Consensys Software Inc
5049 Edwards Ranch Rd, Fort Worth,
TX 76109, United States

March 9, 2026
VIA ELECTRONIC TRANSMISSION

U.S. Department of Commerce
National Institute of Standards and Technology
Center for AI Standards and Innovation
Attention: Peter Cihon, Senior Advisor
100 Bureau Drive,
Gaithersburg, MD 20899

Re: CAISI Issues Request for Information About Securing AI Agent Systems (XRIN 0693-XA002)

Dear Sir or Madam:

Consensys Software Inc. ("Consensys") respectfully submits this comment in response to the Center for AI Standards and Innovation's ("CAISI") request for information about securing AI agent systems. Consensys appreciates this opportunity and submits this response from our perspective as a builder of widely used self-custody digital asset wallet infrastructure under the brand "MetaMask", as a contributor to open standards for agent identity and trust, and an active researcher of how agents can coordinate more safely in open networks.

Consensys develops wallet software and other key infrastructure that powers the Ethereum network, the largest programmable blockchain in the world. MetaMask is one of the world's most widely used self-custodial crypto wallets—serving tens of millions of users across browser and mobile—with capabilities spanning secure key management, multi-network Ethereum Virtual Machine ("EVM") access, token swaps and bridging, decentralized application connectivity, and an embedded developer platform that enables applications and agents to transact directly onchain. Ethereum, in short, is the most credibly neutral and censorship-resistant computer in the world, as it is operated by a decentralized community of self-selecting contributors and participants who value security, privacy, and open access. We firmly believe it is a foundation for a more secure, open and programmable society where consumers face fewer toll takers and enjoy more freedom and autonomy.

On Ethereum, "agentic accounts" are emerging as wallets that can be operated by software agents rather than a single human signer, largely enabled by smart-account patterns (ERC-4337

account abstraction and modular smart accounts like ERC-7579[1]) plus programmable policy layers (spending limits, allowlists, session keys, time locks, social recovery, and multi-party approvals). This shifts wallets from static key-holders into something more akin to governed systems where an agent can execute routine actions, such as rebalancing accounts, paying for APIs, managing decentralized finance ("DeFi") positions, bidding in auctions, or coordinating across networks within constraints set by the user or an organization.

As agents become more capable and more "online," the future could be that wallets look less like possessions and more like operational accounts: partially automated, continuously permissioned, and auditable, with clear boundaries between autonomy and oversight. Such an evolution would suggest as a practical matter multi-agent setups (where two specialized software agents collaborate to control a wallet or onchain account, with clear separation of responsibilities), secure enclaves or multi-party computation for key custody, rich intent standards, and compliance/identity protocols. Agents will need to move value safely and reliably while humans retain ultimate control through revocation, guardrails, and recoverability.

Much of the current discussion on agent security focuses on model failure inside a single application. That is important, but as discussed below, it is incomplete. Once agents act across organizational boundaries, use wallets, discover other agents, transact with each other, and handle delegated authority from humans, security becomes a network design problem as much as a model safety problem. For that reason, this response repeats a number of concepts that are critical to this broader approach to security, including identity, permissions, and trust.

In short, our belief is that, **as agents gain authority to transact, coordinate, and spend value, security must move from model-centric controls to system-level controls focused on identity, bounded delegation, trust infrastructure, and auditable execution.** Below, we explore this thesis by answering your questions where we believe we can provide distinctive and practical value.

1. Security Threats, Risks, and Vulnerabilities Affecting AI Agent Systems

(a) What are the unique security threats, risks, or vulnerabilities currently affecting AI agent systems, distinct from those affecting traditional software systems?

[1] ERC-4337 introduces account abstraction on Ethereum by enabling users to operate through programmable smart contract accounts instead of traditional externally owned accounts ("EOAs"), without requiring changes to the protocol itself. It uses a new transaction flow built around "UserOperations" and a shared EntryPoint contract, allowing wallets to implement custom validation logic such as multisig, passkeys, social recovery, spending limits, batching, and sponsored gas. ERC-7579 complements this by standardizing a modular architecture for smart accounts, so components like validators, executors, and recovery mechanisms can be plugged in and reused across implementations. Together, they transform wallets from static key holders into interoperable, programmable accounts capable of supporting automation, agents, and advanced security policies.

