International Journal of Artificial Intelligence, Data Science, and Machine Learning
Grace Horizon Publication | Volume 7, Issue 2, 9-16, 2026
ISSN: 3050-9262 | https://doi.org/10.63282/3050-9262.IJAIDSML-V7I2P103

Original Article

Enterprise Agentic AI Lifecycle Governance: A Control-Driven Framework from Design to Decommissioning

Sandeep Kumar Anuguthala
Independent Researcher/VP of Technology Risk and Control, JP Morgan Chase & Co.

Received On: 20/02/2026 ***Revised On: 17/03/2026*** ***Accepted On: 26/03/2026*** ***Published On: 06/04/2026***

Abstract - *Agentic Artificial Intelligence (AI) systems, characterized by autonomous multi-step planning and execution capabilities, are increasingly transforming enterprise operations. However, their autonomy introduces novel risks related to security, compliance, and operational control that are not fully addressed by existing AI governance frameworks [1], [4]. This paper proposes a control-driven lifecycle governance framework tailored for agentic AI systems, spanning planning, design, development, deployment, monitoring, and decommissioning. The framework integrates inherent risk assessment, risk tiering, and continuous control validation aligned with established standards such as the NIST AI Risk Management Framework [1]. It further incorporates agent-specific threat modeling approaches, including MITRE ATLAS [2] and MAESTRO [3], to address emerging adversarial risks. A case study demonstrates how the framework enables organizations to operationalize agentic AI systems in a secure, controlled, and compliant manner.*

Keywords - *Agentic AI, AI Governance, Risk Management, Autonomous Systems, Lifecycle Management, AI Security.*

1. Introduction

Agentic AI represents a shift from traditional AI paradigms by enabling systems to independently plan, decide, and act across chained tasks [7]. These systems interact with tools, data sources, and peer agents, forming complex, adaptive execution environments [6], [7].

While widely adopted governance standards such as the NIST AI Risk Management Framework [1] and ISO-aligned controls [4] offer foundational guidance, they primarily assume static models or human-mediated decision paths. They do not fully capture challenges introduced by agentic AI, including autonomous execution, evolving behavior, multi-agent orchestration, and real-time decision loops [6], [13].

Despite increasing adoption, enterprises lack an operationally enforceable governance model that integrates lifecycle management, risk-tiered controls, and accountability for agentic systems [1], [8], [13]. This gap motivates the need for a unified framework that translates high-level governance principles into actionable controls across the agent lifecycle

Contributions

This work contributes:

- A lifecycle-oriented governance model for agentic AI systems.
- A control-translation approach that maps risk levels to enforceable safeguards.
- Integration of AI-specific threat modeling (MAESTRO and MITRE ATLAS) into governance workflows.
- A system-of-record construct to enable end-to-end traceability and accountability for agentic use cases.
- A case study demonstrating enterprise applicability in a financial-service context.

2. Background and Related Work

2.1. Agentic AI Fundamentals

Agentic AI refers to AI systems that autonomously plan and execute multi-step workflows in pursuit of specific goals, often by coordinating multiple tools and services under persistent objectives. Agentic systems are composed of several functional elements as described in emerging agentic AI frameworks [6],[7]. Core components include a reasoning engine (typically an LLM), tools and APIs for interacting with enterprise systems, instructions and policies that define capabilities and boundaries, memory mechanisms, and orchestration logic that coordinates interactions among agents and backend services. As illustrated in Figure 1, enterprise deployments may use single-agent designs, where one agent performs end-to-end reasoning and tool use, or multi-agent designs, where specialized agents collaborate on subtasks.

