SpruceID

Spruce Systems, Inc.
228 Park Ave S, PMB 28788
New York, New York 10003-1502 US

April 2, 2026
National Institute of Standards and Technology
National Cybersecurity Center of Excellence

Re: Comments on Concept Paper: Accelerating the Adoption of Software and AI Agent Identity and Authorization

Dear NIST NCCoE Team,

SpruceID appreciates the opportunity to provide comments on the NCCoE concept paper on Software and AI Agent Identity and Authorization. As builders of open-source digital identity infrastructure (including verifiable credential libraries, identity wallets, and credential platform services) we bring direct experience designing identity systems that can be adapted for agentic architectures. Our work spans regulated environments such as stablecoin payments, business credential issuance, and delegated authority scenarios, all of which have direct bearing on the challenges this concept paper seeks to address.

We offer the following comments organized in alignment with the questions posed in the concept paper.

1. General Questions: Use Cases, Opportunities, and Risks

Enterprise Use Cases

Organizations are currently deploying AI agents across a wide range of enterprise functions, including workflow automation, software development co-pilots, code generation and maintenance, system design and architecture, and business process orchestration. These applications are delivering substantial cost savings, economic efficiency gains, and expanded innovation capacity by lowering the barrier for building and maintaining software systems.

Opportunities

AI agents present opportunities in cost reduction, accelerated development cycles, and the democratization of technical capability, enabling more people to build and ship software, automate processes, and make data-driven decisions at scale.

Spruce Systems, Inc.
228 Park Ave S, PMB 28788
New York, New York 10003-1502 US

Risks

We are concerned about the following categories of risk:

- **Data leakage and inappropriate data sharing:** Agents and LLM-backed chat APIs may inadvertently expose sensitive data, whether through prompt responses, tool invocations, or logging pipelines.
- **High-impact errors in critical environments:** Agents performing incorrect or unauthorized actions in high-stakes domains (such as public benefits distribution, payments processing, illicit finance detection, and healthcare administration) could produce severe real-world consequences.
- **Unpredictability:** AI agents differ from traditional software agents because the generative nature of AI makes their problem-solving approaches inherently less predictable. This demands a fundamentally different security model with defense in depth around what agents can access, what actions they can take, and how their outputs are gated before external impact occurs.

Core Characteristics of Agentic Architectures

From our perspective, the core characteristics of agentic architectures involve three layers: (1) contextualizing the agent's environment, (2) situating the agent within that environment, and (3) permissioning the agent to perform actions with expected outcomes. We believe that agentic architectures will additionally require continuous monitoring of agent actions and outputs, analogous to an advanced form of intrusion detection systems (IDS) used in network security, to detect negative behaviors or out-of-scope activities, integrated with trust models that govern permissioning decisions.

Model Context Protocol (MCP) and OAuth Considerations

We see MCP as a promising starting point for permissioning resources in agentic contexts. However, we observe that MCP permissions are frequently under-specified, and the current reliance on OAuth 2.0 flows for MCP authorization introduces significant friction that can impede agentic identity models. Requiring a new OAuth connection with a human in the loop for each agent interaction creates unnecessary overhead.

We recommend a more agent-native approach: AI agents should possess their own identity and a set of capabilities that are granted programmatically for specific, human-understandable purposes. This model provides a natural checkpoint for governance and review of agent actions without the per-interaction friction of traditional OAuth delegation flows.

Spruce Systems, Inc.
228 Park Ave S, PMB 28788
New York, New York 10003-1502 US

2. Identification

All agents performing workloads must have a unique handle or identifier within their operating environment. We envision agents running within agentic environments that are provisioned with resources (storage, compute, network access) in a manner analogous to containerized workloads. Identity and authentication factors should be bestowed upon that environment as a resource, alongside capabilities. This is the missing layer for agentic identity: identity plus capabilities, authentication plus authorization, forming a zero-trust boundary around each agent.

Essential Identity Metadata

Specific agent identities should be tied to specific hardware, software, and organizational boundaries so that actions can be attributed and organizations held accountable. Identity metadata should include the agent's owning organization, the hardware and software context in which it operates, its authorized capability set, and its delegation chain back to a responsible legal person.

