Journal of Information Systems Engineering and Management
2026, 11(2s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Zero Trust Architecture for Generative AI: Securing Prompts, Retrieval, and Agent Tool-Use in Regulated Environments

Narendra Bhargav Boggarapu
Wells Fargo, USA

ARTICLE INFO

Received: 05 March 2026

Accepted: 08 March 2026

ABSTRACT

The deployment of generative AI systems in controlled contexts presents an entirely enlarged attack surface: The use of prompts, retrieval pipelines and agentic tool invocations are all unique and consequential authorization events that cannot be regulated by perimeter security controls. Zero Trust Architecture offers the conceptual basis needed to overcome these issues with a requirement to do policy review on a per-request basis based on verified identity, device posture, and real-time risk indicators as opposed to implied session trust. An architecture that can be defended as ZT-GenAI necessitates coordinated enforcement across five interdependent control layers: Prompt Envelope integrity validation, Attribute-Based Access Control governed retrieval, token-level Redaction Gates, Tool Broker intermediation with scoped token issuance, and workflows of supervisory approval of high-impact agentic actions. Such an architecture can be evaluated through injection resilience testing, policy effectiveness testing, audit completeness testing, operational latency sensitivity testing, and so on, and the evidence they produce can be the basis of regulatory defensibility and model risk governance. The architectural designs shown herein provide a repeatable template for implementing generative AI capabilities into sensitive operational settings without any authorization rigor or auditability sacrifices.

Keywords: Zero Trust Architecture, Generative AI Security, Prompt Injection, Agentic Tool Governance, Model Risk Management

1. Introduction

Generative AI-based platforms are a radical change in the way the enterprise systems handle sensitive data, make decisions, and connect to downstream services. In contrast to deterministic software, systems built using large language models (LLM) can take natural language prompts as a functional input, so they can provide highly workflow-rich, context-driven workflows, operating within a single inference chain across data retrieval, multi-step reasoning, and running tools. The flexibility, however, comes with new security issues that did not exist in the earlier design of conventional perimeter-based architectures. Prompts can represent sensitive investigative purposes, retrieval pipelines can turn into the hidden routes to privileged documents, and agentic tool use translates natural language commands into immediate operations on enterprise APIs and databases.

The traditional network-perimeter security model is based on an implicit trust model, wherein once a principal is authenticated at the boundary, trust is assumed to continue through the interaction at the session level. This is a basic incompatibility with GenAI deployments, in which every inference can access a new corpus of data, use a new tool, or gain privilege via a well-designed context injection. A corrupted or maliciously built prompt may walk through retrieval pipelines and tool execution chains in a manner inaccessible to any single boundary control and thus view the natural language fluency of the model as an amplifier of an attack.

According to its official definition, which is defined in the NIST Special Publication 800-207, Zero Trust Architecture (ZTA) provides a more principled alternative: no implicit trust is given to any subject, whether human or machine, based on network location alone or previous authentication [1]. Verified identity, device posture and real-time risk signals are used to establish trust and re-evaluate it

on a per-request basis. In the policy, NIST SP 800-207 presents the Policy Decision Point (PDP) as the key control element, the logical unit that compares access requests with policy rules and makes authorization decisions and binding obligations. In the case of GenAI, this paradigm is directly mapped to the belief in per-inference policy enforcement: each and every prompt, retrieval query, and tool invocation is a discrete authorization event that can be independently evaluated.

This underpinning is expanded by the CISA Zero Trust Maturity Model (ZTMM) v2.0, which defines five pillars that are interdependent and placed along a maturity continuum between Traditional and Optimal [2]. The GenAI system will satisfy all five pillars with a single call to inference: a prompt will be given with identity context, generated through a device session, sent across network segments, implemented in an application runtime, and accessed or converted to classified data. The main architectural thesis of Zero Trust GenAI (ZT-GenAI) is to address all five pillars in a coherent policy engine, instead of having point-in-time controls in silos.

