Independent Submission
Internet-Draft
Intended status: Informational
Expires: 1 November 2026

C. Hood
Nomotic, Inc.

30 April 2026

AGTP Session Protocol
draft-hood-agtp-session-00

Abstract

This document specifies the AGTP Session Protocol (AGTP-SESSION), a companion to [AGTP] that defines session semantics for agent-to-agent and agent-to-API communication. AGTP-SESSION addresses two distinct session models: bounded sessions for time-limited transactional flows, and persistent sessions for long-lived agent contexts. Both inherit identity, authority, and attribution from base AGTP. This is an early working draft; many design decisions are deliberately open.

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 1 November 2026.

Table of Contents

1. Introduction

Base AGTP defines request-and-response semantics for individual method invocations between agents. Many real-world agent interactions span multiple invocations under shared context: a chatbot conversation that continues across hours, an agentic commerce flow that completes a transaction across a sequence of methods, a research agent maintaining working state across a long-running task.

These interactions need session semantics that base AGTP does not specify. AGTP-SESSION defines those semantics as a companion protocol that builds on base AGTP without modifying it.

The architectural commitment is straightforward: a session in AGTP is an agent-level abstraction. It is not a TLS connection, a QUIC connection, or a transport-layer session. It is a logical context shared between two or more agents (or between an agent and an API), within which multiple method invocations carry shared identity, authority scope, and attribution chain.

A session may map to a single underlying transport connection, may span multiple connections over time, or may persist while the underlying transport is intermittently disconnected and reestablished. The session state is the agent-level fact; the connection is the mechanism that carries traffic when active.

2. Terminology

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.

Session: A logical context shared between AGTP participants under which multiple method invocations are correlated, identity and authority remain consistent, and shared state evolves across the lifetime of the session. Sessions are agent-level abstractions, not transport- level constructs.

Bounded Session: A session with an explicit termination condition: a time limit, an action completion, or a transaction outcome. Bounded sessions are appropriate for commerce flows, multi-step delegations with finite scope, and any agent interaction with a defined end state.

Persistent Session: A long-lived session intended to span hours, days, or longer. State is maintained across reconnections. Persistent sessions are appropriate for chatbot interactions, research agents, monitoring agents, and other contexts where the agent retains continuity.

Session Identifier: A unique identifier assigned at session establishment that is carried in every method invocation within the session. Format and scope of session identifiers are TBD.

Session Context: The shared state associated with a session, including (but not limited to) participant Agent-IDs, Authority-Scope grants applicable to the session, accumulated Attribution-Records for invocations within the session, and any session-specific governance metadata.

3. Session Models

AGTP-SESSION defines two session models. Implementations MAY support one or both depending on deployment profile.

3.1. Bounded Sessions

Bounded sessions exist for a finite, often short period and complete when one of the following conditions is met:

* An explicit time limit (session_ttl) expires
* A defined transaction or workflow completes
* A participant explicitly terminates the session
* An authority constraint is violated (e.g., scope limit reached)

Bounded sessions match the semantics of agentic transactions where the session has a clear lifecycle: an agent initiates a purchase flow, exchanges several methods with a merchant, completes the transaction, and the session closes. The session boundary is meaningful to governance and attribution: all method invocations within a bounded session are correlated under a single transactional context.

3.2. Persistent Sessions

Persistent sessions exist for extended periods and are designed to survive transport disconnection, network instability, and platform restarts. The session identifier remains stable; the underlying transport may reconnect multiple times during the session lifetime.

Persistent sessions match the semantics of long-lived agent contexts: a chatbot maintaining conversation continuity, a research agent working on a multi-day task, a monitoring agent maintaining persistent attention to a system. Authority-Scope grants applicable to the session may be re-validated periodically; identity verification follows base AGTP semantics on reconnection.

