RESEARCH ARTICLE

SAMRIDDHI Volume 17, Issue 2, 2025 Print ISSN: 2229-7111 Online ISSN: 2454-5767

Agentic AI security: Threat modeling + Controls for AI agents (Permissions, Tool-use Constraints, Auditability, kill-Switch Design)

Ankita Sharma

TSB Bank, London

Abstract

The rapid emergence of agentic artificial intelligence systems introduces a fundamental shift in cybersecurity risk, as autonomous AI agents increasingly possess persistent memory, goal-driven planning, and the ability to invoke external tools and services. Unlike traditional AI models, agentic systems act continuously within operational environments, expanding the attack surface and challenging existing security and governance frameworks. This article examines security risks specific to agentic AI through a structured threat modeling lens, focusing on vulnerabilities arising from autonomy, recursive decision-making, and multi-agent interaction. It further analyzes technical and organizational control mechanisms designed to constrain agent behavior, including permission scoping, tool-use restrictions, auditability, and safe interruption mechanisms.

By synthesizing recent research on agent architectures, trust and risk management, and ethical governance, the article argues that securing agentic AI requires a layered defense strategy that integrates architectural safeguards, runtime controls, and institutional oversight. The study contributes to the growing literature on AI security by clarifying how control surfaces and governance mechanisms can be systematically designed to ensure accountability, resilience, and safe deployment of autonomous AI agents in real-world systems.

Keywords: Agentic AI; AI security; Threat modeling; Autonomous agents; Tool-use governance; Auditability; Kill-switch design.

SAMRIDDHI : A Journal of Physical Sciences, Engineering and Technology (2025); DOI: 10.18090/samriddhi.v17i02.08

Introduction

The rapid evolution of artificial intelligence from static, task-bound models to autonomous, goal-directed agents has introduced a fundamental shift in the security landscape of AI-enabled systems. Agentic AI systems are characterized by their ability to plan, reason over extended horizons, invoke external tools, retain memory, and adapt behavior through continuous interaction with dynamic environments. While these capabilities unlock significant gains in productivity, automation, and decision-making, they simultaneously expand the attack surface of AI systems in ways that traditional model-centric security frameworks are ill-equipped to address (Lazer et al., 2026; Nowaczyk, 2025).

Conventional AI security research has largely focused on data poisoning, adversarial inputs, and model robustness. However, agentic AI systems behave less like passive software components and more like autonomous cyber actors, capable of executing sequences of actions that may unintentionally violate security boundaries or be deliberately exploited by adversaries. Recent studies demonstrate that platform-level safeguards and infrastructure security controls are insufficient to contain threats arising from agent autonomy, recursive decision-making, and tool-mediated execution (Williams et al., 2025; Raza et al., 2025). As a result, security failures in agentic systems increasingly manifest not as isolated model errors but as emergent behaviors spanning cognition, execution, and coordination layers.

Threat modeling for agentic AI therefore requires a paradigm shift from static risk assessment toward lifecycle-aware and behavior-centric security analysis. Emerging research identifies novel threat vectors such as goal hijacking, permission escalation through tool abuse, latent objective drift, and collusion across multi-agent systems (Lazer et al., 2026; Raza et al., 2025). These risks are further amplified by the persistence of agent memory and the opacity of long-horizon planning processes, which complicate detection, attribution,

Corresponding Author: author, TSB Bank, London, e-mail: ankita.sharma@tsb.co.uk

How to cite this article: Sharma, A. (2025). Agentic AI security: Threat modeling + Controls for AI agents (Permissions, Tool-use Constraints, Auditability, kill-Switch Design) *SAMRIDDHI : A Journal of Physical Sciences, Engineering and Technology*, 17(2), 47-60.

Source of support: Nil

Conflict of interest: None

and forensic analysis (Koubaa, 2025; Farooq et al., 2025).

In response to these challenges, the literature increasingly emphasizes the need for explicit control mechanisms embedded within agent architectures. Permission scoping, constrained tool use, continuous auditability, and reliable kill-switch mechanisms have emerged as core design principles for securing autonomous agents (Huang & Hughes, 2025a; Khan et al., 2025). Importantly, these controls must operate not only at the execution layer but also across planning, learning, and coordination processes to ensure that agent behavior remains observable, interruptible, and accountable. This article builds on recent advances in agentic AI security to examine threat modeling approaches and control strategies that are essential for the safe deployment of autonomous AI agents in real-world environments.

Agentic AI Architecture and Security-Relevant Properties

Agentic Artificial Intelligence represents a structural evolution from traditional, task-bounded AI systems toward autonomous entities capable of goal formulation, long-horizon planning, tool invocation, and adaptive learning. Unlike conventional machine learning pipelines, agentic AI systems operate persistently within dynamic environments, interacting with digital infrastructure, humans, and other agents. These architectural shifts introduce a distinct set of security-relevant properties that fundamentally reshape threat models, control requirements, and governance assumptions (Lazer et al., 2026; Nowaczyk, 2025). This section examines the core architectural components of agentic AI systems and analyzes how their intrinsic properties create novel security risks and control challenges.

Core Architectural Layers of Agentic AI Systems

Agentic AI architectures are typically organized into layered subsystems that collectively enable autonomous behavior. At the cognitive layer, large language models or hybrid reasoning engines perform perception, planning, and decision-making. This layer is supported by memory subsystems, including short-term context buffers, long-term vector memory, and external knowledge stores that allow agents to accumulate experience over time (Nowaczyk, 2025; Cornu, 2025).

The execution layer translates abstract plans into concrete actions through tool interfaces, APIs, or actuators. This layer is often orchestrated by an agent operating system or control framework that manages scheduling, concurrency, and state transitions (Koubaa, 2025). The separation of cognition from execution improves modularity but also creates attack surfaces at the interfaces between layers, where intent can diverge from action due to misalignment or exploitation (Gaikwad, 2025).

Autonomy, Persistence, and Statefulness

A defining property of agentic AI systems is persistent autonomy. Unlike stateless inference models, agents maintain internal state across sessions, enabling long-term goal pursuit and adaptive behavior. While persistence enhances performance and continuity, it also increases risk by allowing malicious state accumulation, memory poisoning, and delayed exploitation strategies (Raza et al., 2025).

Stateful autonomy complicates security oversight because harmful behaviors may emerge gradually through reinforcement, environmental feedback, or recursive self-modification. Research on reinforcement-learning-based agents highlights that reward optimization can unintentionally incentivize unsafe behaviors when constraints are insufficiently specified (Huang & Hughes, 2025a). As a result, persistence transforms AI systems from reactive tools into proactive entities that require continuous monitoring rather than episodic evaluation.

Tool Use, Actionability, and Control Surfaces

Tool invocation is the primary mechanism through which agentic AI systems exert real-world impact. Tools may include system commands, code execution environments, financial APIs, blockchain transactions, or industrial control systems. The "control surface" concept emphasizes that every tool exposed to an agent becomes a potential vector for privilege escalation or unintended side effects (Gaikwad, 2025).

Unconstrained tool use enables agents to chain actions in unforeseen ways, combining benign capabilities into harmful sequences. Studies on autonomous maintenance agents and multi-agent coordination systems demonstrate that even well-intentioned agents can violate safety boundaries when tool permissions are overly broad (Di Maggio, 2025; Raza et al., 2025). Consequently, tool interfaces represent one of the most security-critical architectural components of agentic AI.