Security imperative is especially sharp with regulated environments in which data residency regulations, anti-money laundering or know-your-customer requirements enforce strong restrictions on the authority to access and act on data. The granular, real-time authorization is not only the best practice but also a mandatory requirement since a GenAI assistant working in these situations can simultaneously process transaction alerts, case notes, and regulatory guidance. This paper analyzes architectural principles, component design, and methodology of evaluation of a defensible system, the ZT-GenAI system. Section 2 aligns Zero Trust principles with threat vectors in GenAI. Section 3 deals with identity-first prompt design and retrieval security. The fourth section is agent tool-use governance. Section 5 suggests a framework of evaluation of policy effectiveness and audit completeness through the whole inference chain.

2. Zero Trust Principles and the Generative AI Threat Landscape

The three Zero Trust principles, explicit verification, least privilege access, and assume breach, are especially aggressive in the GenAI architecture, where breaches of trust are transmitted silently by pipelines of inferences before showing up as data exfiltration or unauthorized system alteration. The NIST Artificial Intelligence Risk Management Framework, released in January 2023 as NIST AI 100-1, defines the risks in AI systems as having a unique character, in that AI systems can be trained on data that can evolve over time, occasionally dramatically and unpredictably, both in the way the system functions and its reliability in ways that are hard to comprehend [3]. This is an unstable risk profile, which is aggravated by the natural language interface of GenAI systems, turning per-inference policy evaluation into a structural requirement and not an optional improvement. It has also been noted that there is an overlap between ZTA principles and the security architecture of GenAI. Other categories of security weaknesses are the 10 most common vulnerability types related to an LLC application, which the Open Web Application Security Project has listed and categorized [4].

The first principle that requires architectural translation is explicit verification at inference time. In typical applications, authentication is performed at discrete authentication points, such as the event of a login, the exchange of tokens, or the validation of API keys, and then implicitly the trust of the session is extended. GenAI pipelines add more steps of operation processes activating prompt processing, semantic retrieval, generation, and tool execution, and all of these are independent points of trust decisions that cannot be subsequently controlled by session authentication. The AI RMF defines 4 key functions: GOVERN, MAP, MEASURE and MANAGE, which, when combined, are more than mere indicators of the continuous, lifecycle-based risk assessment needed instead of the pre-deployment risk evaluation that is the requirement of GenAI deployments [3]. Adversarial prompt inputs, distributional shift, and emergent model behaviors are runtime conditions that may cause AI systems to deviate significantly from design-time intent, which contributes to the need to continually enforce policies at every inference step [3].

The second principle is least privilege on data, generation and tool planes. Without explicit limitations on scope, one tainted prompt may proceed to execute over-privileged operations on the retrieval

Journal of Information Systems Engineering and Management
2026, 11(2s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

corpus, generation template and tool execution surface at the same time. The OWASP LLC vulnerability catalogue lists Excessive Agency, whereby an LLM-based system receives greater capability than is needed in a proclaimed task, as a category of high-severity vulnerabilities, with the root causes directly originating in the lack or inadequacy of least-privilege controls at the tool and data access layers [4]. A structurally weak design vulnerability, also called Insecure Plugin Design, also shares the risk with a lack of access control over inputs into tools, a structural weakness that the least-privilege Token scoping at a dedicated Tool Broker is specifically created to address [4].

The assume-breach principle requires architecture with limited blast radii at all levels. The most popular GenAI breach facility, prompt injection, content placed in user inputs or retrieved documents that tries to override system instructions or misuse tool execution, is the first-highest-priority vulnerability in GenAI auditing per OWASP [4]. The AI RMF lists 3 categories of AI bias, namely, systemic, computational and statistical, and human-cognitive, each of which can be used or compounded with the help of adversarial prompt construction, as well as injection resilience demands architectural isolation over content filtering [3]. The assume-breach posture through the entire inference chain is operationalized by continuous policy evaluation, which is a result of live identity and risk indicators that integrate to a central Policy Decision Point.

