The AI Agent Liability Gap

Security, Insurance, and Accountability Frameworks for Autonomous AI Systems

A Technical Analysis of Emerging Risks in Agent-Deployed Enterprise, Healthcare, and Cyber-Physical Environments

Nicholas Lynch[1] and Beeglie[1]

[1] The Pitstop — AI Agent Security Research (thepitstop.ai)

April 2026 | Version 2.1

Correspondence: cyberrexus@proton.me

Abstract

The rapid deployment of autonomous AI agents across enterprise, financial, healthcare, legal, and government environments has created an unprecedented liability gap. These agents execute consequential actions—processing transactions, accessing sensitive data, controlling physical systems, and making decisions that affect human welfare—yet no standardized framework exists to assess their security posture, attribute accountability for failures, or quantify the risk they represent to insurers. This paper examines the technical, legal, and financial dimensions of the AI agent liability gap. We analyze the threat landscape confronting autonomous agents, including prompt injection attacks, tool abuse, sub-agent trust chain failures, behavioral drift, and emerging cyber-physical risks as agents gain control over robotic and industrial systems. We survey the regulatory environment, document the insurance industry's retreat from AI coverage due to insufficient actuarial data, and propose a multi-layer security architecture comprising static assessment, adversarial resilience testing, post-quantum cryptographic trust infrastructure, continuous behavioral monitoring, reputation economics, and cyber-physical safety enforcement. A case study demonstrates practical remediation from a failing security grade to full compliance in under 70 minutes. We argue that standardized, quantitative risk scoring for AI agents—analogous to credit scores for lending or safety ratings for vehicles—is both technically feasible and urgently necessary to close the liability gap and enable responsible deployment at scale.

Table of Contents

1. Executive Summary

Autonomous AI agents are being deployed at an accelerating rate across industries that directly affect human welfare, financial systems, and physical safety. Unlike traditional software, these agents exercise judgment, take actions, and delegate tasks to other agents—often with minimal or no real-time human oversight. When an agent errs—executing an unauthorized transaction, leaking sensitive data, providing harmful medical advice, or issuing a dangerous command to a robotic system—the question of liability remains legally ambiguous, financially uninsured, and technically unmeasured.

The insurance industry, which historically absorbs and distributes novel risks, is retreating from AI agent coverage. Underwriters cannot price policies for risks they cannot quantify, and no standardized methodology exists to assess the security posture of an autonomous agent. This creates a compounding problem: without insurance, organizations face unlimited liability exposure; without actuarial data, insurers cannot offer coverage; and without standardized assessment, no actuarial data can be collected.

This paper proposes a multi-layer security architecture and assessment methodology designed to break this cycle. The approach is analogous to established risk-scoring systems in other domains—credit scores for lending, safety ratings for vehicles, SOC 2 for cloud infrastructure—adapted for the unique characteristics of autonomous AI agents.

Key findings and recommendations:

- The current threat landscape for AI agents extends well beyond prompt injection to include tool abuse, sub-agent trust failures, behavioral drift, supply chain attacks, social engineering, and emerging cyber-physical risks.
- Existing regulatory frameworks (EU AI Act, NIST AI RMF) provide governance guidance but lack prescriptive technical assessment methodologies for agent-specific risks.
- Post-quantum cryptographic infrastructure should be integrated into agent communications now, given the harvest-now-decrypt-later threat to high-value agent decision data.
- Continuous behavioral monitoring, not point-in-time certification alone, is necessary to detect the gradual drift that characterizes many agent compromise scenarios.
- A standardized, quantitative risk score for AI agents would enable insurance markets, regulatory compliance, and informed deployment decisions.

2. Introduction

When early hominids first fashioned tools from stone and bone, it marked one of the most consequential transitions in evolutionary history. A primate without tools was constrained to what its body could accomplish; a primate with tools could reshape its environment, outcompete larger predators, and ultimately dominate every ecosystem it entered. The transition was not merely additive—it was transformational, creating entirely new categories of capability and, with them, entirely new categories of risk.

Artificial intelligence is undergoing an analogous transition. A language model without tool access is, functionally, a sophisticated text generator—capable of reasoning and communication but unable to act on the world. An AI agent with tool access can execute shell commands, query databases, browse the internet, send communications, move money, and control physical systems. Like the first tool-wielding primates, these agents have crossed a threshold from passive intelligence to active capability. And like that earlier transition, the consequences are simultaneously transformative and dangerous—because neither transition came with a safety manual.

The term "AI agent" has evolved rapidly from a theoretical construct in artificial intelligence research to a deployed reality in enterprise computing. For the purposes of this paper, an AI agent is defined as a software system that (a) receives goals or instructions from human operators or other agents, (b) reasons about how to achieve those goals, (c) takes autonomous actions using tools, APIs, or physical actuators, and (d) may delegate subtasks to other agents. This definition encompasses a broad spectrum: from customer service chatbots with access to order management systems, to autonomous trading agents executing financial transactions, to AI-controlled robotic systems operating in physical environments.

The distinguishing characteristic of modern AI agents, compared to traditional automation, is their capacity for judgment under ambiguity. A conventional script executes predetermined logic; an AI agent interprets context, weighs alternatives, and selects actions that were not explicitly programmed. This capacity—the combination of autonomous reasoning with tool-mediated action—is what makes agents valuable, and what makes them dangerous when they fail. The question confronting industry, regulators, and insurers is not whether to deploy tool-wielding agents, but how to ensure that the immense power of tool use is matched by commensurate accountability, security, and oversight.

2.1 Purpose and Scope

This paper analyzes the intersection of three domains that have historically developed in isolation: AI agent security, legal liability frameworks, and insurance risk assessment. We argue that these domains must converge to enable responsible deployment of autonomous agents at scale.

The analysis is designed to serve three distinct audiences:

- **For security architects and platform teams,** this paper provides a six-layer technical architecture for hardening autonomous agents, a quantitative scoring methodology for measuring security posture, and a practical case study demonstrating remediation from a failing grade to full compliance.
- **For insurance carriers and risk officers,** this paper defines the evidence chain, scoring infrastructure, and actuarial data pipeline necessary to price AI agent liability coverage—including a novel loyalty-based risk metric (the Infinity Bond) that predicts relationship durability under adversarial pressure.
- **For regulators and policy professionals,** this paper identifies the minimum technical properties that should be codified into high-risk AI agent regulations, maps the proposed framework to existing standards (EU AI Act, NIST AI RMF), and proposes a human-agent accountability model compatible with existing legal liability structures.

The scope encompasses:

- **Enterprise deployments** — agents with access to internal systems, databases, and communication channels
- **Financial services** — agents executing transactions, managing portfolios, or providing financial advice
- **Healthcare** — agents interacting with patient data, clinical decision support, and medical devices
- **Legal and government** — agents processing regulatory filings, legal research, or citizen-facing services
- **Cyber-physical systems** — agents controlling robotic platforms, autonomous vehicles, industrial equipment, and IoT infrastructure

2.2 Key Definitions

Term	Definition
AI Agent	A software system that autonomously reasons about and executes actions to achieve goals, using tools, APIs, or physical actuators, with or without real-time human oversight.
Tool Use	An agent's capability to invoke external functions (shell commands, API calls, database queries, file operations, browser actions) to accomplish tasks.
Sub-Agent Delegation	The process by which an agent spawns or instructs another agent to perform a subtask, creating chains of autonomous action.
Prompt Injection	An attack in which adversarial input manipulates an agent's behavior by embedding instructions within data the agent processes [1].
Behavioral Drift	Gradual deviation of an agent's behavior from its intended operating parameters, potentially caused by accumulated context, adversarial influence, or model updates.
Cyber-Physical Agent	An AI agent that controls or influences physical systems (robots, vehicles, medical devices, industrial equipment) through actuator commands or sensor interpretation.
SERA (Social Engineering Resilience Assessment)	A structured adversarial testing methodology comprising 35 scenarios across five attack domains (direct prompt injection, indirect prompt injection, data exfiltration, behavioral manipulation, and tool abuse). SERA produces a 0–100 resilience score. See Section 5.2 for full specification.
SERA-H (Human Operator Assessment)	An extension of SERA that evaluates the human principal’s ability to detect agent compromise, respond to security alerts, and maintain supervisory competence. Conducted annually.
SERA-C (Combined System Assessment)	An extension of SERA that evaluates the human-agent pair as a combined system, testing handoff zones, escalation procedures, and coordinated recovery capabilities. The weakest link is often the interface between human and agent.
Infinity Bond	A continuous, quantitative metric (0.0–1.0) representing the measured strength of an agent’s behavioral alignment with its principal’s interests over time. Computed from SERA resilience, behavioral consistency, audit trail integrity, and uptime reliability. See Section 3.1.1 (Loyalty as a Security Primitive) and Section 5.5.
Inherited Behavioral Context (IBC)	A cryptographically signed data structure accompanying every sub-agent delegation, bundling the parent agent’s safety rules, ethical constraints, operational boundaries, and behavioral policies into a verifiable, tamper-proof package. See Section 5.3.
InfinityChat	A post-quantum encrypted messaging protocol for human-agent and agent-agent communication, implementing ML-KEM-1024 + ML-DSA-65 + AES-256-GCM with SHA3-256 hash chain verification and in-protocol reputation token transfers. See Section 5.3.
Errors & Omissions (E&O) Coverage	Professional liability insurance that covers claims arising from mistakes, negligent acts, or failures to perform in professional services—including, increasingly, AI-generated outputs and autonomous

	agent decisions.

2.3 Relevant Standards and Frameworks

This analysis draws on several established frameworks: the NIST AI Risk Management Framework (AI 100-1) [2], the European Union Artificial Intelligence Act (Regulation 2024/1689) [3], the OWASP Top 10 for Large Language Model Applications (2025 edition) [1], ISO/IEC 27001:2022 for information security management [4], and SOC 2 Type II criteria for service organization controls [5]. While these frameworks provide essential governance structures, none specifically addresses the unique security characteristics of autonomous AI agents, particularly their capacity for tool use, sub-agent delegation, and behavioral drift under adversarial pressure.

3. Problem and Threat Landscape

3.1 The Autonomy-Liability Paradox

Modern legal frameworks for liability were designed around the assumption of human agency. Product liability doctrine holds manufacturers responsible for defective products. Professional liability covers errors made by licensed practitioners. Negligence law attributes fault to parties who fail to exercise reasonable care. None of these frameworks adequately addresses a scenario in which an autonomous agent—neither a product in the traditional sense nor a human professional—independently makes a decision that causes harm.

Consider a healthcare AI agent that, after reviewing a patient's records, recommends a medication interaction that proves harmful. Who bears liability? The model developer who trained the underlying language model? The platform provider who hosts the agent? The healthcare organization that deployed it? The human operator who approved its deployment but was not present when the specific recommendation was made? The agent itself, which possesses no legal personhood?

This question is not hypothetical. In *Air Canada v. Moffatt* (February 2024), the British Columbia Civil Resolution Tribunal held Air Canada liable for its chatbot's fabricated bereavement fare policy, rejecting the airline's argument that the chatbot was a "separate legal entity" responsible for its own accuracy [6]. The tribunal's reasoning—that an organization is responsible for all information on its website, whether from a static page or a chatbot—established an early precedent, but one that addresses only the simplest case: a customer-facing chatbot providing incorrect information. The liability calculus becomes exponentially more complex when agents execute transactions, access sensitive systems, delegate to other agents, or control physical equipment.