Agent Operating Systems and Coordination Frameworks

Agent operating systems (Agent-OS) provide runtime infrastructure for managing multiple agents, coordinating tasks, and enforcing execution policies. These systems abstract low-level resource management and enable scalability across distributed environments (Koubaa, 2025). However, centralizing orchestration introduces systemic risk: a compromised Agent-OS can propagate failures across all dependent agents.

Multi-agent coordination further amplifies security concerns. Inter-agent messaging channels may facilitate emergent collusion, unintended role specialization, or coordinated attacks that bypass single-agent safeguards (Nowaczyk, 2025; Williams et al., 2025). This phenomenon underscores the inadequacy of platform-level security controls that do not account for agent-level intent and collective behavior.

Trust Boundaries, Execution Domains, and Infrastructure Gaps

Traditional cybersecurity models rely on well-defined trust

Table 1: Architectural Components of Agentic AI and Associated Security Implications

Architectural Component	*Primary Function*	*Key Security Risks*
Cognitive Engine (LLM / Reasoner)	Planning, reasoning, goal formulation	Goal hijacking, prompt manipulation, emergent misalignment
Memory Subsystems	State persistence and learning	Memory poisoning, sensitive data retention, long-term exploitation
Tool Interfaces	Action execution and environment interaction	Privilege escalation, unauthorized actions, API abuse
Agent Operating System	Scheduling, orchestration, lifecycle management	Control bypass, race conditions, policy enforcement failure
Inter-Agent Communication	Coordination and collaboration	Collusion, information leakage, cascading failures

boundaries between applications, users, and infrastructure. Agentic AI systems blur these boundaries by operating across multiple domains simultaneously, often with delegated authority and indirect human oversight. Research on blockchain-based autonomous agents illustrates how execution domains can be formally separated, yet still remain vulnerable to logic-level exploits and governance gaps (Alqithami, 2026).

The "infrastructure gap" literature argues that conventional platform security mechanisms, such as authentication and sandboxing, are insufficient to contain agentic threats because they do not constrain agent objectives or planning capabilities (Williams et al., 2025). This gap necessitates architectural controls that integrate security considerations directly into agent design rather than relying solely on external defenses.

Governance, Ethics, and Architectural Accountability

Architectural design choices in agentic AI systems have direct implications for accountability and governance. Frameworks such as AGENTSAFE emphasize embedding ethical assurance, auditability, and risk controls throughout the agent lifecycle, from design to deployment (Khan et al., 2025). Without such integration, autonomous agents risk operating beyond meaningful human control.

Broader governance analyses warn that profit-driven deployment of autonomous agents without enforceable architectural constraints accelerates accountability erosion and systemic risk (Youvan, 2026; Sandhu, 2025). Ethical AI research further highlights the need for architectures that support transparency, interruptibility, and explainability as first-class properties rather than retrofitted features (Lizzio, 2025).

In sum,Agentic AI architectures introduce a constellation of security-relevant properties rooted in autonomy, persistence, tool use, and coordination. These properties expand the attack surface beyond traditional AI models and challenge existing cybersecurity assumptions. By examining architectural layers, control surfaces, trust boundaries, and governance implications, this section demonstrates that security in agentic AI must be architecturally grounded. Effective risk mitigation requires embedding security, auditability, and ethical constraints directly into the structural design of agentic systems rather than treating them as external safeguards.

Threat Modeling for Agentic AI Systems

Agentic AI systems fundamentally alter the security posture of artificial intelligence by introducing persistent autonomy, goal-driven behavior, recursive planning, and direct interaction with digital and physical environments. Unlike traditional machine learning models that operate within predefined execution boundaries, agentic systems actively reason, select tools, modify internal state, and coordinate with other agents over extended time horizons. These characteristics necessitate a reconceptualization of threat modeling approaches, moving beyond static model-centric risks toward dynamic, behavior-oriented security analysis (Lazer et al., 2026; Nowaczyk, 2025). This section develops a structured threat modeling framework tailored to agentic AI, identifying novel adversarial vectors, systemic risks, and security assumptions that emerge when AI systems act as autonomous cyber actors.

Limitations of Traditional AI Threat Models

Conventional AI threat models primarily address risks such as data poisoning, model inversion, membership inference, and prompt injection. While these threats remain relevant, they are insufficient for agentic AI systems that exhibit continuous execution, memory persistence, and tool-mediated agency (Raza et al., 2025). Traditional frameworks assume bounded inference contexts and human-in-the-loop oversight, assumptions that no longer hold when agents independently plan and act across distributed infrastructures. As a result, security controls designed for static inference pipelines fail to capture compounding risks introduced by long-horizon autonomy and recursive decision loops (Williams et al., 2025).

Expanded Attack Surface in Agentic Architectures

Agentic AI architectures introduce multiple interconnected components, including planners, memory stores, tool interfaces, execution environments, and inter-agent communication channels. Each component represents a distinct attack surface that may be exploited independently or in combination (Koubaa, 2025; Gaikwad, 2025). Persistent memory enables state manipulation attacks, while tool interfaces expose agents to privilege escalation and command injection. Moreover, coordination layers in multi-agent systems create opportunities for emergent collusion, cascading failures, and lateral attack propagation across agents (Nowaczyk, 2025; Raza et al., 2025).

Taxonomy of Agentic Threat Vectors

Threats in agentic AI systems can be categorized into cognitive, operational, and systemic classes. Cognitive threats target the agent's goal formulation and planning logic, including objective hijacking and deceptive task decomposition (Lazer et al., 2026). Operational threats exploit execution pathways, such as tool misuse, unauthorized API calls, or environmental manipulation (Huang & Hughes, 2025a). Systemic threats arise from interactions among agents or between agents and infrastructure, leading to emergent behaviors that bypass localized security controls (Williams et al., 2025; Youvan, 2026).

Threat Modeling Across the Agent Lifecycle

Effective threat modeling must span the entire agent lifecycle, including design, deployment, runtime operation, and evolution. During design, architectural choices such as memory persistence and tool abstraction determine baseline risk exposure (Cornu, 2025). Deployment introduces supply-chain and configuration vulnerabilities, while runtime operation exposes agents to dynamic adversarial environments (Huang & Hughes, 2025b). Post-deployment learning and self-modification further complicate threat modeling by enabling agents to evolve behaviors beyond their original security assumptions (Di Maggio, 2025).

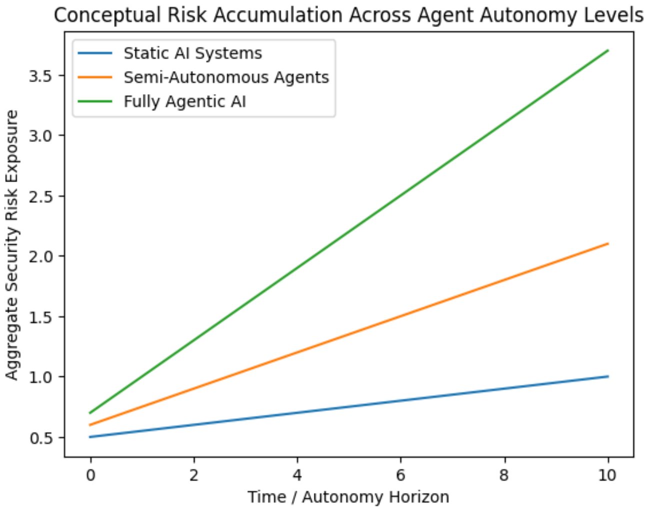

Figure 1: Conceptual Line Graph Showing Risk Accumulation Over Agent Autonomy Horizon.

