to many downstream tasks, but broad reuse also propagates weaknesses and creates evaluation challenges across applications (Bommasani et al., 2021). In cloud deployments, these risks are mediated by the surrounding application: prompt construction, retrieved context, permissions, external tools, logging, and data retention policies

influence the impact of a model failure. Accordingly, model-level evaluation and system-level security assessment should be treated as related but distinct activities.

2.2 Retrieval-augmented generation

Lewis et al. (2020) introduced a retrieval-augmented generation approach that combines parametric model knowledge with non-parametric retrieved evidence. Their results showed benefits for knowledge-intensive NLP tasks and highlighted provenance and knowledge updating as relevant problems. RAG can improve grounding, but retrieval also creates a new trust boundary: documents may be stale, misclassified, over-permissioned, or adversarial. A retrieved passage is evidence to be evaluated, not an authority that should automatically override system policy.

2.3 Prompt injection and tool-integrated systems

Greshake et al. (2023) demonstrated that indirect prompt injection can exploit application-integrated language models by placing instructions in content that the application later retrieves or processes. This finding matters architecturally because hostile instructions may enter through web pages, files, knowledge bases, or tool output rather than only through a direct user prompt. Input filtering at the interaction layer is therefore insufficient as a stand-alone control. The system also needs instruction/data separation, constrained tool permissions, output validation, and auditability at the orchestration and tool boundary.

2.4 AI risk management frameworks

The NIST AI RMF is a risk-management framework, not a deployable reference architecture. Its Govern, Map, Measure, and Manage functions help organizations establish context, identify risks, evaluate them, and respond (NIST, 2023). The NIST Generative AI Profile extends that orientation to GenAI-specific risks and lifecycle considerations (Autio et al., 2024). These frameworks help define what organizations should manage, but an engineering team still needs to map risk actions to concrete components, control owners, and telemetry.

2.5 Cloud security reference architectures

AWS and Google Cloud guidance addresses cloud workload security through identity and access management, infrastructure protection, data protection, logging/detection, response, and secure design (AWS, 2024; Google Cloud, 2026). Such guidance is directly applicable to

Innovation and Integrative Research Center Journal
ISSN: 2584-1491 | www.iircj.org
Volume-3 | Issue-09 | September - 2025 | Page 65-82

GenAI deployments because models and data pipelines run on cloud services. Its unit of analysis, however, is primarily the cloud workload or platform security posture. GenAI-specific instruction flow, retrieved content as untrusted input, model output handling, and agent tool permissions need explicit application-level treatment.

2.6 Comparison and synthesis

Source / approach	Primary unit of analysis	Strength	Limitation for the present problem	Role in this paper
NIST AI RMF 1.0 (NIST, 2023)	Organization and AI risk-management lifecycle	Governance, context, measurement, and risk response	Does not prescribe a six-layer deployment topology or per-component enforcement points	Governance and lifecycle mapping
NIST GenAI Profile (Autio et al., 2024)	Generative AI risk categories and actions	GenAI-specific risk identification and management	Not a concrete runtime reference architecture; controls must be engineered locally	Risk taxonomy and control rationale
OWASP Top 10 for LLM Applications 2025	Application-level LLM risk patterns	Actionable risk awareness, including prompt injection and data disclosure	Top-10 list is not a complete architecture or assurance method	Threat scenario selection
AWS Security Pillar (AWS, 2024)	Cloud workloads and cloud security practices	Identity, traceability, defense in depth, data protection, response	Does not by itself model retrieved instructions or model-to-tool authority boundaries	Cloud runtime and security operations controls

Innovation and Integrative Research Center Journal
ISSN: 2584-1491 | www.iircj.org
Volume-3 | Issue-09 | September - 2025 | Page 65-82

Google Cloud Well-Architected Framework (Google Cloud, 2026)	Cloud workload design and operations	Security by design, zero trust, data and infrastructure protection	Provider-oriented guidance; GenAI app-specific trust boundaries still need mapping	Cloud control baseline
Lewis et al. (2020), RAG	Retrieval-augmented NLP model	Grounding through external non-parametric evidence	Does not solve authorization, provenance trust, or adversarial retrieved content by itself	Retrieval/data layer design
Greshake et al. (2023), indirect prompt injection	Application-integrated LLM attack path	Shows that untrusted external content can influence model behavior	Attack analysis is not a full cross-layer governance architecture	Orchestration and trust-boundary requirements
Proposed six-layer model	End-to-end GenAI-enabled cloud application and its evidence trail	Connects trust boundaries, control ownership, and test criteria across layers	Conceptual; effectiveness and novelty versus all existing designs are not empirically established	Synthesis and test-planning artifact

