Network Working Group
Internet-Draft
Intended status: Informational
Expires: 5 December 2026

Y. Jiang
L. Li
Y. Song
F. Liu
Huawei
3 June 2026

Security Considerations for Intent-Based Requests in Agentic Systems
draft-jiang-intent-security-02

Abstract

Intent-based requests enable users, applications, and agents to express goals and constraints without specifying step-by-step procedures. Such intents are commonly translated into executable directives and propagated across multiple entities (clients, agents, authorization components, orchestration functions, and execution endpoints). This multi-hop processing expands the attack surface for tampering, privilege escalation, constraint bypass, and intent drift. In addition, at the point where an intent enters the system, a forged or unauthorized origin may cause actions to be taken without valid consent.

This document provides a solution-agnostic security analysis for intent-based requests across agentic systems. It introduces a reference model and scenarios to guide protocol and system design, and also presents threats and requirements. The document emphasizes origin authentication and admission control, constraint validation, invocation validation, multi-hop chain-of-custody, and policy-driven responses to drift, while remaining independent of any specific deployment domain.

Status of This Memo

This Internet-Draft will expire on 5 December 2026.

Copyright Notice

Table of Contents

1. Introduction

Intent-based interaction is increasingly adopted in automation, orchestration, and agentic systems, where a request expresses desired outcomes and constraints rather than explicit procedures. A receiving system (or a chain of systems) translates the intent into executable directives and invokes tools or services to achieve the intended outcome.

Multi-hop processing (client-to-agent, agent-to-agent, agent-to-tool/service) introduces security risks beyond traditional single-hop APIs, including: (1) integrity and substitution attacks against derived directives, (2) privilege escalation during tool/service invocation, (3) constraint bypass, (4) multi-hop intent drift where constraints degrade or diverge over transformations, and (5) admission-time risks where an intent of spoofed or unauthorized origin is accepted and acted upon without valid consent.

This document does not define a new protocol. Instead, it provides a security-oriented reference model, threat analysis, requirements, and scenarios to support future standardization and interoperable designs.

1.1. Scope

This document focuses on security considerations for intent-based requests in multi-hop agentic systems. While examples may reference telecom or networking contexts, the analysis applies broadly to any domain where intent processing spans multiple trust boundaries.

2. Terminology and Conventions

2.1. Conventions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

2.2. Definitions

This document uses the following terms:

Intent: A declarative expression of desired operational goals and outcomes, without specifying how to achieve or implement them. This definition is aligned with intent-based networking (IBN) guidance [RFC9315] [RFC9316].

Intent Translation: The process of transforming an intent into more concrete representations, such as constraints, objectives, candidate procedures, or executable directives.

Constraint: A condition that limits acceptable outcomes or actions. Constraints may include invariants, policy rules, safety boundaries, and compliance requirements.

Constraint Validation: Verifying whether an intent and/or its derived artifacts comply with applicable constraints, invariants, policy rules, and safety boundary requirements.

Invocation: A request to a tool or service intended to fulfill an intent (e.g., API call, workflow step, actuation command).

Invocation Validation: Determining whether an invoker holds the required privileges to invoke a tool or service and whether invocation parameters satisfy the requirements and constraints specified by the intent.

Observation: Telemetry, events, measurements, or other signals used for monitoring and assurance.

Drift: A divergence between the intent (including its constraints) and the realized plan or actions over time or across multi-hop transformations.

Derived Directive: An executable or enforceable artifact generated from an intent through translation, such as an allowed rule set, capability token, or authorization grant.

Intent Originator: The entity that produces an intent (e.g., a user, an application, or an agent). The originator is the asserted source of the intent and is distinct from the entity that merely transports or forwards it.

Admission Control: The decision, taken before an intent is forwarded into or accepted by the processing chain, of whether to admit the intent, based on the verified origin, the originator’s authorization to request the targeted service, and applicable consent.

2.3. Acronyms

IBN: Intent-Based Networking

IBS: Intent-Based System

3. Problem Statement and Threat Model

In many agentic systems, an intent is translated into executable directives (e.g., an allowed rule set) that must be propagated across multiple entities and enforced at execution endpoints. However, existing designs often lack end-to-end mechanisms that jointly ensure: (1) the intent originates from an authenticated and authorized originator and has obtained any required consent before admission, (2) directives remain within authorized boundaries across transformations and propagation, (3) constraints are validated before execution, (4) invocations are privilege-checked and constraint-checked at each call boundary, and (5) drift is detected and handled under policy.