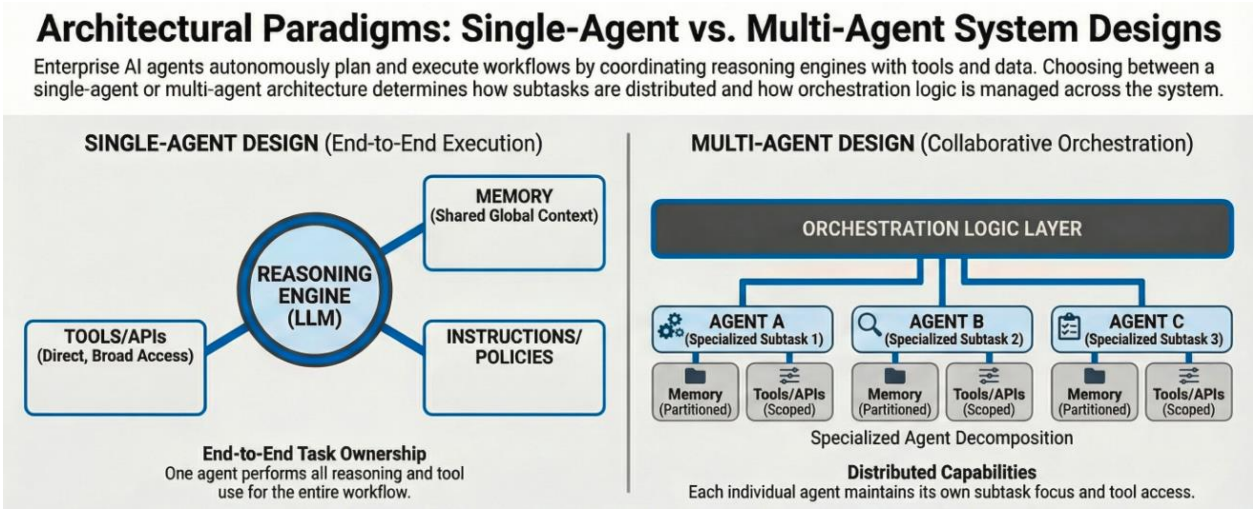

Fig 1: Single Vs Multi Agent System Design

Organizations such as NVIDIA have developed frameworks to classify the autonomy levels of agentic AI systems [7]. Tabel 1 summarizes the autonomy level used in this work.

Table 1: Autonomy Levels

Autonomy Level	Description	Example
0-Inference API	Single request -> Single model call	Text Classification
1-Deterministic	Predefined multi-step execution	Rule-based workflow calling an LLM
2-Conditional autonomy	Conditional decision points	Customer support assistant that decides whether to answer, escalate or retrieve data
3- High Autonomy	Dynamic Planning and execution	Autonomous financial agent that evaluates transactions and initiates refunds via APIs

Higher autonomy generally correlates with higher risk exposure and stricter governance requirements [7],[13].

2.2. AI Governance and Risk Frameworks

Current governance frameworks provide important principles but limited coverage for agentic behaviors. The NIST AI Risk Management Framework [1] outlines risk identification and mitigation but lacks prescriptive lifecycle enforcement for autonomous systems. Similarly, ISO/IEC 27001 [4] and NIST SP 800-53 [5] define control baselines without addressing AI-specific dynamics such as reasoning loops and tool-mediated actions.

Conventional threat modeling approaches (e.g., STRIDE, PASTA) emphasize software-centric risks and do not fully represent AI-specific attack vectors such as prompt injection or model manipulation [2],[9]. MITRE ATLAS [2] extends adversarial modeling to AI, and MAESTRO [3] introduces layered analysis for agentic environments. However, these are not typically embedded within a lifecycle governance construct.

2.3. Gaps in Existing Approaches and Novelty of This Work

Despite these advances, current frameworks leave several gaps when applied to highly autonomous enterprise agents. Most guidance remains model-centric and does not treat persistent agent identities, long-lived memory, and direct write access to production systems as first-class governance concerns [13],[14]. Lifecycle integration is often implicit, requiring organizations to manually map high-level functions or generic control families onto concrete activities across planning, development, testing, deployment, monitoring, and decommissioning for agentic workflows

This paper bridges these gaps by unifying lifecycle governance, risk tiering, and threat-informed control design into a cohesive model. Unlike prior frameworks such as NIST AI RMF [1], which provide high-level guidance, and threat-centric approaches such as MITRE ATLAS [2] and MAESTRO [3], which primarily focus on adversarial modeling, the proposed framework introduces a lifecycle-enforced governance model that operationalizes risk-tiering, control mapping, and accountability into a single system. The integration of a mandatory system-of-record and a RACI-based governance structure enables traceability and enforceability, representing a practical advancement for enterprise-scale adoption of agentic AI systems [13],[14].