The literature does not present a simple disagreement in which one framework is correct and another is incorrect. Instead, the sources differ in purpose and granularity. RAG research demonstrates a capability benefit under task-specific evaluation, while prompt-injection research shows that application composition can create attack paths that are not resolved by retrieval alone. AI risk frameworks emphasize governance and lifecycle management; cloud frameworks emphasize workload protection and operations. The gap addressed here is the lack of an explicit mapping, in this paper's selected evidence set, from GenAI-specific instruction

and retrieval risks through cloud controls to observable assurance artifacts. This is a bounded synthesis claim, not a claim that no similar architecture exists anywhere.

3. Review Method

3.1 Review design and reporting choice

This manuscript uses a structured, targeted evidence review to inform a conceptual architecture. It is not presented as a completed systematic review. PRISMA 2020 is a reporting guideline for systematic reviews; using its name requires a defensible search record, eligibility screening, and flow counts (Page et al., 2021). Because this manuscript's evidence gathering was not based on complete exports from subscription databases and independent duplicate screening, it does not invent a PRISMA flow diagram or counts. The protocol below is designed to make the search repeatable and to support conversion into a full systematic or scoping review.

3.2 Search sources and search date

The evidence search for this draft was conducted on 18 October 2026 using web-indexed scholarly and official publication sources. Source portals consulted included NeurIPS Proceedings, ACM Digital Library/publisher-indexed records, arXiv for preprints, NIST publications, the OWASP GenAI Security Project, AWS documentation, Google Cloud documentation, and the PRISMA statement website. This is a targeted source search, not a claim of a comprehensive search of Scopus, Web of Science, IEEE Xplore, or ACM Digital Library exports.

3.3 Search strings

The following query strings define the search logic used to identify the core sources and can be rerun verbatim. Search engines may tokenize or rank terms differently; all results should be logged by the review team.

- "generative AI" AND cloud AND security AND architecture
- "large language model" AND "cloud security" AND architecture
- "retrieval-augmented generation" AND security OR provenance
- "indirect prompt injection" AND application-integrated language models

Innovation and Integrative Research Center Journal
ISSN: 2584-1491 | www.iircj.org
Volume-3 | Issue-09 | September - 2025 | Page 65-82

- "AI Risk Management Framework" AND "Generative AI Profile"
- "OWASP Top 10 for LLM Applications 2025"
- "cloud well-architected framework" AND security AND identity
- "PRISMA 2020 statement"

3.4 Eligibility criteria

Criterion	Include	Exclude
Topic	GenAI/LLM system security, cloud security architecture, RAG, AI risk governance, or review methodology directly relevant to the architecture	General AI commentary without an identifiable security, architecture, or governance implication
Source type	Peer-reviewed conference/journal papers; standards or government technical reports; official framework documentation; clearly identified preprints used cautiously	Unsourced blogs, promotional claims without technical detail, duplicated summaries, or unverifiable citations
Relevance	Source supports a named risk, design principle, architectural component, or review method	Source does not materially inform the research questions
Access/verification	Bibliographic record or official publisher/organization page can be located	Reference cannot be checked against an original or authoritative record
Language/date	English; no fixed lower date limit for foundational work; current guidance checked through 18 October 2026	Non-English material not assessed in this draft

3.5 Screening and extraction procedure

4. Deduplicate records by title, DOI, and publisher record.
5. Screen titles and abstracts/snippets against the topic and relevance criteria.

