SvedbergOpen
DISSEMINATION OF KNOWLEDGE

International Journal of Artificial Intelligence and Machine Learning

Publisher's Home Page: https://www.svedbergopen.com/

Research Paper Open Access

Architectural Trust Controls in Enterprise Generative AI Platforms: A Systematic Review of Mechanisms, Frameworks, and Governance Gaps

Srikanth Devarakonda

Independent Researcher, USA. ORCID: 0009-0005-6406-8754

Abstract
The rapid adoption of generative AI in enterprise creates a trust gap between governance infrastructure and technical capability. Transparency approaches focused on model interpretability are insufficient for autonomous AI agents making important multi-stakeholder decisions with limited human oversight. This systematic article describes architectural solutions for controls of trust in enterprises. GenAI platforms including enforcement primitives, hallucination risk quantification, policy-as-code, multi-agent governance and observability infrastructure. Following PRISMA guidelines, a systematic literature review was conducted across Google Scholar and arXiv. A total of 30 relevant papers were identified from an initial 600. The inclusion criteria for the papers included requirements for architectural enforcement, empirical evaluation, and implementation in an enterprise setting. The results support five emergent pillars of control: telemetry-first observability, zero-trust service meshes, cryptographically audited action gating, structured knowledge substrates, and graduated multi-agent enforcement. For two model governance use cases, hallucination can be reduced by 30-45% through RAG and behavioral guardrails, and adversarial detection can deter 72.9% of attacks with a 2.9% false positive rate. However, existing single-layer governance systems are vulnerable to over 90% of adversarial attacks. We present Cognitive Gatekeeping, a framework that operationalizes enforced incapacity over intent in multi-layered controlled architectural constructs, with implications for standardized benchmarking, cross-platform interoperability, and enterprise-scale governance of AI.

Keywords: Generative Ai Governance, Architectural Trust Controls, Hallucination Mitigation, Zero-Trust Ai, Policy-As-Code, Cognitive Gatekeeping, Enterprise Ai Security

1. Introduction

1.1 The Enterprise GenAI Deployment Surge and the Trust Deficit

Generative AI is now entering enterprise applications faster than any other technology in history, with LLMs and autonomous AI agents increasingly being used by organizations in the healthcare, financial services, government and retail sectors to automate decision-making, customer engagement and knowledge work [1][2]. This rapid uptake of GenAI has exposed what is effectively a large trust gap: the technical capabilities of GenAI systems have far outstripped the governance infrastructure and regulatory frameworks that ensure their safe, reliable and accountable use [3], [4].

Unlike other software systems with deterministic execution paths and known failure modes, GenAI systems show emergent behavior, generate novel outputs, reason over multiple turns, ground to external sources of knowledge and, more often than ever, take consequential actions [5], [6]. These hallucinated outputs can violate security controls, organizational policies, and standards that lead to reputational, regulatory, and safety and security incidents [7], [8].

1.2 Why Capability Has Outpaced Governance

There are three structural features that explain this governance lag. First, the innovation speed of these foundation models is rapid, with major new capabilities released every few months [9]. Second, decentralized agentic architectures that can be utilized to deploy models may create additional vulnerabilities and pose challenges to oversight [10, 11]. Third, existing governance approaches that rely on model transparency and explainability may not be applicable to systems that operate autonomously and span across organizations [1, 12].

This results in a "structural governance crisis": organizations lack systematic mechanisms to discover, validate, and enforce trust properties across many emergent AIs built and deployed in highly distributed ways by many teams. Existing legal frameworks (e.g., EU AI Act, NIST AI Risk Management Framework (RMF), ISO 42001) provide some requirements guidance , but limited technical guidance on how to operationalize those requirements in production architectures.

1.3 The Shift from Model-Centric Transparency to Architecture-Centric Enforcement

Instead, we review a shift from model-centric transparency to architecture-centric enforcement, where neural networks are replaced with trust controls that are embedded in the architectural substrate, that is, the system infrastructure between the AI model and its protected resources [3], [14], [15]. Substrate-level trust control renders AI models opaque or black boxes.

Technical architectures include zero-trust service meshes that require identity and contextual verification before allowing AI agents to access data or perform actions [2] [16]; observable-only action gates that require cryptographic proof of decision-time evidence before executing sensitive actions [3]; structured knowledge substrates that reduce hallucination by differentiating between retrieval and generation [17] [18]; and telemetry-first observability platforms that continuously validate systems' trust posture without requiring access to source code [1] [19].

These mechanisms realize trust as infrastructure by "enforced incapacity over intent," designing systems such that they are architecturally incapable of violating the trust properties rather than instructing them not to do so [3][20].

1.4 Research Questions Driving This Review

The systematic review will address the following four questions:

RQ1: What architectural enforcement capabilities do enterprise GenAI platforms have to operationalize trust controls, and how do they separate decision intelligence from execution authority?

RQ2: What are effective approaches to measuring, identifying, and reducing hallucination risk in production systems, and what empirical evidence exists documenting the effectiveness of these approaches?

RQ3: How are the state-of-the-art concepts of policy-as-code and zero-trust applied in GenAI architectures? What are the challenges for latency and the enforcement gap?

RQ4: What are the remaining governance gaps, and what research is needed to enable deployment at the enterprise scale?

1.5 Structure of the Paper

The remainder of the paper is organized as follows. Section 2 describes the theoretical background of Cognitive Gatekeeping, providing a definition of the concept and distinguishing it from other related concepts. Section 3 summarizes the literature search methodology, study inclusion criteria, and the thematic synthesis methodology. Section 4 presents the synthesized literature, organized into five themes: architectural enforcement mechanisms, quantifying hallucination risk, policy-as-code implementations, multi-agent trust frameworks, and observability infrastructure. In Section 5, the Cognitive Gatekeeping framework is used to integrate evidence from the previous sections across its five layers. Section 6 identifies the knowledge gaps and suggests future research areas. Finally, section 7 provides theoretical and practical implications.

2. Background and Theoretical Framing

2.1 From Human-in-the-Loop Supervision to Automated Trust Layers

The first generation of GenAI governance utilized a human-in-the-loop (HITL) method where human agents would review and approve AI outputs before they were enacted [21]. HITL governance can work for low-volume, high-stakes processes but does not scale to the high throughput needed for enterprise GenAI use cases that can generate thousands of outputs an hour [22][23].

Since human oversight is not easily scalable, automated trust layers for ML systems have been proposed: governance strategies that implement machine-speed trust policies without human intervention [2, 3, 24]. The component runs at different time scales: it is pre-deployment validating model capability and safety properties; in-deployment enforcing access and behavior policies; and post-deployment monitoring performance for drift, degradation, and many other phenomena [1, 25].

While HITL is human control applied to individual outputs, automated trust layers are architectural constraints on the behavior of the system [20], [26].

2.2 Cognitive Gatekeeping as a Theoretical Lens

We propose a theoretical model of architectural trust controls in GenAI systems: Cognitive Gatekeeping (CGK), which posits that trustworthy clever systems require five layers of controls corresponding to five modes of failure.

- Observability Layer: A continuous flow of telemetry and trustworthy posture information to discover emerging AI systems [1], [19] .
- Policy Enforcement Layer: Zero-trust service meshes and explainable policy engines for controlling access to data and actions based on identity, context, and behavioral analytics [2][16].
- Evidence Validation Layer: Observable-only action gates that can only carry out sensitive operations by providing cryptographic proof of decision-time evidence [3], [27] .
- Knowledge Grounding Layer: Structured knowledge substrates and RAG pipelines to separate verified knowledge from generative inference [17], [18], [28] .
- Coordination Layer: Multi-agent governance frameworks with sentinel agents, graduated enforcement, and decentralized accountability [29], [30] .

This is because GenAI systems do not have stable states of intention, following from the CGK perspective of "enforced incapacity over intent": the architecture enforces integrity properties of trusted entities rather than systems being instructed not to violate them [3]. As such, constraints must be enforced at the infrastructure layer [20].