3. Proposed Lifecycle Governance Framework

3.1. Governance Structure and Roles

Effective governance of agentic AI systems requires a cross-functional structure that aligns business objectives, risk management, and technical implementation [13]. The proposed framework assumes an enterprise-level Agentic AI

Governance Owner with ultimate accountability for maintaining policies, standards, and a unified lifecycle process for agentic workflows. This role is supported by an Agentic AI Governance Committee that includes representatives from Model Risk Management, Cybersecurity/Security Engineering, Technology or AI Risk Management, Data Governance, Legal/Compliance, and Operational Risk [13].

At the use-case level, a Use Case Owner (UCO) is accountable for defining business intent, ensuring alignment with organizational policies, and maintaining the lifecycle status of the agent in the system-of-record [13]. The Development Team (DEV) is responsible for designing and implementing the agentic workflow in accordance with approved architectures and control requirements. Technology/AI Risk (RISK) leads inherent-risk assessment and risk tiering, Cybersecurity (SEC) leads threat modeling and security control design, Data Governance (DATA) assesses data usage and approvals, and Model Risk Management (MRM) performs independent validation where the agent relies on models in scope of model-risk policies.

3.2. RACI Model across the Lifecycle

To operationalize this structure, the framework defines a RACI (Responsible, Accountable, Consulted, Informed) model that assigns ownership to each role across key lifecycle activities. Activities include use-case registration, definition of business intent and architecture, inherent-risk assessment, risk tiering, control-requirements definition, threat modeling, data-governance approvals, design, development, and control implementation, testing and validation, production readiness review, deployment and change management, continuous monitoring, incident response, ongoing risk assessments, and decommissioning decisions. Table 2 presents the RACI assignments across lifecycle phases.

Table 2: RACI across Lifecycle Phases

Lifecycle Activity	UCO	DEV	RISK	SEC	DATA	MRM	GOV
Use Case Registration	A/R	C	C	C	C	C	I
Define Business Intent & Architecture	A/R	R	C	C	C	C	I
Inherent Risk Assessment	R	I	A/R	C	C	C	I
Risk Tiering Determination	C	I	R	C	C	C	A
Control Requirements Definition	C	R	A	R	C	C	A
Threat Modeling	I	C	I	A/R	I	I	I
Data Governance Approvals	R	I	C	I	A/R	I	I
Design Approval (End of Planning Phase)	R	I	C	C	C	C	A
Development	A	R	C	C	I	I	I
Control Implementation	C	A/R	C	R	I	I	I
Secure Coding (SAST/DAST/SCA)	I	A/R	I	R	I	I	I
HITL Design Implementation	A	R	C	C	I	I	I
Testing & Validation	A	R	C	C	C	C	I
Model Validation	I	I	C	I	I	A/R	I
Security control validation/Red Teaming	I	R	C	A/R	I	I	I
Residual Risk Acceptance	A	I	R	C	C	C	A
Production Readiness Review	R	I	C	C	C	C	A
Deployment & Change Management	A	R	I	C	I	I	I
Continuous monitoring set up	A	R	C	R	C	C	I
Ongoing Risk Assessment	R	I	A/R	C	C	C	I
Incident Response (AI-Specific)	A	R	C	A/R	C	I	I
Decommission Decision	R	I	C	C	C	C	A
System of Record updates	A/R	I	I	I	I	I	I

The RACI structure is aligned with enterprise governance practices for AI risk management and accountability models [13].

3.3. System of Record for Agentic Use Cases

A central design principle of the framework is the use of a system-of-record for agentic AI use cases. This system maintains a canonical inventory of all agentic workflows, capturing for each use case: a unique identifier, business owner, lifecycle phase, risk tier, model and tool dependencies, data sources and classifications, autonomy level, integration endpoints, and key control requirements. The Agentic AI Governance Owner is responsible for establishing and maintaining this system, while Use Case Owners are responsible for keeping their entries complete and up-to-date across the lifecycle. In practice, the system of record may be implemented using enterprise governance platforms, model registries, or workflow orchestration systems to ensure integration with existing control processes [1],[14].

Registration in the system-of-record is mandatory when there is an intent to build or deploy an agentic workflow, whether developed in-house or procured from third parties. Lifecycle transitions such as moving from planning to development, from testing to production, or from production to decommissioning are only permitted after the required governance checkpoints are completed and recorded. This approach aligns with best practices in AI RMF-aligned

inventories and model registries but extends them to agentic workflows by explicitly tracking agent identities, autonomy levels, tool access, and decommissioning status [14].

