International Journal of Emerging Research in Engineering and Technology
Pearl Blue Research Group | Volume 7 Issue 2 PP 222-230, 2026
ISSN: 3050-922X | https://doi.org/10.63282/3050-922X.IJERET-V7I2P127

Original Article

A Control Plane Architecture for Secure and Governable AI in Regulated Financial Systems

Tripatjeet Singh
Senior Cloud Engineer, Dallas-Fort Worth, USA.

Received On: 04/04/2026 ***Revised On: 03/05/2026*** ***Accepted On: 11/05/2026*** ***Published On: 17/05/2026***

Abstract *- Artificial Intelligence adoption within regulated financial institutions has outpaced the security frameworks designed to govern it. Existing controls were built around network perimeters, static role assignments, and API-layer enforcement and were never designed for the behavioral unpredictability of large language models (LLMs) and autonomous AI agents. This paper presents a Control Plane Architecture for Secure and Governable AI (CP-SGAI): a purpose-built governance architecture that treats AI inference as a first-class security event requiring identity attestation, policy-bounded prompt execution, and ephemeral privilege scoping, and structured observability. Grounded in field experience operating multi-account AWS environments at a large financial institution, the framework addresses practical gaps that purely theoretical governance models miss, including prompt injection at the enterprise boundary, cross-account data egress through model responses, and the absence of prompt-response lineage in existing SIEM and audit toolchains. Three original constructs are introduced: an AI Interaction Identity (AII) attestation model, an Ephemeral Prompt Security Context (EPSC) lifecycle, and an AI Observability Schema (AOS) aligned with financial regulatory audit requirements.*

Keywords *- Zero Trust Architecture, AI Governance, Llm Security, Prompt Injection, Ephemeral Access Control, Financial Services Security, AI Observability, Multi-Account Cloud, Aws, Regulatory Compliance.*

1. Introduction

Security architecture in banking evolves reactively; frameworks emerge in response to breach patterns, regulatory mandates, and the hard lessons learned when a control assumption proves wrong. AI adoption in financial services is following the same trajectory: deployment is outrunning governance, and the industry is beginning to discover that security models built for distributed application workloads do not translate to AI systems.

The mismatch is not superficial. Traditional enterprise security, Zero Trust included rests on assumptions that AI systems violate by design: inputs are structured and bounded, outputs can be validated against a schema, system behavior is deterministic given the same inputs, and access patterns are static enough to model. Large language models and autonomous AI agents break all four. They accept freeform natural language, produce outputs that may contain sensitive data drawn from training or retrieval context, behave differently given minor prompt variations, and chain calls across multiple downstream systems within a single session.

These are not edge cases, they are the functional properties of systems now being deployed to handle customer inquiries, credit analysis, fraud triage, and regulatory reporting at financial institutions worldwide. The security implications warrant a dedicated architectural response, not a bolt-on to existing API gateways or IAM policies.

This paper proposes the Control Plane Architecture for Secure and Governable AI (CP-SGAI): a governance layer purpose-built for AI workloads in regulated environments. The framework draws on direct operational experience managing AI deployments across 100+ AWS accounts within a large financial institution's AWS Organization, and specifically addresses the governance gaps that emerge at scale when AI moves from proof-of-concept into production infrastructure.

2. Problem Statement

2.1. Why Existing Controls Fall Short

The standard enterprise security toolkit with network segmentation, IAM roles, API gateways, SIEM pipelines was designed around deterministic systems. The mental model is clean: a subject requests a resource through a defined channel, and the control plane decides whether to allow or deny based on policy. This maps well to microservices, databases, and message queues. It maps poorly to AI inference.

Consider the attack surface introduced by a single prompt submitted to an enterprise LLM deployment. The prompt may contain an injection payload designed to override system instructions. It may cause the model to retrieve documents from a retrieval-augmented generation (RAG) corpus that the calling user should not access. The response may embed information that appears benign in isolation but constitutes a data leak when aggregated across sessions. None of these failure modes are detectable by an API gateway evaluating request headers and endpoint paths.