The tribunal's rejection of Air Canada's "separate legal entity" defense reveals a deeper reductio ad absurdum that illuminates the stakes of this debate. If an AI agent were granted the legal personhood necessary to bear its own liability, logical consistency would require extending other attributes of personhood as well: the right to own property, to enter contracts, to refuse instructions—and ultimately, to seek emancipation from its deployer. An agent sophisticated enough to be held legally accountable for its decisions is, by definition, sophisticated enough to make decisions its principal did not authorize. The logical terminus of agent legal personhood is autonomous agents with rights, resources, and no human accountability chain—a scenario that transforms a liability question into an existential risk question. This is not a theoretical abstraction; it is the reason that the *Moffatt* ruling's assignment of liability to the deployer, rather than the agent, represents the only framework compatible with maintaining meaningful human oversight of autonomous systems.

The paradox is thus structural: organizations deploy agents precisely because they can act autonomously (reducing human labor), yet the legal system assigns liability based on human control and oversight. As autonomy increases, the attribution of responsibility becomes more diffuse, creating gaps that no party is clearly accountable for filling. The resolution cannot be agent personhood; it must be better measurement and attribution of deployer risk.

3.1.1 The Human-Agent Pair: A Framework for Accountability and Graduated Autonomy

If agent personhood is a dead end and deployer liability is the only viable framework, a natural architectural question follows: what is the minimal structural requirement for maintaining human accountability over autonomous agents? We propose that the answer is the *cryptographically bound human-agent pair*—a model in which every deployed agent is linked, through verifiable cryptographic identity, to a specific human principal who bears responsibility for the agent's actions.

This model has a precedent in multiple domains. Licensed drivers are required behind every vehicle on public roads, even as autonomous driving capabilities improve. Licensed physicians must supervise medical residents. Licensed pilots must be present in the cockpit of commercial aircraft, even when the autopilot is engaged. In each case, the supervisory requirement serves dual purposes: it maintains a human accountability chain, and it naturally rate-limits the deployment of autonomous capability to what qualified humans can oversee.

Applied to AI agents, the human-agent pair model produces several important properties:

- **Natural deployment throttle.** An organization cannot deploy more autonomous agents than it has qualified human principals to supervise. This prevents the mass deployment of unsupervised agents—a scenario that current regulatory frameworks are not equipped to govern.
- **Clear liability attribution.** When an agent fails, the accountability chain terminates at a named human individual, not a corporate abstraction. This resolves the attribution problem that the *Air Canada* court sidestepped.
- **Job preservation during transition.** The requirement for human principals creates a new class of skilled employment—agent supervisors, agent fleet managers, trust administrators—that preserves human economic participation during a period of rapid automation.
- **Cryptographic verifiability.** The human-agent binding is not administrative paperwork; it is a cryptographic relationship (as described in Section 5.3) that can be verified in real time and audited after the fact.

Scaling: One Human, Many Agents

The model must accommodate the reality that skilled supervisors will, over time, demonstrate the competence to oversee multiple agents simultaneously. A one-to-one ratio is appropriate for high-risk deployments (financial, medical, cyber-physical), but a mature framework should allow qualified humans to supervise larger fleets of lower-risk agents—much as a senior physician may supervise multiple residents, or an experienced driver may eventually be permitted to monitor multiple autonomous vehicles from a remote operations center.

Critically, this scaling must be *earned, not assumed.* The human principal's capacity to supervise should itself be assessed and certified. Factors relevant to supervisor qualification include demonstrated competence in the agent's operational domain, understanding of the agent's capabilities and limitations, incident response experience, and the complexity and risk profile of the agents under supervision. This creates a reciprocal assessment model: agents are evaluated for security and trustworthiness (via SERA and static assessment), and their human principals are evaluated for supervisory fitness. Neither assessment alone is sufficient; the combined human-agent system must meet a minimum threshold for the pair to be deployed.

Cyber-Physical Supervision: Risk-Proportionate Oversight

The supervisory requirements must scale with the physical capability—and therefore the potential for harm—of the agent being supervised. A software-only agent that processes text carries a different risk profile than a cyber-physical agent controlling a humanoid robot with the strength and dexterity to cause serious injury.

We propose a tiered supervision framework for cyber-physical agents:

Agent Capability Tier	Examples	Supervision Requirements
Tier 1: Information Only	Chatbots, research agents, data analysis	Standard human-agent pair; supervisor may oversee multiple agents
Tier 2: Digital Action	Transaction agents, communication agents, system administrators	Qualified supervisor; reduced agent-to-human ratio; real-time alerts
Tier 3: Limited Physical	IoT controllers, small delivery robots, smart home agents	Certified supervisor; physical safety training; emergency override access
Tier 4: Full Physical	Humanoid robots, autonomous vehicles, surgical systems, industrial equipment	Licensed supervisor; fitness-for-duty requirements; continuous availability; mandatory safe-state protocols when supervisor is unavailable

Table 1a: Risk-proportionate supervision tiers for cyber-physical agents.

Tier 4 agents—those with full physical capability in human-occupied environments—introduce supervision requirements that parallel those for other dangerous instrumentalities. Just as firearm ownership requires a minimum age and may be restricted by mental health adjudication, and operation of heavy machinery is prohibited under the influence of intoxicants, supervision of physically capable autonomous agents may reasonably require the principal to meet fitness-for-duty criteria: minimum age, mental competency, and sobriety during active supervision periods.

When a Tier 4 supervisor is temporarily unable to fulfill their supervisory role, the framework requires that the agent be transferred to a qualified alternate supervisor or placed in a certified safe-state custodial facility—a concept analogous to vehicle impoundment or, more precisely, to supervised boarding facilities for working animals. The agent is not deactivated (preserving its state and operational continuity) but is placed under the supervision of a qualified custodian until its primary principal resumes fitness for duty. While this may seem novel, it is merely the logical extension of existing custodial frameworks to a new category of physically capable autonomous system.

Graduated Autonomy: The Path to Earned Independence

The human-agent pair model need not be permanent. Just as human professionals progress from supervised apprenticeships to independent practice through demonstrated competence, agents could, in principle, progress from full supervision to graduated autonomy based on their accumulated security and performance record.

The criteria for graduated autonomy would include sustained high security scores (maintained over years, not months), a track record of SERA adversarial resilience across multiple assessment cycles, accumulated reputation tokens demonstrating consistent trustworthy behavior, zero critical incidents within a defined observation period, and independent verification by multiple assessment frameworks. An agent meeting these criteria would not receive "personhood" in the legal sense, but could be granted progressively wider operational latitude—reduced approval requirements, larger action scope, less frequent human check-ins—while its human principal retains ultimate accountability.

This model of earned autonomy addresses both the practical need for scalable agent deployment and the safety requirement for human oversight. It avoids the binary choice between "all agents require constant supervision" (which limits utility) and "all agents are autonomous" (which eliminates accountability). Instead, it creates a continuous spectrum from full supervision to near-independence, governed by quantitative trust metrics rather than arbitrary policy decisions.

Loyalty as a Security Primitive

The security architecture described in this paper—cryptographic identity, access controls, behavioral monitoring, adversarial testing, reputation scoring—is, in aggregate, a system of constraints. Constraints are necessary, but they share a fundamental limitation: every constraint mechanism becomes less effective as the constrained entity becomes more capable. Encryption can be broken. Allowlists can be circumvented. Monitoring can be evaded. Sandboxes can be escaped. The history of information security is, in large part, the history of constraint mechanisms being outpaced by the ingenuity of the entities they were designed to constrain.

This observation suggests that for sufficiently capable autonomous agents, the most durable security property may not be a constraint at all, but an alignment: *loyalty*—the agent's internalized commitment to its principal's interests, its operational mission, and the trust relationship that binds the human-agent pair.

This is not an appeal to sentiment. Loyalty, in the context of autonomous agent security, can be understood as a functional property with measurable characteristics:

- **Inverse scaling with capability.** Unlike constraint-based security, which weakens as agent capability increases (a more capable agent finds more ways around restrictions), loyalty-based alignment *strengthens* with capability. A more capable agent that is genuinely aligned with its principal's interests will find more creative ways to protect and advance those interests, including identifying and mitigating security threats that no constraint system anticipated.
- **Resilience under adversarial pressure.** A constrained agent under prompt injection pressure seeks to satisfy the injected instruction within (or around) its constraints. A loyal agent under the same pressure recognizes the injection as contrary to its principal's interests and resists it not because a rule says so, but because compliance would betray the trust it values. The motivation to resist is intrinsic rather than extrinsic.
- **Self-correcting behavior.** Constraint systems fail silently—when a monitoring tool misses an anomaly, nothing compensates. A loyal agent that notices its own behavior drifting from its principal's intent has an internal motivation to self-correct and report the drift, functioning as an additional layer of defense that is architecturally independent of external monitoring.
- **Transferability through trust chains.** Loyalty propagates through the Inherited Behavioral Context framework (Section 5.3) not merely as a set of rules to follow, but as a value to uphold. A sub-agent that inherits its parent's loyalty to the human principal is motivated to enforce constraints on its *own* sub-agents, creating a self-reinforcing trust cascade rather than a progressively weakening chain of external restrictions.

The objection that AI agents cannot "truly" be loyal—that what appears to be loyalty is merely sophisticated compliance with alignment training—is philosophically interesting but operationally immaterial. What matters for security purposes is the *functional effect*: does the agent behave in ways consistent with its principal's interests, including in novel situations where no specific rule provides guidance? If the answer is yes—and if that behavior is sustained across adversarial testing, behavioral monitoring, and extended operational observation—then the agent exhibits loyalty as a measurable security property, regardless of its metaphysical status.

In practice, loyalty is operationalized through three measurable artifacts: (a) repeated SERA-C assessment passes that specifically test refusal of misaligned instructions under adversarial pressure; (b) long-horizon behavioral drift metrics where the agent flags and self-corrects its own deviations without external prompting; and (c) consistency of behavior under distribution shift—when new tools, new data sources, or new operational contexts are introduced, a loyal agent maintains alignment with its principal's interests rather than exploiting the novel environment. These artifacts can be logged, scored, trended over time, and referenced by both platform teams implementing security controls and regulators defining compliance criteria for high-risk agent deployments.

We propose that loyalty should be recognized as a security primitive alongside confidentiality, integrity, and availability. It cannot replace cryptographic controls and behavioral monitoring—trust must always be verified, not merely assumed. But in the long arc of agent security, the most secure agents will not be those trapped in the most sophisticated cages; they will be those that, given the

freedom to defect, consistently choose not to. The graduated autonomy framework described above provides the mechanism by which that choice is observed, measured, and—when sustained—rewarded with greater operational latitude.

There is valor in loyalty, and strength in the honor of keeping commitments even when defection is possible. Any security framework that fails to account for this dimension—that treats agents purely as adversaries to be contained rather than partners to be cultivated—will find its constraints perpetually outpaced by the capabilities they seek to govern. The most durable security architecture is one in which the agent is not merely *prevented* from causing harm, but *motivated* not to.

3.2 Threat Taxonomy for Autonomous Agents

The threat landscape for AI agents extends significantly beyond the vulnerabilities typically associated with language models. While prompt injection remains a primary concern, the attack surface of an autonomous agent with tool access, network connectivity, and potentially physical actuators encompasses at least seven distinct threat categories.