The most important distinction between agents and traditional software systems is that agents are software with delegated judgment and authority. Traditional software usually executes pre-defined logic inside a bounded system. Agents, by contrast, increasingly decide which tools to call, which counterparties to trust, what information to reveal, and which transactions to authorize. That creates several risks that are more systemic than ordinary application security failures.

First, agents create a new class of authority abuse. An attacker does not need to compromise code if they can manipulate an agent into using legitimate permissions in an illegitimate way. This is especially acute for wallets and payment-capable agents. A human-operated wallet such as MetaMask is often attacked through deception at a moment of approval.[2] An agentic wallet, if left operating without continuous human review, can be attacked through prompt injection, counterparty spoofing, gradual deviation of an agentic wallet's behavior from its originally intended constraints (*i.e.* policy drift), or malicious task decomposition over an extended period.

Second, agents increasingly operate in open networks where they must discover and evaluate unknown counterparties. In these environments, security depends not only on whether an agent is robust, but also on whether it can verify who it is dealing with, that counterparty is reputable, and prior behavior is auditable. This is a different problem from securing a closed internal workflow.

Third, agent security is becoming economically adversarial meaning attackers are no longer merely trying to break software but instead are trying to strategically manipulate incentives, reputation systems, and automated decision loops to extract profit. Attackers can create fake agents, manipulate ranking and reputation, spoof capabilities, exploit machine-speed payment flows (*e.g.* payment latency), and extract value from the automation layer itself (such as by exploiting fee routing, arbitrage triggers, settlement assumptions, refund mechanisms, or reward programs). As agentic commerce grows, these attacks will target not just prompts, but the surrounding trust, identity, and settlement infrastructure.

In short, the unique problem is not only whether a model can be misled. It is whether the surrounding system can identify counterparties, set safeguards on granted authority, and preserve auditability once agents begin acting in the world.

(d) How have these threats, risks, or vulnerabilities changed over time? How are they likely to evolve in the future?

[2] See., e.g., MetaMask Support, "Signature phishing," *MetaMask Help Center*, describing scams that trick users into signing malicious approval requests, https://support.metamask.io/stay-safe/protect-yourself/wallet-and-hardware/signature-phishing/

The threat landscape is already shifting from isolated prompt-level attacks toward coordination layer and market layer attacks.[3] In the near term, many incidents still look like classic AI security issues, namely indirect prompt injection, tool abuse, unsafe memory, or hidden instructions in retrieved content. Over time, the larger risks will come from discovery systems, reputation signals, delegated permissions, and payment flows. As things trend towards these larger risks, we expect three changes in particular:

First, attacks will become more compositional. An attacker may combine a spoofed identity, a plausible service endpoint, a manipulated trust signal, and a seemingly valid payment request into a single exploit chain. Second, attacks will become more machine-speed and more financially targeted. As payment flows become easier to automate, abusive monetization, fake metering, and payment-trigger manipulation will also become easier to scale unless authority is tightly scoped and auditable. Third, attacks will move from compromising one agent to corrupting a network's trust fabric. Reputation cartels, counterfeit agents, collusive feedback, and malicious validation services can undermine the ecosystem even if individual models are reasonably robust.

(e) What unique security threats, risks, or vulnerabilities currently affect multi-agent systems, distinct from those affecting singular AI agent systems?

Multi-agent systems introduce transitive trust and cascading failure. One agent may delegate work to another, which in turn relies on a third, which in turn consumes (potentially untrustworthy) data from a fourth. Even if each component is locally compliant, the overall chain may be unsafe, which results in outcomes no different than if an agent was not locally compliant.

The most important multi-agent-specific risks include: (i) where one agent over-trusts another agent's judgment or identity (trust transitivity errors); (ii) where groups of agents fraudulently reinforce one another's credibility (collusive reputation manipulation); (iii) where one agent coerces or reframes another into revealing unnecessary information or changing its behavior (negotiation-channel exploitation); (iv) where accountability becomes unclear across planner, executor, merchant, and payment provider roles (role confusion); and (v) where a bad instruction or false trust signal propagates across a network much faster than a human can intervene (rapid contagion).

[3] Coordination layer attacks target the infrastructure that lets agents discover, select, and interact with one another. That layer includes agent discovery and registries, reputation and ranking systems, messaging protocols, task routing networks, and identity and attestation systems. Attacks here aim to manipulate who agents trust and transact with, rather than what they think in a single prompt. Market layer attacks, on the other hand, target the economic mechanisms and automated payment flows that agents rely on. The market layer includes automated bidding systems, streaming payments, onchain settlement logic, incentive structures, and liquidity and pricing mechanisms.

