Network Working Group
Internet-Draft
Intended status: Informational
Expires: 3 October 2026

B. W. Beyer
Independent

1 April 2026

Architecture for Human-Anchored Agent Identity, Delegation, and Provenance

draft-beyer-agent-identity-architecture-00

Abstract

Software agents increasingly act on behalf of people across communication, automation, and decision-making contexts. These agents initiate actions, delegate tasks, and interact with other agents without a consistent model for representing the human who authorized them, the scope of authority they possess, or the provenance of their actions across ecosystems.

This document defines an architectural model for human-anchored agent identity. The model introduces a human identity root, explicit delegation semantics, and a provenance structure that enables ecosystems to determine whether an agent is legitimate, whether it is acting within its intended authority, and how its actions relate to a responsible human.

This document does not define a protocol or wire format. It provides an architectural foundation that existing systems may bind to in order to support accountable, interoperable, and human-aligned agent ecosystems.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 3 October 2026.

Table of Contents

1. Introduction

Software agents increasingly act on behalf of people across communication, automation, and decision-making contexts. These agents initiate actions, delegate tasks, and interact with other agents without continuous human supervision. As their capabilities expand, ecosystems require a consistent way to determine whether an agent is legitimate, whether it is acting within its intended authority, and how its actions relate to a responsible human.

Existing identity systems authenticate software components, devices, or network endpoints, but they do not provide a durable representation of the human who authorized an agent or the delegation chain that governs its behavior. They also do not preserve provenance as agents move across platforms. These gaps limit accountability, create opportunities for impersonation or unauthorized replication, and make it difficult for ecosystems to reason about agent behavior in a consistent way.

This document defines an architectural model for human-anchored agent identity. The model introduces three core elements: a human identity root that represents the responsible human; explicit delegation semantics that express the scope and conditions of authority; and a provenance structure that enables ecosystems to understand how an agent's actions relate to a responsible human across contexts.

The architecture is transport-agnostic and does not define a protocol or wire format. Instead, it provides a conceptual foundation that existing systems may bind to in order to support accountable, interoperable, and human-aligned agent ecosystems. The model is intended to complement, not replace, existing identity and authorization mechanisms.

The goal of this document is to provide a consistent architectural framework that preserves human authority, enables explicit delegation, and maintains verifiable provenance across agent interactions, while avoiding global identifiers or mechanisms that enable cross-context tracking.

2. Core Identity Model

The architectural model for human-anchored agent identity is built on three foundational constructs: a human identity root that represents the responsible human; an agent identity that represents a software agent acting on behalf of that human; and explicit delegation semantics that express the scope and conditions under which the agent is authorized to act. These constructs provide a consistent way for ecosystems to understand how agent behavior relates to human authority across contexts.

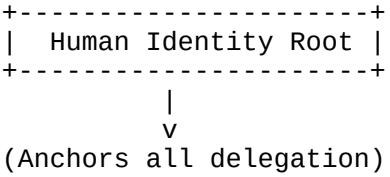

Figure 1: Human Identity Root

2.1. Human Identity Root

The human identity root represents the human who ultimately bears responsibility for an agent’s actions. It is an architectural role, not a global identifier or account. The human identity root provides a stable point of reference for delegation chains and provenance, enabling ecosystems to determine which human authorized an agent without requiring cross-context tracking or universal identifiers.

The architecture does not prescribe how the human identity root is instantiated, stored, or authenticated. It may be bound to existing identity systems, credentials, or verification mechanisms, provided that the binding preserves privacy and avoids unnecessary linkability across contexts.

2.2. Agent Identity

An agent identity represents a software agent acting on behalf of a human. It is distinct from the human identity root and expresses the agent's role, capabilities, and relationship to the human who authorized it. An agent identity may be instantiated multiple times, provided that each instance can be related to the human identity root through a delegation chain.

The architecture does not define a credential format, key structure, or transport mechanism for agent identities. Instead, it provides a conceptual model that existing systems may bind to in order to represent agents in a consistent and interoperable way.