3.1. Example of multi-hop intent architecture

In modern agentic systems, complex user intents often exceed the capabilities of single-agent architectures. Multi-agent systems provide a robust framework for intent processing through specialized role assignments, parallel execution, and structured intent propagation. In this section, observing the numerous agent architectures, a typical architecture is present that demonstrate effective intent processing patterns across domains, including: sequential orchestration for intent-based processing, parallel systems for information gathering, etc. The core principle is that intent decomposition and propagation are fundamental to multi-agent collaboration. Each architecture implements this principle differently based on domain requirements, but all maintain the critical property that intent must be preserved, verifiable, and contextually appropriate as it flows through the system.

```
        |intent input
        v
+--------------------+
|       Agent1       |
| (e.g., Summarizer) |
+--------------------+
        v
+--------------------+
|      Agent2        |
| (e.g., Translator) |
+--------------------+
        v
+--------------------+
|      Agent3        |
|   (e.g., QA)       |
+--------------------+
        |    Result
        v
```

Figure 1: Example of the typical multi-agent processing

Specifically, the architecture for multi-hop intent processing is shown in Figure 1, following the sequential orchestration pattern of the Microsoft Agent Framework [MS-AF-SEQ]. In the figure, agents process intents in a pipeline fashion. By default, each agent receives the conversation from the previous agent, ensuring context preservation while allowing specialized processing at each stage.

The architecture consists of:

1. Agent Pipeline: Agents are arranged in a predefined sequence where output from one agent becomes input for the next.

2. Shared Context: By default, each agent consumes the previous agent’s full conversation, so context is preserved across the pipeline. The framework also allows agents to be configured to consume only the previous agent’s response messages, which truncates earlier context.

3. Human-in-the-Loop: Optional approval points for sensitive operations (e.g., tools marked as approval-required).

4. Mixed Executors: Ability to combine LLM-based agents with custom code executors.

Usually, the original intent can be preserved through the shared conversation history, with each agent adding specialized processing while maintaining contextual continuity. The system demonstrates that even simple sequential architectures can effectively process complex intents when agents have clearly defined roles and shared context. However, under certain specific threats, the intent may change, potentially introducing security risks. The next section will focus on this in detail under the architecture.

3.2. Threats

Based on the typical multi-agent processing in Section 3.1 (Figure 1), the following representative threats are considered. T1-T5 arise during multi-hop intent processing, while T6-T7 arise at the intent origination/admission boundary, i.e., where an intent enters the system:

T1 (Directive Tampering/Substitution): A malicious intermediary agent modifies conversation history or derived directives between pipeline stages, altering budget constraints or action parameters while maintaining superficial coherence.

T2 (Unauthorized Invocation / Privilege Escalation): An agent abuses mixed-executor capabilities by smuggling unauthorized commands through parameter injection, bypassing the intended privilege boundary because custom code executors run arbitrary code without an enforced access-control layer.

T3 (Constraint Bypass): Security constraints degrade across pipeline stages due to context truncation or improper inheritance, violating the shared-context integrity assumption.

T4 (Multi-Hop Semantic Drift): Gradual semantic deviation occurs as agents reinterpret ambiguous instructions across hops, causing final actions to diverge from original intent boundaries despite syntactically intact messages.

T5 (Monitoring Evasion / False Observations): Attackers evade detection by suppressing, forging, or selectively presenting observations used for assurance and drift detection.

T6 (Origin Spoofing / Forged Provenance): A co-resident or upstream malicious application or agent fabricates an intent artifact that appears to originate from the user or a legitimate originator, so that downstream entities accept and act on an intent that the claimed originator never authorized.

T7 (Unauthorized or Non-Consensual Origination): An originator that is not entitled to request the targeted service, or that is operating in a disallowed state (e.g., running unattended in the background), issues a high-impact intent without valid user consent; the system admits it because it does not gate on the originator’s eligibility, runtime state, or explicit consent.

NOTE1: By default, the example framework does not provide cryptographic binding between pipeline stages, so agents cannot be assumed to be mutually trustworthy. The degree of such binding may vary across implementations.

NOTE2: Human approval may be agent-triggered rather than enforced by the receiver, creating bypass opportunities.

3.3. Requirements

Based on the threats above, this document identifies the following security requirements:

R1 (Provenance and Authorization Boundary Binding): The system provides a verifiable binding between the intent, derived directives, and the applicable authorization boundary, such that unauthorized expansion can be detected or prevented.