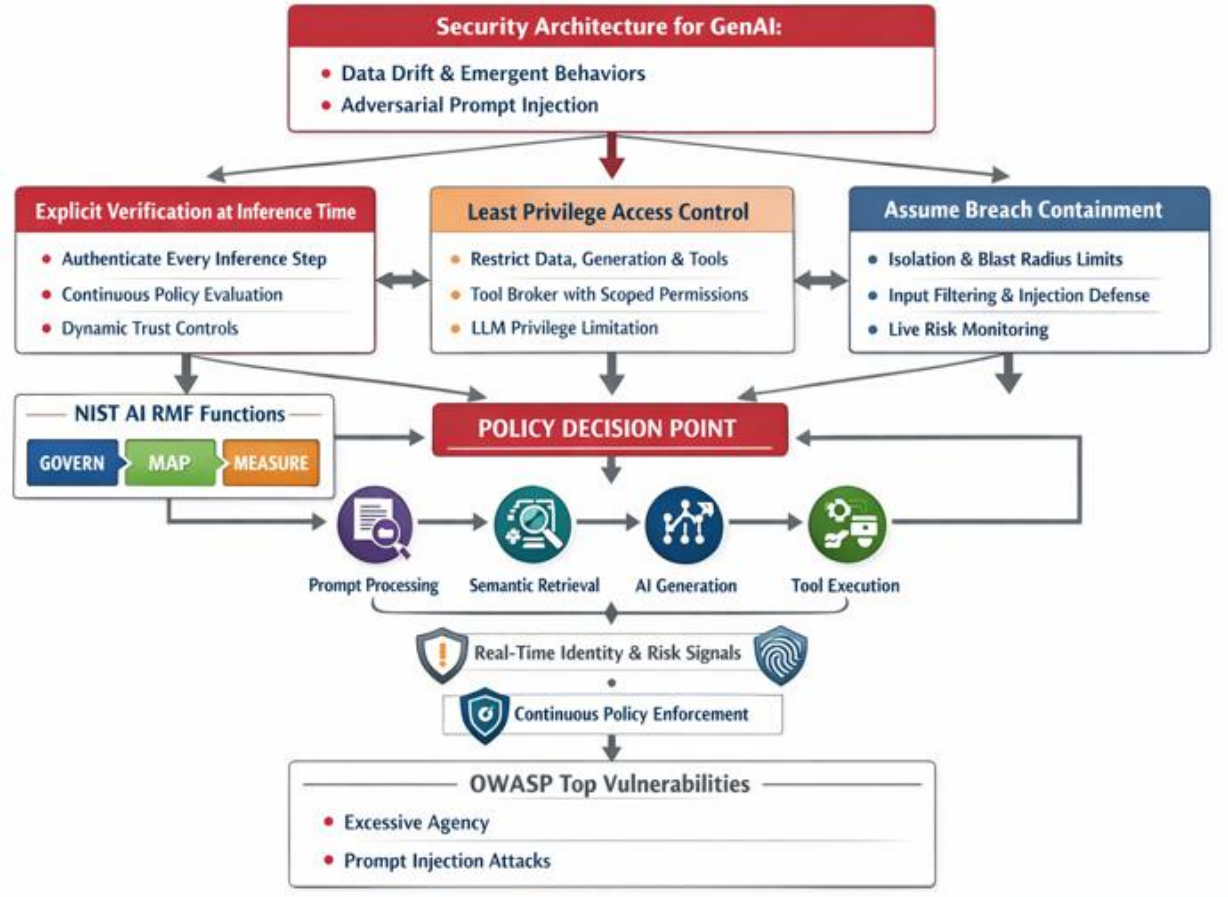

Fig 2: Zero Trust Architecture for GenAI Inference Pipelines [3, 4]

3. Prompt Envelopes, Retrieval Security, and Data Classification

The principle of identity-first design states that all operational requests, such as a database query, an API call, or an LLM prompt, have to represent a verifiable identity context that may be inspected, assessed, and documented by the authorization layer prior to processing being carried out. This principle is implemented in ZT-GenAI by 3 complementary principles: an Attribute-Based Access Control (ABAC) regulated layer of Retrieval Security, the Prompt Envelope pattern, and a Redaction

Journal of Information Systems Engineering and Management
2026, 11(2s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Gate that operates based on the labels of data classification. The combination of these mechanisms can be used to convert abstract Zero Trust principles into specific, enforceable controls at each point of the GenAI inference pipeline.