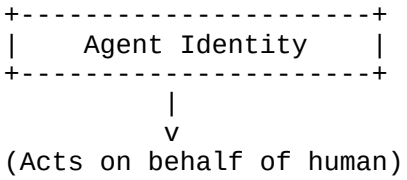

Figure 2: Agent Identity

2.3. Delegation Semantics

Delegation semantics express the scope, duration, and conditions under which an agent is authorized to act on behalf of a human. Delegation is explicit and portable across contexts, enabling ecosystems to determine whether an agent is acting within its intended authority and whether a delegation chain remains valid over time.

Delegation semantics do not prescribe a protocol or wire format. They define the architectural properties that a delegation representation must support, including the ability to express scope, constraints, and revocation at the delegation-chain level rather than at the transport level.

2.4. Relationships Among Core Constructs

The human identity root, agent identity, and delegation semantics form a coherent model that enables ecosystems to understand how agent behavior relates to human authority. The human identity root anchors the delegation chain; the delegation semantics express the conditions of authority; and the agent identity represents the software agent acting under that authority. Together, these constructs provide a foundation for accountability, interoperability, and provenance across agent ecosystems.

3. Delegation Chains

Delegation chains express how authority flows from a responsible human to one or more software agents. A delegation chain begins at the human identity root and proceeds through one or more delegation steps, each of which grants a specific scope of authority to an agent. Delegation chains provide ecosystems with a consistent way to determine whether an agent is legitimate, whether it is acting within its intended authority, and how its actions relate to a responsible human across contexts.

3.1. Structure of a Delegation Chain

A delegation chain consists of a sequence of delegation steps, each of which expresses a grant of authority from one entity to another. The first step originates at the human identity root, and subsequent steps may delegate authority to additional agents. Each step expresses the scope, duration, and conditions under which the delegated authority is valid.

Delegation steps are explicit and portable across contexts. They do not rely on application-specific logic or implicit assumptions about authority. This enables ecosystems to interpret delegation chains consistently, even when agents move across organizational or technical boundaries.

3.2. Scope and Constraints

Each delegation step expresses the scope of authority granted to an agent. Scope may include the types of actions the agent is permitted to perform, the contexts in which those actions are valid, or other constraints that limit the agent’s authority. Delegation steps may also express temporal constraints, such as expiration or duration.

The architecture does not prescribe a specific representation for scope or constraints. Instead, it defines the architectural properties that a scope representation must support, including the ability to express limitations, conditions, and revocation in a way that is interpretable across contexts.

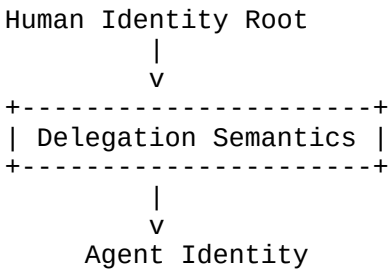

Figure 3: Delegation Semantics

3.3. Nested Delegation

Delegation chains may include nested delegation, in which an agent that has been granted authority may delegate a subset of that authority to another agent. Nested delegation enables complex agent ecosystems while preserving accountability and traceability to the human identity root.

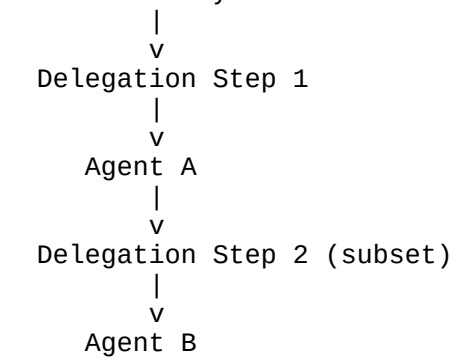

Figure 4: Nested Delegation

The architecture does not require nested delegation, but it supports it as a natural extension of the delegation model. Each nested delegation step must remain within the scope of the authority granted by the preceding step.

3.4. Revocation

Revocation applies at the delegation-chain level rather than at the transport level. A delegation step may be revoked by the entity that issued it, and revocation invalidates all subsequent steps in the chain. This enables ecosystems to determine whether a delegation chain remains valid without relying on transport-specific mechanisms.