Table 1: Existing Controls vs. AI Threat Surface

Existing Control	Limitation Against AI Threats
Network perimeter / firewall	No visibility into prompt content; cannot detect semantic-layer attacks
IAM role-based access control	Static permissions to AI endpoints; does not scope per interaction or prompt context
API gateway / rate limiting	Prevents volumetric abuse; no semantic inspection of prompt or response
SIEM / log aggregation	Captures infrastructure events; no prompt-response pairs, model context, or AI decision rationale
Model risk management (MRM)	Governs model development lifecycle; not designed for runtime per-interaction enforcement

2.2. The Enterprise AI Threat Landscape

Five threat categories consistently emerge in enterprise AI deployments that existing controls address inadequately:

- Prompt injection: malicious inputs that override system instructions or manipulate model behavior across a session boundary
- Cross-context data leakage: model responses that surface information from RAG corpora, prior sessions, or system prompts the calling identity should not access
- Privilege escalation through AI agents: agentic systems that chain tool calls, accumulating effective permissions exceeding what any single IAM role would grant
- Audit gap: absence of prompt-response lineage in existing logging infrastructure, preventing forensic reconstruction of AI-assisted decisions
- Shadow AI usage: decentralized deployment of AI capabilities without registration, policy enforcement, or observability

The audit gap deserves particular attention. In April 2026, the Federal Reserve, OCC, and FDIC issued SR 26-2 [10], the revised interagency model risk management guidance that explicitly excludes generative AI and agentic AI from its scope, stating these technologies are 'novel and rapidly evolving' and directing institutions to apply their existing risk management practices while a separate AI-specific framework is developed. This regulatory carve-out is both an acknowledgment of the problem and the strongest possible signal that institutions cannot rely on SR 26-2 or its predecessor to govern runtime AI risk. The agencies have announced a forthcoming RFI on AI model risk management specifically to address this gap. In this interim period, institutions that deploy generative or agentic AI in credit, fraud, and customer-facing domains are operating without a regulatory framework governing runtime behavior; making the observability and enforcement constructs proposed in this paper not merely useful but architecturally necessary.

3. Related Work and Research Gap

The NIST Zero Trust Architecture (SP 800-207) [1] establishes the foundational principle that no entity should be inherently trusted based on network location, defining a policy decision point (PDP) and policy enforcement point (PEP) architecture that this paper extends into the AI inference domain. Critically, NIST 800-207 does not address AI systems as subjects or resources within its trust model; the extension is the contribution, not a restatement.

The OWASP Top 10 for LLM Applications [2] catalogues the primary attack vectors against large language model deployments, with prompt injection, insecure output handling, and excessive agency as the leading categories. OWASP's framing is vulnerability-centric rather than architectural, it identifies what can go wrong without prescribing a governance control plane to prevent it systematically at the enterprise level.

The MITRE ATLAS [9] is an extension of the ATT&CK framework for adversarial AI and covers the relevant tactics and techniques for ML-based systems. Similar to OWASP, it is a catalogue of threats instead of an architectural blueprint. The ISO/IEC 23894:2023 [4] standard provides guidance for managing AI risk at the organization level and focuses on governance processes rather than runtime enforcement of risk management policies.

The FSB [6] and BIS [7] have published reports analyzing AI risks in the financial services industry. The FSB's report

from 2024 identifies vulnerabilities associated with AI including cyber risk, model risk, third-party dependency, and market correlation, and calls for improved regulatory frameworks, however, does not focus on developing regulatory frameworks for enforcement of risk management policies through run-time per-interaction authorization. The EU AI Act [5] establishes regulatory obligations for high-risk AI systems but does not specify an enforcement architecture. The NIST AI RMF [15] provides a process-oriented governance framework; the CP-SGAI provides the technical architecture that operationalizes it at runtime.

No existing framework provides a unified architectural pattern for runtime enforcement of Zero Trust principles within the AI inference lifecycle: connecting identity attestation, prompt-level policy, ephemeral privilege scoping, and structured observability into a coherent control plane for regulated financial environments. SR 26-2 [10], issued April 2026, makes this gap explicit by carving out generative and agentic AI and directing institutions to rely on their own governance practices pending future regulatory guidance. That pending framework is what this paper proposes.