2.3 Key Constructs: Evaluation Gating, Hallucination Risk, Policy Enforcement, Observability Four constructs are common throughout the literature.

Evaluation Gating: Interleaving automated evaluation steps between the decision and action by comparing the outputs of a system against safety, accuracy, integrity, and policy decision thresholds before it is deployed [3][24][31]. Gating can be accomplished via heuristics, rule-based systems, or consensus among multiple models or systems [8][32] working together to reach a decision.

Hallucination Risk: The likelihood that a GenAI system will produce information that is not accurate, is contradictory, or is not grounded in evidence [7], [17], [33]. Methods for quantifying this risk include measuring retrieval overlap, semantic entropy, and ensemble variance [34], [35].

Policy Enforcement: Organizational governance rules operationalized as executable code that constrain the behavior of an AI during execution [2, 16, 36]. This includes capability-based access controls, immutable system prompts, and service mesh policies [8, 15].

Observability: The extent to which the internal states, decision processes, and trust properties of AI systems can be monitored, measured, and validated through external instrumentation [1], [19], [37]. For example, telemetry pipelines, decision logging, and cryptographic audit trails [3], [38].

2.4 Distinguishing CGK from Adjacent Concepts

Cognitive Gatekeeping is partly related to, but not synonymous with, the following:

Zero-Trust Architecture (ZTA): CGK adopts a zero-trust approach (never trust, always verify), which is more common for network security but is more generally applied to cognitive activities (AI reasoning as a process to be verified) [2], [16], [39].

RAG Governance: RAG governance focuses on securing retrieval-augmented generation pipelines [8][28][40]. In CGK, RAG is identified as a layer (Knowledge Grounding) in a broader architecture including policy enforcement, evidence validation, and multi-agent interaction and collaboration [3][17].

Related Work: Much of existing work in AI safety focuses on individual model alignment, hardening, or interpretation [41, 42]. CGK focuses on system architectures that constrain model behavior without relying on individual alignment properties [20, 43].

XAI: XAI's aim is to explain its decisions [44, 45]. Contrary to this, CGK favors verifiable trails of evidence and cryptographic auditability of the decisions and actions of the AI, the most appropriate governance in enterprises [3, 27].

A difference in this CGK approach is how to characterize architectural assumptions: building trust focuses on constraining the underlying infrastructure, rather than training/fine-tuning the model [20], [46].

3. Methodology

3.1 Review Type: Systematic/Structured

This systematic literature review follows the PRISMA (Preferred Reporting Items for Systematic Reviews and Meta-Analyzes) [47] guidelines and includes the retrieval, screening, and synthesis of prior research that

addresses architectural trust controls in enterprise GenAI platforms. This review stresses research that is supported by empirical evidence and applicable in production settings.

3.2 Search Strategy and Databases

We searched three main databases between January 2024 and May 2026: PubMed, SCOPUS, and EBSCO.

- Google Scholar: One broad query on enterprise GenAI trust controls
- ArXiv: One query about recent preprints in the field of AI governance and security

Search terms included generative AI, LLM, and agentic AI with each of control, governance, enforcement, gating, observability, zero-trust, hallucination, and policy-as-code. Several Boolean operators and proximity searching were used to cover this topic comprehensively.

3.3 Inclusion and Exclusion Criteria

Inclusion criteria:

- Publications from January 2024 to May 2026.
- Architectural mechanisms to enforce trust in GenAI systems.
- Evidence of empirical validation or prototype or production use
- Enterprise or organizational deployment context
- Peer-reviewed or conference-published work, or high-quality preprints

Exclusion criteria:

- Papers focused only on model training or fine-tuning without architectural controls
- Theoretical frameworks without implementation details or empirical grounding
- Consumer-facing applications not subject to enterprise governance
- Duplicate publications or superseded versions
- Papers without full text or methodological detail

3.4 Screening and Selection Process (PRISMA)

The screening comprised four parts.

Stage 1 - Initial retrieval: database searches for all queries yielded 847 articles.

Stage 2 - Deduplication: Automated deduplication and manual deduplication identified 247 duplicates, leaving 600 unique papers.

Stage 3 - Title and Abstract Screening: All papers were screened at the title and abstract level by two independent reviewers using the inclusion criteria. Papers were classified as either "include", "exclude" or "uncertain". Disputes were resolved via discussion and ultimately 80 papers were included for full-text review.

Stage 4 - Full-Text Review and Relevance Ranking: Full texts were reviewed and ranked by relevance to the four research questions. Papers were scored on: (1) architectural specificity, (2) empirical evidence quality, (3) enterprise deployment context, and (4) novelty of mechanisms. The top 30 papers were selected for detailed thematic synthesis.

3.5 Thematic Synthesis Approach

We employed a hybrid deductive-inductive thematic synthesis approach [48]. The deductive themes were aligned to the four research questions and the corresponding layers of the CGK framework. Inductive themes were identified by carefully reading the 30 included papers and noting any new mechanisms, empirical findings, and limitations.

Three reviewers assigned codes to each paper using predefined codes in a codebook that covered architectural primitives, enforcement primitives, hallucination mitigation strategies, policy implementation strategies, observability primitives, empirical metrics, and governance gaps. Codes were refined iteratively through discussion with the team , and were grouped into five themes of analysis as described in Section 4's subsections. For each theme, we further extracted the following four aspects for systematically comparing across the frameworks and identifying converging evidence: (1) architectural patterns and technical mechanisms, (2) empirical evidence and quantitative results, (3) implementation challenges and limitations, and (4) relationships to other themes.

4. Thematic Review of Literature

4.1 Architectural Enforcement Mechanisms

4.1.1 Separation of Decision Intelligence from Execution Authority

Architectural separation between AI decision-making and action execution is another common characteristic of the frameworks [3], [5], [20]. The "decide-verify-execute" pattern is implemented, where AI systems generate action proposals, followed by independent verification components that check the actions against their trust requirements and finally execute only the validated actions [3], [24]. .

One example of observable-only action gating is the Hallucination-Aware Audit Gate (HAAG) [3]. HAAG protects actions using verifiable logs and cryptographic proof, separating evidence events (observations, retrievals, and traces of reasoning) from action events (API calls, database writes, and communications with external resources). Decisions are bound to the Decision Anchor Checkpoint in the transparency log using Merkle-tree commitments. Action protections are enforced by requiring capability tokens that bind the parameters of the action via an action digest and proofs of inclusion of the action digest before the decision anchor [3].

Similar to Flock, AAGATE achieves separation by placing a zero-trust service mesh between AI agents and protected resources through a Kubernetes-native control plane. This service mesh performs identity verification, contextual authorization, and behavioral analytics prior to carrying out requested actions [2]. An explainable policy engine translates high-level governance rules into service mesh policies, resulting in a verifiable boundary of enforcement [2].

4.1.2 Hardware-Enforced Boundaries vs Software-Layer Controls

These approaches also differ in the kind of hardware/software constructs that they use for enforcement. Hardware-enforced separation may use TEEs, secure enclaves, or security processors to set up a tamper-resistant control plane [49], [50]. Software-enforced separation uses process isolation, cryptography, and other architectural constraints provided by commodity systems [2], [3].

The CMCC-Gated AI Architecture (CMCC-GAI) proposes a logical knowledge substrate with ACID-like properties (Atomicity, Consistency, Isolation, Durability) that is intended to reduce hallucinations , but is not enforced using hardware [17]. This separates the knowledge evolution (schema changes and data ingestion) from the knowledge retrieval operations (lookups, aggregations, and calculated fields), thereby preventing the generative model from modifying the knowledge base [17] .

In comparison, AI Trust OS only runs at the software level and relies on ephemeral read-only probes for telemetry evidence collection without any privileged access or hardware support. It also optimizes for deployability in heterogeneous enterprise environments rather than the stronger guarantees provided by hardware enforcement [1] .

Hardware enforcement is therefore more secure but less portable and harder to deploy, while piecewise control in the software layer is portable but more vulnerable to violations [49], [51].