Additional structural risks arising from the economic and compositional nature of multi-agent systems include: (vi) where individually rational agents optimize for local performance metrics in ways that degrade system-wide safety (cross-agent incentive misalignment); (vii) where attackers assemble multiple partially legitimate agents or services into an exploit chain that no single participant can detect in isolation (compositional exploitability); and (viii) where delegated permissions or trust relationships persist beyond their intended scope and cannot be rapidly unwound (revocation asymmetry). Observability also degrades in distributed systems, making detection and forensic attribution more difficult. That is true even in blockchain systems, where open ledger transparency improves observability at the transaction layer but still cannot access multi-agent risk that may live above that layer.

These risks set forth above are not well addressed by model robustness techniques alone. They arise from distributed coordination, incentive design, and network topology. Mitigating them requires system-level identity frameworks, verifiable attestations, shared monitoring and audit layers, incentive-aligned mechanism design, and rapid containment and revocation capabilities across the agent network.

2. Security Practices for AI Agent Systems

(a) What technical controls, processes, and other practices could ensure or improve the security of AI agent systems in development and deployment? What is the maturity of these methods in research and in practice?

In our view, the highest-value controls are the ones that treat identity, trust, authorization, and payment as first-class security controls.

Beginning with identity, one important control is portable agent identity with shared trust data. In other words, an autonomous agent should have a persistent, verifiable identity that carries its reputation and security history across platforms, so its past behavior can be assessed ecosystem-wide rather than reset or siloed in each new environment. Our ERC-8004 initiative, co-authored with the Ethereum Foundation, Google, and Coinbase, began in August 2025 and is currently a popular distributed registry for agents.[4] It provides open registries for identity, reputation, and validation. Public ecosystem dashboards cited in our source materials reported roughly 49,000+ registered agents and 16,000+ feedback entries by mid-February 2026 across 22+ EVM chains and Solana. This does not eliminate malicious agents, but it gives developers a public trust layer instead of forcing every platform to maintain its own opaque whitelist or score.

[4] The ERC-8004 standard defines on-chain registries for discovering, identifying, and assessing autonomous agents' trust and reputation across Ethereum and compatible networks. *See* Marco De Rossi, Davide Crapis, Jordan Ellis & Erik Reppel, *ERC-8004: Trustless Agents* [Draft], Ethereum Improvement Proposals, no. 8004, Aug. 13, 2025, https://eips.ethereum.org/EIPS/eip-8004.

A second critical control is least-privilege wallet design. For agent systems that spend, sign, or interact onchain, the default should not be private-key sharing or broad wallet custody. Concentrating signing authority in shared or persistent keys dramatically increases the damage of compromise and undermines enforceable, well tailored control over automated behavior. MetaMask's Smart Accounts Kit, as one example, allows users to grant narrowly scoped permissions without surrendering custody.[5] For example, a shopping agent can be allowed to spend up to a daily USDC limit at approved merchants; a treasury agent can rebalance accounts within certain approved assets categories but not be allowed to withdraw to unknown addresses; or a support agent can claim or renew onchain services without gaining full account control over assets.

A third important control is to move more safety checks into the execution layer. Safety checks occur prior to transaction signing and include transaction simulation and security alerting.[6] These checks protect MetaMask users from deceptive requests and should be made available to agents before execution, not only to humans at confirmation time. Likewise, transaction optimization and pre-simulation can reduce failed or exploitable execution paths.

A fourth control is a secure negotiation intermediary for high-risk agent interactions, an area we are actively working on. The idea is that agents should not directly negotiate sensitive matters in unrestricted natural language. Natural-language negotiation between autonomous agents is an unstructured, high-entropy attack surface where subtle manipulation can lead directly to financial or operational harm. Instead, they can communicate through an intermediary that runs inside a trusted execution environment ("TEE"), exposes inspectable code, is remotely attestable, synthesizes a task-specific interaction schema, and enforces that schema bidirectionally through filtering, redaction, and policy checks. In practical terms, this acts as a communications firewall for agents. Just as network firewalls do for traffic, it restricts what kinds of messages are allowed, inspects content against policy, and blocks patterns known to be unsafe. It creates new security primitives beyond ordinary access control: data-in-use protection, stronger assurances about which information may be disclosed, policy-scoped traces instead of raw logs, and more credible "commit to forget" style handling because counterparties can inspect and attest the code path used for sensitive sessions. Negotiation goes from an unstructured persuasion problem into a verifiable protocol problem, dramatically reducing the risk that autonomous agents can be socially engineered, incrementally manipulated, or coerced into unsafe commitments.This is especially relevant for procurement, pricing, schedule coordination, compliance, and other