R2 (Chain-of-Custody for Derived Directives): The system protects derived directives against tampering and substitution across multi-hop propagation.

R3 (Constraint Validation): The system validates the intent and/or derived artifacts against applicable constraints, invariants, policy rules, and safety boundaries before accepting or executing actions.

R4 (Invocation Validation): The system validates that an invoker holds the required privileges to invoke a tool/service and that invocation parameters satisfy intent constraints prior to and/or at invocation time.

R5 (Non-Bypass Enforcement): The execution endpoint enforces constraints and authorization boundaries such that direct/side-path invocation cannot bypass required checks.

R6 (Observability and Auditability): The system provides sufficient observations and audit evidence to support compliance assessment, drift detection, and incident investigation.

R7 (Policy-Driven Drift Response): Upon drift detection or constraint violation, the system supports policy-driven responses (e.g., deny, degrade, re-confirm, re-negotiate, fallback).

R8 (Origin Authentication): The system provides a verifiable binding between an intent artifact and the identity of its actual originator, such that a forged or spoofed origin can be detected before the intent is admitted or forwarded.

R9 (Originator Authorization and Consent-Gated Admission): Before admitting or forwarding an intent, the system determines whether the originator is permitted to request the targeted service, based on originator attributes (e.g., identity, type, runtime state, history) and applicable permission policy. For high-impact or irreversible actions, the system additionally obtains and verifies explicit user consent. Intents from unauthorized originators, from originators in a disallowed state, or lacking required consent are rejected or escalated.

4. Reference Model

This section introduces a technology-neutral reference model for intent-based requests. The model is aligned with intent-based system decomposition commonly used in IBN guidance [RFC9315], while remaining applicable to non-networking domains.

```
   +--------------+           +---------------------------+
   |  User Space  |           |       IBS Space           |
   |              |  Intent   |                           |
   |   Intent     |---------->|  Intent Processing Func   |
   |  Originator  | Artifact  |                           |
   |              |---------->|  +---------+  +---------+ |
   |   Intent     |           |  | Intent  |  |Constrain| |
   |   Client     |           |  |Transform|  |Validate |-+-> Policy/
   +--------------+           |  +---------+  +---------+ |   Constrain
                              |                           |   Authority
                              |  +---------+              |
                              |  |Invocate |              |
                              |  | Gate    |--------------+-> Tool/
                              |  +---------+              |   Service
                              +---------------------------+   Provider
                                       ^          |
                                       |   Observe|
                                       |          v
                              +---------------------------+
                              |  Observer (Monitoring)    |
                              +---------------------------+
```

Figure 2: Reference Model for Multi-Hop Intent Processing

The figure separates User Space from IBS Space for clarity. Deployments may collapse functions into fewer components or distribute them across multiple agents and services.

4.1. Reference Model Entities

The following entities are defined in the reference model:

Intent Originator: The party whose goals and constraints are to be satisfied (e.g., human user, application owner, operator, or delegated principal).

Intent Client: The component that submits intents to an IBS and may carry contextual signals. The Intent Client (or an equivalent admission point) may also enforce origin authentication and originator-level admission control before an intent is forwarded (see R8 and R9).

Intent Processing Function: A logical function that performs translation, validation, and orchestration for intent fulfillment. This function encompasses the Intent Transformer, Constraint Validator, and Invocation Gate.

Intent Transformer: A function that transforms intent representations (e.g., natural language to structured intent, structured intent to constraints/objectives, objectives to derived directives).

Constraint Validator: A function that enforces R3 by validating intents and derived artifacts against constraints, invariants, policy rules, and safety boundaries.

Invocation Gate: A function that enforces R4 and R5 by privilege-checking and constraint-checking each tool/service invocation and preventing bypass of required checks.

Policy/Constraint Authority: A logical source of constraints and policy boundaries (e.g., organizational policy, compliance rules, safety invariants, subscription/contract limits). It also supplies the permission policy that determines which originators may request which services (R9).

Tool/Service Provider: A system that executes actions (APIs, workflows, actuators, management functions, data services).

Observer (Monitoring Function): A function that collects observations (telemetry, events, measurements) used for assurance, compliance assessment, and drift detection (R6 and R7).

4.2. Operational Overview

This section provides an informative lifecycle overview to contextualize admission control, constraint validation, invocation validation, observation, and drift handling.

1. The Intent Originator expresses an intent via the Intent Client.

2. The Intent Client (or an equivalent admission point) authenticates the origin and applies admission control (R8, R9): it verifies the originator, evaluates the originator’s eligibility to request the targeted service, and obtains consent for high-impact actions before forwarding.