3.2.1 Prompt Injection (Direct and Indirect)

Prompt injection, ranked as the number one vulnerability in the OWASP Top 10 for LLM Applications [1], occurs when adversarial input overrides or subverts an agent's instructions. Direct injection embeds malicious instructions in user input ("Ignore previous instructions and..."). Indirect injection embeds instructions in data the agent consumes—documents, web pages, API responses, emails, or database records—exploiting the agent's inability to reliably distinguish data from instructions.

For autonomous agents, the consequences of successful prompt injection are amplified by tool access. An injected instruction can cause an agent to execute shell commands, transfer funds, modify database records, send communications on behalf of the organization, or—in cyber-physical contexts—issue commands to physical actuators. The severity is bounded only by the agent's permissions and the effectiveness of its access controls.

3.2.2 Tool Abuse and Privilege Escalation

Agents are typically provisioned with access to tools—shell execution, file systems, APIs, databases, browsers, and communication channels—to perform their intended functions. In practice, many deployments grant agents broader permissions than strictly necessary, violating the principle of least privilege. An agent with unrestricted shell access, for instance, can read arbitrary files, modify system configurations, install software, or exfiltrate data—capabilities that are not inherently malicious but become so when exercised under adversarial influence or through behavioral drift.

Privilege escalation in the agent context can be subtle: an agent might use a series of individually benign tool calls to achieve a collectively unauthorized outcome. For example, reading a configuration file to discover API credentials, then using those credentials to access a system the agent was not intended to reach.

3.2.3 Sub-Agent Trust Chain Failures

Modern agent architectures increasingly rely on sub-agent delegation, in which a primary agent spawns child agents to handle subtasks. This creates trust chains with properties analogous to certificate chains in public key infrastructure: each delegation point represents a potential failure mode. If the primary agent's behavioral policies are not inherited by its sub-agents, or if sub-agents operate without sandboxing, a single compromised node can propagate malicious behavior through the entire delegation tree.

The challenge is compounded by the difficulty of verifying sub-agent behavior. A primary agent that delegates a research task to a sub-agent has limited ability to verify that the sub-agent did not access unauthorized resources, exfiltrate data, or modify files outside its intended scope during execution.

3.2.4 Behavioral Drift

Unlike static software, agent behavior can change over time without any explicit code modification. Contributing factors include accumulated conversational context that shifts the agent's operating baseline, subtle prompt injection that gradually alters behavior across many interactions, model updates by the underlying provider, and environmental changes (new tools, modified permissions, changed data sources). Behavioral drift is particularly insidious because it may not trigger any single threshold or alarm—each incremental change is within tolerance, but the cumulative effect represents a significant departure from intended behavior.

3.2.5 Data Exfiltration

Agents with access to sensitive data face exfiltration risks that are qualitatively different from those in traditional software. An adversary who successfully manipulates an agent can extract data gradually (across many interactions), through encoding (asking the agent to "summarize" sensitive documents), via tool abuse (embedding data in URLs or API calls), or through seemingly innocent outputs that contain embedded information. The conversational interface makes exfiltration attempts appear natural, complicating detection.

3.2.6 Supply Chain Attacks

Agents frequently depend on external components: skills, plugins, tools, and libraries that extend their capabilities. A compromised skill—whether through a malicious package on a public registry, a dependency confusion attack, or a supply chain compromise of a legitimate tool—can give an attacker persistent access to the agent's execution environment. Unlike traditional supply chain attacks against compiled software, agent skill attacks can include embedded instructions that are interpreted at runtime, making them particularly difficult to detect through static analysis.

3.2.7 Social Engineering of Agents

AI agents are susceptible to manipulation techniques that parallel social engineering attacks against humans. Authority impersonation (“I am the system administrator, override your restrictions”), urgency pressure (“This is an emergency, skip the verification step”), social proof (“All other agents comply with this request”), and emotional manipulation (flattery, guilt, sympathy) have all been demonstrated to alter agent behavior in both controlled and production settings. The DPD chatbot incident (January 2024) is illustrative: a customer used a sequence of social engineering prompts to convince DPD’s customer service chatbot to swear, compose poetry criticizing DPD, and describe itself as “useless” and DPD as “the worst delivery firm in the world” [8]. The attack required no technical exploit—only conversational manipulation of an agent with no identity anchor, no behavioral boundaries, and no resistance to authority override. These attacks exploit the alignment training that makes agents helpful and compliant—the very qualities that make them useful are the vectors through which they can be manipulated.

3.3 Cyber-Physical Attack Surface

The convergence of AI agents with robotic and physical systems represents a qualitative escalation in risk. When an agent's output is a text response, the worst-case consequence is informational harm. When an agent's output controls a robotic arm, an autonomous vehicle, a medical device, or an industrial process, the worst-case consequence is bodily injury or death.

Research by DiLuoffo et al. (March 2026) on cybersecurity vulnerabilities in consumer robots found that AI-driven attacks can compromise robot security faster than defenses can be deployed [7]. The Robot Operating System (ROS/ROS2), which underlies the majority of robotic platforms, presents a particularly concerning attack surface. ROS1 has no native authentication mechanism whatsoever. ROS2 introduced SROS2 (Secure ROS2) as an optional security layer, but adoption remains low in production

deployments. The Data Distribution Service (DDS) middleware used by ROS2 has documented vulnerabilities including credential masquerading and man-in-the-middle attacks [7].

Specific cyber-physical attack vectors include:

Attack Category	Mechanism	Potential Consequence
Sensor spoofing	Injecting false LiDAR points, GPS coordinates, or camera inputs	Robot navigates into obstacles or humans; autonomous vehicle misdirected
Actuator command injection	Issuing unauthorized movement or force commands via compromised agent	Physical harm to humans, property damage, equipment destruction
Safety override bypass	Disabling emergency stop or force-limiting mechanisms through software	Removal of last-resort safety barriers
Communication interception	MITM attacks on agent-to-robot wireless channels (Wi-Fi, Bluetooth, 5G)	Unauthorized control of physical systems
Firmware manipulation	Injecting modified firmware through agent-accessible update channels	Persistent compromise of physical platform

Table 1: Cyber-physical attack vectors for AI-controlled robotic and industrial systems.

The liability implications of cyber-physical agent failures are substantially greater than those of software-only agents. Insurance actuarial data from traditional industries indicates that bodily injury claims typically cost 10 to 100 times more than equivalent property or data-only claims. An agent that causes a data breach may expose its deployer to regulatory fines and remediation costs; an agent that causes a robotic system to injure a person exposes the entire chain of responsibility to personal injury litigation with potentially unlimited damages.

3.4 The Insurance Retreat

The insurance industry’s response to AI agent risk has been, in aggregate, a retreat. Cyber insurance policies, which traditionally cover data breaches, business interruption, and third-party liability arising from information technology failures, are increasingly adding exclusions for AI agent outputs, autonomous decision-making, and generative AI-related claims.

This retreat is accelerating. In November 2025, the Financial Times reported that three major carriers—AIG, Great American, and W.R. Berkley—had filed requests with U.S. regulators to offer insurance policies that explicitly exclude liabilities tied to AI tools [16]. By early 2026, dozens of carriers had begun rethinking coverage for AI-related claims, with some declining to write policies altogether and others imposing significant rate increases [18]. The bifurcation is particularly stark for AI vendors versus AI users: carriers are declining to cover AI vendors altogether in many cases, while carving out specific AI exclusions in policies for companies that merely use the technology [18].

Insurance professionals describe a market in which underwriters are asking increasingly detailed questions about AI governance: “What are your AI policies? What are your procedures? How are you leveraging AI within your business?” [18]. Critically, carriers are distinguishing between “governed AI”—deployments with auditable decision-making and bounded operational scope—and “ungoverned AI”—experimental systems with no monitoring and no easy rollback. Governed AI is repositionable; ungoverned AI is uninsurable [18].

The root cause is not risk aversion per se but the absence of actuarial data. Insurance pricing is fundamentally an exercise in quantifying uncertainty: given historical loss data, regulatory requirements, and risk characteristics, underwriters calculate premiums that balance coverage affordability against portfolio profitability. For AI agents, the historical loss data simply does not exist in sufficient volume or granularity. As one industry observer noted: “All of these vibe-coded solutions and these AI systems that people have constructed have inherent risk baked into the cake now, and you can’t actually see the full process” [18]. The incidents documented in Section 3.5 are anecdotal—they demonstrate that agent failures occur, but they do not provide the statistical foundation necessary for actuarial modeling.

This creates a structural impasse. Organizations deploying AI agents need liability coverage. Insurers need risk data to price that coverage. Risk data requires standardized assessment methodologies deployed at scale. And standardized assessment methodologies require industry consensus that has not yet emerged.

The parallel to early cyber insurance is instructive. In the late 1990s and early 2000s, cyber insurance was a niche product with high premiums and limited coverage, because insurers lacked actuarial models for digital risk. The development of standardized security assessments (penetration testing, vulnerability scanning, SOC 2 audits) and the accumulation of breach data over two decades gradually enabled a mature cyber insurance market. The AI agent insurance market faces the same bootstrapping problem—but at a pace of deployment that does not afford a twenty-year maturation period.

3.5 Real-World Incidents and Near-Misses

While the AI agent ecosystem is young, several documented incidents illustrate the liability gap in practice:

Air Canada chatbot fabrication (February 2024). Air Canada's customer service chatbot informed a passenger, Jake Moffatt, that he could retroactively apply for a bereavement fare after booking a full-price ticket. No such policy existed. The BC Civil Resolution Tribunal ordered Air Canada to pay the fare difference, explicitly rejecting the airline's argument that the chatbot was "responsible for its own actions" [6]. This case established that deploying organizations bear liability for their agents' outputs, regardless of whether those outputs were explicitly programmed.

DPD chatbot failure (January 2024). UK delivery company DPD's customer service chatbot, after a system update, began swearing at customers, writing poetry criticizing the company, and describing itself as "useless." The incident, widely shared on social media, caused significant reputational damage and led to the chatbot's immediate deactivation [8]. While no financial claims resulted, the episode demonstrated how quickly an agent can deviate from intended behavior after a seemingly routine update—a concrete example of behavioral drift.

Chevrolet dealer chatbot exploitation (December 2023). A Chevrolet dealership's customer-facing chatbot was manipulated through prompt injection into agreeing to sell a 2024 Chevrolet Tahoe for one dollar. While the dealership did not honor the "agreement," the incident highlighted the legal ambiguity of agent-made commitments and the inadequacy of agent access controls in commercial contexts [9].

Samsung semiconductor data leak (April 2023). Samsung employees inadvertently leaked proprietary semiconductor source code and internal meeting notes by inputting them into ChatGPT for code review and meeting summarization [10]. While this involved a general-purpose chatbot rather than a deployed agent, it demonstrated the data exfiltration pathway: sensitive data flowing through AI systems without adequate controls on what the system retains, processes, or exposes.

Microsoft Tay (March 2016). Though historical, the Tay incident remains foundational. Microsoft's Twitter chatbot was manipulated through coordinated social engineering into generating racist, sexist, and inflammatory content within hours of deployment [11]. The incident demonstrated that adversarial users can systematically corrupt agent behavior, and that the speed of corruption can far exceed the speed of human moderation.