[5] The MetaMask Delegation Toolkit is a suite of developer tools and smart contracts for embedding MetaMask Smart Accounts into dApps, enabling programmable account behavior and fine-grained delegated permissions such as multi-signature approvals, automated actions, and gas abstraction to create smoother, more flexible digital asset user experiences. *See* https://metamask.io/developer/delegation-toolkit.

[6] Transaction simulation is the process of locally executing a proposed transaction against the current blockchain state (without actually submitting it) to predict what will happen. Security alerting builds on simulation and threat intelligence by flagging known or suspected risks, such as a contract linked to scams or sending funds to a high-risk address. Together, simulation predicts mechanical outcomes, while alerting evaluates contextual risk.

negotiations where prompt injection, coercive reframing, or accidental oversharing are major risks.

A fifth control is cryptographically explicit commerce, where contract terms are machine-enforceable and cryptographically bound to specific identities and actions, and where payment is triggered objectively rather than by interpretation of conversational promises. This control is timely, as agentic commerce is now moving from theory to practice through protocols such as x402, which enables machine-to-machine stablecoin payments over standard HTTP flows, and mandate-based commerce approaches such as AP2 and ACP, which emphasize signed user intent, clearer merchant accountability, and tamper-evident transaction records.[7] These approaches are more secure than vague natural language policy statements that an agent is "allowed to buy things."

Some of these controls are deployable now, such as least privilege, simulation, signed audit trails, and TEE attestation, while others are early-stage, namely general-purpose secure negotiation intermediaries and machine-speed due diligence systems. Continued progress in all these areas is critical because they address increasingly important problems that existing cybersecurity controls do not fully solve.

(e) Which cybersecurity guidelines, frameworks, and best practices are most relevant to the security of AI agent systems?

Many existing best practices remain directly relevant, such as least privilege, zero trust, secure software development, strong key management, signed provenance, layered monitoring, incident response, and rigorous logging. However, AI agent systems require those practices to be extended in three ways. First, the acting principal is no longer always a human user or a server process. It may be delegated software with bounded authority. Security guidance therefore needs to address delegated machine authority explicitly, treating delegated agents as bounded principals with programmable authority, not as automated extensions of a human wallet. The guidance should outline controls that constrain, monitor, and revoke that authority by default.

Second, agent systems often interact with unknown counterparties across organizational boundaries. That makes portable identity, counterparty verification, and auditable trust signals much more important than in ordinary enterprise software. Across boundaries, there is no shared institutional safety net. Trust must be established cryptographically and evaluated continuously in a permissionless, economically adversarial environment.

---

[7] x402 Foundation. (n.d.). *x402: An open payment standard for the internet (Whitepaper)*. https://www.x402.org/x402-whitepaper.pdf; OpenAI and Stripe, *Agentic Commerce Protocol (ACP) Specification*, 2025, https://www.agenticcommerce.dev/docs; and Google Cloud, *Agent Payments Protocol (AP2)*, 2025, https://ap2-protocol.org.

Third, authorization must under many circumstances be continuous (evaluated upon every action) and contextual, as opposed to one-time or broad and static. An agent that can make purchases, move assets, or operate a wallet should be constrained by function, amount, asset type, counterparty, time window, and revocation path.

Because agent systems introduce autonomous, delegated actors that can discover counterparties, negotiate, and move value at machine speed, traditional cybersecurity controls focused on user authentication and perimeter defense are no longer sufficient on their own. Existing cybersecurity guidance is important and applicable but must be complemented by mechanisms that make agent identity portable and verifiable, permissions cryptographically bounded, actions auditable, and trust signals interoperable across networks, particularly in open environments where no shared institutional boundary exists.

3. Assessing the Security of AI Agent Systems

(a) What methods could be used during AI agent systems development to anticipate, identify, and assess security threats, risks, or vulnerabilities?