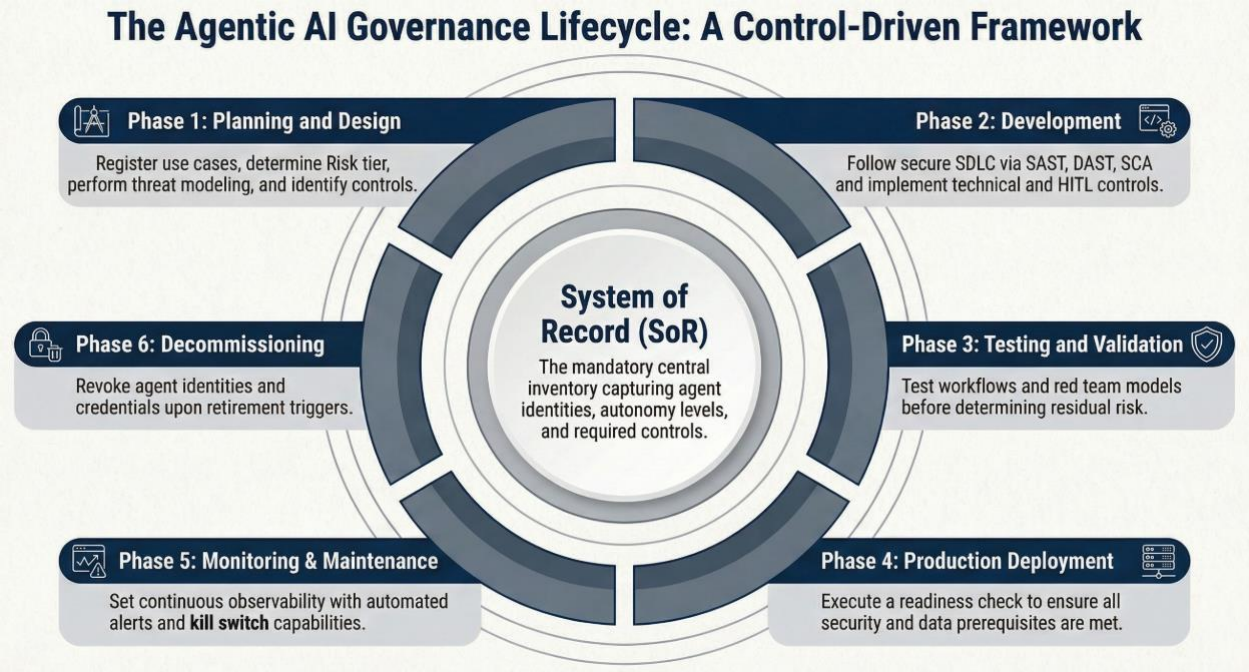

Fig 2: Proposed Agentic AI Lifecycle Governance Framework

As illustrated in Figure 2, the proposed framework embeds governance checkpoints across all lifecycle phases, with the system of record serving as the central control plane. Risk-tiering and control enforcement act as cross-cutting layers, ensuring that agentic workflows cannot progress without satisfying defined governance requirements.

4. Phase-Wise Governance and Controls

Each phase includes defined entry and exits criteria enforced through governance gates recorded in the system of record. The framework defines governance activities and control expectations across six lifecycle phases: (1) Planning and Design, (2) Development, (3) Testing and Validation, (4) Production Deployment, (5) Operational Monitoring and Maintenance, and (6) Decommissioning. Each phase builds on the previous ones and is gated by reviews recorded in the system-of-record, ensuring that agentic workflows cannot progress without the required risk assessments, approvals, and control implementations [1],[13].

4.1. Phase 1: Planning and Design

Phase 1 establishes the foundation for safe deployment of agentic workflows by covering use-case registration, inherent-risk assessment, risk tiering, baseline control definition, threat modeling, and data-governance approvals [1],[13],[4].

4.1.1. Use Case Registration

When there is an intent to build or deploy an agentic workflow, the Use Case Owner must register the use case in the enterprise system-of-record. The registration includes the business objective, high-level description of the workflow, expected autonomy level, target users, and preliminary information on the technical stack (LLMs, tools, datasets, orchestrators, and integration endpoints). Both in-house developments and third-party solutions must be registered. The system-of-record assigns a unique identifier and records ownership, lifecycle phase, and initial risk assumptions, creating traceability from the earliest design decisions.