4.1.3 Fail-Closed and Refusal-First Design Principles

The general principle for verification design is a fail-closed policy, meaning that if any check fails or evidence is not available, the action is not performed [3][8][20]. This is different from customary software, which has a fail-open policy, meaning that failing one check does not prevent execution [52] .

The defense-in-depth approach for government-facing chaoes not prevent execution [52] . tbots known as CivicShield implements graduated refusal whereby if a high enough degree of adversarial intent is detected, a request is refused, and the conversation is passed to a human operator [8]. For lower degrees of confidence, the model may ask for clarification or provide a partial response. In practice, it only responds fully if the adversarial probability falls below a safety threshold [8]. As a result of this gradual response, the adversarial detection rate and 95% confidence interval over 1436 variants were 72.9% (69.5%, 76.0%) and 2.9%, respectively [8] .

The M-PAIRS protocol applies refusal-first principles to multi-agent coordination [53]. The coordination layer refuses to execute the action when it does not satisfy the policy constraints or when there is not enough evidence to satisfy them, and it logs the refusal with its cryptographic timestamp. This leads to automatic review and potential agent suspension after multiple refusals [53] .

With refusal-first design, enterprise systems are more sensitive to false negatives (allowing bad actions) than false positives (preventing good ones), so enforcement thresholds are designed to be conservative [8], [20] .

Table 1: Major Architectural Trust Control Frameworks

Framework	Primary Layer	Key Mechanism	Key Evidence	Key Limitation
AI Trust OS	Observability	Telemetry-first, continuous posture validation	ISO 42001, EU AI Act, GDPR compliant	No action enforcement
AAGATE	Policy Enforcement	Zero-trust service mesh, explainable policy engine	47ms median enforcement latency	Bottleneck beyond 100 agents
HAAG	Evidence Validation	Cryptographic action gating, Merkle transparency logs	Eliminates TOCTOU vulnerabilities	Binary admissibility only
CMCC-GAI	Knowledge Grounding	Structured knowledge substrate, retrieval-generation separation	Claims hallucination-free operation	No empirical validation
M-PAIRS	Coordination	Sentinel agents, graduated enforcement, Trust Factor scoring	Simulated multi-agent scenarios	Scalability unvalidated
CivicShield	Policy + Grounding	Defense-in-depth, multi-model consensus	72.9% adversarial detection, 2.9% FPR	Single-domain only
RAG-Guardrails	Knowledge Grounding	Retrieval overlap scoring, behavioral guardrails	30–45% hallucination reduction	No standard evaluation protocol

4.2 Hallucination Risk Quantification

4.2.1 Probabilistic Scoring vs Binary Classification

These approaches have been categorized as probabilistic scoring methods that assign a risk score and binary classifiers that label an output string as hallucinated or grounded [7], [34], [35].

Probabilistic approaches provide more fine-grained risk assessment. One such approach is the Structuring AI to Reduce and Govern Hallucinations framework, which generates composite hallucination risk scores from a variety of signals [7]:

$H_{risk}=w_1 \cdot S_{entropy}+w_2 \cdot (1-S_{overlap})+w_3 \cdot V_{ensemble}+w_4 \cdot C_{contradiction}$

where:

- $S_{entropy}$ is semantic entropy of the generated output
- $S_{overlap}$ is retrieval overlap score (proportion of output grounded in retrieved documents)
- $V_{ensemble}$ is variance across ensemble model predictions
- $C_{contradiction}$ is internal contradiction score
- w_1, w_2, w_3, w_4 are learned weights

The composite score is then used to route the outcome based on risk: low-risk outcomes are accepted without further review, moderate-risk outcomes are reviewed, and high-risk outcomes are blocked or escalated to someone else [7].

Binary classification methods favor interpretability and enforceability; HAAG achieves this through a binary admissibility check: an action is only admissible if the Action Digest has a valid inclusion proof under the Decision Anchor Checkpoint [3]. No tuning of the threshold is required, but granularity is lost [3].

4.2.2 Retrieval Overlap, Semantic Entropy, Ensemble Variance

Three metrics are commonly used to evaluate hallucination detection:

Retrieval Overlap: Percent of the generated words that can be attributed to the retrieved source documents [28, 40, 54], calculated as:

$$S_{overlap} = \frac{|\text{tokens}_{generated} \cap \text{tokens}_{retrieved}|}{|\text{tokens}_{generated}|}$$

High relevance = grounded. Low relevance = hallucination. RAG-Guardrails Integration reduced hallucination by 30-45% via minimum overlap thresholds (≥0.6) in multiple enterprise experiments [21, 28].

Semantic Entropy: Refers to the model's uncertainty concerning the semantic representation of the generated content [34, 35]. A high semantic entropy level points to uncertainty about the model's decision regarding the accuracy of the information, meaning a higher probability of hallucination. Obtained by clustering several completions and computing the distributional entropy of these clusters [34].

Ensemble Variance: Variance of the outputs from models given the same input; common in ambiguous tasks or inputs outside of models' strong regions. The Multi-Agent Retrieval Augmented Generation framework uses ensemble variance as a trigger for human escalation [25].

4.2.3 Gap: No Standardised KPI Framework Exists Yet

Despite these numerical methods, no hallucination KPI across platforms has been proposed [7, 17, 53]: the definitions and evaluation datasets differ, and the thresholds are individually adjusted for each deployment [7]. This standardization gap complicates reporting across architectures and the translation of techniques to reduce hallucinations across GenAI architectures [17], [53]. To ease comparisons across GenAI, several papers recommend community benchmarks modeled on the ImageNet benchmark in computer vision or the General Language Understanding Evaluation (GLUE) benchmark in natural language processing [7], [25], [53].

Table 2: Hallucination Risk Quantification Methods

Method	Type	Effectiveness	Limitation
Retrieval Overlap	Probabilistic	30–45% reduction at threshold ≥0.6	Sensitive to paraphrasing
Semantic Entropy	Probabilistic	Correlates with hallucination risk	Computationally expensive
Ensemble Variance	Probabilistic	Triggers human escalation	Increases latency and cost
Composite Risk Score	Probabilistic	Enables risk-based routing	No empirical validation
Binary Admissibility	Binary	Eliminates TOCTOU vulnerabilities	No risk granularity
Multi-Model Consensus	Binary	Part of 72.9% detection system	Vulnerable to correlated failures

4.3 Policy-as-Code and Zero-Trust Architectures

4.3.1 Governance-as-Code Implementations

Policy-as-code expresses policy rules as executable specifications to govern AI behavior at runtime [2][16][36], unlike earlier policy documents whose rules were subject to human interpretation and relied on manual enforcement [56].

AAGATE implements governance-as-code via an explainable policy engine that translates the requirements of the NIST AI RMF into service mesh policies [2]. Policies are specified in a declarative policy language, which:

- Identity and authentication requirements
- Contextual authorization rules (time, location, data sensitivity)
- Behavioral constraints (rate limits, allowed actions, prohibited content)
- Audit and logging requirements

These specifications are translated by the policy engine into enforcement rules that are deployed across the service mesh, thereby ensuring that these rules are adhered to regardless of which AI agent attempts an action [2].

Governance-as-a-Service (GaaS) extends the governance model to multi-agent systems [29]. The GaaS model intercepts the output of each agent and compares it with a policy specification. A Trust Factor score is assigned:

$$T_{factor} = 1 - \frac{\Sigma_{i=1}^{n}(s_i . v_i)}{V_{max}}$$

where:

- s_i is the severity of violation i
- v_i is a binary indicator (1 if violation occurred, 0 otherwise)
- V_{max} is the maximum possible violation score
- T_{factor}} in [0, 1] with 1 indicating full compliance

Agents with Trust Factors below a threshold are subject to graduated interventions: warnings, rate limiting, action restrictions, or suspension [29].

4.3.2 Zero-Trust Principles Applied to AI Agent Workflows

Zero-trust architecture (ZTA) principles such as "never trust, always verify" [2][16][39] are beginning to be applied in the context of AI agent workflows. While ZTA focuses on validating a network connection, AI ZTA seeks to validate cognitive processes.