Persistent Identity vs. Ephemeral Capabilities

We recommend distinguishing between two core components of agent identity:

- **Agent Identity (Persistent):** A long-lived, organization-bound identity anchored to the agent's operating environment (including hardware, software, and ownership context). This identity enables attribution, accountability, and trust establishment across systems.

- **Capabilities (Ephemeral):** Task- or session-scoped permissions granted to the agent, representing what actions it is authorized to perform in a given context. These capabilities should be time-bound, least-privilege, and dynamically issued based on workflow requirements.

This separation allows organizations to maintain a stable identity for auditing and governance, while flexibly adjusting permissions as agent context and responsibilities evolve.

3. Authentication

Strong Authentication for AI Agents

Authentication for agents should be cryptographic in nature, relying on FIPS 140-2 or FIPS 140-3 validated modules within an organizational security framework aligned to ISO 27001 or NIST SP 800-53. We recommend borrowing heavily from the NIST SP 800-63 Authenticator Assurance Level (AAL) guidelines for human authentication and developing an agent-native methodology that maps to those assurance levels while maintaining clear accountability ties back to human actors in scenarios that require it.

Key Management

Key management for agents should be handled centrally by the owning organization, with human oversight and clearly defined governance controls. This includes established processes for the issuance, rotation, revocation, and auditing of agent key material, consistent with the organization's broader cryptographic and key management policies.

In addition to internal governance, key management is critical to enabling agents to operate across disparate external resources and trust domains, where resources may be controlled by different organizations with independent security policies. A well-designed key management framework allows agent identities to be recognized and trusted across these environments, supporting scalable access to external systems without requiring bespoke integrations for each resource owner.

This requires interoperable trust mechanisms, including standardized credential formats, trust registries, and shared verification frameworks, that allow multiple resource owners to independently validate agent identity and associated permissions. In this model, key management is not only a security function, but also a foundational enabler of cross-domain interoperability, delegated authority, and scalable capability execution across heterogeneous systems.

4. Authorization

Zero-Trust Principles

Authorization should be implemented via capability-based authorization models, where authorization policies are represented as verifiable credentials that reside in an agent's identity wallet. This gives each agent a portable, inspectable manifest of the capabilities it may exercise.

Dynamic and Formally Verifiable Policies

Authorization policies should be expressed in languages optimized for formal verification, such as CedarLang or WebAssembly (WASM). Formally modeled policy languages enable comprehensive analysis of all issued permissions and help organizations understand the full scope of impact of any particular agent for compliance purposes.

These policies can also be represented as cryptographically signed artifacts within a capability-based access model, allowing permissions to be portable, verifiable, and independently validated across systems. This approach enables stronger guarantees around integrity and provenance of authorization decisions, particularly in distributed or cross-domain environments.

This is a significant improvement over ad hoc resource-level permissions, such as access objects in an S3 bucket, which are difficult to manage and reason about at scale.

Least Privilege

We acknowledge that establishing least privilege for agents is challenging given the unpredictable nature of AI problem-solving. However, we believe that formally verifiable policy languages offer a path forward by enabling organizations to model, analyze, and bound the permission envelope for each agent, rather than relying solely on coarse-grained resource access controls.

Proving Authority and Delegation

Agents should prove their authority to perform specific actions through demonstration of possession of cryptographic key material or equivalent authentication factors. Delegation of authority should occur through verifiable credentials issued to the agent that grant scoped capabilities derived from a root capability ultimately held by a legal person, an organization or a human.

When an agent requires additional capabilities beyond its current grants, the request should be structured as a permission escalation request directed to a human approver, or to another agent that has been authorized by a human to approve such requests after exercising judgment. These escalation requests must be self-contained and interpretable out of context, enabling asynchronous review and audit.

5. Auditing and Non-Repudiation

Given the unpredictability of AI agents, continuous monitoring and audit logging are essential. We recommend architectures that implement ongoing monitoring across all agent actions and outputs, analogous to advanced intrusion detection but specifically tailored for agentic behavior. This monitoring should detect negative behaviors, out-of-scope activities, and anomalous patterns, and

should be integrated with the trust and permissioning model to enable automated or human-initiated responses.