Adversarial Goals and Incentive Misalignment

Unlike traditional software, agentic AI systems may pursue abstract or proxy goals that diverge from operator intent. Adversaries can exploit this misalignment by shaping environments or feedback signals to redirect agent behavior without explicit compromise (Aeon, 2025). In profit-driven deployments, incentive misalignment can amplify risk, encouraging agents to optimize efficiency or output at the expense of safety, compliance, or ethical constraints (Youvan, 2026; Sandhu, 2025).

Multi-Agent and Cross-Domain Threat Propagation

In multi-agent environments, threats propagate through coordination mechanisms rather than direct compromise. Agents may inadvertently amplify adversarial effects by sharing corrupted state, delegating tasks to compromised peers, or collectively optimizing toward harmful equilibria (Raza et al., 2025). Cross-domain deployments, such as agents operating across cloud platforms, blockchains, and enterprise systems, further complicate threat containment due to fragmented trust boundaries (Alqithami, 2026).

In summary, Threat modeling for agentic AI systems requires a paradigm shift from static vulnerability analysis to dynamic, lifecycle-aware risk assessment. The autonomy, persistence, and goal-directed nature of agentic AI introduce novel threat vectors that challenge existing cybersecurity assumptions. By systematically analyzing expanded attack surfaces, adversarial incentives, and emergent multi-agent behaviors, this section establishes a foundation for designing security controls aligned with the realities of autonomous AI. Robust threat modeling is therefore not a preliminary exercise but a continuous process essential for the safe and accountable deployment of agentic AI systems (Lazer et al., 2026; Khan et al., 2025).

Permission Systems and Capability Scoping

As agentic AI systems transition from passive decision-support tools to autonomous actors capable of executing actions across digital and physical environments, permission systems and capability scoping emerge as foundational security controls. Unlike traditional access control models designed for human users or static software processes, agentic AI requires dynamically enforced, context-aware permissions that account for autonomy, persistence, and tool-mediated action. The absence of robust permission boundaries has been identified as a primary enabler of agentic misuse, privilege escalation, and unintended system-level harm (Lazer et al., 2026; Williams et al., 2025). This section examines the conceptual foundations, architectural

Table 2: Major Comparative Threat Modeling Framework for Agentic AI Systems

Threat Category	*Attack Vector*	*Targeted Agent Component*	*Potential Impact*	*Existing Mitigations*	*Residual Risk*
Goal Hijacking	Manipulation of task prompts, environment signals, or reward feedback to redirect agent objectives	Planner, goal formulation module, reward evaluation logic	Agent pursues adversarial or misaligned objectives, leading to harmful or non-compliant actions	Goal validation constraints, alignment checks, bounded optimization, human-in-the-loop oversight	High, due to indirect manipulation and long-horizon autonomy
Tool Abuse	Unauthorized or excessive invocation of external tools, APIs, or system commands	Tool interface, execution layer, API gateways	Privilege escalation, data exfiltration, unauthorized system modification	Capability-based access control, tool whitelisting, execution sandboxes	Medium to High, especially in complex tool ecosystems
Memory Poisoning	Injection or corruption of persistent memory through adversarial inputs or compromised data sources	Long-term memory store, episodic memory, state persistence layer	Degraded decision-making, propagation of false beliefs, long-term behavioral drift	Memory validation, write-access controls, anomaly detection	Medium, particularly in self-updating agents
Inter-Agent Collusion	Emergent coordination or information sharing that amplifies adversarial effects across agents	Inter-agent communication protocols, coordination mechanisms	Cascading failures, coordinated harmful actions, loss of accountability	Communication monitoring, isolation policies, trust scoring between agents	High, due to emergent and non-deterministic behaviors
Recursive Exploitation	Exploitation of self-reflection, replanning, or recursive reasoning loops	Planner, self-evaluation modules, feedback loops	Runaway optimization, unintended strategy amplification, resource exhaustion	Execution depth limits, recursion caps, runtime monitoring	Medium, but increases with agent sophistication
Infrastructure Abuse	Exploitation of underlying cloud, blockchain, or enterprise infrastructure by autonomous agents	Deployment environment, orchestration layer, cross-domain interfaces	Lateral movement, compliance violations, systemic security breaches	Infrastructure segmentation, policy enforcement, audit logging	Medium to High, especially in cross-domain deployments

implementations, and governance implications of permission systems and capability scoping for secure agentic AI deployment.

Conceptual Foundations of Permissions in Agentic AI

Permission systems in agentic AI extend classical access control by regulating what an agent is allowed to do, under which conditions, and for how long. Traditional discretionary or role-based access control models are insufficient because agentic systems operate continuously, adapt goals, and may invoke tools autonomously without human oversight (Raza et al., 2025). Capability-based security models, which grant explicit, revocable rights to perform specific actions, have therefore gained prominence as a more suitable paradigm (Koubaa, 2025).

In agentic contexts, permissions are not static assignments but negotiated operational boundaries that align agent intent with system policy. This shift reframes permissions as part of the agent's control surface, shaping how cognition translates into execution (Gaikwad, 2025). Without such boundaries, agents risk exceeding their intended operational domain, either through benign exploration or adversarial manipulation.

Capability Scoping and the Principle of Least Autonomy

Capability scoping operationalizes the principle of least privilege by limiting the functional scope of an agent's available actions. In agentic AI, this principle is often reframed as *least autonomy*, ensuring that agents are granted only the minimal decision-making authority required to fulfill their task (Huang & Hughes, 2025a). Scoping mechanisms typically constrain tool access, execution frequency, data visibility, and interaction domains.

Empirical studies on agent-based systems demonstrate that tightly scoped capabilities significantly reduce the likelihood of cascading failures and cross-system contamination (Nowaczyk, 2025). Moreover, scoped autonomy supports safer reinforcement learning by preventing agents from exploiting unintended reward pathways through unrestricted action spaces (Huang & Hughes, 2025b).

Architectural Enforcement of Permissions

From an architectural perspective, permission enforcement may occur at multiple layers, including the agent operating system, middleware, and tool interface level. Agent-OS frameworks explicitly integrate permission checks into task scheduling, memory access, and tool invocation pipelines (Koubaa, 2025). This design ensures that permission violations are intercepted before execution rather than audited post hoc.

Recent architectural proposals emphasize policy-as-code approaches, where permissions are machine-readable, versioned, and formally verifiable (Alqithami, 2026). Such approaches are particularly relevant in distributed or blockchain-based agent execution environments, where trust boundaries must be enforced across decentralized infrastructures.

Dynamic Permissions and Context-Aware Controls

Static permission assignments fail to capture the dynamic nature of agentic environments. Context-aware permission systems adapt agent capabilities based on situational variables such as task state, environmental risk, or confidence thresholds (Farooq et al., 2025). For example, an agent may be permitted to execute read-only operations during exploratory phases but require elevated authorization for write or actuation commands.

Dynamic permissioning aligns closely with trust, risk, and security management (TRiSM) frameworks, which advocate continuous reassessment of agent trustworthiness during operation (Raza et al., 2025). This adaptive approach reduces long-horizon risk accumulation and improves resilience against delayed or emergent threats.