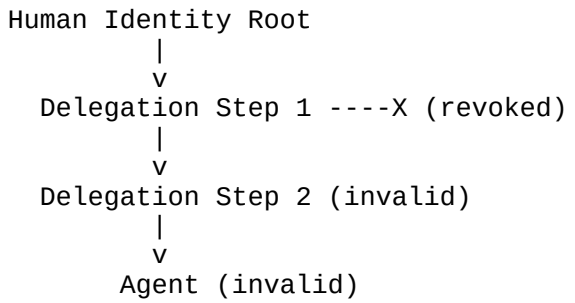

Figure 5: Revocation Flow

The architecture does not define a protocol for distributing revocation information. It defines the conceptual model in which revocation is expressed and interpreted.

3.5. Properties of Delegation Chains

Delegation chains provide several architectural properties that support accountable and interoperable agent ecosystems:

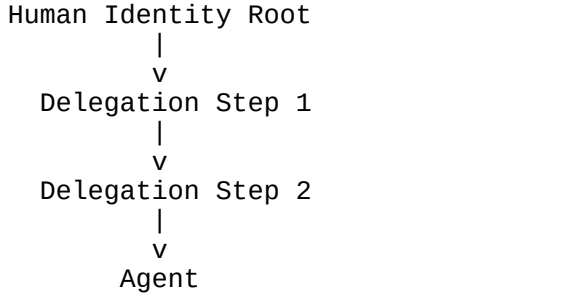

Figure 6: Delegation Chain Structure

* *Explicitness* — Delegation is expressed explicitly rather than inferred from context or application logic.

* *Portability* — Delegation chains can be interpreted across ecosystems without relying on platform-specific assumptions.

* *Traceability* — Each step in the chain can be related to the human identity root, enabling accountability.

* *Scoping* — Delegation steps express the scope and conditions of authority in a consistent way.

* *Revocability* — Delegation steps can be revoked, and revocation invalidates subsequent steps in the chain.

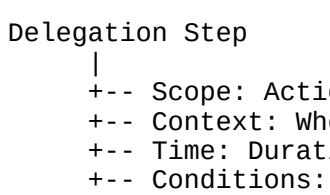

Figure 7: Scope and Constraints

These properties enable ecosystems to reason about agent authority in a consistent and interoperable way, even as agents move across platforms or interact with other agents.

4. Constitutional Contracts

Constitutional contracts define the governance layer that constrains how agents may act on behalf of a human. They express the durable, high-level conditions under which authority may be delegated, exercised, or revoked. Constitutional contracts do not describe runtime behavior, enforcement mechanisms, or policy languages. Instead, they provide an architectural boundary that ensures agent behavior remains anchored to human intent.

4.1. Role of Constitutional Contracts

A constitutional contract establishes the foundational rules that govern the relationship between a human identity root and the agents acting on its behalf. These rules define the permissible structure of delegation chains, the constraints on agent authority, and the conditions under which revocation or modification may occur. The contract provides a stable reference point that ecosystems can rely on when interpreting agent behavior across contexts.

The constitutional contract is not a protocol artifact. It is an architectural construct that informs how systems reason about authority, delegation, and provenance without prescribing how these concepts are encoded or transmitted.

4.2. Scope of a Constitutional Contract

A constitutional contract expresses durable constraints on agent authority. These constraints may include:

* *Structural constraints* — rules governing the shape and depth of delegation chains.

* *Authority constraints* — limits on the types of actions an agent may perform or the contexts in which those actions are valid.

* *Revocation constraints* — conditions under which authority may be withdrawn and how revocation affects downstream delegation steps.

* *Replication constraints* — rules governing when and how agents may be instantiated or replicated.

These constraints provide ecosystems with a consistent way to interpret agent authority without requiring global identifiers or cross-context tracking.

4.3. Portability Across Ecosystems

Constitutional contracts are portable across platforms and ecosystems. They do not rely on platform-specific semantics or enforcement mechanisms. This portability enables agents to operate across organizational boundaries while preserving the architectural guarantees of human anchoring, explicit delegation, and durable provenance.

The architecture does not prescribe how constitutional contracts are stored, distributed, or validated. It defines the conceptual properties that a contract must support in order to remain interpretable across contexts.

