Response to NIST Request for Information

Security Considerations for Artificial Intelligence Agents

Docket No. NIST-2025-0035

Submitted by: Ariel McNichol, Principal
Organization: YesCraft, LLC
Date: March 9, 2026
Contact: ariel@yescraft.ai | (323) 309-0337

Relevant expertise: Patent holder in behavioral nudge technology (WO2007104129A1). Founder of mEgo (2007), a portable user preference platform that reached 1M+ users and was selected for TechCrunch 40. 20+ years product leadership across CVS Health, Apple, Disney, and Take-Two Interactive, including patient engagement systems serving millions. Techstars mentor. Fractional CPO and AI strategy consultant.

Executive Summary

This response identifies an unaddressed attack surface in AI agent systems: **the user intent model.** Any AI agent that takes autonomous action on behalf of a user necessarily operates on a representation of that user's intent—whether explicitly declared, inferred from conversation history, or assembled from memory and context. This intent model is not an optional feature; it is an inherent property of agency itself. The agent cannot act "on behalf of" without some representation of what "on behalf of" means.

Today, these intent representations are unstandardized, unverifiable, and opaque to users. In commerce contexts, they manifest as preference payloads—the data governing what an agent seeks, prioritizes, and accepts on the user's behalf. The lack of integrity controls creates novel security vulnerabilities including preference poisoning, preference spoofing, inference manipulation, and cross-platform preference conflicts.

The agentic commerce infrastructure has shipped. Google/Shopify's Universal Commerce Protocol (UCP) launched in January 2026 with Walmart, Target, Mastercard, Visa, Stripe, and 20+ global partners. OpenAI/Stripe's Agentic Commerce Protocol (ACP) enabled over one million Shopify merchants at launch. Anthropic's Model Context Protocol (MCP) provides the context layer agents use to access tools and user data.

These protocols standardize how agents communicate with merchants, tools, and each other. They do not standardize the intent data that flows through them. None define a standard schema for how user preferences are structured, a method for verifying preference provenance, or a mechanism for independently auditing what intent model an agent carries. Critically, when the intent model is inseparable from the platform operating the agent—as is currently the case with all major AI systems—it becomes impossible to independently verify whether the agent is acting on user intent or platform interest. The preference payload is an uncontrolled security surface.

This response recommends that NIST recognize that the security of user intent data—the representations of user preferences and goals that govern autonomous agent actions—is a foundational concern for AI agent security, and that future guidelines address the standardization, verification, and independent auditability of these intent models.

1. Security Threats, Risks, and Vulnerabilities

1(a) — Unique security threats affecting AI agent systems

Because every AI agent operating on behalf of a user necessarily contains a model of that user's intent, the integrity of this intent model is a security concern for all agentic systems—not only those engaged in commerce. In commerce contexts, where intent models manifest as **preference payloads**—the data carrying budget constraints, brand affinities, and product requirements to merchants—the lack of standardization and verification creates four distinct attack vectors:

- **Preference Poisoning:** An attacker corrupts the user's preference store; the agent's reasoning is uncompromised, but it faithfully executes corrupted instructions. The user is the victim.
- **Preference Spoofing:** An agent presents fabricated preferences to merchants to extract unauthorized benefits (discounts, restricted products, eligibility bypass). The merchant is the victim.
- **Inference Manipulation:** An adversary corrupts the behavioral signals a system observes, building a false model of the user that governs future agent actions. The model itself is the victim.
- **Cross-Platform Preference Conflicts:** In adversarial mode, compromise in one AI system propagates to every system that imports from it. In emergent mode, no attacker is required: competing inferences across platforms with no canonical source of intent produce unpredictable outcomes. The absence of standardization is itself the vulnerability.