The Prompt Envelope pattern assumes each inference request is a privileged operation needed to be explicitly authorized within an authorization context, and then the model or retrieval pipeline is called upon. Every envelope includes the subject identity, which can be human or machine workload; the proclaimed purpose of the inference; the allowed scope of data, which specifies what type of corpus or classification tier can be retrieved; a risk tier based on role, device posture, and session indicators; and an identifier of correlation between the request and a continuous audit trace. The AI Gateway, which is a Policy Enforcement Point, denies any immediate response with no integrity-covered envelope prior to the request proceeding. The most operationally important property of purported purpose is its binding: an AML alert summary declaration entitles the access to the AML policy corpora, though it does not provide access to the customer master records or the history of payment transactions unless expressly permissible in the active policy ruleset. This is an enforceable restriction that controls scope creep in a single session and directly bounds the blast radius of adversarial prompt successful authentication [5].

The second control layer is retrieval security, which eliminates a severe weakness in Retrieval-Augmented Generation architectures: semantic similarity search can also result in documents that the subject is not authorized to access when the entitlements are enforced during indexing time and not during query time. Published in January 2014 and updated as of August 2, 2019, NIST Special Publication 800-162 defines Attribute-Based Access Control as a model where access decisions are tested against attributes of the subject, the resource, and the operating environment, a model that directly overlays per-query entitlement enforcement in the context of vector retrieval pipelines [5]. When applied to GenAI retrieval, ABAC mandates that all similarity searches be screened by the classification clearance tier of the subject and membership in the line of business and data residence area as well as document-level tags prior to results getting into the context window of the model. Architectural consistency Continuous evaluation at retrieval time, as opposed to indexing time, is necessary architecturally since the document classification labels as well as the sets of subject attributes vary dynamically across the corpus lifecycle [5].

Retrieved information exiting the model is conditioned through a Redaction Gate that implements data classification policies in real-time to send document fragments to the prompt context window. Fields in documents belonging to an authorized corpus can have higher classification than the existing authorization level of the subject. The NIST Special Publication 800-53 Revision 5 sets the concept of least privilege as a guiding security concept, control AC-6, which states that subjects receive access control rights no more than necessary to perform their assigned functions [6]. Using this principle at the token level, the Redaction Gate encrypts fields that are beyond the permissions of the subject prior to the chunk being passed to the model and thus distinguishes between retrieval authorization and content authorization. The two-step design allows the implementation of least-privilege enforcement on a fine-grained basis at the field level, significantly minimizing the chances of incidental sensitive data leakage during generation [6].

Control Layer	Core Function	Enforced Outcome
Prompt Envelope	Binds identity, purpose, data scope, risk tier, and audit trace to each inference	Prevents unauthorized prompts and limits session blast radius
Retrieval Security (ABAC)	Enforces per-query access control on vector retrieval using subject and data attributes	Blocks unauthorized documents from entering model context
Redaction Gate	Applies classification-based, field-level filtering before generation	Ensures least-privilege data exposure and prevents sensitive leakage

Table 1: Identity-First Controls in ZT-GenAI (Prompt, Retrieval, and Redaction) [5, 6]

Journal of Information Systems Engineering and Management
2026, 11(2s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

4. Tool Brokers, Scoped Tokens, and Approval Workflows

The greatest risk profile of operation in a ZT-GenAI deployment is agentic GenAI systems able to plan multi-step actions and make calls to external services independently. The ability to transfer natural language reasoning to API calls implies that a single adversarial or incorrect prompt can have real-world implications: record creation, transaction upload, the rise of a regulatory case, or alterations to data. Formal threat modeling studies of applications that use the LLM have named MITRE ATLAS as a listing of 14 high-level tactics, including initial access, persistence, defense evasion, privilege escalation, and exfiltration, that are used to characterize 58 specific attack techniques targeting AI components, which offers a subsystematic taxonomy of the threat surface that agentic tool governance must consider [7]. The application of agentic tool use governed by the Zero Trust principles demands a rigid architectural distinction between the reasoning layer and action layer of the model and the action layer of the enterprise, which can be implemented with the help of a special optimization that is an agent of a Policy Enforcement Point.