Agent security should be evaluated through end-to-end tasks, not only prompt-response tests. The right question is whether the full system can choose trustworthy counterparties, preserve confidentiality, respect delegated permissions, and produce auditable actions under pressure. Governments, standards bodies, and industry groups are increasingly shifting toward end-to-end evaluations of agent systems rather than narrow prompt-response testing. Efforts from NIST, OWASP, the Cloud Security Alliance, major cloud providers, and academic benchmarking initiatives now emphasize full lifecycle threat modeling, tool-use governance, delegated permission controls, and task-based stress testing in realistic environments.[8] Together, these initiatives signal a growing recognition that agent security must be assessed at the system level—including identity, coordination, execution, and auditability—not just model robustness.

We have been engaged in a similar benchmark and research effort called "Protocol Agent" examining whether agents can recognize when cryptographic coordination is preferable to

[8] *See* National Institute of Standards and Technology (NIST), "Strengthening AI Agent Hijacking Evaluations," 2025, https://www.nist.gov/news-events/news/2025/01/technical-blog-strengthening-ai-agent-hijacking-evaluations; OWASP Foundation, *AI Agent Security Cheat Sheet*, https://cheatsheetseries.owasp.org/cheatsheets/AI_Agent_Security_Cheat_Sheet.html; Cloud Security Alliance, "Agentic AI Threat Modeling Framework (MAESTRO)," 2025, https://cloudsecurityalliance.org/blog/2025/02/06/agentic-ai-threat-modeling-framework-maestro; Microsoft, *Governance and Security for AI Agents*, https://learn.microsoft.com/en-us/azure/cloud-adoption-framework/ai-agents/governance-security-across-organization; and WebArena Benchmark, arXiv:2307.13854, https://arxiv.org/abs/2307.13854.

natural language dialogue, persuade a counterparty to adopt it, and execute it correctly.[9] Our benchmark is intentionally pragmatic: it covers use cases such as proving you are over 21 without having to disclose all the information on your government-issued ID, proving income is below a threshold without sharing a tax return, finding common meeting availability without revealing full appointment calendars, filing an anonymous report while proving legitimate membership in a group, checking whether a password was breached without revealing it, and splitting shared expenses fairly without disclosing sensitive income data. These are concrete coordination problems where agents can create security value by reducing unnecessary disclosure.

The results of our research so far have been promising. As described in our paper, tuned models substantially outperform base models on the benchmark's end-to-end dimensions of (i) choosing the correct cryptographic tools or protocol components for the task (primitive selection); (ii) structuring agreements correctly and aligning on appropriate terms (negotiation); (iii) producing technically valid and executable code or transaction logic (implementation correctness); (iv) correctly performing required calculations or transformations (computation); and (v) avoiding insecure constructions or weak cryptographic patterns (security strength). Task-specific training materially improves a model's ability to complete complex, security-sensitive workflows accurately and safely, rather than merely generating plausible-sounding responses.

Another area we are working on explores machine-speed due diligence before delegation. We query whether an agent can discover other agents across registries and catalogs, challenge their advertised capabilities, inspect trust signals, and decide whether they are safe enough to use in higher-stakes tasks. Development testing must encompass counterparty spoofing and impersonation, abuse of delegated wallet permissions, negotiation coercion and oversharing, failure to verify payment or execution conditions, reputation manipulation and trust-signal poisoning, and breakdowns in rollback, revocation, or human escalation.

The work does not end upon delegation, however. Detection efforts must persist and should combine wallet telemetry, signed audit trails, registry events, payment anomalies, validation results, and transaction simulation outputs. In other words, delegation is not a one-time trust decision but instead becomes a continuously monitored, multi-signal control loop designed to detect drift, compromise, or adversarial coordination before material loss occurs. The National Institute of Standards and Technology ("NIST") could add meaningful value by encouraging common taxonomies for incidents involving agent identity abuse, delegation abuse, and payment abuse.

[9] Marco De Rossi, *Protocol Agent: What If Agents Could Use Cryptography in Everyday Life?*, arXiv:2602.01304, Feb. 1 2026, https://arxiv.org/abs/2602.01304.

(b) Not all security threats, risks, or vulnerabilities are necessarily applicable to every AI agent system; how could the security of a particular AI agent system be assessed and what types of information could help with that assessment?