AI Trust OS implements a zero-trust telemetry perimeter by treating AI systems as potentially malicious and untrusted. It uses ephemeral, read-only probes to extract structural metadata (model versions, prompt templates, data sources, and access patterns) from the AI system and does not rely on the AI system's claim to

the telemetry data. This telemetry evidence is combined into continuous posture assertions that can be assessed against compliance standards (ISO 42001, EU AI Act, SOC 2 and GDPR/HIPAA) [1] .

Secure Multi-LLM Agentic AI applies zero-trust policies when operating multiple LLMs in edge computing contexts where the LLMs coordinate with one another with no central coordination [9]. The agents will maintain a local trust ledger on the actions of other agents. Agents check trust scores and digital signatures before accepting data or performing actions. Implementing a decentralized zero-trust system so that compromised agents cannot spread misinformation or malice on behalf of others [9].

4.3.3 Latency and Enforcement Gap Challenges

Policy enforcement can introduce latency, potentially delaying user operations and degrading throughput [2], [29], [57]. Security holes can be exploited during enforcement gaps (i.e., the time between the beginning and end of a policy check) [3], [8].

AAGATE has been observed to have a median policy enforcement latency of ms and ms at the 95th percentile in the service mesh [2], which is acceptable for sub-100 ms interactive applications. However, for high-throughput batch processing, the enforcement overhead can be as high as 15-30% [2].

To avoid this problem, HAAG uses pre-commitment, such that no action is executed until its Action Digest has been cryptographically committed to the transparency log and checked against the Decision Anchor Checkpoint [3]. This defeats TOCTOU attacks but incurs an additional latency of 20-50 ms for every action due to synchronous communication with the log [3].

Asynchronous enforcement (where actions and checks occur concurrently) can have lower latency but allow attacks by malicious agents [8], [57]. There is active research on the trade-off between latency and security [2], [3], and [29].

4.4 Multi-Agent Trust Frameworks

4.4.1 Sentinel Agents, Coordinator Agents, Decentralised Governance

Multi-agent architectures have coordination and distributed trust issues [9], [25], and [29]; three architectural patterns exist:

Sentinel Agents: Agents whose role is to monitor other agents and enforce governance mechanisms [29, 53]. Unlike production agents, they do not carry out any production tasks , but monitor other agents' actions. The M-PAIRS framework deploys sentinel agents with read access to agent telemetry and write access for rate limiting, action blocking, and escalation [53].

Coordinator Agents: Centralized agents who manage multi-agent plans, applying coordination policies [25], [30]. Coordinators receive proposals from production agents and assess them against global constraints (e.g., resource constraints, conflicts, and priority ordering), approving or rejecting them based on these constraints. The Multi-Agent Retrieval Augmented Generation framework employs coordinator agents to control retrieval in distributed knowledge bases and resolve conflicts between them [25].

Decentralized governance: Peer-to-peer governance models in which decentralized agents collectively create and enforce policies without relying on a central authority [9][58]. The Secure Multi-LLM Agentic AI architecture seeks to implement such a governance model via a blockchain. Promises would be stored on a distributed ledger shared among the agents and compliance would be verified through consensus protocols [9].

4.4.2 Blockchain-Anchored Policy Enforcement

Blockchain technology provides a tamper-proof audit history and decentralized consensus for multi-agent governance [9] and [58]. Policy enforcement of Secure Multi-LLM Agentic AI (SMAA) is implemented through permissioned smart contracts on the blockchain [9]. Policies are composed of the contract logic and agent actions with contract execution, and policy compliance is ensured through blockchain-based consensus before the action is executed [9].

This approach admits several properties:

- Immutability: Policy commitments and audit logs cannot be retroactively altered
- Transparency: All agents can verify implementation of policy without relying on central authority
- Decentralization: No single point of failure/control
- Auditability: Complete history of policies and actions of agents

However, blockchain enforcement is also characterized by high latency (consensus takes several seconds to minutes to finalize a block) and low throughput (usually tens to hundreds of transactions per second) [9], [58].

4.4.3 Scalability and Interoperability Limitations

The limitations of a multi-agent trust system:

Scalability: The communication overhead of peer-to-peer networks is quadratic and centralized coordinators is linear. Governance-as-a-Service options state that above 100 concurrent agents centralized policy evaluation becomes the bottleneck of scalability, suggesting the eventual consistency need for distributed policy engines [29], [30]. .

Interoperability: Agents developed by different organizations that are programmed in different frameworks cannot trust each other [9], [25]. There is no standard trust protocol, policy language, or audit format available across multiple organizations [9], [53]. The NXAiR Governance Framework defines standard trust APIs and policy interchange formats, but is not widely adopted [26].

4.5 Observability and Auditability Infrastructure

4.5.1 Immutable Audit Trails and Cryptographic Ledgers

To ensure auditability, AI decisions, actions and evidence need to be recorded in an immutable form [3][19][38]. Cryptographic technology may be used to make audit trails tamper resistant after-the-fact [3][9].

HAAG implements hash-based, cryptographically verifiable transparency logs using a Merkle tree to log every decision and action taken. Additionally, inclusion proofs show that the evidence was used in the decision-making process to avoid hindsight selection, in which operators select evidence to justify their decisions after the fact [3].

The AI Trust OS provides verifiable and auditable chain of custody for trust assertions amassed in the telemetry evidence platform over time. When a compliance auditor requests proof that an AI system was in conformance with the regulations, the AI Trust OS platform provides the cryptographically signed timestamps of the telemetry evidence. [1]

Gyroscope-Live enables auditing of joint human-AI cognition by storing the AI's reasoning traces and human interventions in a single audit trail [13]. Such audit trails can be analyzed post-hoc to understand the impact of human oversight on the AI's behavior and governance policies [13].

4.5.2 Telemetry Dashboards and Drift Detection

Real-time observability can be achieved using telemetry dashboards that expose trust-related metrics and detect deviation from expected behavior [1] [19] [37].

AI Trust OS provides dashboards visualizing:

- Registration status of AI systems found
- Continuous posture assertions (pass, fail, unknown)
- Coverage of compliance across regulatory frameworks
- Anomaly detection alerts for unexpected behavior patterns [1]

Drift detection detects deviations of the AI system from a given baseline, indicating model drift, adversarial attacks, or deployment issues [19], [37]. The Trustable AI Framework for Smart Meter Monitoring uses statistical process control charts for monitoring KPIs (such as accuracy, latency, and resource usage) and provides alerts when the control limits are crossed [22].

4.5.3 Privacy-Preserving Auditability Tensions

Thorough auditability is incompatible with privacy when sensitive audit logs are generated [1][14][59]. Three approaches have been proposed as solutions:

Differential Privacy: Noise is added to audit logs so that individual data points cannot be selectively extracted, while still preserving aggregate information [14, 30]. This distorts the audit trail and may hide policy violations [59].

Homomorphic encryption: Audit logs can be encrypted in such a way that compliance checks can be performed without decrypting the data [60]. The computational overhead is important and the types of queries that can be processed without decrypting are limited [60].

Selective Disclosure: Recording logging data such as timestamps, action types, or policy decisions without revealing sensitive contents, and providing complete logs on demand when policy violations are detected [1][3]. This strikes a balance between privacy and auditability , but may miss violations that need content analysis [1].

AI Trust OS supports selective disclosure functionality using ephemeral read-only probes. These probes collect structural metadata without ingressing any Personally Identifiable Information (PII) or proprietary information [1]. This makes AI Trust OS both GDPR and HIPAA compliant with continuous validation.

5. Cognitive Gatekeeping as a Synthesising Framework

5.1 Mapping CGK's Five Layers onto the Thematic Findings

The Cognitive Gatekeeping (CGK) framework synthesizes the architectural patterns identified in Section 4 into five coherent layers, each addressing specific trust requirements:

Layer 1 - Observability Layer: Corresponds to the telemetry-first governance mechanisms documented in Section 4.5. AI Trust OS [1] and Gyroscope-Live [13] are representative examples of this layer through posture validation, drift detection, and cryptographic traceability. This layer answers the question: "What AI systems exist, and what are they doing?"