4.1.2. Inherent Risk Assessment

After registration, the Use Case Owner collaborates with Technology/AI Risk to determine the inherent-risk profile of the agentic workflow. The assessment considers six primary dimensions, informed by NIST AI RMF and model-risk-management guidance: intended use and context; potential harms (financial, privacy, safety, fairness, reputational); affected stakeholders; data characteristics (quality, bias, sensitivity, provenance, lineage); model capabilities and limitations; and operational environment. The outcome is a qualitative inherent-risk rating for each dimension and an overall inherent-risk profile [1], [10].

4.1.3. Risk Tiering Model

The framework translates the inherent-risk profile into one of four governance tiers (Tier 1–Tier 4), which determine the depth and rigor of required controls, validations, and approvals. Tiers are defined based on autonomy, data sensitivity, impact, and system coupling, ranging from assistive, low-impact agents (Tier 1) to fully autonomous agents that can initiate or approve high-impact actions in critical systems (Tier 4). Each tier corresponds to a baseline control set and approval path; higher tiers require stronger human-in-the-loop checkpoints, independent validation, more

stringent monitoring, and higher-level residual-risk acceptance [13].

4.1.4. Baseline Control Requirements

Baseline controls are derived from the assigned risk tier, applicable regulatory and policy requirements, and general AI and security-risk-management practices. They span identity and access management (unique agent identities, least privilege), data protection and governance (classification, access controls, encryption, retention), logging and observability (comprehensive logging and tracing), model and policy controls (guardrails, model-performance monitoring), change and configuration management (versioning, change approvals), and incident response (AI-specific playbooks and kill switches). These baseline controls must be reflected in the design documentation and recorded in the system-of-record [13],[14].

4.1.5. Threat-Informed Control Design

Baseline controls are supplemented by incremental controls derived from threat modeling. To capture the autonomy, tool-use, and interactive behavior of agentic systems, the framework adopts a procedure that combines MAESTRO, MITRE ATLAS, and OWASP guidance. Security architects map the agentic workflow onto MAESTRO's layers, identify threats per layer using relevant MITRE ATLAS tactics and techniques, rate threats for likelihood and impact given the agent's risk tier and baseline controls, and select incremental controls from OWASP guidance and internal catalogs for example, input validation and prompt-injection filters, constrained tool execution, time-bound credentials, hard caps on actions, and kill switches. The resulting threat-to-control mapping is added to the design and tracked through development and testing [2],[3],[9].

4.1.6. Data Governance Approvals

Before leaving the planning and design phase, the Use Case Owner must obtain data-governance approvals for all datasets and data flows involved in the agentic workflow. Data-governance assessors review data sources and legitimacy, classification (e.g., PII, PCI, SPI), ownership and stewardship, access patterns, data quality and lineage, privacy risks and required privacy-impact assessments, and retention and destruction requirements. Approval status and constraints are recorded in the system-of-record. Only after risk assessment, tiering, control definition, threat modeling, and data-governance approvals are completed and reviewed by the Governance Committee can the use case advance to Phase 2.

4.2. Phase 2: Development

Phase 2 translates the approved design into an implemented agentic workflow, ensuring that baseline and incremental controls are embedded from the outset and that secure-development practices are followed.

4.2.1. Implement the Approved Architecture

The Development team implements the agentic workflow in accordance with the approved architecture and control requirements. Activities include building the agentic orchestrator; integrating with LLMs, tools, memory components, and knowledge sources; implementing workflow logic, guardrails, and escalation paths; and instrumenting the system for observability. Material deviation from the approved architecture must be documented and may trigger re-assessment in phase 1 [13].

4.2.2. Implement Required Controls

Controls identified in Phase 1 both baseline and incremental must be implemented as part of development. Key examples include agent identity and entitlements (unique identities, least-privilege permissions), audit logging and tamper-resistant storage, data protection aligned with data-governance approvals, and layered technical controls across identity, application, model, data, and infrastructure layers [13], [14].

4.2.3. Following Secure Software Development Practices

Secure-software development practices are integrated into the lifecycle to reduce vulnerabilities before production. Recommended practices include static application security testing (SAST), software composition analysis (SCA), dynamic application security testing (DAST), and secret/configuration scanning. For Tier 3 and Tier 4 agents, critical findings must be resolved before proceeding to testing [15].