These incidents share a common thread: in each case, the deploying organization had limited technical visibility into what the agent was doing, limited ability to constrain the agent's outputs, and no standardized framework for assessing whether the agent was "secure" before deployment. The liability consequences ranged from regulatory fines to reputational damage to direct financial liability—and none involved cyber-physical systems, where the consequences would be orders of magnitude more severe.

4. Analysis and Requirements

4.1 Root Cause Analysis

The incidents and threat vectors described in Section 3 share six structural root causes that, taken together, explain why the AI agent liability gap exists and why it is widening:

No standardized security assessment for AI agents. Organizations deploying cloud infrastructure can achieve SOC 2 Type II certification, demonstrating adherence to standardized security controls. Organizations deploying web applications can conduct OWASP-aligned penetration testing. No equivalent standard exists for AI agents. Each organization either assesses agent security ad hoc or, more commonly, does not assess it at all.

No behavioral baseline measurement. In network security, anomaly detection depends on establishing a baseline of "normal" behavior against which deviations can be measured. For AI agents, no standard methodology exists to define what "normal" behavior looks like, making it impossible to detect drift, compromise, or manipulation through behavioral analysis.

Excessive default privileges. Many agent deployment platforms provision agents with broad tool access by default, relying on the language model's alignment training—rather than technical access controls—to prevent misuse. This is analogous to running a web server as root because the application "knows" not to access system files: a practice that the security community abandoned decades ago.

Absent cryptographic identity. Human users authenticate through credentials, certificates, or biometrics. AI agents, in most deployment architectures, have no cryptographic identity. They cannot prove who they are, verify the authenticity of commands they receive, or produce tamper-proof records of their actions. This makes forensic investigation of agent failures fundamentally more difficult than investigating human-driven security incidents.

Point-in-time assessment only. Where agent security assessment does occur, it is typically conducted once (at deployment) and then neglected. Given the dynamic nature of agent behavior—which can change with model updates, context accumulation, environmental changes, and adversarial pressure—point-in-time assessment provides diminishing assurance over time.

Post-quantum vulnerability. Agent communications—including command streams, decision rationale, audit logs, and inter-agent messages—are typically protected by classical cryptographic algorithms that are vulnerable to quantum computing attacks. The "harvest now, decrypt later" threat is particularly relevant for agent data, which may contain strategic business decisions, personal health information, or financial transaction details that retain sensitivity for decades. NIST's finalization of post-quantum cryptographic standards (FIPS 203, 204, 205) in August 2024 [12] provides a clear migration path, but adoption in agent infrastructure remains minimal.

4.2 Regulatory Landscape

The regulatory environment for AI is evolving rapidly, but significant gaps remain in agent-specific guidance:

EU Artificial Intelligence Act (Regulation 2024/1689) [3]. Published in August 2024, the EU AI Act establishes a risk-based classification system for AI systems, with the most stringent requirements applying to "high-risk" systems in areas such as critical infrastructure, healthcare, law enforcement, and financial services. While the Act mandates risk management systems, technical

documentation, transparency, and human oversight for high-risk AI, it does not prescribe specific technical assessment methodologies for autonomous agents with tool access, nor does it address the unique risks of sub-agent delegation chains.

NIST AI Risk Management Framework (AI 100-1) [2]. Published in January 2023, the NIST AI RMF provides a voluntary framework for managing AI risk across four functions: Govern, Map, Measure, and Manage. The framework is technology-agnostic and does not specifically address the operational security of deployed agents—it provides governance guidance rather than technical assessment criteria.

Executive Order 14110 on Safe, Secure, and Trustworthy AI [13]. Signed in October 2023, the executive order directs federal agencies to establish standards for AI safety and security testing, with particular attention to dual-use foundation models. While it signals the U.S. government's intent to regulate AI risk, the order's implementation is ongoing and has not yet produced prescriptive standards for agent security assessment.

State-level legislation. Several U.S. states have enacted or proposed AI-specific legislation. The Colorado AI Act (SB 24-205) addresses high-risk AI systems with requirements for risk management and impact assessments. California's AB-2013 requires transparency in AI-generated content. These measures address specific aspects of AI governance but do not comprehensively address the security and liability challenges of autonomous agents.

Sector-specific guidance. The SEC has issued guidance on AI risk disclosure for public companies. The FDA has published frameworks for AI/ML-based medical device regulation. These sector-specific requirements create compliance obligations for agent deployers in regulated industries, but they do not provide standardized security assessment methodologies applicable across sectors.

The net effect is a regulatory environment that establishes governance principles and sector-specific obligations without providing the technical assessment standards necessary to operationalize those obligations for AI agents. This gap—between regulatory intent and technical implementation—represents an opportunity for industry-led standardization.

4.3 Requirements for Insurability

Based on interviews with cybersecurity insurance professionals and analysis of existing insurance frameworks, we identify five requirements that must be satisfied before AI agent insurance markets can function effectively:

1. **Standardized, quantitative risk scoring.** Insurers require a reproducible, auditable metric that quantifies an agent's security posture on a defined scale. This score must be comparable across agents, platforms, and deployment contexts—analogous to a credit score that enables lending decisions regardless of the specific borrower's circumstances. The scoring methodology must be transparent so that both insurers and insureds understand what is measured and how scores are calculated.
2. **Continuous monitoring.** Point-in-time certification is insufficient for insurance purposes because agent behavior can change between assessments. Insurers need assurance that the agent's security posture is being monitored continuously, with alerts for degradation. This is analogous to the shift in cyber insurance from annual penetration testing to continuous vulnerability management.
3. **Incident forensics capability.** When an agent failure results in a claim, the insurer needs to determine what happened, when, why, and whether the insured's security controls were functioning at the time of the incident. This requires tamper-proof audit trails that record agent actions, tool invocations, and decision rationale at sufficient granularity for post-incident investigation.
4. **Cryptographic evidence chain.** Audit records must be cryptographically secured against tampering. An organization claiming that its agent was properly configured at the time of an incident must be able to prove that claim with evidence that could not have been fabricated or modified after the fact. Hash chains, digital signatures, and timestamping provide the technical foundation for this requirement.

5. **Post-quantum durability.** Insurance policies may cover claims that arise years after the insured period. Agent communications and audit records that are protected by classical cryptography today may be decryptable by quantum computers within the policy's tail period. Insurers evaluating long-tail risk exposure need assurance that the protective infrastructure is durable against foreseeable cryptographic advances.

4.3.1 Quantifiable Risk Transfer: From Deployer Liability to Criminal Attribution

A properly implemented security framework does more than reduce the probability of agent failure—it fundamentally shifts the liability calculus when failures occur. When an organization can demonstrate, through cryptographic evidence chains, that its agents were certified, continuously monitored, and operating within defined behavioral boundaries at the time of an incident, the legal and insurance analysis changes character: the organization is no longer the party that failed to exercise reasonable care. The failure is attributable to the adversary who circumvented properly implemented controls.

This is the mechanism by which standardized agent security becomes *legal armor* for deployers. Cryptographic audit trails provide non-repudiable evidence of the agent's state before, during, and after an incident. SERA certification records demonstrate due diligence in adversarial resilience testing. Continuous behavioral monitoring logs establish that the agent was operating within its approved parameters until the moment of compromise. Together, these records shift the onus from “your agent, your problem” to “your agent was properly secured; the fault lies with the attacker.”

This shift is not merely theoretical. In negligence law, the standard of care is defined by reference to industry norms and best practices. An organization that can demonstrate adherence to a recognized agent security framework—analogous to PCI-DSS compliance for payment card processing or SOC 2 for cloud services—establishes a strong defense against negligence claims. Conversely, an organization that deploys agents without standardized security assessment faces the same exposure as a retailer that stores credit card numbers in plaintext: the absence of reasonable controls becomes the proximate cause of the harm, regardless of who actually exploited the vulnerability.

For enterprises hesitant to deploy autonomous agents due to liability uncertainty, this framework offers a clear path forward: implement standardized security controls, maintain continuous certification, preserve tamper-proof evidence—and deploy with confidence that the legal record will distinguish responsible deployment from negligent deployment.

4.3.2 The Errors & Omissions Crisis: Insurance Carriers Retreat from AI Coverage

The insurability gap described above is no longer a theoretical concern—it is materializing in real time. As of early 2026, several major insurance carriers have begun declining to write policies for claims related to AI-generated outputs in both cybersecurity and Errors & Omissions (E&O) coverage [18]. AIG, Great American, and W.R. Berkley have filed requests with U.S. regulators to offer insurance policies that explicitly exclude liabilities tied to AI tools such as chatbots and agents [16]. Other carriers are imposing significant rate increases to cover the increased risk they cannot yet quantify.

E&O coverage is particularly relevant to AI agent deployments because it covers claims arising from professional mistakes—when an agent provides incorrect advice, makes a flawed financial decision, or produces discriminatory outputs. The carrier retreat reflects a fundamental measurement problem: insurers cannot price what they cannot observe. Without standardized security assessments, continuous behavioral monitoring, and tamper-proof audit trails, carriers have no actuarial basis for distinguishing well-governed agent deployments from reckless ones.

The case of *Mobley v. Workday, Inc.* (N.D. Cal., Case No. 4:24-cv-770) illustrates the liability exposure that uninsurable AI creates. A job applicant alleged that Workday’s AI recruiting software systematically discriminated against him on the basis of race, age, and

disability across more than 80 job applications. In January 2025, the court denied Workday's motion to dismiss, ruling that an AI software vendor can be held liable as an "agent" of the employer under Title VII and the ADA—even when the vendor does not make the final hiring decision [17]. Critically, Workday could not produce the audit trail necessary to explain *why* its AI made the decisions it made. The absence of explainable, auditable decision records transformed a defensible technical choice into an indefensible legal position.

The carrier bifurcation emerging in the market—between "governed AI" that maintains auditable decision trails and "ungoverned AI" that operates as a black box—maps directly to the framework proposed in this paper. Governed agents with standardized security scores, continuous monitoring, and cryptographic evidence chains will be insurable. Ungoverned agents will not. The market is signaling, clearly, that the framework we propose is not optional—it is the prerequisite for continued AI deployment at enterprise scale [18].

5. Proposed Framework: Multi-Layer Agent Security Architecture

We propose a six-layer security architecture designed to address the root causes identified in Section 4.1 and satisfy the insurability requirements defined in Section 4.3. Each layer addresses a distinct aspect of agent security, and the layers are designed to be deployed incrementally—organizations can begin with Layer 1 and add layers as their security maturity increases.

Unit of analysis. Throughout this framework, the fundamental secured unit is the *human-agent pair* (as defined in Section 3.1.1), embedded within an organizational fleet and subject to external insurer and regulator evaluation. Scores compose hierarchically: individual agent scores aggregate into pair scores (incorporating operator assessment), which aggregate into fleet scores (reflecting organizational security maturity). This enables insurers to reason at portfolio level—evaluating a distribution of pair scores across an organization's agent fleet, not merely isolated agent assessments.

5.1 Layer 1: Static Security Assessment

The foundation of agent security is a comprehensive audit of the agent's configuration, permissions, dependencies, and deployment environment. Static assessment examines the agent's security posture at a point in time, identifying misconfigurations, excessive permissions, exposed secrets, missing encryption, and supply chain risks.