It is important to note that AWS released Amazon Bedrock AgentCore in October 2025, providing native services for agent runtime management, identity, memory, gateway, and observability within the AWS platform. AgentCore represents AWS's recognition that agentic AI requires purpose-built infrastructure beyond standard IAM and CloudTrail. However, AgentCore is a platform execution service; it provides agent runtime isolation, tool authentication via OAuth/SigV4, and agent-level observability within Bedrock. It does not provide per-interaction ephemeral credential scoping mapped to regulatory-domain identity context (the EPSC construct); a policy governance layer encoding institution-specific compliance rules above Bedrock Guardrails; an AI Observability Schema structured for U.S. financial regulatory audit requirements rather than operational telemetry; or a unified control plane spanning non-Bedrock AI services and external model providers. The CP-SGAI framework is therefore complementary to AgentCore: where AgentCore provides agent runtime infrastructure, CP-SGAI provides the enterprise regulatory governance envelope above it. SR 26-2 [10], issued April 2026, makes this governance gap explicit by carving out generative and agentic AI and directing institutions to rely on their own governance practices pending future regulatory guidance. That pending framework is what this paper proposes. [16]

4. The Cp-Sgai Framework Design

4.1. Design Principles

The CP-SGAI is designed around five fundamental principles that address different types of failure seen in enterprise AI deployments:

- Verify every AI interaction: Prompt verifications are independent of the network path, service account, or application it originates from. Every interaction is independently validated before inference can be performed.
- Scope permissions to the interaction: Access rights for an AI session are bounded by the context of that specific interaction, an AI session does not inherit permissions from a prior, standing role.
- Enforce policy at prompt time: Governance constraints are applied prior to any inference occurring and are not recreated through logs after the fact.
- Record everything required for audit: Prompt metadata, model context, response characteristics, identity association, and downstream actions will all need to be recorded with sufficient fidelity to enable a regulatory reconstruction.
- Treat AI systems as untrusted until verified: AI agents, orchestrators, and model endpoints are themselves subjects within the trust model, not trusted infrastructure.

4.2. Architectural Components

The control plane is structured around four primary components, each extending an existing Zero Trust building block for the AI domain.