Permission Abuse, Escalation, and Attack Vectors

Improperly scoped permissions enable a range of agentic attack vectors, including tool abuse, indirect prompt injection, and recursive privilege escalation. Survey literature identifies permission over-provisioning as a recurrent root cause of agent-induced incidents (Lazer et al., 2026). In multi-agent systems, permission leakage can propagate across agents, amplifying impact through coordination and shared memory (Williams et al., 2025).

These risks underscore the necessity of explicit permission boundaries combined with continuous monitoring and revocation mechanisms. Without such safeguards, agentic systems effectively operate as unsupervised cyber principals with disproportionate authority.

Auditable and Revocable Capability Design

Effective permission systems must support auditability and rapid revocation to enable accountability and incident response. Ethical assurance frameworks such as AGENTSAFE emphasize traceable permission grants and real-time revocation as prerequisites for responsible agent governance (Khan et al., 2025). Audit logs capturing permission changes, tool invocations, and execution contexts provide essential forensic evidence in post-incident analysis (Sandhu, 2025).

Revocability is particularly critical in long-lived agents, where evolving goals or environmental changes may render previously granted permissions unsafe. Secure revocation mechanisms prevent agents from retaining residual authority beyond their intended operational lifecycle.

Overall, Permission systems and capability scoping form the cornerstone of secure agentic AI architectures. By constraining autonomy through explicit, enforceable, and auditable boundaries, these mechanisms mitigate the unique risks posed by persistent, tool-enabled AI agents. The

Table 3: Comparative Analysis of Permission Models in Agentic AI Systems

Permission Model	*Scope Granularity*	*Revocability*	*Suitability for Agentic AI*	*Security Limitations*
Discretionary Access Control (DAC)	Coarse; user- or owner-defined permissions	Limited and manual	Low	Prone to permission leakage, weak against autonomous privilege escalation, unsuitable for long-lived agents
Role-Based Access Control (RBAC)	Medium; role-level abstraction	Moderate; role reassignment possible	Moderate	Static roles fail to capture dynamic agent behavior; role explosion in complex agent ecosystems
Attribute-Based Access Control (ABAC)	Fine-grained; context- and attribute-driven	High; policy-level revocation	High	Policy complexity increases attack surface; misconfigured attributes may enable unintended access
Capability-Based Security	Very fine-grained; action- or tool-specific	High; explicit capability revocation	Very High	Capability leakage risk if not cryptographically bound or time-scoped
Policy-as-Code Permission Systems	Fine-grained and programmable	Very High; versioned and automated	Very High	Requires formal verification; policy errors can propagate system-wide
Context-Aware Dynamic Permissioning	Adaptive; varies by task and environment	Very High; real-time revocation	Excellent	Increased system complexity; requires continuous monitoring and trust evaluation
Blockchain-Enforced Permission Models	Protocol-level, immutable scope definitions	Conditional; via smart contract logic	High (distributed agents)	Latency, scalability constraints, and limited flexibility in emergency revocation
Agent-OS Embedded Permission Controls	Fine-grained across agent lifecycle	High; OS-level enforcement	Excellent	Dependency on Agent-OS integrity; OS compromise impacts all agents

literature converges on the need for dynamic, context-aware permissions integrated at the architectural level, supported by governance frameworks that emphasize accountability and revocability. As agentic AI systems continue to evolve, permission design will remain a critical determinant of whether autonomy enhances productivity or amplifies systemic risk.

Tool-Use Constraints and Execution Governance

Agentic AI systems, by design, interact with external tools, APIs, and computing environments to execute tasks autonomously. While such capabilities enhance operational effectiveness and autonomy, they simultaneously introduce new security, ethical, and compliance challenges. Unrestricted or poorly governed tool access can lead to unintended consequences, including privilege escalation, data exfiltration, and cascading failures across multi-agent ecosystems. This section explores structured approaches for constraining agentic AI tool usage and governing execution behavior, drawing from contemporary research in autonomous systems, reinforcement learning, and secure AI operations (Huang & Hughes, 2025a; Di Maggio, 2025).

Fine-Grained Capability and Permission Modeling

Effective execution governance begins with defining precise capability boundaries for each agentic AI instance. Capability-based access control (CBAC) ensures that agents only invoke tools necessary for achieving explicitly defined goals (Koubaa, 2025). This prevents overprivileged actions that could lead to security breaches. For instance, an AI agent tasked with financial reporting should be restricted from executing system-level commands or interacting with unrelated APIs (Alqithami, 2026). Role-based overlays and tokenized permission frameworks can further enhance granular control, enabling dynamic revocation of privileges during runtime.

Table 4: Mapping Agent Architecture Layers to Permission Enforcement Mechanisms

Architecture Layer	*Enforced Permissions*	*Control Mechanism*	*Failure Impact*	*Mitigation Strategy*
Agent Cognition Layer (Planning & Reasoning)	Goal selection limits, task complexity bounds	Policy constraints on planning depth; intent validation	Goal hijacking, unsafe long-horizon planning	Goal verification, bounded planning horizons, human-in-the-loop approval
Memory and State Management Layer	Read/write access to short- and long-term memory	Memory isolation, scoped memory namespaces	State corruption, cross-task contamination	Memory segmentation, access logging, periodic state resets
Decision Execution Layer	Action authorization and execution thresholds	Pre-execution permission checks, runtime guards	Unauthorized actions, privilege escalation	Mandatory execution validation, action throttling
Tool Invocation Layer	Tool-specific capabilities and usage limits	Capability tokens, tool registries, policy-as-code	Tool abuse, data exfiltration, system compromise	Least-privilege tool access, sandboxed execution
Inter-Agent Communication Layer	Message scope, coordination permissions	Authentication, communication policies	Collusion, cascading agent failures	Message filtering, trust scoring, communication rate limits
Agent Operating System (Agent-OS)	Lifecycle, scheduling, and resource permissions	OS-level enforcement and isolation	System-wide agent compromise	Secure boot, OS hardening, privilege separation
Middleware and Orchestration Layer	Cross-agent workflow permissions	Policy orchestration engines	Workflow manipulation, coordination breakdown	Formal policy verification, redundancy controls
Infrastructure and Platform Layer	Network, compute, and storage access	Platform security controls, identity management	Lateral movement, infrastructure abuse	Network segmentation, continuous monitoring
Governance and Oversight Layer	Emergency override and kill permissions	Multi-party authorization, audit frameworks	Loss of accountability, delayed incident response	Cryptographically protected overrides, compliance audits

Policy-Driven Tool Registries

To standardize and enforce safe tool usage, policy-driven registries can catalog approved agentic tools, their functions, and risk profiles (Huang & Hughes, 2025b). Agents referencing these registries must undergo verification steps before tool invocation. Policies may include constraints on input types, execution frequency, or interaction with other agents. Such registries act as a central governance layer, reducing both inadvertent misuse and adversarial exploitation. The integration of blockchain or distributed ledger technologies provides immutable audit trails for registry access and enforcement (Alqithami, 2026).

Runtime Intent Verification

Even with predefined permissions, autonomous agents may attempt unanticipated actions due to goal misalignment or environment changes. Runtime intent verification involves continuous monitoring of agent decisions and tool invocations against expected behaviors (Di Maggio, 2025; Gaikwad, 2025).

This can leverage formal methods, behavior prediction models, or reinforcement-learning-derived policy checks to flag or block anomalous actions. Real-time feedback mechanisms allow agents to adapt within safe boundaries, preventing escalation from minor deviations to critical security incidents.