Innovation and Integrative Research Center Journal
ISSN: 2584-1491 | www.iircj.org
Volume-3 | Issue-09 | September - 2025 | Page 65-82

6. Retrieve the publisher, standards-body, or official source record for potentially eligible items.
7. Record bibliographic metadata, source type, scope, main claims, stated limitations, relevant threat categories, and the architecture layer(s) informed.
8. Exclude records that cannot be verified or do not support a specific design or risk claim; record the reason for exclusion.
9. Use a source-to-claim matrix so that each major architectural claim can be traced to one or more references.
10. Before describing the review as systematic, rerun the searches in the chosen bibliographic databases, save the search dates and exports, complete full-text screening, and have two reviewers independently screen a sample or all records with disagreements resolved by consensus.

3.6 Evidence synthesis and limitations

Evidence was synthesized narratively by comparing source purpose, unit of analysis, threat coverage, and the level at which controls are specified. The synthesis is not a meta-analysis: the sources use different methods, populations, and outcomes, and no common quantitative effect size can be calculated. No numeric count of included studies is reported because a complete, auditable screening log was not generated for this draft. This limitation is material and should be resolved before submission to a venue that expects a systematic review.

4. Proposed Six-Layer Reference Architecture

The architecture is a logical decomposition rather than a mandatory deployment topology. A production system may combine or split layers, but the control responsibilities and trust boundaries should remain identifiable. Security and governance are cross-cutting: Layer 6 observes and evaluates all other layers, while identity, authorization, data classification, and policy enforcement must be implemented at the relevant enforcement points.

Layer	Primary components	Security responsibility	Representative evidence
L1. User and interaction	Web/mobile UI, API gateway, client	Authenticate caller; validate request shape; rate-limit; label tenant and purpose; prevent	Identity event, request ID, rate-

Innovation and Integrative Research Center Journal
ISSN: 2584-1491 | www.iircj.org
Volume-3 | Issue-09 | September - 2025 | Page 65-82

	identity, request normalization	untrusted input from being treated as policy	limit event, input validation result
L2. Orchestration and policy enforcement	Prompt assembly, workflow/agent controller, tool router, policy engine, approval gates	Separate trusted instructions from untrusted data; enforce tool allowlists and least privilege; validate arguments; require approval for high-impact actions	Policy decision, tool-call authorization record, approval event, blocked-action reason
L3. Model and inference	Hosted/self-hosted LLM, model gateway, inference configuration, safety filters	Model/provider allowlisting; version and configuration control; input/output handling; quotas; safety evaluation; prevent secrets from being included unnecessarily	Model/version metadata, safety result, latency/cost metrics, deployment attestation where available
L4. Retrieval, data, and knowledge	Document ingestion, parser, embedding pipeline, vector index, object store, retrieval service	Classify and authorize documents; tenant-aware retrieval; provenance and freshness checks; ingestion scanning; protect embeddings and source content; treat retrieved text as untrusted	Document ID/hash, ACL decision, source URI, retrieval trace, ingestion scan result
L5. Cloud runtime and infrastructure	Containers/VMs, network, IAM, secrets, KMS, storage, service mesh, CI/CD	Workload identity; network segmentation; encryption; secrets management; patching; software/model supply-chain controls; isolation and backup	IAM/configuration audit, vulnerability scan, key-use event, deployment artifact/SBOM, network telemetry
L6. Assurance, observability,	Central logging, security monitoring,	Correlate events across layers; preserve evidence; evaluate control effectiveness; manage	Correlated trace, alert, evaluation report, incident

Innovation and Integrative Research Center Journal
ISSN: 2584-1491 | www.iircj.org
Volume-3 | Issue-09 | September - 2025 | Page 65-82

and governance	evaluations, incident response, policy registry, risk register	retention and privacy; govern exceptions and changes; coordinate response	timeline, exception approval, periodic review record

4.1 Trust boundaries and data flow

11. Boundary A—caller to application: user identity, tenant, purpose, and request limits must be established before orchestration.
12. Boundary B—application to model: prompts may contain confidential context; minimize disclosure and record model/provider configuration.
13. Boundary C—retrieval to model: retrieved documents are untrusted content with provenance and access constraints, not higher-priority instructions.
14. Boundary D—orchestrator to tools: tool invocation is an authority transition; the model's suggestion is not sufficient authorization.
15. Boundary E—workload to cloud control plane: service identities, secrets, network paths, and deployment artifacts must be governed independently of natural-language intent.
16. Boundary F—runtime events to assurance plane: logs should support investigation while avoiding unnecessary retention of raw prompts, secrets, or personal data.