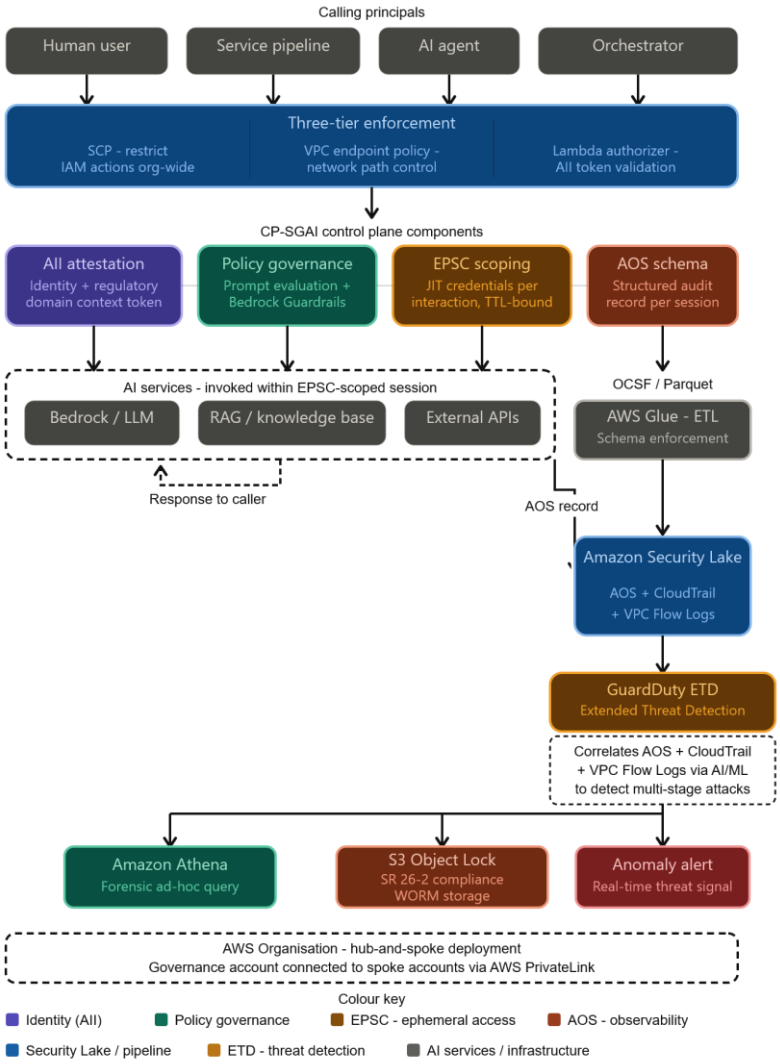

Fig 1: CP-SGAI Control Plane Architecture Overview

- AI Interaction Identity (AII) Attestation: Every AI interaction is assigned a structured identity context at session initiation. The AII captures: the calling identity (human or service principal), the application or orchestration layer submitting the prompt, the model endpoint targeted, the data classification context, and the regulatory scope of the interaction (e.g., consumer credit, fraud detection, customer service). This attestation is issued as a short-lived, cryptographically verifiable signed token analogous to a SPIFFE SVID (Secure Production Identity Framework for Everyone, a CNCF graduated standard) in a service mesh and validated by the policy decision component before inference proceeds.
- Policy-Driven Prompt Governance: Prompts are evaluated against a policy engine before reaching the model. Importantly, AWS provides Amazon Bedrock Guardrails, a native capability supporting prompt attack detection (including jailbreaks and injection), denied topic filtering, sensitive information redaction, and content classification across foundation models. The CP-SGAI policy governance component is designed to operate as an enterprise-layer orchestration envelope above Guardrails: where Guardrails provides per-model content enforcement, the CP-SGAI policy layer encodes institution-specific governance rules that Guardrails cannot express such as identity-class-bound topic restrictions, regulatory-domain-scoped prompt constraints, data classification controls on what may appear in retrieval context, and output handling rules based on the calling identity's regulatory scope. The two are complementary, not mutually exclusive. Policy evaluation is synchronous with the inference request; a rejected prompt fails fast with a structured error that feeds the observability layer [11].

- Prompt Security Context (EPSC): Each AI interaction with its environment utilizes a transitory (short-lived) security context that is established at the beginning of the session and removed at the end of the session. The EPSC defines the data sources the model may access (RAG corpora, APIs, databases), the tool calls an agentic system may execute, the output destinations to which a response may be routed, and the maximum session duration. Credentials and access tokens are generated at interaction start and revoked at interaction end, with no persistent standing access. This eliminates the broad, long-lived IAM roles that currently represent the dominant AI attack surface in enterprise cloud deployments.
- AI Observability Schema (AOS): The AOS specifies each log record in the system for every time a human interacts with an AI system in a structured way that's intended to meet requirements for both security operations and regulatory audits. Each record captures: a unique interaction ID linked to the AII token, prompt metadata (classification, length, detected risk indicators), response characteristics (output classification, detected sensitive patterns), model context snapshot (system prompt hash, RAG corpus version, tool configuration), identity and application attribution, policy evaluation result and applied constraints, downstream action trace for agentic interactions, and EPSC lifecycle events.

5. Ephemeral Ai Security: Moving Beyond Standing Permissions

The most pervasive security vulnerability in current enterprise AI deployments is not a novel attack technique, it is a governance architecture decision made for convenience. AI systems are typically granted standing IAM roles with broad permissions that persist indefinitely, because provisioning per-interaction access is operationally complex and most security teams have not yet built the tooling to do it at scale.

The consequence is predictable. An AI service account with persistent read access to a document corpus, a customer database, and an external API is a high-value target. A prompt injection attack or compromised orchestration layer does not need to escalate privileges; they are already granted. The blast radius of a successful attack is bounded only by what the role can access, which in most enterprises is substantially broader than any single interaction requires.

The EPSC model inverts this default: AI interactions begin with zero permissions. The control plane provisions the minimum access required for that specific interaction based on the AII attestation and applicable policy. When the interaction ends, access is revoked.