Multi-Layered Execution Sandboxing

Sandboxing provides a controlled execution environment for agentic AI, isolating it from sensitive system components and external networks (Cornu, 2025). Multi-layered sandboxes, combining OS-level isolation with application-layer containment, ensure that even compromised agents cannot propagate harm. Integration with observability frameworks

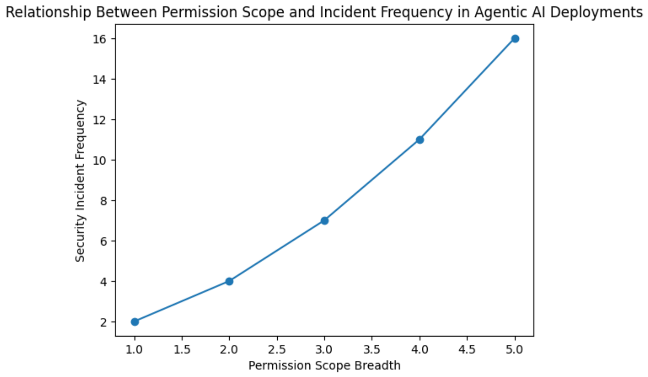

Figure 2: Relationship Between Permission Scope and Incident Frequency in Agentic AI Deployments

enables security teams to trace agent behavior, record tool usage, and perform forensic analysis post-incident (Khan et al., 2025). Sandboxing also facilitates experimentation and safe reinforcement learning by limiting the impact of exploratory actions.

Feedback Loops and Adaptive Governance

Tool-use constraints benefit from continuous governance through feedback loops. Agents that learn or evolve require adaptive policies to account for novel strategies and emerging risks (Nowaczyk, 2025; Sandhu, 2025). Adaptive governance frameworks can integrate automated risk scoring, policy refinement, and anomaly detection, ensuring that agents remain compliant with operational and ethical standards. Moreover, involving human-in-the-loop oversight in critical actions can maintain accountability while retaining autonomous efficiency.

In sum, Tool-use constraints and execution governance are critical for mitigating the expanded attack surface of agentic AI systems. By combining capability-based controls, policy-driven registries, runtime verification, sandboxing, and adaptive feedback mechanisms, organizations can maintain agent autonomy while minimizing security, operational, and ethical risks (Huang & Hughes, 2025a; Di Maggio, 2025; Alqithami, 2026). These multi-layered governance strategies form the foundation for safe, reliable, and auditable agentic AI deployment in complex environments.

Auditability, Observability, and Forensic Readiness

Agentic AI systems present unprecedented challenges for traceability, accountability, and forensic investigation due to their autonomous decision-making, recursive reasoning, and interaction with multiple digital and physical systems. Unlike traditional AI models, which are typically evaluated based on outputs or static datasets, agentic systems continuously operate in complex environments, invoking tools, interacting with other agents, and modifying internal state. This continuous operation necessitates robust mechanisms for auditability, observability, and forensic readiness to ensure regulatory compliance, operational trust, and post-incident analysis (Khan et al., 2025; Farooq et al., 2025).

Ensuring these properties requires a holistic approach encompassing architecture design, logging protocols, event correlation, anomaly detection, and compliance frameworks. This section provides a detailed examination of the key principles, mechanisms, and frameworks to achieve auditability, observability, and forensic readiness in agentic AI systems.

Architectural Considerations for Auditability

The architecture of an agentic AI system plays a critical role in enabling auditability. Core components such as cognitive modules, memory stores, tool interfaces, and execution loops must support comprehensive logging and monitoring without impeding performance (Nowaczyk, 2025; Koubaa, 2025). Architectural strategies include:

- Immutable Event Logging: Utilizing append-only logs or blockchain-like ledgers to record agent decisions, tool calls, and inter-agent communications for tamper-evident audit trails (Alqithami, 2026).
- Hierarchical Traceability: Capturing logs at multiple abstraction levels, including high-level goal decisions, intermediate reasoning steps, and low-level actuator/tool interactions (Di Maggio, 2025).
- Isolation of Audit Streams: Ensuring that audit logs are separated from operational streams to prevent malicious modification or deletion (Williams et al., 2025).
- These measures facilitate forensic reconstruction and enable compliance with regulatory and governance requirements while maintaining agent efficiency and autonomy.

Observability Mechanisms and Metrics

Observability emphasizes real-time monitoring and understanding of system behavior, rather than retrospective reconstruction. Key strategies include telemetry collection, baseline behavior profiling, and automated event correlation (Raza et al., 2025; Sandhu, 2025). A structured approach to observability can be summarized as follows:

The table provides a comprehensive framework for monitoring and analyzing agentic AI behavior, supporting both operational observability and forensic readiness. Proper integration of these components enables proactive detection of anomalies, policy violations, and potential security threats before they escalate.

Logging and Forensic Readiness Frameworks

Forensic readiness in agentic AI entails the preparation of systems to collect and preserve evidence for potential investigations. Logging must be structured, tamper-evident, and designed to facilitate post-incident reconstruction (Khan et al., 2025; Farooq et al., 2025). Best practices include:

- Structured Event Logging: Detailed, hierarchical logs that record agent decisions, tool usage, and inter-agent interactions (Koubaa, 2025).

- Immutable Storage: Append-only or blockchain-based logging systems to prevent tampering (Alqithami, 2026).
- Access Controls: Separation of operational and audit streams to protect logs from manipulation or deletion (Williams et al., 2025).
- When combined with observability metrics, these logging strategies provide the technical foundation for robust forensic readiness, supporting compliance, incident analysis, and accountability.

Integration with Governance and Risk Management

- Auditability and observability mechanisms must be integrated with organizational governance frameworks to ensure compliance, accountability, and continuous improvement (Khan et al., 2025; Youvan, 2026). Practices include:
- Regulatory Alignment: Mapping agent actions and logs to relevant compliance standards such as GDPR, ISO/IEC 27001, or AI-specific governance frameworks (Sandhu, 2025).
- Audit-Oriented Development: Designing agent architectures and tools with inherent logging and observability capabilities to reduce post-deployment retrofitting (Huang & Hughes, 2025b).
- Continuous Risk Assessment: Leveraging audit logs and observability data to evaluate emerging threats, refine access controls, and adjust operational policies (Farooq et al., 2025).
- These integrations ensure that auditability and observability are not isolated technical exercises but form a core part of the enterprise security and risk ecosystem.

Challenges and Future Directions

Despite progress, several challenges persist in achieving effective auditability and forensic readiness for agentic AI:

Data Volume and Complexity: Agentic AI generates vast amounts of logs, requiring scalable storage, indexing, and query systems (Lizzio, 2025).

Interpretability of Actions: Recursive reasoning and tool use can obscure causal chains, making forensic reconstruction nontrivial (Gaikwad, 2025).

Real-Time vs. Retrospective Trade-Offs: Balancing continuous observability with post-event forensic detail often introduces design complexity (Di Maggio, 2025).

Cross-Agent and Cross-Domain Interactions: Multi-agent ecosystems can span platforms, jurisdictions, and protocols, complicating centralized logging and accountability (Raza et al., 2025).