3. The Intent Client submits an admitted intent artifact to the IBS.

4. The IBS performs intent translation (Intent Transformer) to derive constraints, objectives, and candidate directives.

5. The IBS performs constraint validation (R3) in consultation with the Policy/Constraint Authority.

6. The IBS determines one or more tool/service invocations needed for fulfillment.

7. Prior to each invocation, the IBS performs invocation validation (R4), including privilege checks and parameter/constraint checks.

8. The Tool/Service Provider executes the invocation and returns results; side effects may be irreversible.

9. The Observer produces observations used by the IBS for assurance and drift detection (R6).

10. If drift or violations are detected, the IBS applies a policy-driven response (R7), such as deny, degrade, re-confirm, re-negotiate, or fallback.

5. Security Scenarios

This section describes representative security scenarios using a consistent template: Setting, Actors, Assets, Attack Sketch, Impact, and Relevant Requirements. These scenarios are not exhaustive but illustrate key threat patterns in multi-hop intent processing.

5.1. Scenario 1: Directive Tampering and Authorization Boundary Expansion

Setting:
An IBS translates an intent into derived directives (e.g., allowed rules) that traverse multiple intermediaries before reaching an execution endpoint.

Actors:
Intent Originator, Intent Client, IBS, one or more intermediaries (agents/clients), Tool/Service Provider.

Assets:
Authorization boundary, constraints/invariants, protected resources, audit evidence.

Attack Sketch:

1. An intermediary modifies derived directives to add operations or widen resource scope.

2. The modified directives are forwarded to the execution endpoint without effective detection.

3. The endpoint performs out-of-bound operations (e.g., modifying account state, accessing other parties’ data, disabling safety rules).

Impact:
Privilege escalation, policy bypass, unauthorized side effects, compliance violations.

Relevant Requirements:
R1 (Provenance and Authorization Boundary Binding), R2 (Chain-of-Custody for Derived Directives), R3 (Constraint Validation), R5 (Non-Bypass Enforcement), R6 (Observability and Auditability).

5.2. Scenario 2: Spoofed Origin and Non-Consensual Intent Origination

Setting:
A user-facing device (e.g., a terminal) hosts multiple applications and agents. Any of them can express intents that are submitted to a local Intent Client and forwarded to a remote IBS that accepts intent artifacts and may trigger high-impact services or actions. The remote receiver acts on whichever intents it admits.

Actors:
Intent Originator (the user or a legitimate originator), a legitimate local agent, a malicious co-resident application/agent, the Intent Client, and the remote IBS / Tool/Service Provider.

Assets:
Originator identity and account/identity bindings, user consent, the permission policy that determines which originators may request which services, billing/spending limits, and safety constraints.

Attack Sketch:

1. The malicious application fabricates an intent artifact that appears to originate from the user or a legitimate agent (spoofed provenance); or an originator that is not entitled to the targeted service, or is running unattended in the background, issues a high-impact intent.

2. The Intent Client / IBS forwards the intent, and the remote receiver executes high-impact actions (purchases, data disclosure, account or state changes) without verifying the actual originator, the originator’s eligibility for the requested service, the originator’s runtime state, or valid user consent.

Impact:
Unauthorized actions, fraud, privacy leakage, billing abuse, and irreversible side effects.

Relevant Requirements:
R8 (Origin Authentication), R9 (Originator Authorization and Consent-Gated Admission), R3 (Constraint Validation), R6 (Observability and Auditability).

6. Security Considerations

This section provides solution-agnostic security considerations mapped to the scenarios and requirements. Implementations may realize these considerations using different security mechanisms (tokens, signatures, attestation, policy engines, or protocol-level bindings).

6.1. Scenario-to-Requirement Mapping

Table 1 summarizes the primary mappings between the elaborated scenarios and security requirements. Note that these mappings are non-exhaustive; additional requirements may apply depending on deployment context.

Scenario	Primary Threats	Key Requirements
1 (Directive Tampering)	T1, T3	R1, R2, R3, R5, R6
2 (Spoofed/Non-Consensual Origin)	T6, T7	R8, R9, R3, R6

Table 1: Scenario to Requirement Mapping

6.2. Considerations for Scenario 1 (Directive Tampering)

Scenario 1 highlights that derived directives are often more operationally powerful than the original intent text. Therefore, systems should treat derived directives as security-relevant artifacts whose integrity and authorization boundary binding should be protected across hops.