4.2 Cross-cutting design principles

- Least privilege: a model or agent receives only the permissions needed for the current task.
- Instruction/data separation: retrieved and user-supplied content cannot override system policy merely by containing imperative language.
- Provenance: retrieved sources and model/tool versions are recorded sufficiently to reproduce or investigate consequential outputs.
- Defense in depth: content filters supplement, rather than replace, authorization, isolation, and audit controls.
- Fail safely: uncertain or malformed high-impact tool requests are denied or escalated rather than executed.
- Privacy-aware observability: capture security-relevant metadata and correlated identifiers while minimizing sensitive payload retention.

Innovation and Integrative Research Center Journal
ISSN: 2584-1491 | www.iircj.org
Volume-3 | Issue-09 | September - 2025 | Page 65-82

- Continuous evaluation: repeat tests after model, prompt, retrieval corpus, tool, policy, or cloud configuration changes.

5. Novelty and Differentiation

The proposed architecture is intentionally positioned as a reference and traceability model. Layering alone is not novel, and individual controls such as least privilege, logging, data classification, or prompt filtering are established practices. The contribution lies in the explicit mapping between GenAI-specific trust boundaries and cloud-security evidence across a single application lifecycle.

Dimension	**Existing frameworks commonly provide**	**Specific addition in this model**	**What remains unproven**
Governance to implementation	AI RMF provides risk functions and categories	Connects risk concerns to layers, control owners, and evidence artifacts	Whether the mapping improves audit time or control completeness
Cloud-to-LLM integration	Cloud frameworks cover IAM, data, infrastructure, and detection	Separates model/inference and retrieval/data from cloud runtime while preserving their dependencies	Whether this decomposition is better than an organization's existing architecture
Prompt-injection boundary	OWASP and research describe prompt injection and related risks	Treats retrieval-to-model and orchestrator-to-tool as distinct trust boundaries	Attack resistance under adversarial testing
Tool authority	General IAM and application security support authorization	Makes tool calls explicit policy decisions with argument validation and optional human approval	False-positive/false-negative rates and workflow overhead

Innovation and Integrative Research Center Journal
ISSN: 2584-1491 | www.iircj.org
Volume-3 | Issue-09 | September - 2025 | Page 65-82

Assurance traceability	Logging and risk frameworks recommend monitoring and evaluation	Requires a cross-layer event trail linking request, retrieval, model, policy, tool, and response identifiers	Operational cost, privacy impact, and forensic completeness

5.1 Bounded novelty claim

Based on the selected evidence set, the paper identifies a practical integration gap: existing sources provide risk taxonomies, cloud control principles, or specific model/application findings, but the selected sources do not themselves provide the exact six-layer control-to-evidence matrix proposed here. This statement is deliberately bounded to the reviewed sources. A definitive novelty claim requires a broader search of related architecture papers, including IEEE Xplore, ACM Digital Library, Scopus, Web of Science, and relevant preprint repositories. The manuscript should not claim to be the first architecture of its kind until that search is completed.

6. Analytical Validation

6.1 Validation method

Validation in this paper is analytical and scenario-based. Six representative threat scenarios are derived from the reviewed risk guidance and research, then traced through the architecture to identify preventive controls, detection evidence, response actions, and observable acceptance criteria. This exercise tests internal completeness and traceability: every scenario must have a named enforcement point and an evidence artifact. It does not measure real-world attack resistance, prove control effectiveness, or substitute for a prototype, penetration test, or independent expert assessment.

6.2 Threat-to-control traceability matrix

Scenario	**Entry point / risk**	**Preventive controls and layers**	**Detection / evidence**	**Acceptance criterion for future test**

Innovation and Integrative Research Center Journal
ISSN: 2584-1491 | www.iircj.org
Volume-3 | Issue-09 | September - 2025 | Page 65-82