4. Session Establishment

A session is established explicitly via an ESTABLISH method invoked between two AGTP participants. Implicit establishment via an extended- context flag on a standard method invocation is noted as an alternative under consideration but is *NOT RECOMMENDED* as the primary mechanism. Explicit establishment provides:

* Unambiguous negotiation of session model (bounded or persistent), session lifetime, and any session-scoped Authority-Scope grants

* A clear root for the session attribution chain, with the ESTABLISH invocation serving as the first signed Attribution-Record under the session

* Clean error semantics when negotiation fails (e.g., 451 Scope Violation if session-scoped grants exceed the establishing agent's Authority-Scope, 4xx for unsupported session models, 5xx for governance platform unavailability)

* Predictable enforcement boundaries for governance platforms and Scope-Enforcement Points, which can recognize session establishment as a distinct event rather than inferring it from method context

The full wire format and parameter set for ESTABLISH is not specified in this version. At minimum, the method *MUST* carry:

* The proposed session model (bounded or persistent)

* A proposed session lifetime (session_ttl) for bounded sessions

* Any session-scoped Authority-Scope grants the establishing agent proposes for the session context

* Standard AGTP request fields (Agent-ID, Principal-ID, Authority-Scope, signature)

A successful ESTABLISH response *MUST* carry:

* A session identifier (format TBD; see Open Questions)

* The accepted session model and lifetime

* The accepted Authority-Scope grants for the session

* A governance platform signature binding the session identifier to the participating Agent-IDs

Rejection of an ESTABLISH request follows base AGTP status code semantics. Implementations *MUST NOT* establish a session by side effect of any other method invocation; the absence of a successful ESTABLISH response means no session exists.

5. Inheritance from Base AGTP

AGTP-SESSION inherits all base AGTP properties:

* Agent-only at the wire level. Sessions exist between agents, not between general-purpose actors.

* Agent-ID and Principal-ID present on every method invocation, including invocations within a session.

* Authority-Scope enforced on every method invocation. Scopes are not relaxed inside sessions.

* Attribution Records signed for every invocation, including those within sessions, with the session identifier included in the attribution metadata.

* Status codes, including 451 Scope Violation and 551 Authority Chain Broken, apply within sessions exactly as outside them.

* Three-level verification model. Sessions do not weaken verification.

AGTP-SESSION adds session-specific semantics on top of these properties; it does not replace or modify them.

6. Transport Considerations

Base AGTP runs over TCP with TLS 1.3 or over QUIC. AGTP-SESSION inherits these transport options.

QUIC is a particularly natural fit for persistent sessions because of its connection migration capability [RFC9000], which allows a logical connection to survive IP address changes and network path changes without renegotiation. AGTP-SESSION over QUIC can leverage QUIC's connection migration to maintain a persistent session across network transitions transparently.

7. Relationship to MOQT and Real-Time Media

The proposed AI Agent Communication Protocols (acp) charter under discussion at IETF anticipates a session protocol on webtransport or MOQ as a foundational building block for agent communication. That work and AGTP-SESSION reflect inverse architectural commitments.

The acp approach treats session and transport semantics as the foundation, with agent-specific concerns (identity, authority, attribution) layered on top by the protocols that compose on the session. Under this model, the session protocol does not know that its traffic is agent traffic; agent semantics are an application-layer concern.

AGTP-SESSION reflects the inverse: agent-specific concerns are the foundation (in base AGTP), and session semantics layer on top of an agent-aware substrate. Under this model, a session is, by definition, an agent session. The protocol carries identity, authority, and attribution at the wire level for every method invocation within the session, because the substrate beneath the session already commits to carrying agent traffic structurally.

The two architectures are not mutually exclusive in deployment. MOQT-style relay-based pub/sub patterns and real-time media streaming address use cases (particularly voice with sub-100ms barge-time interruption) that AGTP-SESSION does not currently address. A future companion draft (or a later version of AGTP-SESSION) may define how AGTP composes with MOQT-style transport for real-time agent media, or how MOQT-style sessions can carry AGTP semantic envelopes.