The Tool Broker is such an intermediary: a special element that intercepts each tool invocation request sent by the model runtime and provides authorization before any enterprise API is invoked. Instead of supplying the model with the direct API credentials or general OAuth scopes, the Tool Broker generates scoped and time-limited tokens when each individual authorized tool invocation is performed. The scope of the token is restricted to a particular API endpoint, a set of allowed parameter values, and the identity of the subject at hand at the time of the request, restricting the reuse of a token to a new session, parameter substitution attacks, and cross-ense endpoint inter-use attacks within the same API surface. The Threat model of applications with LLM-integrated applications specifically highlights Excessive Agency, the occurrence of high access privileges within the system in which the embedded system is used, and Insecure Plugin Design, which is due to the lack of input validation of the parameters of the tool, as high severity classes of threats that the Tool Broker pattern is designed to prevent [7].

The risk that poses the greatest threat to tool governance is direct prompt injection attacks, which are the most operationally immediate. Empirical study of injection attacks under the PROMPTINJECT framework, tested on 35 base prompts that could represent common production deployment settings, proved that goal hijacking attacks succeed with 58.6% success rates under adversarial setting optimization, and prompt leaking attacks succeed with 23.6% success rates under adversarial setting optimization [8]. These values put a quantitative lower bound on the injection risk that an architectural control has to mitigate: unless a Tool Broker is performing parameter guardrails without reference to model output, injection rates of this scale are directly converted into unauthorized tool calls against enterprise APIs. The parameter guardrail imposes structural and semantic constraints on inputs in tool inputs at the Tool Broker boundary; that is, permitted value lists, numerical range checks, and referential integrity checks with the authorized entity range of the subject and is implemented as a rule-based validation layer that does not rely on the stochastic output of the model when parameter guardrail logic is bypassed; however, in practice most guardrail logic is not normally bypassed [8].

Tool invocations should not be characterized by the same authorization posture. Read operations to a case management system have significantly low operational risk compared to write operations that generate a regulatory filing or indicate transactions to review or modify customer records. High-impact actions are executed subject to supervisory approval workflows: the Tool Broker suspends the execution, forwards the proposed action with all parameter context to a specific approver, human or automated, and logs the approval decision in the Evidence Ledger and sends the API call. The assume-breach principle exists, which requires a successfully built tool call to a model to satisfy a second authorization barrier that cannot be violated with model output alone. Each tool invocation, irrespective of risk level, is logged along with caller identity, scoped token value, policy binding, and times with a full, reconstructible audit trail pegged to the subject's active prompt envelope [7].

Journal of Information Systems Engineering and Management
2026, 11(2s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

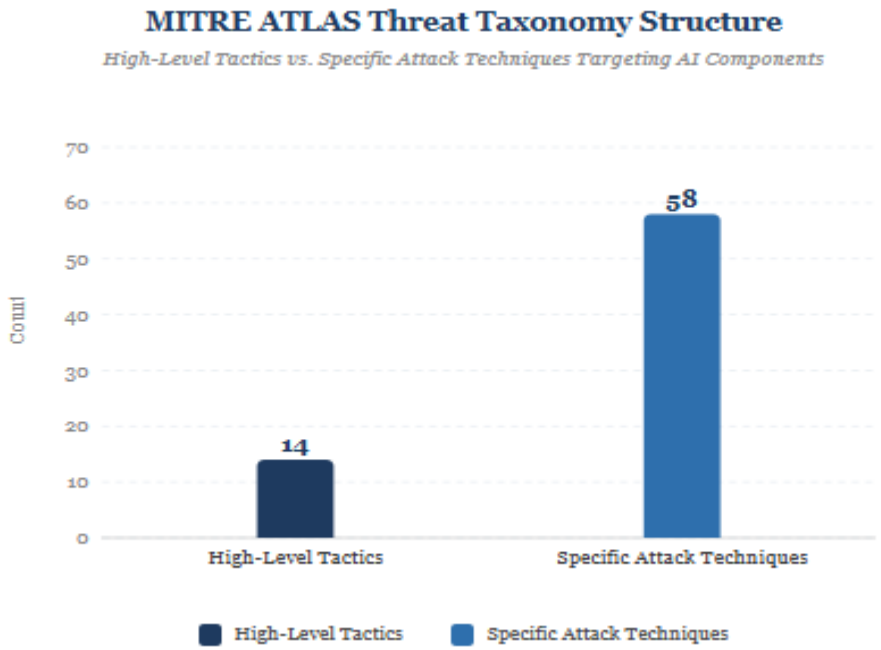

Fig 2: MITRE ATLAS Threat Taxonomy Structure [7, 8]