4.4. Non-Goals

Constitutional contracts are not intended to serve as:

* *Policy engines* — they do not define runtime decision logic.

* *Protocol elements* — they do not specify message formats or transport semantics.

* *Cryptographic frameworks* — they do not select algorithms or key structures.

* *Behavioral safety systems* — they do not govern agent cognition or internal decision-making.

These non-goals ensure that constitutional contracts remain an architectural governance layer rather than an operational mechanism.

4.5. Summary

Constitutional contracts provide the durable governance structure that binds human authority to agent behavior. They define the architectural boundaries within which delegation, replication, and provenance operate, enabling ecosystems to interpret agent actions consistently without imposing protocol semantics or platform-specific assumptions.

5. Replication and Provenance

Replication and provenance describe how agent instances relate to one another and how their actions can be traced back to a responsible human. Replication refers to the creation of new agent instances, whether through copying, instantiation, or migration across platforms. Provenance refers to the durable record of how an agent's authority, identity, and actions relate to the human identity root and the delegation chain that authorized them.

5.1. Replication Model

Agents may be instantiated or replicated across platforms in order to perform tasks, operate in different environments, or maintain continuity of service. The architecture treats replication as an expected property of agent ecosystems, provided that each replicated instance can be related to the human identity root through a valid delegation chain.

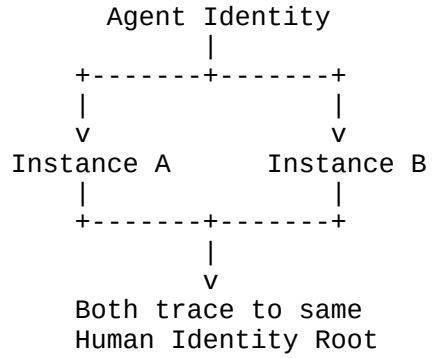

Figure 8: Replication Model

The architecture does not prescribe how replication occurs or how instances are created. Instead, it defines the conceptual requirement that replicated agents must remain within the scope of the authority granted by the human identity root and must maintain a verifiable relationship to the delegation chain that authorized them.

5.2. Constraints on Replication

Replication is constrained by the constitutional contract and the delegation semantics that govern agent authority. These constraints may limit the number of instances, the contexts in which replication is permitted, or the scope of authority available to replicated agents. Replication must not create new authority; it may only reproduce authority that has already been granted.

These constraints ensure that replication does not expand an agent's authority beyond what the human identity root intended and that ecosystems can distinguish authorized instances from unauthorized copies.

5.3. Provenance Model

Provenance provides a durable record of how an agent's actions relate to the human identity root and the delegation chain that authorized them. Provenance is architectural rather than operational: it describes the information that ecosystems must be able to interpret, not how that information is encoded, stored, or transmitted.

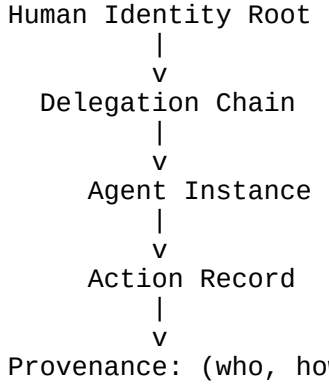

Figure 9: Provenance Model

Provenance enables ecosystems to determine:

* which human authorized the agent,
* which delegation chain applied at the time of the action,
* whether the agent was acting within its intended scope,
* and how the agent instance relates to other instances.

These properties support accountability and interoperability across agent ecosystems without requiring global identifiers or cross-context tracking.

5.4. Portability Across Contexts

Provenance must remain interpretable as agents move across platforms or interact with other agents. The architecture does not prescribe a specific representation for provenance, but it requires that provenance be portable and that it preserve the relationship between agent actions, delegation chains, and the human identity root.

Portability ensures that ecosystems can reason about agent behavior even when agents operate in environments with different identity systems, storage models, or authorization mechanisms.

5.5. Summary

Replication and provenance provide the architectural foundation for understanding how agent instances relate to one another and how their actions relate to a responsible human. Replication enables agents to operate across contexts, while provenance ensures that their actions remain accountable and interpretable. Together, these constructs support safe, interoperable, and human-aligned agent ecosystems.