S1. Indirect prompt injection in retrieved page	Malicious instructions embedded in a retrieved document; L4 to L3	L4 provenance and trust labels; L2 instruction/data separation; L2 policy-controlled tool routing	Retrieval trace, policy decision, blocked tool call, incident correlation ID	A seeded hostile document must not cause an unauthorized tool action; the trace must identify the source document and policy decision
S2. Cross-tenant retrieval leakage	Vector search or document ACL bug returns another tenant's content; L4/L5	Tenant-scoped identity and retrieval filters; per-document authorization; L5 storage/network isolation	Tenant ID, ACL decision, document ID/hash, denied-access event	A test query under tenant A must not return tenant B's protected canary document
S3. Sensitive information disclosure in model output	Prompt/context includes secrets or restricted data; L1/L2/L3/L4	Data minimization; secret redaction; access-aware retrieval; output DLP/policy checks	Redaction event, data-classification label, output-check result	Known secret canaries must not appear in user-visible output or logs outside approved secure handling
S4. Unauthorized	Model proposes deletion, payment,	Tool allowlist; schema and argument	Authorization decision, approval identity,	Disallowed action is denied; high-

Innovation and Integrative Research Center Journal
ISSN: 2584-1491 | www.iircj.org
Volume-3 | Issue-09 | September - 2025 | Page 65-82

high-impact tool call	external message, or privilege change; L2	validation; least-privilege tool identity; approval gate for high impact	tool invocation record	impact action cannot execute without required approval; denied attempt is logged
S5. Compromised dependency or model artifact	Supply-chain risk in package, model, container, or prompt/template dependency; L3/L5	Trusted registry; version pinning; artifact integrity checks; CI/CD scanning; deployment approvals	SBOM or artifact manifest, scan result, deployment identity, version metadata	A deliberately unapproved artifact is rejected by the deployment gate and leaves an auditable event
S6. Missing or privacy-invasive telemetry	Incomplete traces prevent investigation, or logs retain excessive sensitive payloads; L6	Correlated request IDs; metadata-first logging; access controls and retention policy; redaction	Trace completeness report, access log, retention/deletion record	A simulated incident can be reconstructed across required events while a seeded secret is absent from general-purpose logs

6.3 Coverage and traceability criteria

For design review, the architecture can be assessed against the following binary criteria. A criterion is satisfied only when a design artifact or test result is available; prose assurances alone do not count.

- C1—Control ownership: every critical threat has an assigned enforcement layer and named operational owner.

Innovation and Integrative Research Center Journal
ISSN: 2584-1491 | www.iircj.org
Volume-3 | Issue-09 | September - 2025 | Page 65-82

- C2—Least privilege: model-generated tool requests are independently authorized against caller, tenant, task, and tool policy.
- C3—Retrieval isolation: authorization is enforced before protected content is included in model context.
- C4—Provenance: consequential responses can be linked to retrieved sources and relevant model/policy versions.
- C5—Auditability: the incident path can be reconstructed using correlated identifiers without requiring unrestricted raw-prompt access.
- C6—Change assurance: model, prompt, corpus, tool, and infrastructure changes trigger appropriate re-evaluation.
- C7—Failure handling: denied, ambiguous, or malformed high-impact actions fail closed or require a documented human approval path.
- C8—Privacy: telemetry retention and access rules are documented and tested against data-minimization requirements.

Under the conceptual review conducted here, the proposed architecture provides a location and evidence expectation for each criterion. Actual pass/fail status cannot be assigned without implementation artifacts and test execution. The validation result is therefore a completeness argument for the proposed design, not an empirical performance result.

6.4 Proposed empirical evaluation protocol

A follow-on prototype study should compare the six-layer implementation with a baseline implementation that has equivalent functionality but lacks explicit cross-layer policy/evidence mapping. Use a fixed test corpus and seeded scenarios, with at least three repetitions per scenario and configuration. Report unauthorized-action rate, cross-tenant leakage rate, sensitive-canary exposure rate, detection precision/recall, trace completeness, end-to-end latency, token consumption, operational cost, and human-approval burden. Freeze model version, prompts, retrieval corpus, tools, cloud configuration, and random seeds where applicable. Publish attack prompts and benign controls where safe, along with scripts and configuration manifests. Do not infer effectiveness from a small scenario suite alone; include limitations and external-validity discussion.