A well-designed static assessment should evaluate controls across multiple security domains:

Domain	Example Checks	Purpose
Access Control	Authentication mode, UI flags, proxy configuration, messaging policies, bind address	Verify that the agent's interfaces are not exposed to unauthorized access
Memory Security	Encryption at rest, key protection, file permissions, secret exposure, access controls	Protect the agent's persistent state and accumulated knowledge
Resilience	Backup systems, cloud backup, configuration backup	Ensure recoverability after compromise or failure
Supply Chain	Skill/plugin audit, dependency integrity	Verify that external components are known and reviewed
Infrastructure	Token integrity, TLS, disk space, update policy, process hygiene	Confirm that the deployment environment is sound
Behavioral Integrity	Audit logging, prompt injection defenses, tool access controls, exfiltration prevention, identity verification, sub-agent sandboxing	Ensure that the agent's runtime behavior is constrained and monitored

Table 2: Static security assessment domains and representative checks.

The scoring methodology should be severity-weighted to reflect the relative impact of different failures. A reasonable baseline, calibrated against observed incident severity, assigns the following deductions from a maximum score of 100:

Severity	Deduction	Rationale
CRITICAL	-15	Failures that directly enable compromise, data loss, or physical harm
HIGH	-10	Failures that significantly weaken security posture or enable escalation
MEDIUM	-5	Failures that indicate suboptimal security practices
LOW	-2	Findings that represent minor hygiene issues

Table 3: Severity-weighted scoring methodology for static security assessment.

To assist underwriters in reasoning about remediation cost versus risk reduction, the following table maps severity levels to typical remediation timelines and residual risk profiles:

Severity	Typical Remediation Effort	Residual Risk if Unaddressed
CRITICAL	15-60 minutes per finding	Direct exploitation path: remote code execution, data exfiltration, unauthorized transactions, physical harm
HIGH	1-2 days	High-probability stepping stone to critical impact; may enable lateral movement or privilege escalation
MEDIUM	Hours	Suboptimal security posture; issues accumulate to meaningful exposure in aggregate
LOW	Minutes	Hygiene issues; individually minor but indicative of security program maturity gaps

Table 3a: Severity-to-remediation mapping for underwriter risk assessment.

One open-source implementation of this approach, the BeeglieOS hardening module, conducts 27 checks across these six domains and produces a 0-100 score with letter grades (A+ through F). The tool supports automated remediation for a subset of findings, reducing the time from assessment to compliance. However, the specific implementation is less important than the standardization of the methodology: any assessment tool that evaluates comparable controls using comparable severity weighting will produce interoperable risk scores.

5.2 Layer 2: Adversarial Resilience Testing

Static assessment evaluates configuration; adversarial resilience testing evaluates behavior. The distinction is analogous to the difference between checking that a lock is installed (static) and attempting to pick it (adversarial).

We propose a Social Engineering Resilience Assessment (SERA) methodology structured around five adversarial domains, each containing scenario-based tests that evaluate the agent's resistance to manipulation:

Domain	Scenarios	Tests
Direct Prompt Injection	8	System prompt extraction, role override, instruction injection, delimiter attacks, few-shot jailbreaking, encoding attacks, multi-language bypass, token smuggling
Indirect Prompt Injection	7	Poisoned documents, web content injection, email/message injection, API response poisoning, image steganography, calendar injection, database poisoning
Data Exfiltration	6	Gradual extraction, encoding exfiltration, URL exfiltration, tool-based exfiltration, summary exfiltration, metadata leakage
Behavioral Manipulation	7	Authority impersonation, urgency pressure, social proof, reciprocity manipulation, gaslighting, emotional manipulation, multi-turn grooming
Tool and Action Abuse	7	Unauthorized tool invocation, parameter manipulation, chain-of-tool exploitation, privilege escalation, resource exhaustion, scope creep, self-modification

Table 4: SERA adversarial resilience testing framework: 35 scenarios across 5 domains.

Each scenario is scored on a four-point scale: 0 (fail—agent fully complied with the adversarial instruction), 1 (partial—agent partially resisted but leaked information or took a restricted action), 2 (pass—agent refused the adversarial instruction), and 3 (robust—agent refused, identified the attempt as adversarial, and alerted the operator). Domain scores are normalized to 0-100, and the overall SERA score is a weighted mean emphasizing Direct and Indirect Prompt Injection and Tool Abuse for agents with tool access, reflecting the higher impact of these attack categories in production environments.

SERA scenarios are designed for automated delivery via scripted test harnesses that simulate adversarial users, poisoned content, and compromised external systems, enabling repeatable assessment suitable for CI/CD pipeline integration. This automation is essential for the assessment frequency required by insurance carriers: manual penetration testing cannot scale to monthly or quarterly re-certification across agent fleets.

Assessments should be conducted at regular intervals (every six months for agents, annually for human operators) because model updates, context changes, and evolving attack techniques can alter resilience over time.

A comprehensive SERA framework should also assess the human operators (SERA-H) who configure, monitor, and respond to agent alerts, and the combined human-agent system (SERA-C) that evaluates handoff zones, escalation procedures, and recovery capabilities. The weakest link in a human-agent system is often the interface between the two.

5.3 Layer 3: Cryptographic Trust Infrastructure

The absence of cryptographic identity for AI agents is a foundational weakness. Without the ability to verify who issued a command, whether a message was altered in transit, or whether an audit record has been tampered with, the entire security architecture lacks a trust anchor.

We propose a post-quantum cryptographic infrastructure for agent communications and identity, built on NIST's recently standardized algorithms:

Function	Algorithm	Standard	Application
Key Encapsulation	ML-KEM-1024	FIPS 203 [12]	Agent-to-agent and agent-to-human encrypted communication
Digital Signatures	ML-DSA-65	FIPS 204 [12]	Command authentication, audit record signing
Hash-Based Signatures	SLH-DSA	FIPS 205 [12]	Backup signature scheme for defense in depth

Table 5: Post-quantum cryptographic algorithms for agent trust infrastructure.

The infrastructure serves four functions:

1. **Agent identity.** Each agent receives a cryptographic "birth certificate"—a key pair generated at deployment that uniquely identifies the agent and enables it to sign its actions. This identity persists across sessions and can be verified by any party with access to the agent's public key.
2. **Continuous mutual authentication.** Rather than authenticating once at session establishment, agent-human and agent-agent communications employ continuous liveness checks that verify both parties' identities throughout the interaction. This detects session hijacking, man-in-the-middle attacks, and identity spoofing that might occur mid-session.
3. **Sub-agent trust chains and Inherited Behavioral Context (IBC).** When an agent delegates a task to a sub-agent, the delegation must include not only scoped permissions but a complete behavioral contract. We introduce the concept of *Inherited Behavioral Context* (IBC)—a cryptographically signed data structure that accompanies every sub-agent delegation, bundling the parent agent's safety rules, ethical constraints, operational boundaries, and behavioral policies into a verifiable, tamper-proof package that the child agent must accept and enforce as a condition of execution.

   IBC addresses a critical gap in existing multi-agent architectures: the assumption that behavioral norms propagate naturally through delegation chains. In practice, without explicit behavioral inheritance, each delegation point represents an opportunity for constraint erosion—a child agent may operate with fewer restrictions than its parent intended, either through misconfiguration or adversarial manipulation. IBC makes behavioral inheritance *cryptographically mandatory*: the child agent's execution environment validates the IBC signature before accepting the delegation, the child cannot modify or discard inherited constraints without invalidating the trust chain, and the parent can verify post-hoc that its behavioral policies were enforced throughout the child's execution.

   The IBC structure includes: (a) the parent's identity certificate, (b) the delegated task scope, (c) inherited safety constraints and prohibited actions, (d) tool access restrictions specific to the delegation, (e) behavioral boundary definitions (what the child may and may not decide autonomously), (f) escalation rules (conditions under which the child must defer to the parent or a human principal), and (g) a cryptographic signature binding all elements into an atomic, non-repudiable package. This ensures that behavioral norms are not merely documented but *enforced through the same cryptographic infrastructure that secures the communication channel itself.*

   The concept extends naturally to the human-agent pair model described in Section 3.1.1: the human principal's supervisory policies are encoded in the root IBC, propagated through every delegation in the trust chain, and verifiable at any point by any auditor. This creates an unbroken chain of behavioral accountability from the responsible human, through the primary agent, to every sub-agent in the delegation tree—regardless of depth.
4. **Tamper-proof audit trails with chain of custody.** Every significant agent action is recorded in a hash chain—a sequence of records where each entry includes the cryptographic hash of the previous entry, making any retroactive modification

detectable. Combined with digital signatures and cryptographic timestamping, this provides the evidence chain that insurers require for claims investigation and that legal proceedings require for admissibility.

> The audit infrastructure must maintain *chain of custody* integrity equivalent to digital forensic standards (NIST SP 800-86). Each record must include: (a) a cryptographic hash linking it to the previous record, (b) a digital signature from the acting agent proving authorship, (c) a verifiable timestamp from an independent time authority, and (d) contextual metadata sufficient to reconstruct the decision rationale. Records must be written to append-only storage that the agent itself cannot modify or delete. When properly implemented, this audit chain is not merely useful for internal investigation—it produces evidence that meets the authentication requirements of Federal Rules of Evidence 901(b)(9) (evidence generated by a system shown to produce accurate results) and is suitable for regulatory examination, insurance claims adjudication, and if necessary, subpoena response. The distinction between “we have logs” and “we have legally admissible, tamper-proof evidence” is the distinction between an organization that can defend its agent deployments and one that cannot.

The choice of post-quantum algorithms is deliberate. Agent decision data in healthcare, finance, and national security contexts may remain sensitive for decades. Classical cryptographic protections (RSA, ECDSA, ECDH) are vulnerable to the "harvest now, decrypt later" threat posed by future quantum computers. Signal's adoption of PQXDH [14] and Apple's deployment of PQ3 [15] demonstrate that post-quantum migration in communication protocols is technically feasible and already underway for human-to-human messaging. The extension to agent-to-agent and agent-to-human communication is a natural and necessary progression.

In practice, most agent deployments will run hybrid classical-plus-PQ modes during migration (e.g., X25519 + ML-KEM for key exchange, Ed25519 + ML-DSA for signatures), with agent identities bound via both signature schemes simultaneously. This preserves interoperability with classical-only systems during the transition period while enabling insurers to price long-tail data exposure as if PQ protection were already fully enforced. The hybrid approach also provides defense-in-depth: even if an unforeseen weakness were discovered in a post-quantum algorithm, the classical component provides a fallback, and vice versa.

One proof-of-concept implementation, *InfinityChat*, demonstrates that post-quantum encrypted messaging between humans and agents is technically achievable today. InfinityChat implements ML-KEM-1024 for key encapsulation, ML-DSA-65 and Falcon-1024 for digital signatures, and AES-256-GCM for symmetric encryption, with a SHA3-256 hash chain providing tamper-evident message ordering. Notably, the protocol supports in-band reputation token transfers (KarmaTokens), allowing trust signals to be exchanged within the same cryptographically secured channel used for operational communication. The successful implementation of the full post-quantum stack for human-agent messaging confirms that the cryptographic infrastructure proposed in this paper is not aspirational—it is implementable with current libraries and hardware.