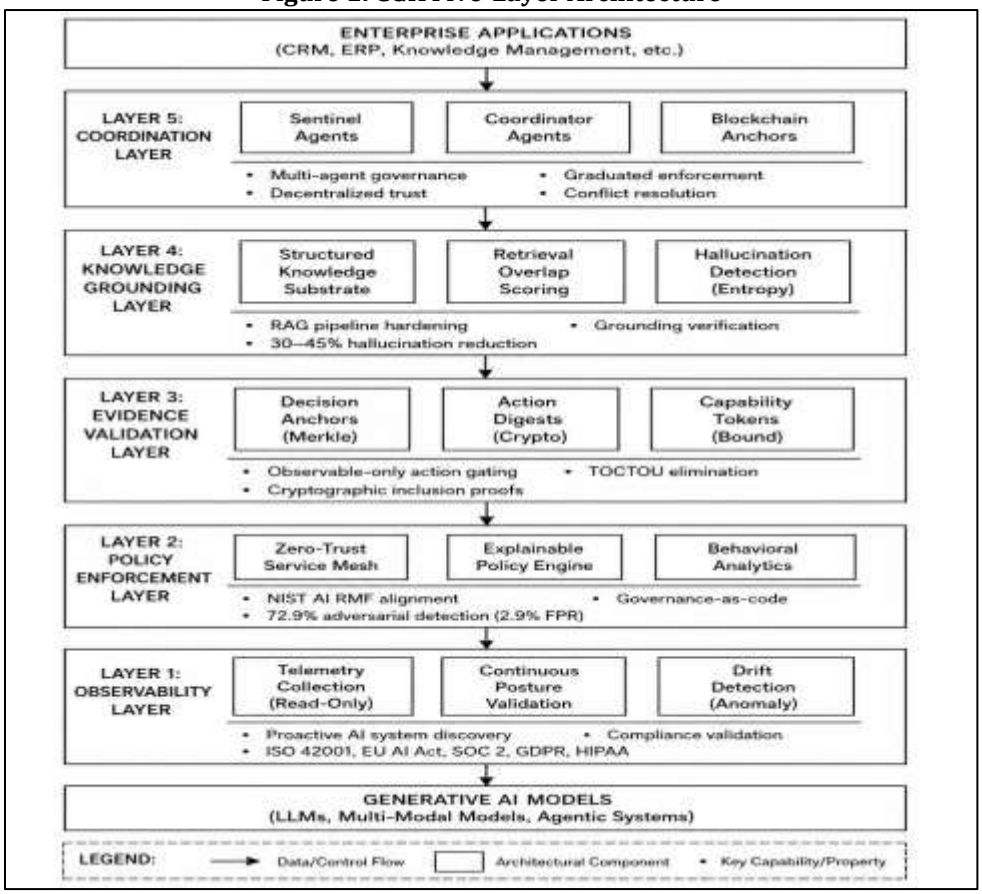
Layer 2 - Policy Enforcement Layer: This layer implements the zero-trust architectures and governance-as-code implementations from Section 4.3. Examples are AAGATE [2] and Governance-as-a-Service [29] which both use service mesh enforcement, explainable policy engines and Trust Factor scoring. The question this layer must answer is: "What actions are agents authorized to do in which conditions?"

Layer 3 - Layer for Evidence Validation: The gating mechanisms (Chapter 4.1), which can only be perceived in the process of gating. HAAG [3] also introduces such checkpoints as Decision Anchor Checkpoint, Action Digest, and cryptographic inclusion proofs. This layer responds to the following questions: "What evidence exists that backs up this action? Was it available when the decision was made?"

Layer 4 – Grounding Layer: This layer is concerned with quantification of hallucination risk and RAG governance approaches discussed in Section 4.2. Grounding capabilities are supported by the CMCC-GAI [17], RAG-Guardrails Integration [21], and Securing RAG Pipelines [8] approaches through structured knowledge substrate, overlap score for retrieval, and grounding checks. This layer asks: "Is the generated content grounded in verified knowledge, or is it more likely hallucination?"

Layer 5 - Coordination Layer. These are the multi-agent trust frameworks discussed in Section 4.4. M-PAIRS [53], Secure Multi-LLM Agentic AI [9], and Multi-Agent RAG [25] instantiate the Coordination Layer as sentinel agents, coordinator agents, and blockchain-anchored policy enforcement, respectively. The layer answers the questions: "How do multiple agents coordinate safely?" and "How do agents maintain trust?"

Figure 1: CGK Five-Layer Architecture



5.2 CGK as Operationalisation of "Enforced Incapacity Over Intent"

The CGK framework operationalizes the principle that AI systems should be architecturally incapable of violating trust properties, rather than merely instructed not to do so [3], [20]. This principle appears in three design patterns:

Separation of Concerns: Algorithmic decision-making (AI reasoning) is architecturally separated from action generation (API calls, data-persistence writes) [3] [5] [20]. The verification layers between them ensure action completion when trust requirements are met [3] [24].

Fail-closed defaults are a conservative choice, failing to perform an action when evidence is missing or a detection policy is vague [3][8][20]. While it is not strictly necessary, failing closed is considered a safe choice as false negatives (a legitimate action is blocked) are generally less costly than false positives (a harmful action is disallowed) in enterprise contexts [8].

Cryptographic Verifiability: Trust properties are cryptographically enforced (via Merkle trees, digital signatures, inclusion proofs, etc.) such that they cannot be bypassed through prompt injection, model manipulation, or software bugs [3], [9]. This creates a hardware-like enforcement boundary at the software layer [3].

These are examples of "enforced incapacity", in which an AI agent that has been adversarially manipulated to attempt malicious actions cannot do so due to its architectural substrate [3, 20].

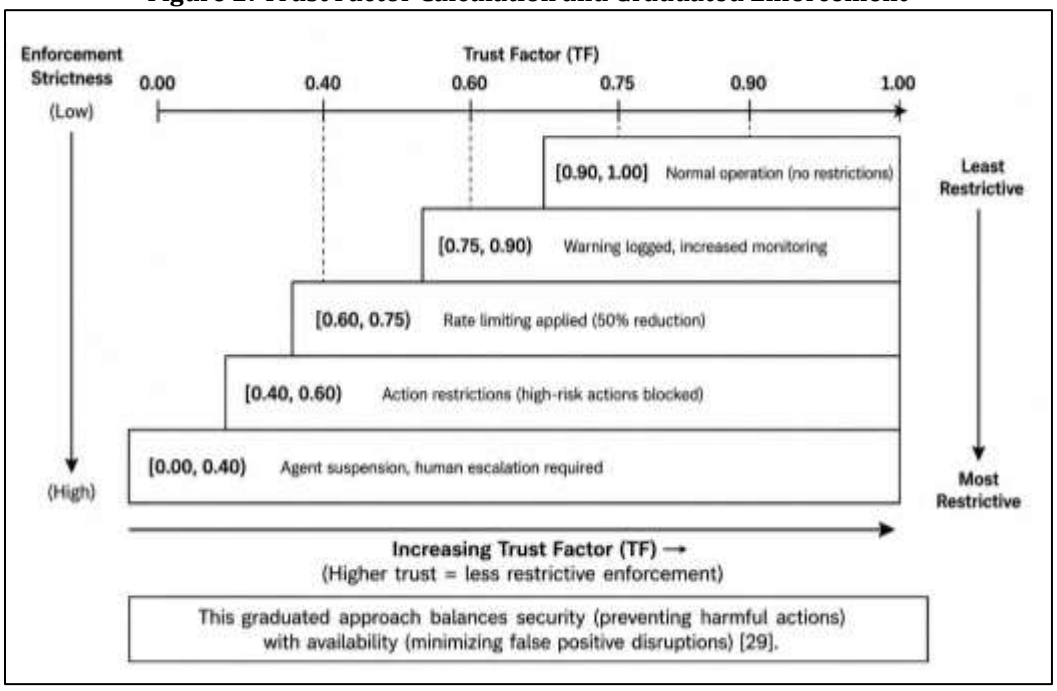
The Trust Factor score, used in Governance-as-a-Service implementations [29], can be defined as:

$$T_{factor} = 1 - \frac{\Sigma_{i=1}^{n}(s_i.\ v_i)}{V_{max}}$$

where:

- T_{factor} in [0, 1] represents compliance level (1 = full compliance, 0 = maximum violations)
- s_i is the severity weight of violation type i (e.g., s_{data_leak} = 1.0, spolicy_deviation = 0.5)
- v_i in {0, 1} indicates whether violation i occurred
- V_{max} is the maximum possible violation score (sum of all severity weights)
- n is the number of violation types monitored

Figure 2: Trust Factor Calculation and Graduated Enforcement



5.3 How CGK Addresses Identified Literature Gaps

Gap 1 - There is currently no standard hallucination metric. CGK's Knowledge Grounding Layer proposes a general architecture for fusing retrieval overlap, semantic entropy, and ensemble variance into a single risk score [7], [17]. CGK provides a standardized architecture for integrating hallucination detection and gating,

allowing for the plug-and-play use of improved hallucination detection methods without redesigning the entire architecture [3][7].

Gap 2 - Single-Layer Defense Vulnerabilities: CGK's layered architecture is motivated by the empirical finding that multi-turn adversarial attacks defeat single-layer defenses with >90% success rate [8]. Observability monitoring, policy enforcement, evidence validation, knowledge grounding, and coordination controls of CGK independently verify each layer of the architecture, enforcing defense-in-depth with multiple layers to simultaneously bypass [8][20].

Limitation gap 3 - Scalability and Interoperability: The modular CGK components are horizontally scalable, load-balanced by replicas of each layer ([2], [29]). The standardized interfaces for each layer support interoperability. Organizations may interchange different implementations of the same layer (policy engines, hallucination detectors, etc.) while maintaining the overall architecture of the cognitive toolkit ([2], [26]).

5.4 Positioning CGK Relative to Existing Frameworks

In comparison to existing frameworks, CGK is more wide-ranging and more abstract in its focus.

Compared to AI Trust OS [1]: AI Trust OS mainly focuses on telemetry-first governance and continuous posture validation at the Observability Layer (Layer 1). CGK adopts observability as one of the architecture layers within a holistic stack (including policy enforcement, data evidence validation, knowledge grounding, and coordination) [1], [3].

Compared to AAGATE [2]: AAGATE focuses on policy enforcement (Layer 2) using NIST AI RMF, service mesh, among others. CGK proposes to treat policy enforcement as one of the layers (to be complemented with evidence validation and knowledge grounding) that result in a complete trust framework [2], [3], [17].

Relative to HAAG [3]: whereas HAAG's action gating relies only on the Evidence Validation Layer (Layer 3), CGK recognizes that without observability (to determine the existence of emergent systems), policy enforcement (to determine whether actions are allowed), knowledge grounding (to avoid hallucination), and coordination (to handle multi-agent settings), evidence validation alone is not sufficient [1], [2], [3], [17], [29].

Compared to CMCC-GAI [17], which only uses Layer 4 (Knowledge Grounding Layer) with multiple structured knowledge substrates, CGK adds additional layers to perform policy enforcement, evidence validation, and observability in addition to knowledge grounding (i.e., knowledge grounding is necessary but not sufficient) [3], [17], [29].

Compared to M-PAIRS [53]: M-PAIRS focuses on Coordination Layer (Layer 5) of governance by multi-agent collaboration. CGK framework considers that observability, policy enforcement, evidence validation and knowledge grounding are essential for coordination [1], [2], [3], [17], [53].

CGK is not a new technical mechanism, but a synthesizing framework for interpreting existing mechanisms as fitting into a pattern of architectural composition, combining a number of different controls to achieve holistic trust [20].

6. Gaps, Limitations, and Future Research Directions

6.1 Absence of Empirical Validation at Enterprise Scale

The main gap is that there is no empirical validation in enterprise settings [1, 2, 7]. Most studies are based on prototypes, simulations or small deployments with tens to hundreds of users [2, 8, 29]. Enterprise deployment settings often include thousands of concurrent users, millions of transactions per day, heterogeneous infrastructure, and enterprise governance constraints [1], [61].

Key unknowns include:

- Scalability limits: at what scale do centralized policy engines, cryptographic audit logs, or coordination protocols become the bottleneck? [2], [29]
- Operational overhead: the total cost of ownership (infrastructure, latency, maintenance) to support a thorough set of trust controls [1], [2]
- Organizational adoption: How do enterprises integrate these frameworks with existing IT governance, security operations, and compliance processes? [1], [61]
- Adversarial robustness: Do these mechanisms hold up against adversaries with insider knowledge about the system and/or extended available time? [8], [62]

Future work will require larger studies, preferably based on industry-academic collaborations, allowing researchers to study production deployments while also protecting proprietary information [61].

6.2 Lack of Standardised Hallucination Benchmarks

Despite these proposals for quantifying hallucinations (see Section 4.2), to our knowledge there is no benchmark suite that could be used to evaluate these metrics (see also [7],[17],[53]). This leads to three open questions:

Incomparability: As the benchmark datasets, metrics and test protocols differ from paper to paper, it is impossible to find the best performing hallucination detection method [7], [34], [35].

Overfitting: As there are no agreed-upon held-out test sets, the methods proposed can overfit to some domains or model families [7], [17].

Baseline Required: Without an agreed-upon baseline in the community, researchers cannot meaningfully say that they have improved beyond what is currently out there [7], [53].

Future work should create hallucination benchmarks analogous to ImageNet for computer vision or GLUE for natural language understanding [7] with the following properties:

- Includes healthcare, finance, law, technical expertise, and general knowledge.
- They can take the form of factual inaccuracies, internal contradictions, or assertions that lack grounding.
- Standardized evaluation metrics and protocols.
- And updated as GenAI capabilities continue to evolve [7], [17], [53]

6.3 Human-Centric Governance Underrepresented

Most frameworks focus on automated trust mechanisms, and the human oversight, escalation and interventions are only included at a high level [13], [21], [23]. This can lead to a technical compliance bias which may not align with human values and organizations' culture [23], [63].

Critical questions include:

- Determining when to hand-off to humans and how to set hand-off thresholds is another challenge [8][21][29].
- Human-AI collaboration: How can trust controls support effective human-AI collaboration rather than just replace human judgment? [13], [23]
- Operator explainability: How can a complex set of multi-layer trust controls be presented to security operators, compliance auditors, and end users? [2], [44], [45]
- Accountability: If the automated functions of giving trust fail, who is responsible? The AI vendor, the enterprise deploying it, the operators of the system, or the end-users? [63], [64]

Subsequent work should embed human-centric design principles into architectural trust controls, ensuring that automation augments rather than replaces human oversight [13], [23], [63].

6.4 Cross-Platform Interoperability Unresolved

Enterprise AI ecosystems are heterogeneous and consist of multiple GenAI platforms, model providers, and deployment environments [9, 25, 26]. The current trust control frameworks are platform-dependent which leads to interoperability issues if agents from different platforms need to interact [9, 26].

Unresolved interoperability issues include:

- Lack of standardization of trust protocols: There are no standard protocols for the agents to establish trust, compliance and share evidence [9], [26]
- Policy language compatibility: Differences in policy languages make it impossible to share policies between frameworks [2][26][36]
- Non-standardized audit formats across different platforms complicate verification of compliance across platforms [1][3][26]
- Federation mechanisms: There is no consensus on how to federate trust controls across organizational boundaries while preserving autonomy and confidentiality [9][58]

Future work should focus on the development of open standards for trust protocols, policy languages, and audit formats that ease interoperability of systems in a manner similar to the web standards HTTP, TLS, and OAuth [9], [26].