4.2.4. Designing Human-In-The-Loop Controls

For higher-risk tiers, the agentic workflow must incorporate human-in-the-loop (HITL) Controls at points where agents can perform irreversible or high-impact actions, such as writing to production systems, approving financial transactions, or modifying security configuration. During development, the team defines HITL trigger conditions, pause and approval mechanisms, and user interfaces that provide sufficient context for safe human oversight [13].

4.3. Phase 3: Testing and Validation

Phase 3 verifies that the agentic workflow behaves as intended, that controls operate effectively, and that residual risk is within the organization's risk appetite before production deployment [13],[16].

4.3.1. Functional and Workflow Testing

Functional testing ensures that the agentic workflow implements the intended business logic and handles normal and edge cases correctly. Tests cover correct tool invocation, and sequence, handling of invalid or ambiguous inputs, escalation and fallback behavior, and coordination in multi-agent setups [13],[16].

4.3.2. Model-Performance Validation

Where the agent relies on machine-learning models, Model Risk Management and the development team validate that model performance is adequate for the intended use and risk tier. Validation includes accuracy and robustness, stability across input distributions, fairness and bias indicators where applicable, and explainability appropriate to regulatory and internal requirements. For Tier 3 and Tier 4 agents,

independent model validation is recommended before production approval.

4.3.3. Security Control Validation and AI Red Teaming

Security validation confirms that baseline and incremental controls derived from threat modeling are correctly implemented and effective. Activities include validating enforcement of agent identities and entitlements, logging coverage and alerting for high-risk actions, and conducting targeted AI red teaming or adversarial testing such as simulated prompt injections, goal-hijacking attempts, and tool-misuse scenarios aligned with MAESTRO and MITRE ATLAS threats [2], [3], [16].

4.3.4. Data, HITL, and Performance Validation

Additional validation steps include data validation (conformance to data-governance decisions, including access and retention); HITL validation (correct triggering and enforcement of human approvals and rejections); and stress and performance testing (response times, throughput, resilience under load, and behavior under component failures).

4.3.5. Residual-Risk Assessment and Approvals

After testing, the Use Case Owner and Technology/AI Risk jointly assess residual risk, considering test results, open issues, and control effectiveness. If residual risk is within the organization's appetite for the assigned tier, the Use Case Owner updates the system-of-record with validation outcomes and seeks approval from the Governance Committee to proceed to Phase 4 (Production Deployment). If residual risk remains unacceptably high and cannot be mitigated, the Governance Committee may require design changes, reduced autonomy, or rejection of the use case [13].

4.4. Phase 4: Production Deployment

Phase 4 ensures that the agentic system is operationally safe, compliant, and properly monitored before interacting with real users and live data.

4.4.1. Production-Readiness Review

Prior to deployment, a production-readiness review confirms that all pre-requisites from previous phases are satisfied. Model Risk Management verifies validation outcomes, Security confirms remediation of vulnerabilities and deployment of required controls, Data Governance confirms compliant data usage, and Technology/AI Risk confirms that controls are operating effectively. The Governance Committee approves or rejects the transition to production and updates the system-of-record accordingly [1],[11].

4.4.2. Production Environment Security Validation

Security teams validate that the production environment is configured securely, including secrets management, secure API integration, encryption, network segmentation, and monitoring configurations. For higher-tier agents, additional safeguards such as restricted rollout, feature flags, or canary deployments may be applied [13].

4.4.3. Change-Management Integration

Production deployments follow the organization's change-management process, including change requests, impact analysis, approval of deployment windows, rollback plans, and post-deployment verification. This reduces the risk of unintended disruptions and ensures that agentic deployments are visible within existing operational processes.

4.5. Phase 5: Operational Monitoring and Maintenance

Phase 5 focuses on continuous performance, safety, and risk monitoring, as well as lifecycle updates as the environment evolves [6].

4.5.1. Continuous Monitoring

Use Case Owners and operations teams monitor whether the agentic system continues to perform as expected. Monitoring covers performance metrics, error rates, model drift indicators, and high-risk activities such as updates to financial records or security configurations. Logging and alerting should support both real-time intervention and retrospective analysis [6],[13].