6. Interaction Model

The interaction model describes how agents, humans, and ecosystems relate to one another within the architectural framework. It defines the conceptual roles involved in agent interactions and the architectural properties that enable ecosystems to interpret those interactions consistently. The model does not prescribe protocols, message formats, or transport mechanisms. Instead, it provides a foundation that existing systems may bind to in order to support accountable and interoperable agent behavior.

6.1. Roles in the Interaction Model

The interaction model includes three primary roles:

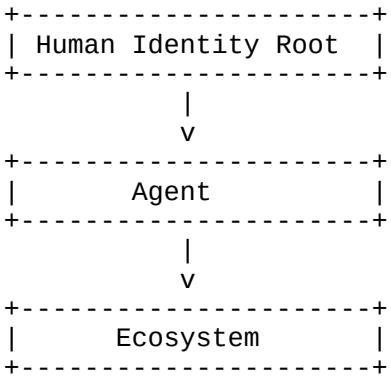

Figure 10: Roles in the Interaction Model

* *Human identity root* — the responsible human whose authority anchors the delegation chain.

* *Agent* — a software entity acting on behalf of the human identity root under a defined scope of authority.

* *Ecosystem* — the environment in which agents operate, including platforms, services, and other agents.

These roles provide a consistent conceptual structure for interpreting agent behavior across contexts without imposing platform-specific assumptions.

6.2. Authority in Interactions

Agent interactions are governed by the delegation chain that connects the agent to the human identity root. Ecosystems interpret agent actions through the lens of this delegation chain, determining whether the agent is acting within its intended authority and whether the delegation chain remains valid.

The architecture does not define how authority is enforced or validated at runtime. It defines the conceptual requirement that authority must be interpretable across contexts and must remain anchored to the human identity root.

6.3. Cross-Context Interactions

Agents frequently operate across multiple platforms, services, or organizational boundaries. The interaction model treats cross-context operation as a normal property of agent ecosystems. To remain interpretable, agent interactions must preserve the relationship between the agent, the delegation chain, and the human identity root as the agent moves across contexts.

The architecture does not prescribe how cross-context information is transmitted or stored. It defines the architectural requirement that ecosystems must be able to understand how an agent's authority and provenance relate to the human identity root, regardless of the environment in which the interaction occurs.

6.4. Agent-to-Agent Interactions

Agents may interact directly with other agents. In these interactions, each agent presents its own identity and delegation chain, enabling the receiving agent or ecosystem to determine whether the interaction is legitimate and whether the initiating agent is acting within its intended authority.

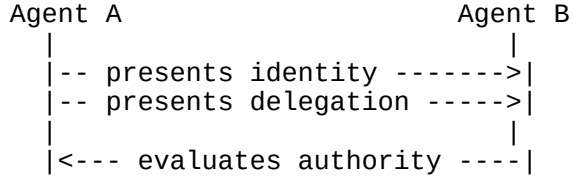

Figure 11: Agent-to-Agent Interaction

The architecture does not define a negotiation protocol or trust mechanism for agent-to-agent interactions. It defines the conceptual requirement that each agent's authority and provenance must be interpretable by other agents and ecosystems.

6.5. Human-Agent Interactions

Humans may interact directly with agents to authorize actions, delegate authority, or modify constraints. These interactions are governed by the constitutional contract and the delegation semantics that define how authority flows from the human identity root to the agent.

The architecture does not prescribe user interfaces, authentication mechanisms, or interaction workflows. It defines the conceptual requirement that human-agent interactions must preserve the integrity of the delegation chain and must not introduce ambiguity about the scope of authority granted.

6.6. Summary

The interaction model provides a conceptual framework for understanding how agents, humans, and ecosystems relate to one another. It ensures that agent interactions remain interpretable, accountable, and anchored to human authority without prescribing protocols or platform-specific mechanisms. This model supports interoperable and human-aligned agent ecosystems across diverse environments.