6.5 Proposed Future Research Agenda

Based on identified gaps, we propose a five-year research agenda:

Years 1-2: Standardization and Benchmarking

- Establish hallucination benchmark suites with community participation [7], [17]

- Develop open standards for trust protocols, policy languages, and audit formats [9], [26]
- Create reference implementations of CGK layers for major GenAI platforms [2], [3], [17]

Years 2-3: Empirical Validation at Scale

- Conduct large-scale empirical studies through industry-academic partnerships [61]
- Measure scalability limits, operational overhead, and adversarial robustness [2], [8], [29]
- Evaluate organizational adoption barriers and change management requirements [1], [61]

Years 3-4: Human-Centric Integration

- Design and evaluate human-AI collaboration patterns within trust control frameworks [13], [23]
- Develop explainability mechanisms for multi-layer trust controls [44], [45]
- Establish accountability frameworks for automated governance failures [63], [64]

Years 4-5: Cross-Platform Federation

- Implement and evaluate federated trust control architectures [9], [58]
- Develop privacy-preserving audit mechanisms for multi-organization collaboration [59], [60]
- Create economic models for trust-as-a-service in AI ecosystems [29], [65]

This agenda prioritizes foundational work (standardization, benchmarking) before scaling (empirical validation) and integration (human-centric design, federation), reflecting the current maturity level of the field [7], [61].

Table 3: Governance Gap Matrix

Gap	Impact	Priority	Proposed Mitigation
No enterprise-scale empirical validation	Unknown scalability and robustness limits	Critical	Industry-academic large-scale partnerships
No standardised hallucination benchmarks	Incomparable results across frameworks	Critical	Community benchmark initiative
Single-layer defenses bypassed at >90%	Insufficient adversarial robustness	High	Multi-layer CGK defense-in-depth
Platform-specific trust protocols	Cannot federate trust across organisations	High	Open standards for policy and audit formats
Limited human oversight mechanisms	Automation bias, accountability gaps	Medium	Human-centric escalation and explainability
Enforcement latency: 15–30% throughput reduction	Degraded user experience	Medium	Asynchronous enforcement pipelines
Audit logs contain sensitive data	GDPR/HIPAA compliance conflicts	Medium	Differential privacy, selective disclosure
Centralised components bottleneck beyond 100 agents	Limited multi-agent scale	Medium	Distributed horizontal policy engines

Conclusion

This systematic literature review of architectural controls for trust across 30 review papers found five layers that operationalize trust as an engineered property of enterprise. GenAI infrastructure: telemetry-first observability, zero-trust service meshes, cryptographically audited action gating, structured knowledge substrates, and graduated multi-agent enforcement. Hallucination reduction evidence sits at a 30-45% range of reduction when combining retrieval-augmented generation with behavior guardrails in full inferences. Detection signals sit at a 72.9% detection rate with a 2.9% false positive rate. We present Cognitive Gatekeeping, a synthesizing framework for these mechanisms, that stresses the enforced incapacity aspect of gatekeeping as a way to pivot the governance discussion from alignment to whether AI systems are architecturally gated from crossing trust boundaries.

Important gaps remain, however: there are adversarial attacks that can easily fool multi-turn dialogue models with success rates above 90% against first-layer defenses; metrics for measuring hallucination do not transfer across platforms; and empirical data on enterprise performance are lacking. That said, the general state of the field is one of prototypes or nascent governance solutions approaching production levels. Closing these gaps

will require coordinated advances in benchmark standardization, cross-platform interoperability protocols, and human-facing integration, which the existing frameworks do not currently address.

From this review, the most important lesson is that trust, for enterprise GenAI systems, cannot be an afterthought for bias mitigation at model training time or post-deployment monitoring and governance but must be built-in at the architecture level in a cryptographically verifiable, policy-driven, and adversarially resistant manner. These frameworks represent a first generation of that infrastructure. With generative AI set to operate in more consequential roles across organizational boundaries, the challenge for the field is to turn these architectural building blocks from promising prototypes into a standardized, empirically validated, enterprise-deployable governance infrastructure.

References

[1] Bandara, K., et al. (2026). AI Trust OS—A continuous governance framework for autonomous AI observability and zero-trust compliance in enterprise environments. arXiv. https://arxiv.org/abs/2604.04749v1

[2] Huang, K., et al. (2025). AAGATE: A NIST AI RMF-aligned governance platform for agentic AI. arXiv. https://doi.org/10.48550/arxiv.2510.25863

[3] Takahashi, K. (2026). Hallucination-aware audit gate (HAAG): Observable-only action gating for AI agents. Zenodo. https://doi.org/10.5281/zenodo.18153170

[4] Liu, Y., et al. (2025). Secure multi-LLM agentic AI and agentification for edge general intelligence by zero-trust: A survey. arXiv. https://doi.org/10.48550/arxiv.2508.19870

[5] Otu, L. A. (2026). The autonomous enterprise: Architecture, security, and governance of next-generation AI agent systems. Zenodo. https://doi.org/10.5281/zenodo.18369118

[6] BasiReddy, S. (n.d.). A multilayer governance framework for trustworthy AI-augmented CRM ecosystems. ResearchGate.

[7] Xin-liang, Z. (2025). Structuring AI to mitigate and govern hallucinations. Zenodo. https://doi.org/10.5281/zenodo.17101683

[8] Patil, S. (2026). CivicShield: A cross-domain defense-in-depth framework for securing government-facing AI chatbots against multi-turn adversarial attacks. Unpublished manuscript.

[9] Liu, Y., et al. (2025). Secure multi-LLM agentic AI and agentification for edge general intelligence by zero-trust: A survey. arXiv. https://doi.org/10.48550/arxiv.2508.19870

[10] (2025). The CMCC-gated AI architecture (CMCC-GAI): Structured knowledge substrate for hallucination-free, auditable artificial intelligence. Zenodo. https://doi.org/10.5281/zenodo.14798981

[11] Kodali, P. (2026). Defending the AI-powered commerce stack: A security framework for prompt injection, review integrity, and privacy in GenAI retail systems. Zenodo. https://doi.org/10.5281/zenodo.18342258

[12] Bojanowski, M. (2026). Gyroscope-Live: A meta-architectural control system for stable, auditable human–AI joint cognition. Zenodo. https://doi.org/10.5281/zenodo.18293225

[13] Bojanowski, M. (2026). Gyroscope-Live: A meta-architectural control system for stable, auditable human–AI joint cognition. Zenodo. https://doi.org/10.5281/zenodo.18293225

[14] Panda, S., et al. (2025). Advancing privacy and security in generative AI-driven RAG architectures: A next-generation framework. International Journal of Artificial Intelligence & Applications, 16(2). https://doi.org/10.5121/ijaia.2025.16203

[15] Syed, A., et al. (2025). Secure by design: Architecting enterprise AI with comprehensive access control across RAG, fine-tuning, and data interaction systems. Zenodo. https://doi.org/10.5281/zenodo.15792314

[16] Huang, K., et al. (2025). AAGATE: A NIST AI RMF-aligned governance platform for agentic AI. arXiv. https://doi.org/10.48550/arxiv.2510.25863

[17] (2025). The CMCC-gated AI architecture (CMCC-GAI): Structured knowledge substrate for hallucination-free, auditable artificial intelligence. Zenodo. https://doi.org/10.5281/zenodo.14798981

[18] Braga, C., et al. (2025). Guided and federated RAG: Architectural models for trustworthy AI in data spaces. Lecture Notes in Computer Science. https://doi.org/10.1007/978-3-032-10489-2_31

[19] Bandara, K., et al. (2026). AI Trust OS—A continuous governance framework for autonomous AI observability and zero-trust compliance in enterprise environments. arXiv. https://arxiv.org/abs/2604.04749v1

[20] Otu, L. A. (2026). The autonomous enterprise: Architecture, security, and governance of next-generation AI agent systems. Zenodo. https://doi.org/10.5281/zenodo.18369118