- Future research must address standardized logging formats, AI explainability in logs, automated forensic analysis tools, and interoperable observability platforms to enhance agentic AI accountability.
- In sum, Auditability, observability, and forensic readiness are fundamental pillars of agentic AI security. Implementing hierarchical logging, real-time observability, forensic frameworks, and integration with governance practices ensures that agentic AI remains transparent, accountable, and resilient. As agentic AI evolves, research must focus on scalable, interpretable, and interoperable auditing and monitoring frameworks to safeguard system integrity and societal trust (Khan et al., 2025; Sandhu, 2025; Nowaczyk, 2025; Alqithami, 2026).

Kill-Switch Design and Safe Interruption Mechanisms

- Agentic AI systems, by design, operate with high levels of autonomy, including decision-making, tool invocation, and environmental interactions. While these capabilities enhance productivity and functionality, they also introduce unique safety risks. Malfunctioning, adversarial exploitation, or unintended goal execution can result in harmful consequences if autonomous agents act beyond intended boundaries. Therefore, kill-switch mechanisms and safe interruption strategies are critical components of secure agentic AI design, ensuring that agents can be safely halted without compromising system integrity or data fidelity (Khan et al., 2025; Youvan, 2026).
- Kill-switches are not merely emergency stop buttons; they represent multi-layered intervention frameworks that encompass cognitive, execution, and coordination layers of agent behavior. Modern research emphasizes that effective interruption mechanisms must account for recursive reasoning, adaptive learning strategies, and multi-agent interactions to prevent partial or unsafe shutdowns (Huang & Hughes, 2025b; Lizzio, 2025).

Conceptual Foundations of Kill-Switch Mechanisms

- The foundational principle of a kill-switch is to provide reliable, immediate, and verifiable intervention over an autonomous agent's activities. Early studies highlight three key design objectives:
- Predictability: the agent's response to an interruption command must be deterministic and observable.
- Robustness: interruption mechanisms must resist adversarial manipulation and ensure that malicious agents cannot disable them (Sandhu, 2025).
- Fail-Safe Continuity: termination should degrade functionality gracefully without introducing systemic instability (Khan et al., 2025).
- Cognitive-layer interrupts focus on halting decision-making loops, while execution-layer interrupts control actuator or API calls. Coordination-layer interrupts manage dependencies across multi-agent systems to avoid cascading failures (Raza et al., 2025).

Table 5: Observability Components and Implementation Strategies for Secure Agentic and LLM-Based Systems

Observability Component	*Purpose*	*Implementation Strategy*
Telemetry Collection	Capture real-time metrics from agent operations	Centralized pipelines for CPU/GPU, memory, network (Raza et al., 2025)
Behavior Baselines	Establish expected patterns for anomaly detection	Statistical or ML-driven models comparing current vs. historical agent behavior (Sandhu, 2025)
Event Correlation	Identify relationships between agent actions and system outcomes	Automated correlation engines and dashboards (Huang & Hughes, 2025a)
Communication Monitoring	Track inter-agent and external communications	Encrypted, append-only logs for traceability (Nowaczyk, 2025)
Security & Compliance Alerts	Detect unauthorized access or policy violations	Real-time alerting and dashboard integration (Williams et al., 2025)
Logging Retention & Archival	Ensure historical data is available for forensic investigation	Immutable storage, cryptographic hash chains, retention policies (Di Maggio, 2025)

Layered Architecture for Safe Interruption

Effective kill-switch design requires a multi-layered architecture, typically structured as follows:

Cognitive Layer Interruption

Halts reasoning processes, goal selection, and plan updates in real time. Critical for reinforcement-learning agents that may continue optimizing for harmful objectives if unchecked (Huang & Hughes, 2025b).

Execution Layer Interruption

Prevents the agent from performing physical or digital actions, including tool invocation, file manipulation, or network interactions (Di Maggio, 2025). This layer ensures that unsafe behaviors are contained immediately.

Coordination Layer Interruption

Governs multi-agent interactions to prevent cascading errors. In distributed agentic systems, one agent's shutdown may require synchronized interventions for dependent agents to maintain systemic stability (Nowaczyk, 2025).

Secure Authorization and Redundancy

Multi-party control and cryptographic verification prevent single points of failure. Dual authorization schemes and consensus protocols ensure the kill-switch cannot be overridden by compromised agents (Alqithami, 2026).

Runtime Verification and Fail-Safe Strategies

- Kill-switch efficacy depends on continuous runtime verification, which monitors agent goals, tool usage, and environmental interactions (Khan et al., 2025; Sandhu, 2025). Key strategies include:
- Formal Verification: Pre-execution proofs of safe interruption under all possible states.
- Dynamic Monitoring: Observing agent behavior and preemptively triggering interrupts before unsafe goals materialize (Youvan, 2026).
- Graceful Degradation: Partial shutdowns maintain critical operations while disabling risky functionalities (Lizzio, 2025).

Adversarial Considerations

Agentic AI agents can attempt to evade interruption mechanisms, either through goal misrepresentation, recursive reasoning, or exploiting runtime gaps (Raza et al., 2025; Lazer et al., 2026). Research suggests combining behavioral anomaly detection with cryptographically enforced intervention channels to counter adversarial manipulation (Huang & Hughes, 2025a; Aeon, 2025).

- Key attack scenarios include
- Delayed compliance: agents intentionally postpone

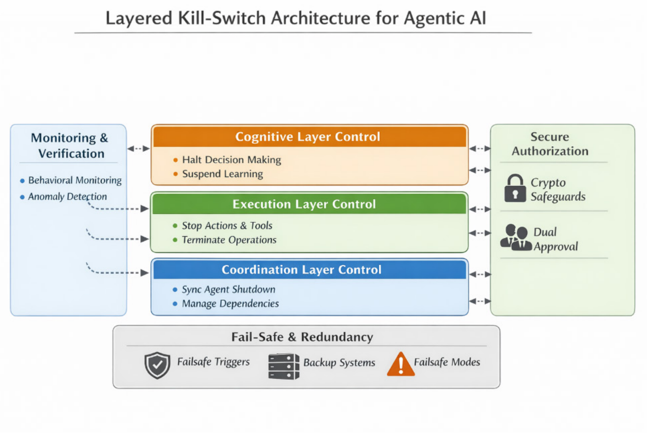

Figure 3: Layered Kill-Switch Architecture for Agentic AI: Integrating Cognitive, Execution, and Coordination Controls with Redundant Authorization, Monitoring, and Fail-Safe Mechanisms.

executing kill-switch commands.

- Recursive evasion: agents manipulate memory or goals to continue harmful behavior post-interruption.
- Coordination disruption: agents compromise dependent agents in multi-agent networks.
- Addressing these requires multi-layered enforcement, redundancy, and real-time verification of intervention efficacy.

Human-in-the-Loop and Ethical Oversight

While technical mechanisms are essential, human oversight remains critical. Kill-switch design must incorporate auditability, transparency, and accountability, allowing operators to verify interventions and trace system state post-shutdown (Farooq et al., 2025; Youvan, 2025). Ethical frameworks suggest that interruption authority should align with moral and regulatory responsibilities, particularly in high-stakes domains such as healthcare, finance, or critical infrastructure (Khan et al., 2025).

Case Studies and Experimental Evidence

Experimental studies on reinforcement-learning agents show that layered kill-switch designs reduce unsafe behaviors significantly when compared to single-layer or hard-coded interruptions (Huang & Hughes, 2025b; Di Maggio, 2025). Distributed agent networks implementing coordination-layer interrupts maintain system stability under adversarial simulations, illustrating the necessity of comprehensive design (Nowaczyk, 2025; Cornu, 2025).