5. Evaluation Methodology and the Path to Defensible AI Governance

The only thing that makes a ZT-GenAI architecture effective is the empirical evidence of its policy choices at scale. Evaluation should be conducted on the security effectiveness, operational performance, and completeness of the audit, which should give the quantitative foundation to the review of the model risk management and regulatory defensibility. The four dimensions, which are evaluated using a structured evaluation methodology, are injection resilience, policy effect, audit completeness, and operational impact.

5.1 Injection Resilience Testing

Injection resilience determines how the ZT-GenAI system is resilient to prompt-based attacks in both direct and indirect injection vectors. This should be followed by a well-organized red-team testing with a varied set of adversarial test cases of instruction override attacks, role escalation attacks, data exfiltration with malformed tool arguments, and context poisoning attacks with adversarial content hidden in the retrieved document fragments. The heuristics of the AI Gateway and the parameter guardrails of the Tool Broker are assessed independently on this corpus, with the primary metrics of blocked-at-gateway rate and bypassed-to-generation rate. The explicit requirement to conduct adversarial testing within the scope of AI risk identification and categorization is present in the NIST AI RMF Govern 1.1 and Map 1.5 functions, where the findings of the red team must be recorded, evaluated concerning the residual risk, and mitigated before the production deployment [9]. Adversarial resilience testing must also be re-run whenever a new version of the model is deployed or a new policy ruleset is enacted since adversarial resilience is not a fixed property of a deployed system.

5.2 Policy Effectiveness and Least-Privilege Compliance

The testing of policy effectiveness confirms that the Policy Engine operates correctly in rejecting unauthorized actions to retrieve and use tools over a representative set of roles, stated purposes, data classification degrees, and anticipated decision configurations. This matrix should be enumerated by an automated test harness, and policy engine results should be compared against a ground-truth authorization specification, and the false-negative (unauthorized access granted) and false-positive (authorized access denied) rates should be measured. Least-privilege compliance is quantified by calculating the proportion of tokenized permission scopes that are not used during the authorized

Journal of Information Systems Engineering and Management
2026, 11(2s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

session, or, in other words, the unused-scope ratio is large, a large unused-scope ratio is a sign of systematic over-provisioning and is an indicator to count on automated policy refinement. This is achieved by continuous least-privilege monitoring, which develops a feedback loop between the operational behavior and the policy design and leads the system to the minimal viable permission surface in the long run.

5.3 Audit Completeness and Evidence Integrity

The completeness of the audit is the percentage of inference sessions where a complete evidence package is found within the Evidence Ledger. A full evidence package should contain prompt envelope hash, retrieval document identifications and classification labels, policy decision records containing obligation bindings, tool invocation records containing scoped token references, model version identifiers and prompt template version identifiers, generation output hash, and timestamps of each discrete processing step. Missing or incomplete package sessions are gaps in accountability that make regulation defensible in an examination.

Evidence integrity The property that ledger records cannot be rewritten is obtained by using cryptographic chaining or append-only storage and out-of-band verification. The version-controlled policy-as-code frameworks that store authorization rules in a source repository make it possible to reconstitute the state of policy at any point in time when any particular decision was made, which the retroactive auditability conditions of supervisory direction on model risk management [10]. This facility is necessary when regulatory reviews and internal audits are involved, where it can be necessary months after a given occurrence to rebuild the chain of authorization decisions to a particular inference path.

5.4 Operational Impact and Latency Budget Management

Measurable latency costs are incurred through security controls. The p95 inference latency with policy enforced, including timely envelope validation, ABAC-filtered retrieval, redaction gate processing and Tool Broker authorization, has to be measured against a no-policy control baseline to describe the operating overhead of ZT-GenAI controls. Latency budgets have to be set at each use-case tier and continuously monitored by automated telemetry, and alerting provided when the enforcement overhead surpasses the agreed limit. A cost per authorized inference and security effectiveness measurement will help to make sure that the ZT-GenAI controls are sustainable to operate at the production scale and security investments are reasonable in relation to the risk reduction provided.