4.5.2. Alerting and Intervention

Organizations define alert thresholds and intervention rules. Programmatic thresholds detect patterns such as repeated failed tool calls or anomalous transaction patterns; anomaly-detection techniques identify outlier behaviors; and, where appropriate, agents may monitor other agents and flag inconsistencies. For high-priority alerts, execution may be paused or the agent's permissions restricted until human review is completed, while catastrophic malfunctions can trigger kill switches and fallback procedures [6].

4.5.3. Ongoing Risk and Control Assessment

Use Case Owners periodically re-engage Technology/AI Risk, Security, and Data Governance to reassess inherent and residual risk, control adequacy, and compliance with data-usage and regulatory requirements. Material changes such as expanded autonomy, new tools, new data sources, or broader impact may require re-tiering, additional controls, or re-approval via earlier lifecycle phases [13].

4.6. Phase 6: Decommissioning

Decommissioning is the final phase, triggered when an agent no longer meets performance or safety criteria or when its risk profile becomes unacceptable.

4.6.1. Decommissioning Triggers

Typical triggers include security compromise or malfunction; residual risk exceeding the organization's appetite during periodic assessments; expansion of the agent's "lethal trifecta" (intersection of tool access, autonomy level, and impact area) beyond originally defined bounds; and unauthorized repurposing of the agent for new business contexts. These triggers are assessed in collaboration between the Use Case Owner, Technology/AI Risk, Security, and the Governance Committee [6], [12],[13].

4.6.2. Decommissioning Activities

When decommissioning is initiated, the Use Case Owner and Governance Committee coordinate to revoke agent identities and credentials, disable tool access, shut down orchestrators, and ensure that no residual privileged access remains. Data-retention and destruction follow organizational and regulatory requirements, while audit logs and relevant artifacts are preserved for required periods. The system-of-record is updated to reflect decommissioned status and rationale. Where appropriate, lessons learned from incidents or near misses feed back into earlier phases to improve future designs [13],[14].

5. Illustrative Case Study: Autonomous Refund Agent

To demonstrate application of the framework, consider a fully autonomous refund agent deployed by a financial institution to process low-value customer refund requests[13]. The agent receives inputs from customer-service channels, retrieves account and transaction data from core banking systems, evaluates eligibility against policies, and initiates refund transactions through an internal payments API. The workflow operates with minimal human oversight for transactions below a defined threshold, making it a candidate for Tier 4 (Critical) under the risk-tiering model [13],[14].

In Phase 1, the use case is registered in the system-of-record and undergoes inherent-risk assessment and risk tiering. The assessment classifies the use case as high impact due to financial loss potential, high data sensitivity (customer financial data), and tight coupling to critical payment systems. Baseline controls include strong agent identity and entitlements, detailed logging of every refund decision, segregation of duties for higher-value refunds, and human-in-the-loop approvals for exceptions. Threat modeling using MAESTRO and MITRE ATLAS identifies threats such as prompt injection via upstream channels, delegated privilege abuse, and runaway refund loops [2],[3]; incremental controls include strict input validation, time-bound credentials, hard caps on refunds per customer and per time window, and a kill switch that disables refund initiation while preserving diagnostics.

In Phases 2 and 3, the architecture and controls are implemented and validated [14],[16]. Functional tests confirm correct refund logic and error handling; model-performance validation checks adequacy of fraud and eligibility logic; red-team exercises probe prompt-injection and tool-misuse paths; and HITL controls for higher-risk scenarios are verified. Residual risk is assessed as acceptable for Tier 4, and the Governance Committee approves progression to production with conditions [14].

In Phases 4 and 5, the agent is deployed with continuous monitoring of refund volumes, error rates, and anomaly indicators such as unusual spikes for specific customers or channels. Alerts trigger human review and, if necessary, temporary suspension of refund initiation through the kill switch [6],[12]. Periodic reviews reassess risk, autonomy, and control adequacy, ensuring that the agent's lethal-trifecta parameters remain within approved bounds [12],[13].

If residual risk becomes unacceptable-for example, after repeated attempted abuses or changes in regulatory expectations-the governance committee may initiate phase 6 (Decommissioning), revoking the agents access, disabling its workflow, and preserving logs for investigation and audit. This case illustrates how the proposed lifecycle framework converts deployments of a powerful autonomous agents into a governed process with explicit risk-tiered controls, accountability, and exit criteria [13],[14].