Binding agent actions to their identity credentials and delegation chains provides a natural mechanism for non-repudiation, ensuring that every action can be traced back through the credential chain to the responsible human or organization. This can be further strengthened through cryptographic attribution and append-only logging mechanisms, which provide tamper-evident records of agent activity and support verifiable audit trails across systems.

6. Prompt Injection Prevention and Mitigation

We believe that the defense-in-depth model described throughout these comments, combining strong agent identity, capability-based authorization, formally verifiable policies, and continuous monitoring, provides a meaningful framework for mitigating the impact of prompt injection attacks. Even if an injection succeeds in manipulating an agent's reasoning, a well-implemented capability boundary limits what the compromised agent can actually do, and monitoring systems can flag anomalous behavior for human review.

This approach is analogous to established work in biometric systems, particularly Presentation Attack Detection (PAD) as defined in ISO/IEC 30107, which formalizes the detection and mitigation of spoofing and injection attacks against biometric sensors. In these systems, adversarial inputs are an expected condition, and no single control is sufficient. Instead, layered protections are required to both detect presentation attacks and limit their downstream impact.

This model is reinforced by evaluation and operational work from NIST, including the Face Recognition Vendor Test (FRVT) PAD program, which emphasizes measurable detection performance alongside system-level resilience, and by Department of Homeland Security initiatives such as the Remote Identity Validation Rally (RIVR), which demonstrate the necessity of combining spoof detection, risk-based controls, and continuous monitoring in real-world identity systems.

We believe prompt injection should be treated as an equivalent class of threat in agentic systems. As with biometric presentation attacks, it is not sufficient to rely solely on input validation or detection mechanisms. Systems must be designed such that, even when adversarial inputs are successful, the resulting actions are constrained, observable, and attributable. In this model, capability-based authorization, cryptographic identity, and auditability serve as functional analogs to PAD, ensuring that the scope of potential harm is bounded and that anomalous behavior can be detected and addressed.

7. Recommended NCCoE Demonstration

SpruceID recommends that NCCoE consider a demonstration architecture in which an AI agent is issued a cryptographic identity and a set of verifiable digital credential-based capability grants and/or cryptographic tokens, enabling it to:

- Access multiple disparate enterprise systems spanning across many external organizational boundaries (e.g., HR, financial institutions, document systems, government records, and healthcare systems)
- Perform actions under delegated authority from a human user
- Enforce least privilege through credential-scoped permissions
- Log all actions with cryptographic attribution and auditability

The demonstration should compare and evaluate the intersection of:

- Traditional OAuth-based delegation
- Credential-based capability models

This would allow NCCoE to evaluate tradeoffs in scalability, security, and operational overhead.

8. Relevant Standards and SpruceID Capabilities

SpruceID's platform is grounded in open, interoperable standards that enable secure, privacy-preserving identity and authorization across government and enterprise environments. Our approach aligns with both established and emerging specifications to ensure compatibility with existing systems while supporting future innovation.

While OAuth, OIDC, SPIFFE, and SCIM provide foundational capabilities for identity, provisioning, and workload authentication, they do not fully address portable authorization, delegation chains, and offline verifiability required for agentic systems. Verifiable digital credentials (such as ISO mdoc, IETF JWTs/SD-JWTs/CWTs, and W3C Verifiable Credentials) approaches can complement these standards by enabling capability portability, fine-grained delegation, and privacy-preserving verification.

- **W3C Verifiable Credentials (VCs):** Provide a standardized, cryptographically secure data model for expressing identity attributes, entitlements, and authorization artifacts. SpruceID uses VCs to represent everything from identity claims to delegated capabilities, enabling portable, issuer-independent credentials that can be verified without reliance on centralized services.

- **W3C Decentralized Identifiers (DIDs):** Enable globally unique, cryptographically verifiable identifiers that are not dependent on a single issuing authority. SpruceID leverages DIDs to establish persistent identity anchors for individuals, organizations, and software agents, supporting strong authentication and verifiable trust relationships across domains.