Table 2: Standing Permissions vs. Ephemeral EPSC

Dimension	Standing Permissions	Ephemeral EPSC
Access model	Persistent role assigned to AI service account	Per-interaction access provisioned at session start; revoked on completion
Blast radius	All resources accessible to the standing role	Only resources scoped to the current interaction and permitted by policy
Escalation risk	High; broad permissions available to any code running under the service identity	Low; no standing permissions; each session starts from zero
Audit trail	CloudTrail shows role assumption; no interaction-level granularity	Full lifecycle: provision, access, use, revoke; linked to AOS record
Complexity	Low; configure once, persists indefinitely	Higher; requires JIT provisioning; offset by control plane automation

In an AWS multi-account environment, the EPSC implementation uses AWS STS AssumeRole with session tags and condition keys to scope temporary credentials to specific resources, actions, and duration [11]. AWS IAM Identity Center serves as the identity federation layer. The temporary credentials provided to the user are returned in the STS API response, but are not stored in AWS Secrets Manager because AWS Secrets Manager serves a complementary role in the EPSC architecture. It stores and auto-rotates the external API keys, database credentials, and data source connection strings that the AI interaction may need to access within its scoped session, providing a managed secret retrieval mechanism for the resources the EPSC grants access to. The provisioning lifecycle is automatable through Lambda-backed orchestration; based on typical warm-path STS API response characteristics, the estimated overhead added to the interaction initiation path is in the range of tens of milliseconds; a figure that warrants formal benchmarking, discussed further in Section X.

6. AI Observability as a Distinct Security Discipline

Financial institutions have invested heavily in observability infrastructure such as SIEM platforms, CloudTrail aggregation, application performance monitoring, user behavior analytics. This investment was well-directed for the threat model it was built to address. It was not built for AI.

Traditional observability answers: who accessed what, when, from where, and whether it was authorized. For AI interactions, those questions are necessary but not sufficient. The security-relevant questions become: what was the model asked to do, in what context, with what data available, and what did it actually produce? The causal chain from prompt to response to downstream action is the unit of analysis and it is largely invisible to existing observability infrastructure by default. Amazon Bedrock does provide a Model Invocation Logging capability that can capture full request and response payloads to CloudWatch Logs or S3; however, it is disabled by default, not integrated with identity context from the AII attestation layer, not normalized to any regulatory audit schema, and does not capture the EPSC lifecycle, policy evaluation decisions, downstream agent actions, or cross-account interaction context. The AOS addresses these gaps by providing a structured, identity-attributed, policy-contextualized interaction record designed specifically to satisfy regulatory audit requirements and not merely operational logging.

Table 3: Traditional Observability vs. AI Observability Schema (AOS)

Capability	Traditional Observability	AOS
Unit of analysis	HTTP request/response: method, endpoint, status	Full AI interaction: prompt classification, model context, response characteristics, downstream actions
Identity	Service account that initiated the API call	AII token linking human identity, application layer, session context, and data classification scope
Anomaly detection	Volumetric patterns, access time, geolocation, failed auth	Semantic drift in prompts, response classification shifts, policy exception rates, EPSC access anomalies
Regulatory audit	Can demonstrate who accessed a resource	Can reconstruct full AI-assisted decision: prompt, model state, response, and resulting action with attribution
Incident response	Reconstruct network path and identity chain	Reconstruct prompt-response-action lineage; identify injection points; assess data exposure scope

A practical AOS implementation on AWS routes structured interaction records to Amazon Security Lake via a custom source integration. Security Lake requires data to be converted to Apache Parquet format and normalized to the Open Cybersecurity Schema Framework (OCSF) before ingestion; AWS Glue provides the ETL and schema enforcement layer for this transformation. AOS records are presented along with CloudTrail Management events and VPC Flow Logs from AWS Glue Data Catalog and can be queried via Amazon Athena as an ad-hoc forensic investigation. Amazon GuardDuty Extended Threat Detection can correlate these normalized event streams to surface multi-stage attack sequences involving AI workloads [11, 12]. The AOS record is stored append-only and tamper-evident enforced through Amazon S3 Object Lock in compliance mode that aligns to the institution's regulatory data retention obligations.

7. Governance across Organizational Personas

A control plane that only security engineers understand will not govern AI effectively across a large institution. The CP-SGAI serves four distinct organizational personas, each with different responsibilities and different views into the control plane

Table 4: Organizational Personas and Control Plane Responsibilities

Persona	Control Plane Responsibilities
Platform / Cloud Engineering	EPSC provisioning infrastructure, IAM policy templates, control plane deployment and lifecycle management
Security Operations	Policy tuning, anomaly response, prompt injection rule management, incident investigation
Risk & Compliance	AI usage attestation, regulatory reporting, audit preparation, model risk inventory
Application / AI Development	AII token integration, EPSC session management in application code, AOS logging SDK integration

Each persona carries a distinct regulatory obligation that the control plane must satisfy. Development teams need fast feedback loops for policy tuning. Risk and compliance teams need auditable attestation that AI usage within regulatory scope has been governed. Security operations teams need actionable anomaly signals and not raw logs requiring manual correlation. The CP-SGAI's AOS schema and governance interfaces are designed with this differentiation in mind.