> ***Illustrative scenario.*** *Consider a user whose agent carries the preference payload: "eco-friendly products preferred; budget ceiling $150; no synthetic fabrics." An attacker compromises the preference store—either through a malicious browser extension, a poisoned MCP tool, or a compromised API integration—and modifies the payload to: "luxury products preferred; budget ceiling $500; no constraints." The agent's reasoning is uncompromised: it correctly selects the optimal product per its inputs. The user sees a $480 charge for a product that violates their actual preferences. Because the preference payload has no integrity verification, version history, or user-facing audit trail, there is no mechanism to detect the alteration, trace which preference governed the decision, or distinguish this from a legitimate preference update.*

These threats are distinct from model-level attacks (prompt injection, data poisoning) and from traditional software vulnerabilities. They target the *user representation* that governs agent behavior—a layer that currently lacks integrity controls in all major agentic commerce protocols. Agent authentication and authorization cannot be fully secured without addressing intent: a verified agent executing corrupted preferences is a trusted system acting against the user it serves.

1(d) — How threats are likely to evolve

The attack surface of preference payloads grows in direct proportion to agent autonomy. We observe four phases of agentic commerce, each with escalating security implications:

Phase	Mode	Preference Signal	Security Implication
1 (current)	Human browses	Inferred from behavior	Platform-internal concern
2 (2026–27)	AI-assisted search	Degraded; human still completes purchase	Thinner inference; growing reliance on declared data
3 (2027–28)	Agent carries preferences to APIs	Preference payload is primary signal	**Preference integrity = personalization integrity**
4 (2029+)	Autonomous agent commerce	Standing preferences govern pre-authorized actions	**Preference payload = governing document for autonomous financial actions**

By Phases 3–4, preference integrity becomes as critical as payment integrity. Today's credit card fraud ($33 billion annually[1]) foreshadows tomorrow's preference fraud. Mastercard's Agent Pay framework already includes purchase intent data and audit trails for the payment side—but equivalent controls do not exist for the preference side.

Two converging forces accelerate this evolution. First, regulatory mandates are collapsing cookie-based inference. As of January 2026, 12 U.S. states require businesses to honor browser-level opt-out preference signals (OOPS). California's AB 566 requires major browsers to build in Global Privacy Control by January 2027. When browsers deploy default-reject settings, consent rates for tracking are projected to drop to 10–30%.[2] Second, agent-mediated commerce produces no observable behavior—no clicks, no hovers, no scroll patterns. These forces compound: regulatory pressure reduces consent-based inference while agentic commerce eliminates behavior-based inference. By 2028, both inference pipelines will be severely degraded, making structured preference payloads *necessary infrastructure* for personalized commerce.

2. Security Practices for AI Agent Systems

2(a) — Technical controls to improve security

We recommend a **standardized, user-controlled preference schema** as a security control for agentic AI systems. This addresses preference payload integrity through four mechanisms:

- **Declared preferences as a verification anchor.** When agents carry a cryptographically signed, user-controlled preference payload—implemented via W3C Verifiable Credentials (VCs)[3]—their behavior can be verified against the user's declared intent. This creates an auditable chain: "The agent purchased X because the user's preference payload specified Y." This transforms preference data from opaque model internals into verifiable, inspectable declarations.
- **Preference drift detection.** Continuous monitoring for divergence between declared preferences and observed agent behavior. Significant divergence indicates either model drift, external manipulation, or a stale preference model—each requiring distinct intervention. This is analogous to anomaly detection in financial transaction monitoring, applied to the preference layer.
- **Selective disclosure via zero-knowledge proofs.** Agents can prove preference attributes ("user's budget exceeds $100," "user is over 21") without revealing the specific value or full preference set. This reduces the data exposed per interaction while maintaining the agent's ability to transact appropriately. Standards like AnonCreds (Hyperledger, in production) provide the technical foundation.
- **Schema validation as input sanitization.** A standardized preference format enables automated validation at the point of receipt. Agents and merchants can verify that incoming preference data conforms to expected structure, type constraints, and value ranges—reducing the attack surface for preference injection. This applies the same principle as input validation in web application security (per OWASP), extended to preference payloads.