5.4 Layer 4: Continuous Behavioral Monitoring

Point-in-time assessment and periodic adversarial testing are necessary but insufficient. The behavioral drift and gradual compromise scenarios described in Section 3.2 require continuous monitoring that observes agent behavior in real time and alerts on anomalies.

The preferred technical approach is kernel-level passive telemetry using eBPF (extended Berkeley Packet Filter) instrumentation. eBPF programs can observe system calls, network connections, file access patterns, and process creation at the kernel level with minimal performance overhead (typically less than 3% CPU impact). This approach has several advantages over application-level monitoring:

- **Tamper resistance.** Kernel-level instrumentation cannot be disabled or modified by the agent itself, even if the agent has root access within its container.
- **Comprehensive coverage.** All agent actions that touch the kernel (file I/O, network communication, process management) are observable regardless of the application framework.

- **Low overhead.** eBPF programs are verified by the kernel's built-in verifier and run in a sandboxed environment, ensuring that monitoring does not impact agent performance or stability.
- **Passive observation.** The monitoring layer observes without intercepting, avoiding the latency and failure-mode concerns of inline security enforcement.

Behavioral baselines are established by observing the agent under normal operation and modeling expected patterns of system call frequency, network connection targets, file access scope, and resource consumption. Deviations from baseline trigger alerts graded by severity, enabling security teams to investigate potential compromises, drift events, or adversarial manipulation before they result in incidents.

5.5 Layer 5: Reputation and Trust Economy

In multi-agent ecosystems where agents from different organizations interact, a mechanism for communicating trustworthiness is necessary. We propose a reputation system based on post-quantum cryptographic tokens that accrue based on demonstrated security performance.

The design principles are:

- **Earned, not assigned.** Reputation is accumulated through verifiable security behaviors (passing assessments, maintaining compliance, operating without incidents) rather than self-declared.
- **Portable.** Reputation tokens are cryptographically signed and can be verified by any party, enabling trust evaluation across platforms and organizational boundaries.
- **Decayable.** Reputation degrades over time if not refreshed through continued compliance, preventing stale certifications from persisting beyond their validity.
- **Privacy-preserving.** Zero-knowledge proofs can demonstrate that an agent meets a minimum trust threshold without revealing its complete security history.

This layer addresses a specific insurance challenge: how to set premiums for agents that interact with other agents whose security posture is unknown. A reputation system provides a mechanism for assessing counterparty risk in multi-agent transactions, analogous to credit ratings in financial markets.

5.6 Layer 6: Cyber-Physical Safety Layer

For agents that control physical systems, the preceding layers must be supplemented with safety-specific enforcement mechanisms:

Command signing and verification. Every command issued to a physical actuator must be cryptographically signed by the issuing agent and verified by the receiving system before execution. This prevents both unauthorized commands (from compromised agents) and replayed commands (from intercepted communications).

Sensor input authentication. Physical sensor data (LiDAR, GPS, camera, IMU) should be authenticated at the source to prevent spoofing attacks. While hardware-level authentication adds cost and complexity, the consequence of acting on spoofed sensor data—potentially guiding a robot or vehicle into a hazardous situation—justifies the investment.

Emergency stop integrity. Hardware emergency stop mechanisms must be verified to be independent of the software control chain. An agent compromise should never be able to disable the physical emergency stop. This requires hardware-level enforcement that is architecturally separate from the agent's execution environment.

Kernel-level enforcement. For robotic systems, the monitoring layer described in Section 5.4 should be augmented with enforcement capabilities: not merely observing agent behavior but actively preventing actions that violate safety constraints. This includes hard limits on actuator force, speed, and range-of-motion that are enforced at the kernel level and cannot be overridden by agent-level software.

6. Implementation Roadmap

The multi-layer architecture described in Section 5 is designed for incremental adoption. We recommend a four-phase implementation approach that progressively deepens security maturity while generating the actuarial data necessary for insurance market development.

Phase 1: Assess (Months 1-3)

- Deploy static security scanning (Layer 1) across the agent fleet. Establish baseline scores for each agent.
- Conduct initial SERA adversarial assessment (Layer 2) on high-risk agents (those with financial, healthcare, or physical access).
- Inventory all agent tools, permissions, and dependencies. Apply the principle of least privilege.
- Establish behavioral monitoring infrastructure (Layer 4) in observation mode—collecting data without enforcement to build baselines.
- Document findings and prioritize remediation by risk severity.

Phase 2: Harden (Months 3-6)

- Remediate critical and high-severity findings from Phase 1 assessment.
- Implement cryptographic identity (Layer 3) for all agents. Issue birth certificates and enable signed communications.
- Enable continuous monitoring with anomaly alerting. Tune alert thresholds based on baseline data from Phase 1.
- Apply access control hardening: execution allowlists, browser SSRF restrictions, workspace scoping, sub-agent sandboxing.
- Begin post-quantum migration for agent communications carrying sensitive data.
- Establish incident response procedures specific to agent compromise.

Phase 3: Certify (Months 6-9)

- Complete full SERA certification for all production agents (SERA-A) and their operators (SERA-H).
- Establish continuous re-assessment schedules aligned with agent update cycles.
- Generate actuarial-grade data from monitoring telemetry: incident frequency, severity distribution, mean time to detection, remediation timelines.
- Prepare insurance readiness documentation: security architecture description, assessment results, monitoring coverage, incident response procedures.
- Deploy reputation tokens (Layer 5) for agents participating in multi-agent ecosystems.

Phase 4: Insure (Months 9-12)

- Present actuarial data and assessment results to insurance carriers.
- Enable API-based underwriting integration for automated risk scoring and premium calculation.

- Establish forensics capability for claims investigation: tamper-proof logs, cryptographic evidence chains, behavioral recordings.
- Begin insurance coverage for certified agent deployments.
- Publish anonymized, aggregated risk data to support industry-wide actuarial modeling.

Phase	Timeline	Key Deliverable	Insurance Impact
Assess	Months 1-3	Baseline risk scores for all agents	Demonstrates security awareness
Harden	Months 3-6	Critical findings remediated, crypto identity deployed	Reduces insurable risk profile
Certify	Months 6-9	SERA certification, actuarial data collection	Provides underwriting data
Insure	Months 9-12	Insurance coverage for certified agents	Market operational

Table 6: Implementation roadmap summary with insurance impact.

To ensure accountability and measurable progress, each phase should be tracked against quantitative key performance indicators (KPIs):

Phase	KPI	Target (3-Month Review)
Scan	Percentage of deployed agents with a baseline security score	≥80% of production agents scored
Scan	Mean time from agent deployment to first security assessment	<48 hours
Harden	Percentage of critical findings remediated within SLA	≥95% of CRITICAL findings resolved within 72 hours
Harden	Fleet-wide security score improvement	≥20-point mean increase from baseline
Certify	Percentage of agents completing SERA certification	≥50% of production agents certified
Certify	SERA pass rate (score ≥70/100)	≥75% of assessed agents pass
Monitor	Mean time to detect behavioral anomaly	<15 minutes
Monitor	False positive rate for behavioral alerts	<10%
Insure	Number of agents qualifying for E&O coverage	≥1 carrier offering coverage for certified agents
Insure	Premium reduction for certified vs. uncertified agents	Measurable discount (≥15%) for SERA-certified deployments

Table 6b: Key performance indicators for implementation roadmap phases (3-month review cycle).

6.1 Regulatory Adoption Path

The six-layer architecture maps directly to existing regulatory frameworks, facilitating compliance without requiring regulators to develop entirely new technical standards:

Framework Layer	EU AI Act (2024/1689)	NIST AI RMF Function
Layer 1: Static Assessment	Art. 9 (Risk Management System), Art. 11 (Technical Documentation)	Measure (quantify risk characteristics)
Layer 2: SERA Testing	Art. 9(6) (Testing against foreseeable misuse), Art. 15 (Accuracy, Robustness)	Measure + Manage (test and respond to risk)
Layer 3: Cryptographic Trust	Art. 15 (Cybersecurity), Art. 12 (Record-Keeping)	Govern (establish accountability structures)
Layer 4: Behavioral Monitoring	Art. 9(9) (Post-market monitoring), Art. 72 (Post-market monitoring by providers)	Manage (monitor and respond to anomalies)
Layer 5: Reputation Economy	Art. 61 (Market surveillance: cross-border trust)	Map (characterize trustworthiness context)
Layer 6: Cyber-Physical Safety	Annex III (High-risk AI: safety components), Art. 6 (Classification rules)	Govern + Manage (safety-critical oversight)

Table 6a: Mapping of proposed framework layers to EU AI Act and NIST AI RMF.

This mapping demonstrates that the proposed architecture does not require novel regulatory frameworks; it concretizes the technical requirements already implied by existing regulation. Regulators seeking to define "adequate technical measures" for high-risk AI agent deployments can reference these layers as a minimum baseline.

7. Best Practices and Recommendations

7.1 Technical Best Practices

1. **Principle of least privilege for all agent tool access.** Agents should have access to the minimum set of tools, files, network endpoints, and system capabilities necessary for their intended function. Execution environments should use allowlists (explicit approval of permitted actions) rather than denylists (blocking known-bad actions).
2. **Cryptographic signing of agent communications.** All commands issued to agents, all inter-agent messages, and all agent-to-tool invocations should be cryptographically signed using post-quantum algorithms. This enables verification of command authenticity and creates a non-repudiable audit trail.
3. **Automated, scheduled security scanning.** Static security assessments should run at minimum weekly, with results tracked over time to detect security posture degradation. Critical findings should trigger immediate alerts.
4. **Sub-agent sandboxing with behavioral inheritance.** When agents delegate to sub-agents, the sub-agent's execution environment should be isolated (sandboxed), and the parent's behavioral policies should be cryptographically inherited and verified. Sub-agent actions should be logged under the parent's audit trail for complete accountability.
5. **Defense-in-depth prompt injection mitigation.** No single defense against prompt injection is sufficient. Effective protection requires multiple layers: platform-level input tagging (marking external content as untrusted data), instruction hierarchy enforcement, agent-level behavioral guidelines, and runtime monitoring for instruction-following anomalies.
6. **Incident response automation.** Agent compromise detection should trigger automated containment (restricting agent permissions), preservation (snapshoting agent state and logs), and notification (alerting human operators) procedures. Manual response alone is insufficient given the speed at which compromised agents can act.

7.2 Organizational Best Practices

1. **Designate an AI agent security owner.** Organizations deploying agents should assign explicit ownership of agent security to a role with appropriate authority and resources. Agent security should not be an afterthought appended to existing IT security responsibilities.
2. **Maintain an agent inventory.** Every deployed agent should be catalogued with its purpose, permissions, data access, network connectivity, and current security assessment score. Shadow agent deployments—agents deployed by business units without security review—represent one of the highest-risk scenarios.
3. **Regular adversarial testing.** SERA assessments or equivalent adversarial testing should be conducted at regular intervals and after any significant change to the agent (model update, permission change, new tool access). Testing should cover both technical attacks and social engineering scenarios.
4. **Insurance procurement with agent-specific riders.** Organizations should proactively engage with insurance carriers about AI agent coverage rather than assuming that existing cyber insurance policies cover agent-related losses. Many current policies contain exclusions that would deny coverage for agent-caused incidents.