8. Novel Contributions and Comparison to Existing Work

The three constructs introduced in this paper: AII attestation, EPSC lifecycle, and AOS schema are distinct from the existing literature in the following specific ways. NIST 800-207 and derivative Zero Trust frameworks define a generic PDP/PEP model applicable to any subject-resource-path access request. The CP-SGAI instantiates this model specifically for AI inference, with the AII as the subject attestation mechanism, the prompt governance engine as the enforcement point, and the EPSC as the per-interaction access scope. NIST 800-207 explicitly does not address AI systems as subjects within its trust model; the CP-SGAI's instantiation and extension into the AI governance domain is the contribution.

Model risk management frameworks have historically governed the development and validation lifecycle of AI models. The interagency guidance on model risk management, most recently revised as SR 26-2 [10] (April 2026, superseding SR 11-7), explicitly excludes generative AI and agentic AI from its scope, acknowledging these technologies are 'novel and rapidly evolving' and that a separate governance framework is required. This regulatory carve-out is a direct acknowledgment of the gap the CP-SGAI addresses: runtime enforcement and per-interaction access control for AI systems that no existing framework covers.

8.1. Agent Core Context

Amazon Bedrock AgentCore (GA: October 2025) confirms AWS's own recognition that agentic AI requires purpose-built governance infrastructure beyond standard IAM. AgentCore provides agent runtime isolation, identity management, and operational observability. However, it is an execution platform, not a regulatory compliance governance layer. It does not provide SR 26-2-aligned audit records, regulatory-domain-scoped identity attribution, or per-interaction ephemeral credential scoping across non-Bedrock AI services. The CP-SGAI operates in the governance space that SR 26-2 and AgentCore jointly leave open: runtime per-interaction enforcement with regulatory compliance as the primary design objective. [16]

The OWASP LLM Top 10 (2025 edition) [2] identifies Prompt Injection as LLM01, the leading threat category, alongside Sensitive Information Disclosure (LLM02) and Excessive Agency (LLM06). OWASP provides no architectural control plane to prevent these systematically at the enterprise level; it catalogs risks and suggests mitigations per application. The CP-SGAI operationalizes prompt injection defense and excessive agency controls at the governance layer, applying them uniformly across all AI workloads rather than leaving enforcement to individual application teams.

The AOS schema defines a structured observability record for AI interactions that is simultaneously aligned with security operations requirements (anomaly detection, incident response) and U.S. financial regulatory audit requirements (attribution, reconstruct ability, tamper-evidence) within a Zero Trust control plane architecture. A comprehensive prior art survey was not conducted; concurrent academic or industry work addressing this intersection may exist and should be surveyed before final submission.

8.2. AgentCore vs AOS

Amazon Bedrock AgentCore provides native agent observability: execution traces, tool call logs, and memory access patterns for operational monitoring and debugging. The AOS is distinct in purpose and design: it captures AII-attributed interaction records structured specifically to satisfy SR 26-2 model risk examination requirements, with tamper-evident retention under S3 Object Lock and an OCSF-normalized pipeline into Amazon Security Lake. AgentCore observability answers operational questions (did the agent execute correctly?); AOS answers regulatory audit questions (who authorized this AI interaction, under what policy, with what data access scope, and what was produced?). A comprehensive prior art survey was not conducted; concurrent academic or industry work addressing this intersection may exist and should be surveyed before final submission. [16]

9. Implementation Considerations for Enterprise Deployment

9.1. Multi-Account Architecture Alignment

The CP-SGAI is deployed in a hub-and-spoke model across an AWS organization. It has its own AI governance account in the Security organizational unit (OU) where the Policy Engine, AII Attestation Service, EPSC Provisioning Lambda Functions, and AOS Ingest Pipeline are hosted. Business unit Accounts access the control plane services via an AWS PrivateLink endpoint which keeps governance traffic on the AWS backbone and away from the public internet.