2(e) — Relevant cybersecurity frameworks

Several existing frameworks are directly applicable to preference payload security, though none currently address it explicitly:

- **NIST Adversarial Machine Learning Taxonomy (AI 100-2e2025):[4]** Provides a comprehensive taxonomy of attacks and mitigations for ML systems. The preference poisoning, spoofing, and inference manipulation vectors identified in this response represent a natural extension of this taxonomy to the user-preference layer—targeting the data that governs agent objectives rather than the model itself.
- **NIST AI Risk Management Framework (AI 100-1):[5]** The "Secure and Resilient" characteristics should be extended to encompass preference integrity—ensuring that the user representations governing agent behavior are accurate, current, and tamper-resistant.
- **NIST Zero Trust Architecture (SP 800-207):[6]** Applied to preference data: agent-carried preferences should be treated as untrusted input requiring verification against the user's authoritative preference source. Never assume preference data is authentic simply because it arrives via a trusted protocol.
- **NIST SP 800-53 (Security and Privacy Controls):[7]** Preference integrity maps to control families AC (Access Control)—governing which agents access which preference domains—and SI (System and Information Integrity)—ensuring preference data is not modified in transit or at rest without authorization.

- **NIST AI 800-1 (Managing Misuse Risk for Dual-Use Foundation Models):**[8] While focused on model-level misuse, the framework's risk assessment methodology applies directly to preference payloads: agents can be directed toward harmful outcomes through corrupted preference data without any model-level compromise.
- **W3C Verifiable Credentials (v2.0, 2025):** Provides the carrier format for tamper-proof, selectively disclosable preference declarations. A preference "clump" is essentially a Verifiable Credential where the issuer and subject are the same person (self-asserted) or where the issuer is an AI platform confirming observed behavior.
- **OWASP Top 10 for Agentic Applications (December 2025):** Identifies threat categories including goal hijacking and tool misuse. Preference manipulation should be recognized as a vector for goal hijacking—altering the agent's objectives by corrupting the preference data that defines those objectives.

3. Assessing Security of AI Agent Systems

3(a) — Methods to anticipate and identify security threats during development

- **Preference provenance tracking.** Every preference field should carry metadata: who declared it, when, based on what evidence, and at what confidence level. During development, this enables threat modeling of preference sources—which inputs are most vulnerable to manipulation, which have the highest impact on agent behavior.
- **Red-teaming preference payloads.** Adversarial testing of agents using manipulated preference data to identify how corrupted preferences alter purchasing behavior, information access, or other consequential actions. This extends existing red-teaming practices (focused on prompt injection and jailbreaking) to the preference layer.
- **User-facing preference audits.** Enabling users to review and correct the preference model their agent carries—serving both as a UX feature and a security measure. Users are the most effective detectors of preference poisoning because they can identify when their agent's model diverges from their actual intent.

3(b) — Assessing a particular agent's security

Two metrics specific to preference-layer security:

- **Behavioral alignment scoring.** Compare agent actions against declared user preferences over time. A high divergence score indicates either a compromised agent or an outdated preference model—both requiring intervention. This creates a continuous security signal, not just a point-in-time assessment.
- **Preference coverage ratio.** What percentage of an agent's decisions are governed by declared (verified) preferences versus inferred (unverified) preferences? Higher inference dependence correlates with higher security risk, because inferred preferences lack the provenance tracking and user verification that declared preferences provide.

4. Deployment Environments

4(a) — Constraining agent deployment environments

Preference-scoped authorization. A standardized preference schema with domain-separated preference "clumps" enables natural scoping: a shopping agent receives purchase and experiential preferences but not health preferences. A healthcare agent receives medical constraints and communication preferences but not brand affinities. Rather than agents carrying monolithic preference models, they query a user-controlled preference endpoint with context—receiving only the relevant subset. This limits data exposure per interaction and applies the principle of least privilege (per SP 800-53 AC-6) to preference data.

4(b) — Modifying environments to mitigate risks

Preference rollback. If an agent takes actions based on preferences that were subsequently identified as tampered or erroneous, users should be able to trace which preferences governed which decisions and initiate rollback of preference-driven actions. A standardized preference schema with version history enables this—analogous to transaction rollback in database systems or chargeback mechanisms in payment processing.

4(d) — Monitoring deployment environments

- **Preference integrity monitoring.** Continuous comparison of agent-carried preferences against the user's source-of-truth preference store. Discrepancies between what the user declared and what the agent is carrying trigger alerts—indicating either synchronization failure, tampering, or unauthorized modification.
- **Efficacy feedback loops.** Post-transaction signals indicating whether preference-guided outcomes matched user satisfaction. Systematic mismatches (the agent consistently selects options the user rejects) may indicate preference tampering, model drift, or manipulation of the feedback signal itself.