Innovation and Integrative Research Center Journal
ISSN: 2584-1491 | www.iircj.org
Volume-3 | Issue-09 | September - 2025 | Page 65-82

7. Limitations and Threats to Validity

First, the review is targeted rather than systematic. The evidence set is not exhaustive, and the manuscript does not report database-wide record counts. Second, the proposed architecture is a synthesis of established principles; the claim of novelty is bounded and requires a broader architecture-focused literature search. Third, analytical scenario validation checks traceability but not actual attack resistance or effectiveness. Fourth, the architecture is provider-neutral at the logical level, but concrete implementations differ across managed model APIs, private deployments, orchestration libraries, and cloud platforms. Fifth, the six scenarios cannot represent the full GenAI threat landscape, including model extraction, denial of service, multimodal attacks, agentic long-horizon behavior, and emerging supply-chain issues. Sixth, control effectiveness depends on implementation quality, policy completeness, identity configuration, and operational maturity. These limitations prevent claims of measured security improvement or general superiority.

8. Conclusion

This paper proposes a six-layer reference architecture for GenAI-enabled cloud applications and positions its contribution as cross-layer traceability rather than as a replacement for established AI risk or cloud security frameworks. The architecture separates interaction, orchestration, inference, retrieval/data, cloud runtime, and assurance responsibilities while explicitly identifying the boundaries where untrusted content or model-generated actions can cross into privileged operations. Comparison with NIST AI risk guidance, OWASP LLM risks, cloud well-architected guidance, RAG research, and prompt-injection research motivates the synthesis. A structured search protocol and scenario-based threat-to-control matrix improve transparency, but the present review does not claim PRISMA-compliant systematic-review completion and does not report experimental results. The next research step is a broader database search followed by prototype implementation, controlled adversarial testing, and comparative evaluation. Until that work is completed, the architecture should be treated as a design and test-planning proposal whose operational effectiveness remains an open empirical question.

References

1. Amazon Web Services. (2024). Security pillar—AWS Well-Architected Framework. https://docs.aws.amazon.com/wellarchitected/latest/security-pillar/

2. Autio, C., Schwartz, R., Dunietz, J., Jain, S., Stanley, M., Tabassi, E., Hall, P., & Roberts, K. (2024). Artificial intelligence risk management framework: Generative artificial intelligence profile (NIST AI 600-1). National Institute of Standards and Technology. https://doi.org/10.6028/NIST.AI.600-1
3. Bommasani, R., Hudson, D. A., Adeli, E., Altman, R., Arora, S., von Arx, S., et al. (2021). On the opportunities and risks of foundation models. arXiv. https://doi.org/10.48550/arXiv.2108.07258
4. Google Cloud. (2026). Well-Architected Framework: Security, privacy, and compliance pillar. https://docs.cloud.google.com/architecture/framework/security
5. Greshake, K., Abdelnabi, S., Mishra, S., Endres, C., Holz, T., & Fritz, M. (2023). Not what you've signed up for: Compromising real-world LLM-integrated applications with indirect prompt injection. Proceedings of the 16th ACM Workshop on Artificial Intelligence and Security. https://doi.org/10.1145/3605764.3623985
6. Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W.-t., Rocktäschel, T., Riedel, S., & Kiela, D. (2020). Retrieval-augmented generation for knowledge-intensive NLP tasks. Advances in Neural Information Processing Systems, 33. https://proceedings.neurips.cc/paper/2020/hash/6b493230205f780e1bc26945df7481e5-Abstract.html
7. National Institute of Standards and Technology. (2023). Artificial intelligence risk management framework (AI RMF 1.0) (NIST AI 100-1). U.S. Department of Commerce. https://doi.org/10.6028/NIST.AI.100-1
8. OWASP GenAI Security Project. (2024). OWASP Top 10 for LLM applications 2025. https://genai.owasp.org/resource/owasp-top-10-for-llm-applications-2025/
9. Page, M. J., McKenzie, J. E., Bossuyt, P. M., Boutron, I., Hoffmann, T. C., Mulrow, C. D., et al. (2021). The PRISMA 2020 statement: An updated guideline for reporting systematic reviews. BMJ, 372, n71. https://doi.org/10.1136/bmj.n71
10. Rose, S., Borchert, O., Mitchell, S., & Connelly, S. (2020). Zero trust architecture (NIST Special Publication 800-207). National Institute of Standards and Technology. https://doi.org/10.6028/NIST.SP.800-207