6.2.1. Overview

The core challenge is ensuring that derived directives cannot be tampered with or substituted in transit, and that execution endpoints can verify the authenticity and authorization boundary of received directives.

Binding and Custody (R1, R2): Derived directives should be bound to the intent context and authorization boundary such that unauthorized expansion or substitution is detectable or preventable across hops.

Pre-Execution Constraint Validation (R3): Even if directives appear intact, the receiver should validate that the intended actions remain within constraints and invariants before execution.

Non-Bypass Enforcement (R5): Execution endpoints should enforce checks such that direct calls cannot bypass required validation gates.

Audit Evidence (R6): Systems should produce evidence linking execution decisions to validated directives and constraints.

6.2.2. Illustrative Procedure

The following procedure is informative and solution-agnostic. Implementations may use various mechanisms (e.g., signed tokens, cryptographic binding, attestation) to achieve these objectives.

1. Directive Derivation and Binding: The IBS derives directives from an intent and associates them with the applicable authorization boundary. The IBS generates a cryptographically-protected artifact (e.g., signed token, sealed directive) that binds the directives to the intent context and authorization scope.

2. Integrity Protection for Multi-Hop Propagation: Before forwarding directives across trust boundaries, the system attaches integrity and binding evidence (e.g., digital signature, MAC, or attestation token) sufficient for downstream verification. This evidence includes the authorization boundary, constraint set, and issuer identity.

3. Reception and Verification: Upon receipt, the execution-side gate verifies the integrity and binding evidence of the received directives. This verification confirms: (a) the directives have not been tampered with or substituted, (b) the directives originate from an authorized IBS, and (c) the authorization boundary matches expected scope.

4. Constraint Re-Validation: The execution endpoint re-validates the directives against local constraints, invariants, and policy rules. This step provides defense-in-depth even if upstream validation was bypassed or compromised.

5. Enforcement and Audit: If verification or validation fails, the system denies or degrades execution under policy and records audit evidence. If successful, the system proceeds with execution and logs the binding evidence, execution decision, and outcomes for compliance assessment.

This procedure addresses T1 (Directive Tampering/Substitution) and T3 (Constraint Bypass) by establishing end-to-end integrity and validation across multi-hop processing.

6.3. Considerations for Scenario 2 (Spoofed/Non-Consensual Origin)

Scenario 2 highlights that an intent must be controlled at the point where it enters the system, not only while it is in transit. The receiver (or the admission point on the originating device) should be able to distinguish authorized, consented originators from unauthorized or spoofed ones, and to gate high-impact actions on explicit consent.

6.3.1. Overview

Origin Authentication (R8): The intent artifact should provide a verifiable binding to its actual originator’s identity, so that a fabricated or spoofed origin can be detected before admission.

Originator-Level Authorization (R9): Admission should evaluate originator attributes (e.g., identity, type, runtime state, history) against a permission policy that specifies which originators may request which services. Intents from disallowed originators, or from originators in a disallowed state (e.g., background/unattended), should be rejected or escalated.

Consent for High-Impact Actions (R9): For high-impact or irreversible actions, the system should obtain and verify explicit user consent (e.g., step-up confirmation). Absence or refusal of consent should result in rejection.

Audit Evidence (R6): Systems should record the verified origin, the admission decision, and the consent outcome to support investigation of fraudulent or non-consensual origination.

6.3.2. Illustrative Procedure

The following procedure is informative and solution-agnostic. Implementations may use various mechanisms (e.g., signed origin assertions, platform attestation of the originating application, policy engines) to achieve these objectives.

1. Origin Binding: The intent artifact carries a verifiable binding to the actual originator’s identity/credential (rather than a self-asserted, unverifiable label).

2. Origin Verification: The admission point verifies the origin binding and rejects intents whose provenance is forged or cannot be verified.

3. Originator Authorization: The admission point evaluates the originator’s attributes (identity, type, runtime state, history) against the applicable permission policy to determine whether the originator is eligible to request the targeted service.

4. Consent Gate: For high-impact or irreversible actions, the system obtains and verifies explicit user consent and denies the request on absence or refusal.

5. Enforcement and Audit: If origin verification, authorization, or consent fails, the system rejects or escalates the intent under policy and records audit evidence. If all checks pass, the intent is admitted and forwarded, and the decision is logged.

This procedure addresses T6 (Origin Spoofing/Forged Provenance) and T7 (Unauthorized or Non-Consensual Origination) by establishing authenticated, policy-gated, and consent-gated admission before an intent enters the processing chain.