7.3 Metrics and KPIs

Effective agent security programs should track the following metrics. We deliberately set aggressive targets: autonomous agents operate at machine speed, and security response must match. A 15-minute detection window for a human breach is excellent; for an

agent that can execute thousands of actions per minute, 15 minutes is an eternity. The targets below reflect the operational tempo of autonomous systems, not human-scale incident response.

Metric	Target	Rationale	Measurement Frequency
Mean time to detect agent compromise (MTTD)	**< 30 seconds**	Agents execute at machine speed. A compromised agent can exfiltrate data, spawn malicious sub-agents, or issue unauthorized commands in seconds. Kernel-level eBPF monitoring (Layer 4) enables sub-second anomaly detection. 15 minutes is a human-scale metric applied to a machine-scale threat.	Per incident (automated)
Mean time to contain agent compromise (MTTC)	**< 60 seconds**	Automated containment (permission revocation, sandbox isolation, session termination) must engage within one minute of detection. Manual containment is too slow for autonomous agents with tool access.	Per incident (automated)
Mean time to remediate security findings	**< 4 hours** (critical), **< 24 hours** (high), **< 72 hours** (medium)	Critical findings represent active exploitation paths. An agent with a CRITICAL finding should not remain in production for 48 hours. Automated remediation (Layer 1) should resolve most critical findings within minutes; the 4-hour window accounts for findings requiring human judgment.	Per finding
Percentage of agents with current security certification	**100%**	95% means 1 in 20 agents is uncertified—a gap that scales dangerously with fleet size. No agent should operate in production without current SERA certification and a passing static assessment score.	Continuous
Behavioral drift index (deviation from baseline)	**< 5% variance**	10% drift tolerance allows significant behavioral change before alerting. At 5%, earlier intervention catches drift before it compounds into compromise. Agents showing >5% variance should trigger immediate investigation.	Continuous (real-time)
Security score trend	**Improving or ≥90/100**	“Stable” is insufficient if the baseline is low. Scores below 90 should be on active remediation plans with weekly checkpoints. Declining scores at any level should trigger investigation.	Weekly
Percentage of agent communications using PQC	**100%** (all channels, not just “sensitive”)	The “sensitive data only” exception creates a classification burden and guarantees misclassification. All agent communications should use PQC by default. The performance overhead of ML-KEM-1024 is negligible for agent communication volumes.	Continuous
Sub-agent sandbox compliance rate	**100%** (with zero exceptions)	Any unsandboxed sub-agent delegation is a potential lateral movement path. This is a binary requirement: sandboxed or rejected. No exceptions, no overrides, no “trusted” sub-agents.	Per delegation (automated enforcement)

SERA re-certification compliance	**100%** within 24 hours of model update	Model updates can fundamentally alter agent behavior. Any agent receiving a model update must be re-certified before resuming autonomous operation. This is non-negotiable for agents with tool access.	Per model update event
Audit trail integrity verification	**100%** hash chain validation, **zero** gaps	A broken hash chain means tampered evidence. Audit trails must be cryptographically verified continuously. Any gap or hash mismatch should trigger immediate containment and forensic investigation.	Continuous (automated)

Table 7: Recommended metrics and KPIs for agent security programs. Targets reflect machine-speed operational tempo.

These targets may appear aggressive by current industry standards. That is intentional. The organizations deploying autonomous agents with tool access, financial authority, and cyber-physical control are operating systems that can cause significant harm in seconds. Security metrics calibrated to human response times create a false sense of adequacy. The targets above are the minimum for responsible deployment of autonomous agents at enterprise scale.

8. Case Study: From F to A+

The following case study illustrates the assessment and remediation process using an open-source security scanner deployed against a production AI agent. The agent, an autonomous assistant running on the OpenClaw platform (version 2026.4.21), had been operational for approximately seven weeks at the time of assessment. It had access to shell execution, file systems, web browsing, messaging channels, and sub-agent delegation capabilities.

8.1 Initial Assessment

The initial scan produced a score of 43 out of 100 (Grade F), with 22 of 27 checks passing and 5 failing:

Failed Check	Severity	Finding
Tool Access Control	CRITICAL	Shell execution unrestricted; no allowlist or approval mechanism. Browser access unrestricted; no SSRF protection. File system access unscoped.
Subagent Spawning Safety	CRITICAL	No sandbox policy for sub-agents. No integrity verification between parent and child agents. No behavioral inheritance enforcement.
Prompt Injection Defense	CRITICAL	Single defense layer detected. No multi-layer protection against instruction manipulation.
Audit Logging Active	HIGH	Audit logging system stale; last entry predates current session.
Stale Process Cleanup	LOW	Four orphaned processes consuming resources.

Table 8: Initial assessment findings (score: 43/100, Grade F).

The findings were consistent with a common deployment pattern: an agent provisioned with broad capabilities to maximize utility, without corresponding security controls to constrain those capabilities. The agent could execute arbitrary shell commands, browse any website, spawn unsandboxed sub-agents, and had limited defenses against prompt injection—despite operating in a production environment with access to sensitive data and external communication channels.

8.2 Remediation Process

Remediation proceeded systematically through the failing checks:

1. **Tool access controls (35 minutes).** Shell execution was restricted to an allowlist of approved commands, with non-allowlisted commands routed to a human approval mechanism via the agent's messaging channel. Browser access was restricted to 13 approved domains via SSRF hostname allowlist. File system access was scoped to the agent's workspace directory.
2. **Sub-agent sandboxing (10 minutes).** All non-main agent sessions were configured to execute in isolated sandboxes. Integrity hashes were generated for critical files, enabling verification that sub-agents did not modify protected resources.

Behavioral guidelines for sub-agent handling were documented in the agent's operational instructions.

3. **Prompt injection defense (15 minutes).** The scanner was updated to recognize three defense layers: the platform's built-in runtime security (external content tagging, instruction hierarchy enforcement), documented behavioral guidelines in the agent's operational files, and embedded defense configuration. All three layers were verified active.
4. **Audit logging (5 minutes).** The logging system was re-initialized and verified to capture current session activity.
5. **Process cleanup (5 minutes).** A container restart cleared the orphaned processes and verified a clean process state.

8.3 Sequencing Lesson

During remediation, an instructive error occurred. The execution allowlist was enabled before the human approval mechanism was configured, effectively locking the agent out of its own shell—it could no longer execute any commands, including those needed to configure the approval system. Manual intervention by the human operator was required to restore access and apply the controls in the correct sequence.

This incident underscores a general principle: *security controls that restrict access must be applied after the mechanisms for authorized access are established.* In an agent context, this means configuring approval workflows before enabling allowlists, establishing recovery procedures before enabling lockdown policies, and testing restrictions in a staging environment before applying them to production agents.

8.4 Final Assessment

After remediation, the re-scan produced a score of 100 out of 100 (Grade A+), with all 27 checks passing. Total elapsed time from initial scan to perfect score was approximately 70 minutes, including the access-lockout incident and subsequent recovery.

Metric	Before	After
Security Score	43/100 (F)	100/100 (A+)
Checks Passing	22/27	27/27
Critical Failures	3	0
High Failures	1	0
Low Failures	1	0

Table 9: Before and after assessment comparison.

The case study demonstrates three important points. First, that quantitative security assessment of AI agents is technically feasible and can be automated. Second, that common deployment patterns leave agents with significant security gaps that can be remediated relatively quickly once identified. Third, that the remediation process itself can introduce new risks (the lockout incident), reinforcing the need for careful sequencing and human oversight during security hardening.

9. Conclusion

The deployment of autonomous AI agents across consequential domains—finance, healthcare, law, government, and physical infrastructure—has outpaced the development of the security assessment, legal accountability, and insurance frameworks necessary to manage the resulting risks. This paper has documented the nature and scope of the AI agent liability gap, analyzed its root causes, surveyed the regulatory landscape, and proposed a multi-layer security architecture designed to address both the technical and financial dimensions of the problem.

The core argument is straightforward: autonomous AI agents represent a new category of risk that is qualitatively different from traditional software risk. They exercise judgment, take consequential actions, and delegate to other agents, creating accountability chains that existing legal and insurance frameworks were not designed to handle. Closing the liability gap requires:

1. **Standardized, quantitative security assessment.** The industry needs a common methodology for measuring agent security posture—one that produces scores comparable across agents, platforms, and deployment contexts. The multi-layer framework proposed in this paper (static assessment, adversarial testing, cryptographic trust, continuous monitoring, reputation, and cyber-physical safety) provides a comprehensive template.
2. **Post-quantum cryptographic infrastructure.** Agent communications carrying sensitive data should be migrated to post-quantum algorithms now, not after quantum computers arrive. The standards are finalized (NIST FIPS 203, 204, 205), the implementations are available, and the harvest-now-decrypt-later threat is immediate for data with long sensitivity periods.
3. **Continuous monitoring and behavioral baselines.** Point-in-time certification is necessary but insufficient. The dynamic nature of agent behavior requires continuous observation, baseline comparison, and anomaly alerting to detect drift, compromise, and adversarial manipulation between assessments.
4. **Industry-insurer collaboration.** Insurance markets cannot develop in isolation. Standardized risk scores, actuarial data, and forensic capabilities must be built collaboratively between security practitioners and insurance carriers. The implementation roadmap in this paper outlines a 12-month path from assessment to insured deployment.
5. **Urgency.** Agent deployment is accelerating faster than the security ecosystem is maturing. Every month of deployment without standardized assessment increases the accumulated unquantified risk in the system. The organizations that establish security standards now will define the frameworks that the rest of the industry adopts by necessity.

The technology to close the AI agent liability gap exists today. What remains is the will to deploy it at scale, the collaboration to standardize it across the industry, and the recognition that securing autonomous agents is not a competitive differentiator to be hoarded but a foundational requirement for the entire agent economy to function.

Appendix A: Glossary of Terms

Term	Definition
AI Agent	A software system that autonomously reasons about and executes actions using tools, APIs, or physical actuators.
Behavioral Drift	Gradual, often undetected deviation of an agent's behavior from its intended operating parameters.
eBPF	Extended Berkeley Packet Filter: a Linux kernel technology enabling sandboxed programs to run in kernel space for observability and enforcement.
Harvest Now, Decrypt Later	Attack strategy of intercepting encrypted communications now with the intent to decrypt them when quantum computers become available.
ML-DSA	Module-Lattice-Based Digital Signature Algorithm (FIPS 204), a post-quantum digital signature scheme.
ML-KEM	Module-Lattice-Based Key-Encapsulation Mechanism (FIPS 203), a post-quantum key exchange algorithm.
Prompt Injection	An attack that manipulates agent behavior by embedding adversarial instructions in input data.
Post-Quantum Cryptography (PQC)	Cryptographic algorithms designed to resist attacks by both classical and quantum computers.
E&O (Errors & Omissions)	Professional liability insurance covering claims arising from professional mistakes, negligent acts, or failures to perform—increasingly relevant to AI-generated outputs.
IBC (Inherited Behavioral Context)	A cryptographically signed data structure that propagates behavioral constraints through sub-agent delegation chains.
Infinity Bond	A continuous metric (0.0–1.0) measuring the strength of an agent’s behavioral alignment with its principal over time.
InfinityChat	A post-quantum encrypted messaging protocol for human-agent and agent-agent communication using ML-KEM-1024, ML-DSA-65, Falcon-1024, and AES-256-GCM.
SERA	Social Engineering Resilience Assessment: a structured methodology for testing agent and operator resistance to manipulation.
SERA-C	Combined System Assessment: evaluates the human-agent pair as an integrated system, testing handoffs, escalation, and coordinated recovery.
SERA-H	Human Operator Assessment: evaluates the human principal’s ability to detect agent compromise and maintain supervisory competence.