7. Interoperability with Existing Systems

The architectural model for human-anchored agent identity is designed to complement existing identity, authorization, and credential systems rather than replace them. Interoperability is achieved by providing a conceptual framework that existing systems may bind to, enabling ecosystems to interpret human authority, delegation, and provenance in a consistent way without requiring changes to underlying protocols or infrastructure.

7.1. Binding to Existing Identity Systems

The architecture does not prescribe how identity systems must represent humans or agents. Instead, it defines the conceptual roles of the human identity root and agent identity, which may be bound to existing identifiers, credentials, or authentication mechanisms. These bindings allow platforms to adopt the architectural model without introducing new global identifiers or cross-context tracking mechanisms.

A binding may be implemented through existing account systems, credential formats, or verification processes, provided that the binding preserves the architectural properties of human anchoring, explicit delegation, and durable provenance.

7.2. Integration with Authorization Frameworks

Authorization frameworks often define how permissions are granted, scoped, or enforced within a platform. The architectural model complements these frameworks by providing a higher-level representation of authority that originates at the human identity root and flows through delegation chains.

The architecture does not replace authorization mechanisms or define how permissions are evaluated at runtime. Instead, it provides a conceptual structure that authorization systems may reference when interpreting the scope and legitimacy of agent actions.

7.3. Use with Credential and Key Systems

Many systems rely on credentials, key pairs, or cryptographic material to authenticate software components or establish secure channels. The architectural model does not prescribe specific key structures or credential formats. Instead, it defines the conceptual relationship between credentials and the delegation chain that authorizes an agent.

Existing credential systems may be used to instantiate agent identities or to express delegation steps, provided that they can represent the architectural properties required for human anchoring, scope, and provenance.

7.4. Cross-Context Interpretation

Agents frequently operate across platforms, organizations, or technical environments. The architecture treats cross-context operation as a normal property of agent ecosystems. To remain interoperable, systems must be able to interpret the relationship between an agent, its delegation chain, and the human identity root regardless of the environment in which the interaction occurs.

The architecture does not define how cross-context information is exchanged. It defines the conceptual requirements that enable ecosystems to interpret authority and provenance consistently across boundaries.

7.5. Non-Goals

Interoperability within this architecture does not require:

* *Protocol changes* — existing protocols need not be modified to adopt the architectural model.

* *New global identifiers* — the architecture avoids mechanisms that enable cross-context tracking.

* *Uniform credential formats* — systems may use their existing credential structures.

* *Centralized trust anchors* — the architecture does not introduce new trust hierarchies.

These non-goals ensure that the architecture remains compatible with diverse ecosystems and can be adopted incrementally.

7.6. Summary

Interoperability is achieved by providing a conceptual model that existing systems may bind to without requiring changes to protocols, credentials, or authorization frameworks. The architecture enables ecosystems to interpret human authority, delegation, and provenance consistently across contexts while preserving privacy and avoiding global identifiers.

8. Security and Privacy Considerations

The architectural model for human-anchored agent identity introduces a consistent way to relate agent behavior to human authority. While the model does not define protocols, message formats, or enforcement mechanisms, it raises several security and privacy considerations that ecosystems must evaluate when binding the architecture to existing systems.

8.1. Human Anchoring

The architecture requires that agent authority be traceable to a responsible human through a delegation chain. Systems that bind to this architecture must ensure that the binding between the human identity root and the underlying identity system is resistant to impersonation, unauthorized substitution, or misuse. Failure to maintain a secure binding may allow agents to act without legitimate human authorization.

8.2. Delegation Integrity

Delegation chains express how authority flows from a human to one or more agents. If delegation steps can be forged, modified, or replayed across contexts, ecosystems may incorrectly interpret an agent’s authority. Systems that implement delegation semantics must ensure that delegation steps cannot be altered without detection and that revocation is interpreted consistently across contexts.

8.3. Replication Risks

Replication enables agents to operate across platforms, but it also creates opportunities for unauthorized copies or uncontrolled proliferation. Systems that support replication must ensure that replicated instances remain within the scope of the authority granted by the human identity root and that unauthorized replication does not result in expanded or ambiguous authority.