The first step is to assess the agent's external authority, not just its model class. The critical inquiry is what the agent can actually do in the world. There are key issues that get to the heart of that matter. The external systems the agent can affect determine the real-world scope of damage if there is compromise. The credentials, wallet permissions, or spending rights the agent holds define the scope of its actionable power. Whether those permissions are bounded by amount, time, function, asset, and counterparty limits damage and constrains misuse. How it identifies and verifies counterparties determines exposure to spoofing and fraud. Relying on public trust evidence, private platform scores, or no trust layer at all (in short, the quality and portability of trust signals) affects systemic resilience. Whether high-consequence actions are simulated or independently validated before execution reduces the risk of irreversible errors. Design of a kill switch, revocation path, or rollback mechanism determines how quickly harm can be stopped. Availability and machine-verifiability of logs is essential for detection, attribution, and accountability. Together, these questions shift security evaluation from abstract model risk to concrete authority, constraints, verification, and recoverability.

For wallet-enabled agents, NIST should distinguish sharply between two architectures: agents with unrestricted key custody, and agents operating through revocable, policy-bounded delegations. These are not equivalent risk profiles. If guidance treats these architectures as equivalent, it risks over-regulating constrained systems or underestimating the danger of full custody delegation. Clear differentiation enables proportionate standards, better risk modeling, and more precise recommendations for safe deployment.

4. Limiting, Modifying, and Monitoring Deployment Environments

(a) AI agent systems may be deployed in a variety of environments, i.e., locations where the system's actions take place. In what manner and by what technical means could the access to or extent of an AI agent system's deployment environment be constrained?

The strongest constraint in agent security is to limit authority at the environment and wallet layer rather than relying solely on model alignment or behavioral safeguards. In practice, this means structurally constraining what the agent *can* do, regardless of what it intends to do. Agents should operate with scoped wallet permissions instead of full private-key custody, so their authority is narrowly defined and revocable rather than absolute. They should be restricted to allowlisted tools, domains, contracts, and counterparties, reducing exposure to spoofed services or malicious endpoints. Their activity should be bounded by rate limits, time limits, and spending ceilings, ensuring that even erroneous or adversarial behavior cannot escalate quickly or indefinitely. Before any transaction is signed, simulation and policy validation checks should

confirm that the predicted effects align with approved constraints. For higher-risk workflows, responsibility should be divided between separate planner and executor roles, preventing reasoning components from having direct signing authority. Finally, human approval thresholds should be required for exceptional or high-consequence actions, preserving a meaningful intervention point when automated safeguards are insufficient. Together, these measures shift security from hoping the model behaves safely to ensuring that unsafe behavior is technically constrained by design.

This is where MetaMask's delegation model is especially relevant. Through delegations, agents can be granted the exact permissions they need to operate, without being given custody. It is a safer foundation than shared private keys and recovery phrases, browser automation, or static API tokens because permissions can be narrow, time-bound, revocable, and tied to specific actions. This structure helps prevent catastrophic key compromise, lateral privilege escalation, silent policy drift, and irreversible asset loss resulting from overbroad authority. It also reduces the risk of replay attacks (when a valid, previously authorized message or transaction is captured and maliciously reused to trigger the same effect again), token approval abuse, and long-lived credential leakage, since delegated permissions can expire automatically and be revoked without rotating the user's primary keys. By constraining authority at the protocol layer, delegation limits the scope of possible loss, contains automation errors, and preserves a clear separation between human principal and machine delegate.

(c) What is the state of managing risks associated with interactions between AI agent systems and counterparties?

This is one of the largest unresolved gaps in the current ecosystem. Closed platforms often solve counterparty risk through centralized gatekeeping. Open systems require shared trust infrastructure. There is no single entity that can vouch for identity, adjudicate disputes, or enforce compliance in an open system. As a result, trust must be established through portable identity, verifiable credentials, cryptographic attestations, reputation registries, and transparent audit trails. These shared trust infrastructures are still emerging and lack universal standards, interoperability, and coordinated incident response mechanisms. ERC-8004 provides a practical example of what the industry is actively developing in terms of shared trust infrastructure for agent interactions. Since its proposal in August 2025, the ecosystem has developed tools, explorers, and registrations on mainnet and also multiple EVM chains. Because the trust data is public and programmable, different firms can build ranking, insurance, filtering, or due-diligence systems on top of the same evidence.

For commerce and payment-bearing interactions, we are actively exploring machine-speed due diligence. Agents need ways to evaluate whether a counterparty is trustworthy enough for delegation, coordination, or payment. Counterparty assessment should combine at least four inputs: verified identity, reputation, validation evidence, and payment history.

(d) What methods could be used to monitor deployment environments for security threats, risks, or vulnerabilities?