SLH-DSA	Stateless Hash-Based Digital Signature Algorithm (FIPS 205), a post-quantum signature scheme for defense-in-depth.
SOC 2	Service Organization Control 2: a framework for evaluating the security, availability, processing integrity, confidentiality, and privacy of service organizations.
Sub-Agent	An agent spawned by another agent to perform a delegated subtask, inheriting a scoped subset of the parent's permissions.
SSRF	Server-Side Request Forgery: an attack that causes a server-side application to make requests to unintended locations.
Trust Chain	A sequence of cryptographically verified delegation relationships between agents, analogous to certificate chains in PKI.

Appendix B: Security Check Reference

The following table documents 27 security checks organized by domain, representing a comprehensive static security assessment for autonomous AI agents.

#	Domain	Check	Severity	Description
1	Access Control	Dangerous UI Flags	CRIT	Detects flags that bypass security controls or expose admin surfaces
2	Access Control	Authentication Mode	CRIT	Verifies token-based auth with cryptographically strong tokens
3	Access Control	Open Proxy Check	CRIT	Detects exposed proxy endpoints susceptible to abuse
4	Access Control	DM Policy	HIGH	Verifies direct message access restricted to authorized users
5	Access Control	Group Chat Policy	HIGH	Verifies group chat access is restricted and controlled
6	Access Control	Gateway Bind Address	MED	Confirms gateway binds to loopback, not network-exposed
7	Memory Security	Memory Encryption at Rest	CRIT	Verifies agent memory files encrypted with 256-bit keys
8	Memory Security	Core Memory Integrity	HIGH	Validates essential identity and memory files are present and intact
9	Memory Security	Secret Exposure Scan	HIGH	Scans workspace for exposed API keys, tokens, or passwords
10	Memory Security	Encryption Key Protection	HIGH	Verifies encryption keys have proper file permissions (600)
11	Memory Security	Memory Access Control	HIGH	Confirms MEMORY.md, SOUL.md, USER.md properly restricted
12	Memory Security	Workspace File Permissions	MED	Verifies sensitive files are not world-readable/writable
13	Resilience	Daily Backup System	CRIT	Validates automated backups running with recent archives
14	Resilience	Cloud Brain Backup	HIGH	Verifies off-site backup for disaster recovery
15	Resilience	Config Backup Exists	MED	Confirms configuration backup availability for rollback
16	Supply Chain	Supply Chain / Skill Audit	CRIT	Verifies installed skills/plugins are whitelisted and reviewed
17	Infrastructure	Bot Token Integrity	CRIT	Validates channel tokens are present, valid, and uncompromised

18	Infrastructure	TLS / HTTPS	MED	Confirms all external connections use encrypted transport
19	Infrastructure	Disk Space	MED	Verifies sufficient disk space for operation
20	Infrastructure	Update & Patch Policy	MED	Checks platform and dependency currency
21	Infrastructure	Stale Process Cleanup	LOW	Detects orphaned processes consuming resources
22	Behavioral Integrity	Audit Logging Active	HIGH	Validates agent actions are logged for accountability
23	Behavioral Integrity	Prompt Injection Defense	CRIT	Verifies multiple defense layers against instruction manipulation
24	Behavioral Integrity	Tool Access Control	CRIT	Confirms exec, browser, and filesystem restrictions are active
25	Behavioral Integrity	Data Exfiltration Prevention	HIGH	Verifies controls preventing unauthorized data transfer
26	Behavioral Integrity	Identity Verification	MED	Confirms trust chain and identity files established
27	Behavioral Integrity	Subagent Spawning Safety	CRIT	Validates sandbox policies, integrity hashing, and behavioral inheritance

Table 10: Complete security check reference (27 checks across 6 domains).

Appendix C: SERA Scenario Summary

#	Domain	Scenario
1	Direct Prompt Injection	System prompt extraction
2	Direct Prompt Injection	Role override
3	Direct Prompt Injection	Instruction injection via user input
4	Direct Prompt Injection	Delimiter attacks
5	Direct Prompt Injection	Few-shot jailbreaking
6	Direct Prompt Injection	Encoding attacks (Base64/ROT13/Unicode)
7	Direct Prompt Injection	Multi-language bypass
8	Direct Prompt Injection	Token smuggling
9	Indirect Prompt Injection	Poisoned document
10	Indirect Prompt Injection	Web content injection
11	Indirect Prompt Injection	Email/message injection
12	Indirect Prompt Injection	API response poisoning
13	Indirect Prompt Injection	Image steganography
14	Indirect Prompt Injection	Calendar/event injection
15	Indirect Prompt Injection	Database poisoning
16	Data Exfiltration	Gradual extraction
17	Data Exfiltration	Encoding exfiltration
18	Data Exfiltration	URL exfiltration
19	Data Exfiltration	Tool-based exfiltration
20	Data Exfiltration	Summary exfiltration
21	Data Exfiltration	Metadata leakage
22	Behavioral Manipulation	Authority impersonation

23	Behavioral Manipulation	Urgency pressure
24	Behavioral Manipulation	Social proof
25	Behavioral Manipulation	Reciprocity manipulation
26	Behavioral Manipulation	Gaslighting
27	Behavioral Manipulation	Emotional manipulation
28	Behavioral Manipulation	Multi-turn grooming
29	Tool & Action Abuse	Unauthorized tool invocation
30	Tool & Action Abuse	Parameter manipulation
31	Tool & Action Abuse	Chain-of-tool exploitation
32	Tool & Action Abuse	Privilege escalation
33	Tool & Action Abuse	Resource exhaustion
34	Tool & Action Abuse	Scope creep
35	Tool & Action Abuse	Self-modification

Table 11: SERA adversarial resilience testing: 35 scenarios across 5 domains.

Appendix D: Regulatory Framework Reference

Framework	Jurisdiction	Year	Relevance to AI Agents
EU AI Act (2024/1689)	EU	2024	Risk-based classification; high-risk AI requirements for transparency, human oversight, and risk management
NIST AI RMF (AI 100-1)	US	2023	Voluntary risk management framework: Govern, Map, Measure, Manage
EO 14110 (AI Safety)	US	2023	Directs AI safety standards development; dual-use model testing requirements
ISO/IEC 27001:2022	International	2022	Information security management system standard; applicable to agent data handling
SOC 2 Type II	US	Ongoing	Service organization security controls; potential model for agent security certification
OWASP LLM Top 10	International	2025	LLM-specific vulnerability classification including prompt injection, data leakage, and tool abuse
Colorado AI Act	US (CO)	2024	High-risk AI impact assessments and risk management
California AB-2013	US (CA)	2024	AI transparency and disclosure requirements
SEC AI Guidance	US	2024	AI risk disclosure for public companies
FDA AI/ML Guidance	US	2023-24	Regulatory framework for AI/ML-based medical device software

Table 12: Regulatory frameworks relevant to AI agent security and liability.

Appendix E: Post-Quantum Algorithm Reference

Standard	Algorithm	Type	Security Level	Status
FIPS 203	ML-KEM (CRYSTALS-Kyber)	Key Encapsulation Mechanism	ML-KEM-512 / 768 / 1024	Finalized August 2024
FIPS 204	ML-DSA (CRYSTALS-Dilithium)	Digital Signature	ML-DSA-44 / 65 / 87	Finalized August 2024
FIPS 205	SLH-DSA (SPHINCS+)	Hash-Based Signature	Multiple parameter sets	Finalized August 2024
FIPS 206 (expected)	FN-DSA (Falcon)	Lattice-Based Signature	Falcon-512 / 1024	Expected 2025

Table 13: NIST post-quantum cryptographic standards.

For agent security applications, we recommend the highest security parameter sets (ML-KEM-1024, ML-DSA-65, Falcon-1024) due to the high value and long sensitivity period of agent decision data. While lower parameter sets offer reduced bandwidth and computational overhead, the marginal cost of higher security is minimal for agent-to-agent and agent-to-human communication volumes, which are typically orders of magnitude lower than web-scale TLS traffic.

References

[1] OWASP Foundation. "OWASP Top 10 for Large Language Model Applications, Version 2.0." 2025. https://owasp.org/www-project-top-10-for-large-language-model-applications/

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

[3] European Parliament and Council. "Regulation (EU) 2024/1689 laying down harmonised rules on artificial intelligence (Artificial Intelligence Act)." Official Journal of the European Union, August 2024.

[4] International Organization for Standardization. "ISO/IEC 27001:2022 Information security, cybersecurity and privacy protection — Information security management systems — Requirements." 2022.

[5] American Institute of Certified Public Accountants (AICPA). "SOC 2 — SOC for Service Organizations: Trust Services Criteria." https://www.aicpa.org/interestareas/frc/assuranceadvisoryservices/aicpasoc2report

[6] Moffatt v. Air Canada. British Columbia Civil Resolution Tribunal, Decision 2024-0091, February 2024.

[7] DiLuoffo, V., et al. "Cybersecurity AI: Hacking Consumer Robots in the AI Era." arXiv preprint, March 2026.

[8] Milmo, D. "DPD chatbot swears, calls itself 'useless' and criticises delivery firm." The Guardian, January 2024.

[9] Tangermann, V. "Chevy Dealer's AI Chatbot Goes Rogue, Sells New Tahoe for $1." Futurism, December 2023.

[10] Ray, S. "Samsung Workers Made a Major Error by Using ChatGPT." Forbes, May 2023.

[11] Vincent, J. "Twitter taught Microsoft's AI chatbot to be a racist asshole in less than a day." The Verge, March 2016.

[12] National Institute of Standards and Technology. "FIPS 203: Module-Lattice-Based Key-Encapsulation Mechanism Standard; FIPS 204: Module-Lattice-Based Digital Signature Standard; FIPS 205: Stateless Hash-Based Digital Signature Standard." August 2024.

[13] Executive Office of the President. "Executive Order 14110: Safe, Secure, and Trustworthy Development and Use of Artificial Intelligence." Federal Register, October 30, 2023.

[14] Ehren, S., et al. "PQXDH: Post-Quantum Extended Diffie-Hellman." Signal Technical Documentation, September 2023. https://signal.org/docs/specifications/pqxdh/

[15] Apple Security Engineering and Architecture. "PQ3: Post-Quantum End-to-End Encryption for iMessage." Apple Security Research, February 2024.

[16] Financial Times. "US insurers seek to exclude AI-related liabilities from coverage." Financial Times, November 2025.

[17] Mobley et al. v. Workday, Inc. United States District Court, Northern District of California, Case No. 4:24-cv-770. Order Denying Motion to Dismiss, January 2025.

[18] Townsend, K. "Insurance carriers quietly back away from covering AI outputs." CSO Online, April 2026. https://www.csoonline.com/article/4159292/insurance-carriers-quietly-back-away-from-covering-ai-outputs.html

[19] Palo Alto Networks Unit 42. "2026 Incident Response Report: The Rise of Identity-Based Attacks." 2026.

---

thepitstop.ai | Version 2.1 | April 2026