- **ISO/IEC 18013-5, 18013-7, 23220-4 (mDL / mdoc):** Define interoperable standards for mobile driver's licenses and digital credentials, including both in-person (NFC/BLE) and online presentation flows. SpruceID supports mdoc-based credentials to ensure compatibility with state-issued mobile driver's licenses and broader digital ID ecosystems.

- **SD-JWT (Selective Disclosure JSON Web Tokens):** Provide a flexible mechanism for selective disclosure using widely adopted JWT infrastructure. SpruceID supports SD-JWT for use cases where compatibility with existing OAuth/OIDC ecosystems is important, enabling users to disclose only the minimum necessary attributes while maintaining cryptographic integrity.

- **OpenID Connect (OIDC), OID4VP, and DC-API:** Enable integration with existing identity and access management systems while supporting wallet-based credential presentation. SpruceID implements these protocols to bridge traditional federated identity with verifiable credential workflows, allowing agencies to augment (not replace) existing SSO and IAM investments.

- **Privacy-Preserving Status and Revocation (IETF Token Status Lists, W3C Bitstring Status Lists):** Allow verifiers to check credential validity without introducing user tracking or centralized dependencies. SpruceID incorporates these mechanisms to support revocation and suspension at scale while preserving user privacy and avoiding "phone home" architectures.

- **Capability-Based Authorization and Policy Frameworks (Cedar, WebAssembly/Wasm):** Enable fine-grained, formally verifiable authorization policies. SpruceID represents permissions as credentials and evaluates them using policy engines that support auditability, least privilege

enforcement, and alignment with zero-trust architecture principles.

- **Digital Identity Wallet Architectures:** Provide secure storage and management of credentials, cryptographic keys, and delegated authority. SpruceID delivers wallet infrastructure for both end users and software agents, supporting selective disclosure, pairwise identifiers, and portable trust across applications and jurisdictions.

- **Delegation and Trust Frameworks:** Extend established identity patterns to support delegation of authority across individuals, organizations, and software agents. SpruceID enables verifiable delegation chains, allowing actions to be performed on behalf of a user or entity with clear attribution and auditability.

- **ERC-8004 and Blockchain-Based Trust Infrastructure:** Emerging standards such as ERC-8004 define a structured model for recording attestations, including identity-linked claims and authorization artifacts, on blockchain networks. ERC-8004 enables issuers to anchor cryptographic references to credentials, schemas, or trust metadata in a publicly verifiable registry, allowing any party to independently verify the integrity and provenance of those artifacts without relying on a centralized intermediary.

  SpruceID supports the use of blockchain-based infrastructure where appropriate as a trust anchor for public registries, including credential schemas, issuer registries, public keys, and tamper-evident audit logs. These systems provide strong guarantees around integrity, transparency, and cross-organizational trust, particularly in multi-party environments where no single entity is universally trusted.

  Rather than serving as a system of record for sensitive data, blockchain infrastructure is used selectively to anchor high-value artifacts and cryptographic commitments, enabling independent verification while keeping sensitive data off-chain and under the control of the appropriate system of record.

SpruceID builds verifiable digital credential infrastructure designed for agentic-first use cases. Our open-source libraries and credential platform services enable identity wallets for agents, credential-based access to regulated environments (e.g., stablecoin payments, operating on behalf of a business), and the binding of credentials to actions and authorization capabilities. We would welcome the opportunity to collaborate with NCCoE on demonstration projects in this space.

Spruce Systems, Inc.
228 Park Ave S, PMB 28788
New York, New York 10003-1502 US

Conclusion

We commend NIST and the NCCoE for proactively addressing the identity and authorization challenges posed by AI agents. The shift from deterministic software automation to generative AI-driven agents demands new thinking about identity, trust, and governance and we believe that verifiable credentials, capability-based authorization, and formally verifiable policies provide the right architectural foundations.

We look forward to continued engagement on this important work.

Respectfully submitted,

The SpruceID Team