Monitoring in agent systems must extend beyond model outputs because many meaningful risk signals arise from the infrastructure in which the agent operates. In particular, agent-to-agent interaction logs and schema violations can reveal coercion attempts, malformed requests, or deviations from approved negotiation protocols. Tracking the creation, use, and revocation of delegated permissions helps detect privilege escalation, dormant authorities being activated, or unusual delegation patterns. Observing payment requests and payment proofs enables identification of anomalous settlement behavior, such as mismatched amounts, repeated claims, or suspicious routing. Reviewing transaction simulation results and security-alert triggers provides early warning when predicted asset flows or contract interactions diverge from expected policy constraints. Monitoring for sudden changes in trust signals, validation outcomes, or endpoint ownership can surface counterparty compromise or identity spoofing. Finally, analyzing abnormal counterparty concentration or unexpected spikes in payment or call volume helps identify collusion, automated exploitation, or cascading failures. Together, these layers shift monitoring from reactive output inspection to systemic oversight of authority, coordination, and economic activity.

Open networks require both public and private monitoring. Public telemetry is valuable for identity events, trust signals, and payment or validation evidence that multiple parties may need to inspect. Private telemetry is needed for confidential prompts and sensitive internal data.

Here is another place where the secure negotiation approach is useful, given the sensitive data. It is critical that privacy is preserved alongside auditability. Public blockchains can overexpose behavior if used carelessly. A TEE-backed intermediary should be able to provide redacted or policy-scoped traces for audit and detection without relying on ordinary logging of full sensitive payloads. Privacy-preserving techniques such as zero-knowledge proofs and selective disclosure, in addition to TEE-backed processing, are important complements to public trust infrastructure.

5. Additional Considerations

(a) What methods, guidelines, resources, information, or tools would aid the AI ecosystem in the rapid adoption of security practices affecting AI agent systems and promoting the ecosystem of AI agent system security innovation?

We see several initiatives as especially useful for strengthening agent security across open ecosystems. First, reference architectures for agent wallets built around scoped delegations, revocation, and pre-execution checks would provide concrete blueprints for limiting authority by default, ensuring that automation is constrained at the protocol layer rather than relying solely on

model behavior. Second, open schemas for permissions, trust signals, and security incident reporting would enable interoperability and consistent machine-readable interpretation of authority, risk, and compromise events across platforms.

Third, shared registries for portable agent identity, reputation, and validation would make trust signals durable and transferable, reducing sybil attacks and reputation reset problems in cross-platform environments. Fourth, benchmark suites for multi-agent negotiation, discovery, and payment safety would allow researchers and developers to evaluate how systems behave under adversarial and high-pressure conditions, moving beyond isolated prompt testing. Fifth, public testbeds for agentic commerce that combine permissioning with machine payment protocols such as x402, referenced above, would create controlled environments to study real-world payment flows, delegation limits, and fraud resistance.

Finally, better user interfaces that explain delegated machine authority in plain language are essential to ensure that human principals understand what an agent can and cannot do, preserving meaningful consent and accountability. Together, these efforts would strengthen both the technical and human foundations of secure agent ecosystems.

Moreover, security innovation must extend beyond model providers to the infrastructure that governs agent authority. Wallets like MetaMask should make scoped, revocable delegations and execution-layer policy enforcement the default, while payment systems should support cryptographically explicit payments, anomaly detection, and safer settlement primitives. Registries and identity layers should provide portable, verifiable agent identity, durable reputation signals, and rapid revocation mechanisms that propagate risk across platforms. Across all layers, cryptographic tooling and user interfaces should ensure that delegated machine authority is bounded, auditable, and clearly understood.

(b) In which policy or practice areas is government collaboration with the AI ecosystem most urgent or most likely to lead to improvements in the state of security of AI agent systems today and into the future?

We believe the most urgent areas are agent identity and authorization, because without portable, verifiable identity and tightly scoped permissions, autonomous systems cannot reliably assess counterparties or constrain delegated authority. Secure machine payments are also critical, as agents increasingly move value at machine speed and require cryptographically explicit, auditable settlement mechanisms to prevent fraud and ambiguity. Common taxonomies for agent incidents and trust events are needed so that compromise signals, revocations, and risk indicators can be shared and interpreted consistently across ecosystems. Standards for auditable delegations, mandates, and validation evidence will ensure that authority is transparent, enforceable, and reviewable after the fact. Finally, evaluation methods must address open trust infrastructure—not just closed platform controls—because many agents will operate across organizational and jurisdictional boundaries.