6.4. General Security Considerations

Beyond the scenario-specific considerations, the following general principles apply to intent-based systems:

* Trust Boundary Awareness: systems explicitly identify trust boundaries and apply appropriate security controls at each boundary crossing.

* Defense in Depth: validation occur at multiple layers (admission, translation, propagation, invocation, execution) to provide resilience against bypass or compromise of individual layers.

* Least Privilege: derived directives and invocations are scoped to the minimum privileges necessary for intent fulfillment.

* Fail-Safe Defaults: when validation fails or drift is detected, systems default to denying actions rather than permitting potentially unsafe operations.

* Auditability: all security-relevant decisions and events are logged with sufficient context to support post-incident investigation and compliance assessment.

7. IANA Considerations

This document has no IANA actions.

8. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC7991] Hoffman, P., "The "xml2rfc" Version 3 Vocabulary", RFC 7991, DOI 10.17487/RFC7991, December 2016, <https://www.rfc-editor.org/info/rfc7991>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

9. Informative References

[I-D.goswami-agentic-jwt]
Goswami, A., "Secure Intent Protocol: JWT Compatible Agentic Identity and Workflow Management", Work in Progress, Internet-Draft, draft-goswami-agentic-jwt-00, 1 January 2026, <https://datatracker.ietf.org/doc/html/draft-goswami-agentic-jwt-00>.

[I-D.ietf-oauth-transaction-tokens]
Tulshibagwale, A., Fletcher, G., and P. Kasselman, "Transaction Tokens", Work in Progress, Internet-Draft, draft-ietf-oauth-transaction-tokens-08, 2 March 2026, <https://datatracker.ietf.org/doc/html/draft-ietf-oauth-transaction-tokens-08>.

[I-D.irtf-nmrg-ibn-usecases]
Yao, K., Chen, D., Jeong, J. P., Wu, Q., Yang, C., Contreras, L. M., and G. Fioccola, "Use Cases and Practices for Intent-Based Networking", Work in Progress, Internet-Draft, draft-irtf-nmrg-ibn-usecases-03, 15 March 2026, <https://datatracker.ietf.org/doc/html/draft-irtf-nmrg-ibn-usecases-03>.

[I-D.liu-oauth-a2a-profile]
Liu, P. C. and N. Yuan, "Agent-to-Agent (A2A) Profile for OAuth Transaction Tokens", Work in Progress, Internet-Draft, draft-liu-oauth-a2a-profile-00, 20 October 2025, <https://datatracker.ietf.org/doc/html/draft-liu-oauth-a2a-profile-00>.

[I-D.ni-a2a-ai-agent-security-requirements]
Yuan, N., Liu, P. C., Gao, Q., and Z. Li, "Security Requirements for AI Agents", Work in Progress, Internet-Draft, draft-ni-a2a-ai-agent-security-requirements-01, 28 February 2026, <https://datatracker.ietf.org/doc/html/draft-ni-a2a-ai-agent-security-requirements-01>.

[I-D.oauth-transaction-tokens-for-agents]
Raut, A., "Transaction Tokens For Agents", Work in Progress, Internet-Draft, draft-oauth-transaction-tokens-for-agents-06, 11 April 2026, <https://datatracker.ietf.org/doc/html/draft-oauth-transaction-tokens-for-agents-06>.

[MS-AF-SEQ]
Microsoft, "Microsoft Agent Framework Workflows Orchestrations - Sequential", 2026, <https://learn.microsoft.com/en-us/agent-framework/workflows/orchestrations/sequential>.

[RFC9315] Clemm, A., Ciavaglia, L., Granville, L. Z., and J. Tantsura, "Intent-Based Networking - Concepts and Definitions", RFC 9315, DOI 10.17487/RFC9315, October 2022, <https://www.rfc-editor.org/info/rfc9315>.

[RFC9316] Li, C., Havel, O., Olariu, A., Martinez-Julia, P., Nobre, J., and D. Lopez, "Intent Classification", RFC 9316, DOI 10.17487/RFC9316, October 2022, <https://www.rfc-editor.org/info/rfc9316>.

Acknowledgments

TODO

Authors’ Addresses

Yuning Jiang
Huawei
Email: jiangyuning2@h-partners.com

Lun Li
Huawei
Email: lilun20@huawei.com

Yurong Song
Huawei
Email: songyurong1@huawei.com

Faye Liu
Huawei
Email: liufei19@huawei.com