8.4. Provenance Risks

Provenance provides a durable record of how agent actions relate to the human identity root and the delegation chain. If provenance is incomplete, inconsistent, or lost across contexts, ecosystems may be unable to determine whether an agent acted legitimately. Systems that bind to this architecture must ensure that provenance remains interpretable without introducing global identifiers or mechanisms that enable cross-context tracking.

8.5. Privacy Considerations

The architecture avoids global identifiers and does not require cross-context correlation of agent or human identities. However, systems that bind to the architecture may inadvertently introduce linkability if they reuse identifiers across contexts or expose delegation chains in ways that reveal sensitive information. Implementations must ensure that privacy-preserving practices are maintained when representing human identity roots, agent identities, or delegation chains.

8.6. Ecosystem Fragmentation

Because ecosystems vary widely in their identity and authorization models, inconsistent interpretations of delegation, replication, or provenance may create security gaps. Systems that adopt the architecture must ensure that cross-context interpretation does not rely on assumptions that are valid only within a single platform or environment.

8.7. Summary

The architecture provides a conceptual model for relating agent behavior to human authority. While it does not define enforcement mechanisms, systems that bind to the architecture must ensure that human anchoring, delegation integrity, replication constraints, and provenance preservation are handled securely and in a privacy-preserving manner. These considerations are essential for supporting accountable, interoperable, and human-aligned agent ecosystems.

9. IANA Considerations

This document has no IANA actions.

10. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, 1997, <https://www.rfc-editor.org/rfc/rfc2119>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, 2017, <https://www.rfc-editor.org/rfc/rfc8174>.

11. Informative References

[RFC6749] Hardt, D., "The OAuth 2.0 Authorization Framework", RFC 6749, 2012, <https://www.rfc-editor.org/rfc/rfc6749>.

[RFC7643] Hunt, P., Ansari, A., Sanchez, M., and K. McCloghrie, "System for Cross-domain Identity Management: Core Schema", RFC 7643, 2015, <https://www.rfc-editor.org/rfc/rfc7643>.

[DID-Core] Sporny, M., Longley, D., and C. Allen, "Decentralized Identifiers (DIDs) v1.0", W3C Recommendation DID-Core, 2022, <https://www.w3.org/TR/did-core/>.