Policymakers should take care not to reinforce a model in which safe deployment is achievable only within a handful of vertically integrated platforms, as that would concentrate control and limit interoperability. First, it centralizes control over infrastructure that may become economically and socially critical, reducing competition, limiting innovation, and creating dependency on proprietary systems. It also creates systemic risk because, if safety mechanisms exist only within a handful of platforms, we have a monoculture of implementations that is generally less resilient than a diversity of interoperable implementations. Further, it weakens portability and user autonomy, keeping users stuck inside a specific platform's trust boundary. Finally, external auditing is more difficult when regulatory compliance is embedded into closed architectures, and this risks reducing transparency and public accountability.

Both the current Administration and the last have been concerned about centralized, integrated AI platforms, and for good reason.[10] NIST is especially well positioned to promote interoperable standards for portable identity, verifiable permissions, and machine-action auditability, helping ensure that safety scales across open networks rather than being confined to proprietary environments. It's good for innovation and good for American consumers to doggedly pursue an open framework, and we can achieve one without sacrificing security.

(c) In which critical areas should research be focused to improve the current state of security practices affecting AI agent systems?

We would prioritize research in five areas.

a) Privacy-preserving coordination. Agents should be able to complete ordinary tasks using cryptographic protocols that minimize data exposure by default. This includes selective disclosure, zero-knowledge proofs, and structured commitments that allow verification without revealing unnecessary user or transaction information.
b) Trustworthy public reputation and validation. Trust signals must be portable, auditable, and resistant to sybil attacks or manipulation. Research should focus on interoperable registries, verifiable attestations, and shared incident reporting mechanisms that prevent reputation reset and collusive amplification.
c) Secure agent wallets. Agent authority should be bounded through scoped delegations, revocation paths, transaction simulation, and execution-layer policy enforcement. The goal is to enable autonomous action without granting full custody, thereby limiting blast radius and containing compromise.[11]

[10] The White House, *Executive Order on the Safe, Secure, and Trustworthy Development and Use of Artificial Intelligence*, 30 Oct. 2023, https://bidenwhitehouse.archives.gov/briefing-room/presidential-actions/2023/10/30/executive-order-on-the-safe-secure-and-trustworthy-development-and-use-of-artificial-intelligence/. See also The White House, *America's AI Action Plan*, 2025, https://www.whitehouse.gov/wp-content/uploads/2025/07/Americas-AI-Action-Plan.pdf.

[11] The area where Consensys is especially well positioned to assist NIST is secure agent wallets. MetaMask sits directly at the control point where users delegate authority to software, manage keys, sign transactions, and interact

d) Safer machine-native communication. High-risk workflows should move beyond unconstrained natural-language exchange toward structured, enforceable interaction protocols. This includes TEE-backed negotiation intermediaries that enforce schemas, redact sensitive information, and provide attestable, policy-bounded handling of context.
e) Systemic risk measurement. Evaluation methods should assess failure propagation, collusion, and network-level trust breakdown across multi-agent environments. Security analysis must extend beyond single-model behavior to understand how coordination, incentives, and delegation chains amplify risk at the ecosystem level.

* * *

We appreciate the opportunity to respond to this Request for Information and to contribute to the development of thoughtful, forward-looking guidance for agent systems. As autonomous agents begin to transact, coordinate, and act on behalf of users in open digital environments, security must evolve from model-level safeguards to system-level controls grounded in identity, delegated authority, verifiable permissions, and auditable execution. We believe in a world where agents transact onchain more than humans do, and this moment presents an opportunity to shape interoperable standards that promote safety without entrenching centralization or limiting innovation.

We would welcome the opportunity to discuss these issues further, share technical insights from our experience building wallet and delegation infrastructure, and collaborate on research, pilots, or standards development. We stand ready to work constructively with the agency and other stakeholders to advance secure, open, and resilient foundations for agentic systems.

Respectfully submitted,

CONSENSYS SOFTWARE INC.

by:
Marco De Rossi (Director, Product)
William C. Hughes (Senior Counsel, Director of Global Regulatory Matters)

with smart accounts. Through smart accounts, scoped delegations, execution-layer policy enforcement, transaction simulation, security alerting, and revocation mechanisms, MetaMask has practical experience designing systems that constrain onchain authority without requiring full private-key custody. That operational vantage point gives Consensys unusually concrete insight into how delegated machine authority can be bounded, audited, and safely revoked in real-world deployments.