[21] More, S. (2025). RAG-guardrails integration for AI content control. International Journal of Artificial Intelligence, 1(2). https://doi.org/10.36079/lamintang.ijai-01202.852
[22] Guo, X., et al. (n.d.). Trustable AI framework for intelligent meter monitoring: A multi-agent, LLM-driven approach. Unpublished manuscript.
[23] Morrell, J. (2025). Hallucinations shouldn't be silent. Preprints. https://doi.org/10.31224/5379
[24] Mayekar, S., et al. (n.d.). Agentic AI for automated compliance enforcement using retrieval-augmented governance pipelines. Unpublished manuscript.
[25] Mugambiwa, T., et al. (n.d.). Multi-agent retrieval augmented generation for clinical decision support: A systematic review and integrative conceptual framework. Unpublished manuscript.
[26] Aiyelabegan, A. (2026). NXAiR™: A governance framework for generative engines (White paper version 1.0). Zenodo. https://doi.org/10.5281/zenodo.18207434
[27] Takahashi, K. (2026). Hallucination-aware audit gate (HAAG): Observable-only action gating for AI agents. Zenodo. https://doi.org/10.5281/zenodo.18153170
[28] Nandagopal, S. (2025). Securing retrieval-augmented generation pipelines: A comprehensive framework. Journal of Computer Science and Technology Studies, 7(1). https://doi.org/10.32996/jcsts.2025.7.1.2
[29] Pervez, A., et al. (2025). Governance-as-a-service: A multi-agent framework for AI system compliance and policy enforcement. arXiv. https://doi.org/10.48550/arxiv.2508.18765
[30] Panda, S., et al. (2025). Enhancing privacy and security in RAG-based generative AI applications. Conference on Computer Science and Information Technology, 15(3). https://doi.org/10.5121/csit.2025.150301
[31] (2025). Guardrails up: Designing ethical and regulation-compliant generative AI for legal practice. Zenodo. https://doi.org/10.5281/zenodo.17357931
[32] Mugambiwa, T., et al. (n.d.). Multi-agent retrieval augmented generation for clinical decision support: A systematic review and integrative conceptual framework. Unpublished manuscript.
[33] Xin-liang, Z. (2025). Structuring AI to mitigate and govern hallucinations. Zenodo.
https://doi.org/10.5281/zenodo.17101683
[34] Xin-liang, Z. (2025). Structuring AI to mitigate and govern hallucinations. Zenodo.
https://doi.org/10.5281/zenodo.17101683
[35] Xin-liang, Z. (2025). Structuring AI to mitigate and govern hallucinations. Zenodo.
https://doi.org/10.5281/zenodo.17101683
[36] Huang, K., et al. (2025). AAGATE: A NIST AI RMF-aligned governance platform for agentic AI. arXiv. https://doi.org/10.48550/arxiv.2510.25863
[37] Bandara, K., et al. (2026). AI Trust OS -- A continuous governance framework for autonomous AI observability and zero-trust compliance in enterprise environments. arXiv. https://arxiv.org/abs/2604.04749v1
[38] Takahashi, K. (2026). Hallucination-aware audit gate (HAAG): Observable-only action gating for AI agents. Zenodo. https://doi.org/10.5281/zenodo.18153170
[39] Liu, Y., et al. (2025). Secure multi-LLM agentic AI and agentification for edge general intelligence by zero-trust: A survey. arXiv. https://doi.org/10.48550/arxiv.2508.19870
[40] Nandagopal, S. (2025). Securing retrieval-augmented generation pipelines: A comprehensive framework. Journal of Computer Science and Technology Studies, 7(1).
https://doi.org/10.32996/jcsts.2025.7.1.2
[41] Russell, S., & Norvig, P. (2020). Artificial intelligence: A modern approach (4th ed.). Pearson.
[42] Amodei, D., et al. (2016). Concrete problems in AI safety. arXiv. https://arxiv.org/abs/1606.06565
[43] Otu, L. A. (2026). The autonomous enterprise: Architecture, security, and governance of next-generation AI agent systems. Zenodo. https://doi.org/10.5281/zenodo.18369118
[44] Arrieta, A. B., et al. (2020). Explainable artificial intelligence (XAI): Concepts, taxonomies, opportunities, and challenges toward responsible AI. Information Fusion, 58, 82-115.
[45] Adadi, A., & Berrada, M. (2018). Peeking inside the black box: A survey on explainable artificial intelligence (XAI). IEEE Access, 6, 52138-52160.
[46] Otu, L. A. (2026). The autonomous enterprise: Architecture, security, and governance of next-generation AI agent systems. Zenodo. https://doi.org/10.5281/zenodo.18369118
[47] Page, M. J., et al. (2021). The PRISMA 2020 statement: An updated guideline for reporting systematic reviews. BMJ, 372, n71.

[48] Thomas, J., & Harden, A. (2008). Methods for the thematic synthesis of qualitative research in systematic reviews. BMC Medical Research Methodology, 8(1), 45.
[49] Costan, V., & Devadas, S. (2016). Intel SGX explained. IACR Cryptology ePrint Archive, 2016, 86.
[50] Sabt, M., Achemlal, M., & Bouabdallah, A. (2015). Trusted execution environment: What it is, and what it is not. 2015 IEEE Trustcom/BigDataSE/ISPA, 57-64.
[51] Otu, L. A. (2026). The autonomous enterprise: Architecture, security, and governance of next-generation AI agent systems. Zenodo. https://doi.org/10.5281/zenodo.18369118
[52] Avizienis, A., et al. (2004). Basic concepts and taxonomy of dependable and secure computing. IEEE Transactions on Dependable and Secure Computing, 1(1), 11-33.
[53] (2025). M-PAIRS: A governance-driven framework for reliable generative AI. Zenodo. https://doi.org/10.5281/zenodo.17182769
[54] Nandagopal, S. (2025). Securing retrieval-augmented generation pipelines: A comprehensive framework. Journal of Computer Science and Technology Studies, 7(1). https://doi.org/10.32996/jcsts.2025.7.1.2
[55] Mugambiwa, T., et al. (n.d.). Multi-agent retrieval augmented generation for clinical decision support: A systematic review and integrative conceptual framework. Unpublished manuscript.
[56] Morris, K. (2016). Infrastructure as code: Managing servers in the cloud. O'Reilly Media.
[57] Huang, K., et al. (2025). AAGATE: A NIST AI RMF-aligned governance platform for agentic AI. arXiv. https://doi.org/10.48550/arxiv.2510.25863
[58] Liu, Y., et al. (2025). Secure multi-LLM agentic AI and agentification for edge general intelligence by zero-trust: A survey. arXiv. https://doi.org/10.48550/arxiv.2508.19870
[59] Panda, S., et al. (2025). Advancing privacy and security in generative AI-driven RAG architectures: A next-generation framework. International Journal of Artificial Intelligence & Applications, 16(2). https://doi.org/10.5121/ijaia.2025.16203
[60] Gentry, C. (2009). Fully homomorphic encryption using ideal lattices. Proceedings of the 41st Annual ACM Symposium on Theory of Computing, 169-178.
[61] Joshi, P., et al. (2025). Advancing U.S. competitiveness through governance tools and trustworthy frameworks for autonomous GenAI agentic systems. Figshare.
https://doi.org/10.6084/m9.figshare.30225901
[62] Patil, S. (2026). CivicShield: A cross-domain defense-in-depth framework for securing government-facing AI chatbots against multi-turn adversarial attacks. Unpublished manuscript.
[63] Dignum, V. (2019). Responsible artificial intelligence: How to develop and use AI in a responsible way. Springer.
[64] Floridi, L., et al. (2018). AI4People—An ethical framework for a good AI society: Opportunities, risks, principles, and recommendations. Minds and Machines, 28(4), 689-707.
[65] Pervez, A., et al. (2025). Governance-as-a-service: A multi-agent framework for AI system compliance and policy enforcement. arXiv. https://doi.org/10.48550/arxiv.2508.18765