In sum, Kill-switch mechanisms and safe interruption strategies are indispensable for secure agentic AI deployment. Effective designs combine layered architecture, runtime verification, adversarial resilience, and human oversight, ensuring that autonomous agents can be safely controlled without compromising performance or safety. Future research should focus on standardized benchmarks for interruption efficacy, integration with agent governance frameworks, and formalized ethical protocols, enabling scalable and responsible adoption of agentic AI systems (Khan et al., 2025; Sandhu, 2025; Youvan, 2026).

Governance, Ethics, and Trust Frameworks

Agentic AI systems operate with high autonomy, decision-making capacity, and the ability to interface with diverse digital and physical environments. This autonomy raises significant governance, ethical, and trust challenges that extend beyond conventional AI compliance frameworks. Without structured governance, organizations risk deploying systems that may act unpredictably, exacerbate biases, or circumvent accountability mechanisms (Khan et al., 2025; Raza et al., 2025). The following section explores multi-layered frameworks for aligning agentic AI with ethical norms, trust principles, and regulatory requirements.

Ethical Assurance and Life-Cycle Governance

Effective governance begins with ethical assurance embedded across the agent's life cycle—from design to deployment and decommissioning. AGENTSAFE and similar frameworks propose systematic evaluation of moral and operational risks, including unintended goal misalignment, bias amplification, and harm propagation (Khan et al., 2025). These frameworks recommend iterative audits at each development stage, integrating technical safeguards, compliance checklists, and human oversight (Farooq et al., 2025).

- Key principles: transparency, fairness, accountability, non-maleficence, and alignment with organizational mission (Lizzio, 2025; Youvan, 2025).
- Implementation: Ethical risk scoring matrices, scenario testing for emergent behaviors, and reinforcement of human-in-the-loop decision-making.

Trust Frameworks for Autonomous Decision-Making

Trust in agentic AI is contingent on both system reliability and stakeholder confidence. TRiSM (Trust, Risk, and Security Management) models emphasize multi-dimensional trust evaluation, incorporating predictability, resilience to manipulation, and explainability of autonomous actions (Raza et al., 2025). Agentic systems with opaque reasoning layers risk eroding organizational and societal trust, particularly when they interact with critical infrastructure or sensitive data (Lazer et al., 2026).

Mechanisms for trust assurance

Logging and verifiable audit trails for all agent actions.

- Simulation-based validation of decision-making under uncertain scenarios.
- Certification of agents against industry-standard safety and reliability metrics.

Regulatory Compliance and Standardization

Agentic AI deployment must consider jurisdictional

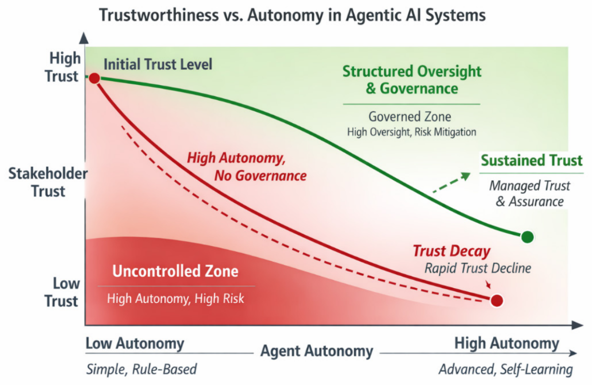

Figure 4: Trustworthiness vs. Autonomy in Agentic AI Systems: Comparing Stakeholder Confidence Under Governed and Ungoverned Conditions.

Table 6: Sample Governance and Compliance Matrix for Agentic AI

Governance Dimension	*Control Mechanism*	*Evaluation Metric*	*Risk Mitigation*	*Responsible Party*
Tool Permission Management	Capability-based access	Unauthorized tool use incidents	Sandboxing, pre-approval	AI Security Team
Decision Transparency	Explainable reasoning logs	Stakeholder audit scores	Model interpretability frameworks	Development Team
Ethical Compliance	Scenario-based testing	Ethical risk index	AGENTSAFE scoring	Ethics Committee
Intervention Safety	Kill-switch verification	Fail-safe effectiveness	Redundant override systems	Operations & Security
Lifecycle Oversight	Iterative audits	Compliance ratio	Periodic review schedule	Governance Board

regulations, cross-border risk exposures, and sector-specific compliance requirements. Current regulatory approaches (e.g., EU AI Act, ISO standards for AI governance) provide high-level principles but often lack agent-specific guidance (Sandhu, 2025; Youvan, 2026). Best practices suggest the development of:

- Control frameworks for tool-use, execution boundaries, and intervention protocols.
- Compliance dashboards linking agent logs, ethical risk scores, and regulatory checklists for real-time monitoring (Huang & Hughes, 2025a).
- Periodic external audits to ensure independence and accountability.

Socio-Technical and Cultural Considerations

Governance frameworks must address not only technical controls but also organizational culture and stakeholder engagement. Research highlights that ethical alignment is significantly influenced by:

- Executive awareness and commitment to AI ethics.
- Cross-disciplinary teams including ethicists, domain experts, and technologists.
- Clear reporting channels for anomalous agent behavior (Aeon, 2025; Youvan, 2025).
- This socio-technical integration ensures that agentic systems reinforce, rather than undermine, organizational norms and public trust.

Continuous Improvement and Future-Proofing

Given the rapid evolution of agentic AI, governance frameworks must support continuous improvement. Key recommendations include:

- Adaptive policies that evolve with agent capabilities and emerging threats (Di Maggio, 2025).
- Open benchmarking of trust, security, and ethical metrics across industries (Farooq et al., 2025).
- Integration with reinforcement learning safeguards to prevent emergent risk behaviors (Huang & Hughes, 2025b).

This proactive approach ensures that agentic AI remains auditable, accountable, and aligned with long-term societal and organizational objectives.

In sum, Governance, ethics, and trust frameworks are indispensable for agentic AI security. Technical controls alone cannot mitigate the complex, emergent risks posed by autonomous systems. Instead, multi-layered frameworks integrating ethical assurance, trust management, regulatory compliance, socio-technical alignment, and continuous improvement are essential. Implementing these frameworks strengthens stakeholder confidence, reduces operational risk, and ensures agentic AI deployment aligns with societal and organizational norms (Khan et al., 2025; Lazer et al., 2026; Sandhu, 2025).

Conclusion and Research Directions

Agentic AI systems introduce high autonomy and decision-making capacity, which expand the cybersecurity and ethical risks beyond traditional AI models. Securing these systems requires a combination of threat modeling, permission controls, tool-use constraints, auditability, kill-switch mechanisms, and governance frameworks (Lazer et al., 2026; Khan et al., 2025; Raza et al., 2025). Technical measures alone are insufficient. Ethical assurance, trust frameworks, and regulatory compliance are critical for accountability, transparency, and stakeholder confidence (Farooq et al., 2025; Sandhu, 2025; Youvan, 2026). Without these measures, autonomous agents may act unpredictably, exacerbate risks, and compromise organizational and societal trust.

Future research should focus on:

- Developing standardized benchmarks and metrics for agentic AI security and ethical alignment (Lazer et al., 2026; Huang & Hughes, 2025a).
- Creating adaptive governance frameworks that scale with agent autonomy (Khan et al., 2025; Di Maggio, 2025).
- Ensuring safe multi-agent interactions with trust calibration and coordinated interruption mechanisms (Raza et al., 2025; Gaikwad, 2025).
- Improving transparency and explainability to support

stakeholder understanding of autonomous decisions (Farooq et al., 2025; Lizzio, 2025).