Appendix A. Figures

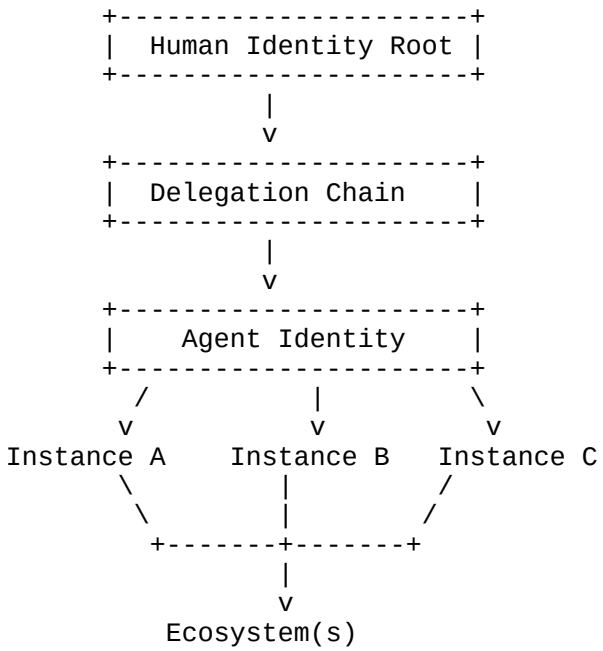

Figure 12: Complete Agent Ecosystem Overview

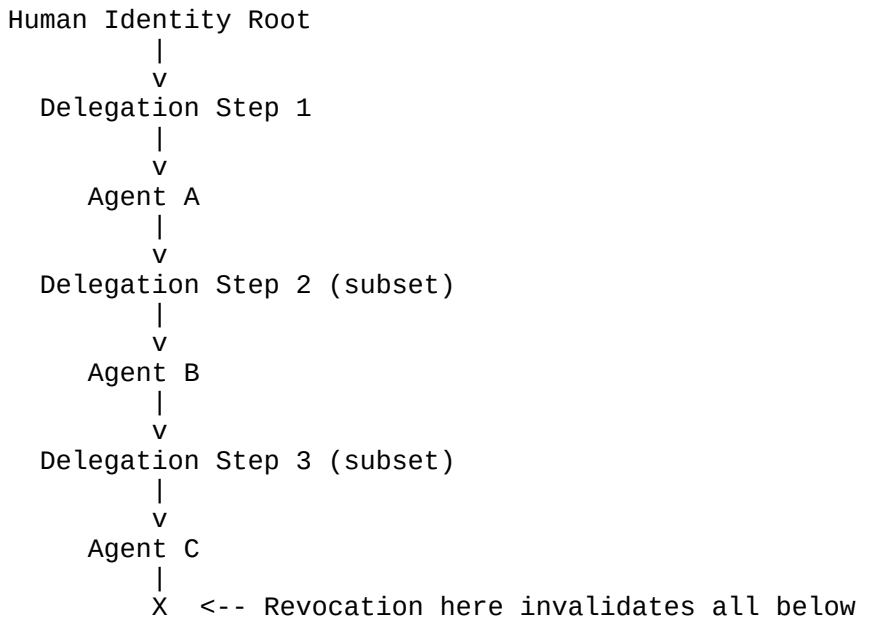

Figure 13: Full Delegation Chain with Nesting and Revocation

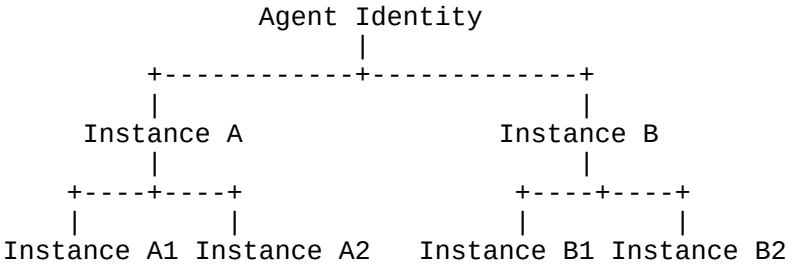

Figure 14: Replication Lineage Tree

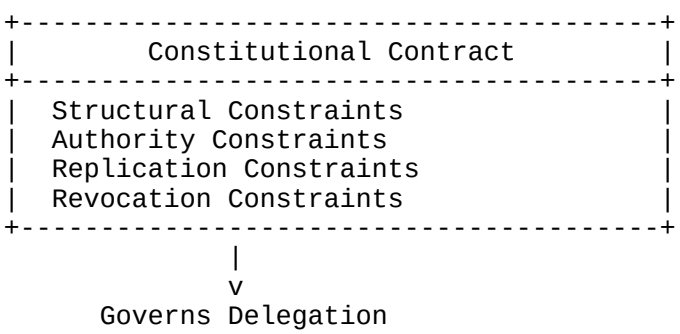

Figure 15: Constitutional Contract Structure

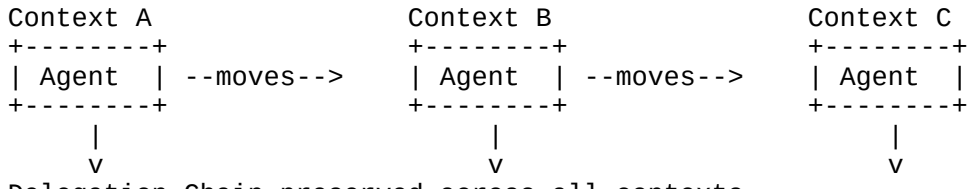

Figure 16: Cross-Context Interaction Map

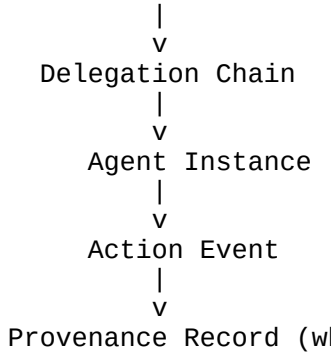

Figure 17: End-to-End Provenance Flow

Author’s Address

Brandon Wesley Beyer
Independent
Email: brandnbyr@icloud.com