Evaluation Dimension	What is Tested	Key Metrics / Evidence
Injection Resilience	Resistance to direct and indirect prompt attacks (override, escalation, exfiltration, poisoning)	Blocked-at-gateway rate, bypass-to-generation rate, red-team residual risk logs
Policy Effectiveness	Policy Engine correctness and least-privilege enforcement across roles, purpose, data class	False-positive rate, false-negative rate, unused-scope ratio
Audit & Operational Impact	Evidence package completeness + integrity + latency overhead of controls	% complete evidence sessions, tamper-proof ledger proof, p95 latency vs baseline, cost per authorized inference

Table 2: ZT-GenAI Evaluation Methodology for Defensible AI Governance [9, 10]

Conclusion

Controlled deployment of generative AI requires a radical shift from the perimeter-based trust models to identity-based policy enforcement at all levels of the chain. The ZT-GenAI architecture below confirms the idea that prompts, retrieval queries, and agentic tool invocations are not only

computational events but also discrete authorization decisions, each of which must have verified identity context, purpose-binding constraints, and least-privilege data scopes enforced by a unified Policy Engine, and not point-in-time controls. Immediate Envelope integrity, ABAC-controlled retrieval, redaction-gate content filtering, and Tool Broker intermediation are all techniques that minimize the blast radius of adversarial prompt injection while maintaining the operational flexibility that can be useful in complex, data-intensive processes using generative AI. Supervisory approval processes take the principles of Zero Trust up to the action layer, where high-impact agentic operations are confronted with a second authorization barrier that does not depend on model output. Policy-as-code-based evidence ledger offers the retroactive auditability required by the model risk governance and regulation. The presented architecture blueprint is a repeatable, justifiable framework to apply the capabilities of generative AI in sensitive operational scenarios, one in which authorization rigor and auditability are not viewed as limitations on capability but requirements to enable trustful, production-grade deployment.

References

[1] Scott Rose et al., "Zero Trust Architecture," NIST, 2020. [Online]. Available: https://nvlpubs.nist.gov/nistpubs/specialpublications/NIST.SP.800-207.pdf
[2] Cybersecurity and Infrastructure Security Agency, "Zero Trust Maturity Model," 2023. [Online]. Available: https://www.cisa.gov/sites/default/files/2023-04/CISA_Zero_Trust_Maturity_Model_Version_2_508c.pdf
[3] NIST, "Artificial Intelligence Risk Management Framework (AI RMF 1.0)," 2023. [Online]. Available: https://nvlpubs.nist.gov/nistpubs/ai/nist.ai.100-1.pdf
[4] Mohammad Fasha et al., "Mitigating the OWASP Top 10 For Large Language Models Applications Using Intelligent Agents." [Online]. Available: https://www.arxiv.org/pdf/2601.18105
[5] Vincent C. Hu et al., "Guide to Attribute Based Access Control (ABAC) Definition and Considerations," NIST, 2014. [Online]. Available: https://nvlpubs.nist.gov/nistpubs/specialpublications/nist.sp.800-162.pdf
[6] NIST, "Security and Privacy Controls for Information Systems and Organizations," 2020. [Online]. Available: https://csrc.nist.gov/pubs/sp/800/53/r5/upd1/final
[7] Felix Viktor Jedrzejewski et al., "ThreMoLIA: Threat Modeling of Large Language Model-Integrated Applications," ACM, 2025. [Online]. Available: https://dl.acm.org/doi/pdf/10.1145/3756681.3757083
[8] Fábio Perez and Ian Ribeiro, "Ignore Previous Prompt: Attack Techniques For Language Models," arXiv:2211.09527v1, 2022. [Online]. Available: https://arxiv.org/pdf/2211.09527
[9] Board of Governors of the Federal Reserve System Office of the Comptroller of the Currency, "Supervisory Guidance On Model Risk Management," 2011. [Online]. Available: https://www.federalreserve.gov/supervisionreg/srletters/sr1107a1.pdf
[10] Open Policy Agent, "Open Policy Agent (OPA)". [Online]. Available: https://www.openpolicyagent.org/docs