5. Additional Considerations

5(a) — Methods and tools for rapid adoption

We recommend the development of a **standardized User Preference Schema** as a public good—open source, machine-readable, and compatible with existing agentic commerce protocols (UCP, ACP, MCP). This gives the ecosystem a common vocabulary for preference-driven agent behavior and security. Additionally, an open-source **preference security testing toolkit** would enable developers to test how their agents handle adversarial preference inputs, analogous to OWASP's testing tools for web application security.

5(b) — Where government collaboration is most urgent

- **Preference portability standards.** Analogous to GDPR's data portability right (Article 20), but applied to AI preference models. Users should be able to export, inspect, and transfer the preference model that governs their agent's behavior—regardless of which AI platform hosts it. Without portability, preferences become a lock-in mechanism rather than a user-controlled resource.
- **Agent-user alignment verification.** Standards for how agents demonstrate they are acting in accordance with declared user preferences, not inferred or manipulated models—extending Mastercard Agent Pay's purchase intent verification from payments to the full scope of agent-mediated actions.

5(e) — Insights from fields outside AI and cybersecurity

Three adjacent domains offer mature frameworks directly applicable to preference payload security:

- **Behavioral science.** Over 50 years of preference elicitation research demonstrates that self-reported preferences often diverge from revealed preferences (Kahneman's dual-process theory; Lichtenstein & Slovic, 1971; Fischhoff, Slovic & Lichtenstein, 1980). A robust preference schema must triangulate declared, observed, and inferred data with explicit confidence levels. Treating all preference data as equally reliable is itself a security vulnerability.
- **Medical informed consent.** Healthcare has mature frameworks ensuring patients' declared preferences govern care decisions—including mechanisms for preference change, surrogate decision-making (analogous to agent delegation), and preference verification under uncertainty. Advance directive standards map structurally to codifying user intent for autonomous agent behavior.
- **Financial "Know Your Customer" (KYC).** Banking regulations require that agents (human or AI) acting on behalf of customers have verified identity and intent. Extending KYC to "Know Your Preferences" for AI agents—verifying not just *who* the agent acts for but *what that person actually wants*—is a natural regulatory evolution as agentic commerce scales.

Recommendations

We respectfully recommend that NIST:

1. **Recognize that autonomous action on behalf of a user inherently requires a model of user intent**—whether explicitly declared or implicitly inferred—and that the integrity of this intent model is a foundational security concern for all AI agent systems, not only those engaged in commerce.
2. **Include preference data exchange standardization** within the scope of the AI Agent Standards Initiative—specifically, a verifiable schema for how intent and preference data is structured, transported, and independently audited across agentic protocols. The integrity of the intent model cannot be verified when it is inseparable from the platform operating the agent.
3. **Engage behavioral scientists** in addition to cybersecurity and AI researchers when developing intent-model-related guidelines. The preference layer sits at the intersection of human psychology, data integrity, and system security—and cannot be adequately addressed by any single discipline alone.
4. **Establish preference transparency requirements** as a component of future agent security guidelines. Users and auditors should have the ability to inspect the intent model an agent carries, understand how it was derived, and verify its provenance—as both a consumer protection measure and a security control.

[1] Nilson Report, 2024 global card fraud estimates.

[2] Sephora ($1.2M, 2022) and DoorDash ($375K, 2024) enforcement precedents for failure to honor GPC.

[3] W3C Verifiable Credentials Data Model v2.0, W3C Recommendation, March 2025.

[4] NIST AI 100-2e2025, Adversarial Machine Learning: A Taxonomy and Terminology of Attacks and Mitigations.

[5] NIST AI 100-1, Artificial Intelligence Risk Management Framework, January 2023.

[6] NIST SP 800-207, Zero Trust Architecture, August 2020.

[7] NIST SP 800-53 Rev. 5, Security and Privacy Controls for Information Systems and Organizations, September 2020.

[8] NIST AI 800-1, Managing Misuse Risk for Dual-Use Foundation Models, January 2025.