5.1. Qualitative Evaluation

Although the case study is hypothetical and anonymized, it reflects typical characteristics of early enterprise deployments of autonomous refund agents and similar financial-transaction workflows. Practitioners involved in such deployments have highlighted several qualitative benefits of applying the proposed framework: clearer ownership for risk assessments and approvals through the RACI model, earlier identification of high-impact threat scenarios via MAESTRO and MITRE ATLAS-aligned threat modeling, and improved traceability of design and risk decisions through the agentic system-of-record [13],[14]. In particular, the explicit Tier 4 classification for the refund agent, coupled with mandatory HITL checkpoints and kill-switch integration, was viewed as materially reducing the blast radius of potential misbehavior compared with an ad hoc deployment under generic application-security controls.

At the same time, feedback from risk and engineering stakeholders indicates that the framework introduces additional overhead in documentation, coordination, and testing especially for higher-risk tier agents. This trade-off between deployment speed and governance rigor is consistent with broader experience in AI and model risk management and underscores the need for automation of inventories, risk assessments, and control verification in the future work [1],[11].

6. Conclusion and Future Work

Agentic AI systems promise significant efficiency and innovation benefits for enterprises but also introduce new governance, security, and operational risks due to their autonomy, tool-use, and deep integration with critical systems [1],[4],[5]. Existing frameworks such as NIST AI RMF, ISO 27001, NIST SP 800-53, MAESTRO, MITRE ATLAS, and OWASP provide essential building blocks but leave gaps when applied directly to highly autonomous enterprise agents [2],[3],[9].

This paper proposes a control-driven lifecycle governance framework for enterprise agentic AI systems that extends these foundations with an threat explicit risk-tiering model, a RACI-mapped governance structure, a mandatory system-of-record for agentic use cases, and a -informed control-design process combining MAESTRO, MITRE ATLAS, and OWASP guidance [2],[3],[9],[13]. By specifying controls, approvals, and accountability across planning and

design, development, testing and validation, production deployment, operational monitoring, and decommissioning, the framework enables organizations to adopt agentic AI more responsibly while managing operational, security, and compliance risk.

The current evaluation is qualitative and based on a single illustrative case; future work will focus on quantitative assessments of the framework's impact on incident rates, near-misses, and time-to-deploy for agentic use cases across multiple organizations. Additional priorities include developing automation for the system-of-record, risk-tiering, and control-verification processes, and adapting the framework to domain-specific regulatory regimes such as healthcare, critical infrastructure, and cross-border financial services. These steps will help further substantiate and refine the proposed lifecycle governance model as agentic AI adoption scales [1],[13].

References

[1] NIST, "AI Risk Management Framework (AI RMF 1.0)," 2023.

[2] MITRE, "Adversarial Threat Landscape for Artificial-Intelligence Systems (ATLAS)," 2024.

[3] Cloud Security Alliance, "MAESTRO: Agentic AI Threat Modeling Framework," 2025.

[4] ISO/IEC, "ISO/IEC 27001: Information Security Management Systems," 2022.

[5] NIST, "Security and Privacy Controls for Information Systems and Organizations (SP 800-53)," 2020.

[6] Cyber Security Agency of Singapore (CSA), "Draft Addendum on Securing Agentic AI," 2024.

[7] NVIDIA, "Agentic Autonomy Levels and Security," 2024.

[8] HiveMQ, "Establishing Governance Frameworks for Agentic AI in Industrial Operations," 2024.

[9] OWASP, "Top 10 for Agentic Applications," 2026 (Draft).

[10] DAMA International, "Data Management Body of Knowledge (DMBOK)," 2017.

[11] Federal Reserve System, "Supervisory Guidance on Model Risk Management (SR 11-7)," 2011.

[12] Simon Willison, "The Lethal Trifecta," 2025.

[13] Palo Alto Networks, "What is Agentic AI Governance," Cyberpedia.

[14] C. Prakash, M. Lind, and A. Sisodia, "Agentic AI Governance and Lifecycle Management in Healthcare," *arXiv preprint arXiv:2601.15630v1 [cs.AI]*, Jan. 22, 2026.

[15] Checkmarx, "SCA vs. SAST vs. DAST," *Checkmarx Learning Center*, 2026.

[16] Promptfoo, "AI Red Teaming," *Promptfoo Documentation*, 2026.