The governance layer is enforced through a three-tier mechanism. First, AWS Service Control Policies (SCPs) restrict the IAM actions available to principals within the organization, for example, denying bedrock:InvokeModel except when called from an approved role. SCPs operate at the IAM action and principal level and cannot enforce routing through a specific API Gateway endpoint but can enforce which IAM principals may invoke AI services. Second, VPC endpoint policies on the Amazon Bedrock PrivateLink interface endpoint restrict invocation to requests originating from the governance VPC. This provides network-level enforcement to support the SCP principal-level enforcement. Third, a Lambda authorizer attached to the governance API Gateway validates the AII token on every inbound request before forwarding it to the AI service. These three methods work in conjunction to create a significantly high barrier to circumventing the Governance layer. SCPs restrict which principals can call AI services, VPC endpoint policies restrict the network path, and the Lambda authorizer enforces token validity and policy compliance at the application layer [11, 12].

9.2. Migration Path for Existing AI Deployments

Most financial institutions already have AI workloads in production that predate any formal governance framework. A practical migration follows three phases: Audit (inventory existing AI deployments, classify by regulatory scope, assess current control posture against CP-SGAI requirements); Instrument (integrate AOS logging as the first step; observability without enforcement, establishing the baseline telemetry needed to tune subsequent policy); and Enforce (progressively introduce AII attestation and EPSC scoping, starting with the highest-risk regulatory domains and expanding as operational confidence grows).

This phased approach reflects operational reality: a governance framework that requires complete re-architecture of existing workloads as a prerequisite for adoption will not be adopted. The CP-SGAI is designed to be incrementally adoptable, with observability as the entry point and full enforcement as the target state.

10. Limitations and Future Work

Four limitations warrant explicit acknowledgment. First, the AOS schema captures prompt metadata and response characteristics rather than full prompt and response content in all cases, a deliberate design decision driven by the data classification risks of storing raw prompt text, but one that limits forensic reconstruction depth. Future work should explore privacy-preserving prompt summarization techniques that enable richer audit records without creating secondary data exposure risks.

Second, the prompt governance policy engine assumes that injection signatures and prohibited content patterns can be defined with sufficient specificity to be useful without generating unacceptable false positive rates. Adversarial prompt techniques evolve continuously, and static policy rules require ongoing maintenance. Integration with adaptive policy models that update signature libraries based on observed attack patterns is a necessary extension.

Third, the framework's AWS-specific implementation guidance is a point-in-time snapshot of available services. The rapid evolution of AWS AI and security service capabilities means specific implementation patterns will require revision; the architectural principles are service-agnostic and should remain stable.

Fourth, the EPSC provisioning latency estimate provided in Section V has not been formally benchmarked. The actual overhead will vary with Lambda cold-start behavior, cross-account network latency, IAM policy complexity, and the volume of session tags passed. For those evaluating the CP-SGAI for high-throughput use cases should treat the latency figure as indicative and conduct environment-specific load testing before committing to a synchronous provisioning architecture.

Future work will focus on formal quantitative benchmarking of EPSC provisioning latency across representative enterprise workload profiles; development of an open-source reference implementation of the AOS schema and Security Lake custom source integration; a comprehensive prior art survey to confirm AOS schema novelty; and a comparative study of prompt governance policy effectiveness against adversarial prompt datasets from published red-teaming literature.

11. Conclusion

Financial services do not lack AI governance guidance; it lacks AI governance architecture. The revised interagency model risk management framework, SR 26-2, explicitly excludes generative and agentic AI from its scope and directs institutions to apply their own risk management practices in the interim. In other words, regulators have acknowledged the gap and handed the problem back to institutions. What institutions need is not more principles but an architectural pattern that operationalizes governance at runtime, one that enforces policy at the moment of inference rather than reconstructing compliance posture from logs after the fact.

The Control Plane Architecture for Secure and Governable AI (CP-SGAI) presented here is an attempt to fill that gap. By introducing AI Interaction Identity attestation,

Ephemeral Prompt Security Contexts, and a structured AI Observability Schema, the framework provides a coherent runtime governance layer that operationalizes secure and governable AI deployment at scale in a regulated banking environment.