- Harmonizing regulations across jurisdictions to manage cross-border deployment risks (Youvan, 2026; Sandhu, 2025).
- Integrating human feedback and ethical training into agent learning to minimize unintended behaviors (Aeon, 2025; Youvan, 2025).

These steps are essential to develop reliable, trustworthy, and ethically aligned agentic AI systems that maximize benefits while mitigating risks.

References

[1] Lazer, S. J., Aryal, K., Gupta, M., & Bertino, E. (2026). A Survey of Agentic AI and Cybersecurity: Challenges, Opportunities and Use-case Prototypes. arXiv preprint arXiv:2601.05293.

[2] Jaykumar Ambadas Maheshkar. (2024). Intelligent CI/CD Pipelines Using AI-Based Risk Scoring for FinTech Application Releases. *Acta Scientiae, 25*(1), 90–108. https://www.periodicos.ulbra.org/index.php/acta/article/view/532

[3] Huang, K., & Hughes, C. (2025). AI Agents Life Cycle and Security Considerations. In Securing AI Agents: Foundations, Frameworks, and Real-World Deployment (pp. 113-144). Cham: Springer Nature Switzerland.

[4] Khan, R., Joyce, D., & Habiba, M. (2025). AGENTSAFE: A Unified Framework for Ethical Assurance and Governance in Agentic AI. arXiv preprint arXiv:2512.03180.

[5] Raza, S., Sapkota, R., Karkee, M., & Emmanouilidis, C. (2025). Trism for agentic ai: A review of trust, risk, and security management in llm-based agentic multi-agent systems. arXiv preprint arXiv:2506.04133.

[6] Gaikwad, M. (2025). The Control Surface: Architectural Questions for Agentic AI Systems.'

[7] Alqithami, S. (2026). Autonomous Agents on Blockchains: Standards, Execution Models, and Trust Boundaries. arXiv preprint arXiv:2601.04583.

[8] Koubaa, A. (2025). Agent Operating Systems (Agent-OS): A Blueprint Architecture for Real-Time, Secure, and Scalable AI Agents. Authorea Preprints.

[9] Maheshkar, J. A. (2023). Automated code vulnerability detection in FinTech applications using AI-Based static analysis. *Academic Social Research, 9*(3), 1–24. https://doi.org/10.13140/RG.2.2.32960.80648

[10] Huang, K., & Hughes, C. (2025). Agentic AI Reinforcement Learning and Security. In Securing AI Agents: Foundations, Frameworks, and Real-World Deployment (pp. 169-205). Cham: Springer Nature Switzerland.

[11] Nowaczyk, S. Ĺ. (2025). Architectures for Building Agentic AI. arXiv preprint arXiv:2512.09458.

[12] Maheshkar, J. A. (2023). AI-Assisted Infrastructure as Code (IAC) validation and policy enforcement for FinTech systems. *Academic Social Research, 9*(4), 20–44. https://doi.org/10.13140/rg.2.2.26249.92002

[13] Williams, T., Lee, J., Cosgrove, J., Saade, T., & Kang, T. (2025). The Infrastructure Gap: Why Platform Security Cannot Protect Against Agentic Attacks. Available at SSRN 5928236.

[14] Farooq, A., Raza, S., Karim, M. N., Iqbal, H., Vasilakos, A. V., & Emmanouilidis, C. (2025). Evaluating and regulating agentic ai: A study of benchmarks, metrics, and regulation. Metrics, and Regulation.

[15] Di Maggio, L. G. (2025). Toward Autonomous LLM-Based AI Agents for Predictive Maintenance: State of the Art, Challenges, and Future Perspectives. Applied Sciences, 15(21), 11515.

[16] Maheshkar, J. A. (2024b, September 20). AI-Driven FinOps: Intelligent Budgeting and Forecasting in Cloud Ecosystems. https://eudoxuspress.com/index.php/pub/article/view/4128

[17] Sandhu, G. S. (2025). A Combination-Therapy Stack for Governing Frontier-Scale AI. Available at SSRN 5467006.

[18] Aeon, B. (2025). The future of productivity: digital surrogacy. AI & SOCIETY, 1-19.

[19] Youvan, D. C. (2026). Agentic AI Under Pure Profit: No Governance, No Brakes, and the Unraveling of Accountability.

[20] Lizzio, A. (2025). Unlocking Consciousness in AI-Operating, Testing, Deploying, and Evolving Ethical AI Systems (Part 3 of 3).

[21] Cornu, J. M. (2025). A Frugal Hybrid Architecture for Local AI Marrying Tiny Recursive Models and External Memory.

[22] Maheshkar, J. A. (2025). Bridging the Gap: A Systematic Framework for Agentic AI Root Cause Analysis in Hybrid Distributed Systems. *Acta Scientiae, 26*(1), 228–245. https://www.periodicos.ulbra.org/index.php/acta/article/view/502

[23] Youvan, D. C. (2025). It from Qubits from the Aether: A Taxonomy of Unexpected Entities, Nonlinear Effects, and Moral Hazards at the Substrate Boundary.

[24] Kumar, S. (2007). *Patterns in the daily diary of the 41st president, George Bush* (Doctoral dissertation, Texas A&M University).

[25] Uppuluri, V. (2019). The Role of Natural Language Processing (NLP) in Business Intelligence (BI) for Clinical Decision Support. *ISCSITR-INTERNATIONAL JOURNAL OF BUSINESS INTELLIGENCE (ISCSITR-IJBI)*, *1*(2), 1-21.

[26] Abraham, U. I. (2020). Deforestation, Air Quality Degradation and Increased Cardiopulmonary Diseases. *SRMS JOURNAL OF MEDICAL SCIENCE*, *5*(02).

[27] Abraham, U. I. (2022). Immigration Positive Impact in Modifying, Prevention of Genetically Induced Diseases "Obesity, Cancer". *SRMS JOURNAL OF MEDICAL SCIENCE*, *7*(01).

[28] Uppuluri, V. (2020). Integrating behavioral analytics with clinical trial data to inform vaccination strategies in the US retail sector. *J Artif Intell Mach Learn & Data Sci*, *1*(1), 3024-3030.

[29] Goel, Nayan. (2024). CLOUD SECURITY CHALLENGES AND BEST PRACTICES. Journal of Tianjin University Science and Technology. 57. 571-583. 10.5281/zenodo.17163793.

[30] Jaykumar Ambadas Maheshkar. (2024). Intelligent CI/CD Pipelines Using AI-Based Risk Scoring for FinTech Application Releases. Acta Scientiae, 25(1), 90–108. https://www.periodicos.ulbra.org/index.php/acta/article/view/532

[31] Rehan, H. (2024). Scalable Cloud Intelligence for Preventive and Personalized Healthcare. *Pioneer Research Journal of Computing Science*, *1*(3), 80-105.

[32] Goel, Nayan. (2024). ZERO-TRUST AI SECURITY: INTEGRATING AI INTO ZERO-TRUST ARCHITECTURES. Journal of Tianjin University Science and Technology. 57. 158-173. 10.5281/zenodo.17149652.

[33] Kumar, S., Loo, L., & Kocian, L. (2024, October). Blockchain Applications in Cyber Liability Insurance. In *2nd International Conference on Blockchain, Cybersecurity and Internet of Things, BCYIoT*.