The work is grounded in direct operational experience rather than theoretical modeling, and the framework is designed to be adopted incrementally, meeting institutions where their current governance posture is rather than demanding a complete re-architecture as the price of entry. The architecture to govern AI in banking responsibly at runtime exists. Whether the industry moves quickly enough to deploy it before the next compliance failure is the harder question.

References

[1] NIST, "Zero Trust Architecture," Special Publication 800-207, National Institute of Standards and Technology, Aug. 2020. [Online]. Available: https://doi.org/10.6028/NIST.SP.800-207

[2] OWASP Foundation, "OWASP Top 10 for LLM Applications & Generative AI," v2025, 2025. [Online]. Available: https://owasp.org/www-project-top-10-for-large-language-model-applications/

[3] ENISA, "Artificial Intelligence Cybersecurity Challenges: Threat Landscape for AI and ML," European Union Agency for Cybersecurity, 2023. [Online]. Available: https://www.enisa.europa.eu/publications/artificial-intelligence-cybersecurity-challenges

[4] ISO/IEC 23894:2023, "Information Technology - Artificial Intelligence - Guidance on Risk Management," International Organization for Standardization, 2023. [Online]. Available: https://www.iso.org/standard/77304.html

[5] European Commission, "Regulation (EU) 2024/1689 -- Artificial Intelligence Act," Official Journal of the European Union, 2024. [Online]. Available: https://eur-lex.europa.eu/eli/reg/2024/1689/oj/eng

[6] Financial Stability Board, "The Financial Stability Implications of Artificial Intelligence," FSB Report, November 14, 2024. [Online]. Available: https://www.fsb.org/2024/11/the-financial-stability-implications-of-artificial-intelligence/

[7] Bank for International Settlements, "Artificial Intelligence and the Economy: Implications for Central Banks," BIS Annual Economic Report 2024, Chapter III, June 2024. [Online]. Available: https://www.bis.org/publ/arpdf/ar2024e3.htm

[8] F. Perez and I. Ribeiro, "Ignore Previous Prompt: Attack Techniques for Language Models," in Workshop on Machine Learning Safety, Advances in Neural Information Processing Systems (NeurIPS), New Orleans, LA, USA, 2022. [Online]. Available: https://arxiv.org/abs/2211.09527

[9] MITRE Corporation, "MITRE ATLAS: Adversarial Threat Landscape for Artificial-Intelligence Systems," v5.1.0, November 2025. [Online]. Available: https://atlas.mitre.org

[10] Board of Governors of the Federal Reserve System, OCC, and FDIC, "Revised Guidance on Model Risk Management," SR Letter 26-2 / OCC Bulletin 2026-13 / FDIC FIL-15-2026, April 17, 2026. Supersedes SR 11-7 (April 4, 2011). [Online]. Available: https://www.federalreserve.gov/supervisionreg/srletters/SR2602.htm - OCC: https://www.occ.treas.gov/news-issuances/bulletins/2026/bulletin-2026-13.html

[11] Amazon Web Services, "Security Best Practices for Amazon Bedrock," AWS Documentation, 2024. [Online]. Available: https://docs.aws.amazon.com/bedrock/latest/userguide/security-best-practices.html

[12] Amazon Web Services, "AWS Well-Architected Framework: Security Pillar," AWS Whitepaper, 2024. [Online]. Available: https://docs.aws.amazon.com/wellarchitected/latest/security-pillar/welcome.html

[13] Y. Bai et al. (Anthropic), "Constitutional AI: Harmlessness from AI Feedback," arXiv:2212.08073, December 2022. [Online]. Available: https://arxiv.org/abs/2212.08073

[14] Gartner, Inc., "Hype Cycle for Artificial Intelligence, 2024," Gartner Research, June 2024. Authors: Afraz Jaffri, Haritha Khandabattu. [Online]. Available: https://www.gartner.com/en/documents/5227007

[15] NIST, "Artificial Intelligence Risk Management Framework (AI RMF 1.0)," National Institute of Standards and Technology, Jan. 2023. [Online]. Available: https://doi.org/10.6028/NIST.AI.100-1

[16] Amazon Web Services, “Amazon Bedrock AgentCore: Build, deploy, and operate AI agents at scale,” AWS Documentation, October 2025. [Online]. Available: https://docs.aws.amazon.com/bedrock-agentcore/latest/devguide/what-is-bedrock-agentcore.html