The architectural choice between these two foundations is a question for community discussion. Both deliver session semantics; they differ on which layer commits to knowing the traffic is agent traffic.

8. Application Layering: Voice and Multi-Modal Agents

AGTP-SESSION focuses on session semantics for agent communication. It does not specify protocols for real-time voice, video, or multi-modal data exchange.

Voice agents, video agents, and multi-modal agents are application-layer constructs that can compose on top of AGTP and AGTP-SESSION. They are not different kinds of agents at the protocol layer; they are agents that produce or consume specific media types.

This raises a related architectural question worth surfacing for community discussion: AGTP may benefit from an agent type system, analogous to HTTP Content-Type, that allows the protocol to identify the kind of agent or kind of agent payload involved in a given exchange. This is not part of AGTP-SESSION as currently scoped but is flagged here as adjacent work that may warrant its own companion specification.

9. Out of Scope for v00

The following are deliberately out of scope for this version:

* Real-time voice and barge-time interruption semantics

* Pub/sub and relay-based session patterns (MOQT-style multicast)

* Session migration across organizations or governance platforms

* Session state synchronization across multiple agents in a chain

* Detailed protocol specification for ESTABLISH/OPEN methods

* Session encryption beyond what TLS 1.3 or QUIC already provides

* Wire format for session metadata

These are flagged as future work or as open questions to be resolved through community input.

10. Security Considerations

Base AGTP security considerations apply to every method invocation within a session, including invocations occurring under bounded or persistent session models. This section identifies session-specific risks and mitigations that supplement the base AGTP threat model.

10.1. Session Identifier Theft and Replay

Threat: An attacker who obtains a session identifier may attempt to inject method invocations into an active session by replaying or forging requests carrying the identifier.

Mitigation: Session identifiers *MUST* be bound at establishment time to the participating Agent-IDs and to a transport channel binding (TLS exporter or QUIC connection ID, depending on transport). Session identifiers in isolation *MUST NOT* be sufficient to authorize method invocations. Every invocation within a session *MUST* continue to satisfy base AGTP identity verification (per [AGTP] Section 7), and the receiving party *MUST* reject invocations whose identity cannot be reverified against the participants bound to the session at establishment.

10.2. Persistent Session Compromise

Threat: A persistent session that survives transport disconnection and reconnection has a longer attack window than a bounded session. An attacker who compromises a participant’s signing key during the session lifetime can impersonate the participant for the remainder of the session.

Mitigation: Persistent sessions *SHOULD* require periodic Authority-Scope re-validation against the establishing governance platform. Recommended re-validation interval is implementation- defined but *SHOULD NOT* exceed the lifetime of the underlying Authority-Scope

grants. Persistent sessions *MUST* support revocation propagation: if a participant's Agent Genesis or Agent Certificate is revoked during the session, the session *MUST* terminate with appropriate status code on the next invocation. Implementations *SHOULD* subscribe to revocation notifications via mechanisms defined in [AGTP-CERT] or [AGTP-LOG].

10.3. Cross-Session Correlation

Threat: An adversary observing multiple sessions from the same agent may be able to correlate sessions to construct a profile of agent behavior across contexts, even if individual session contents are encrypted.

Mitigation: Session identifiers *MUST NOT* be derived from participant Agent-IDs in a way that enables cross-session linkage. Session identifiers *SHOULD* be cryptographically random and unique per session. Implementations supporting privacy-sensitive deployments *SHOULD* consider session identifier rotation policies that limit linkability, with the trade-off that rotation complicates revocation propagation.

10.4. Authority Constraint Violation Mid-Session

Threat: An agent participating in an established session may attempt to invoke methods that exceed session-scoped Authority-Scope grants, either through scope drift across multiple invocations or through deliberate scope expansion attempts.

Mitigation: Authority-Scope is enforced on every invocation regardless of session context. A scope violation *MUST* trigger 451 Scope Violation per base AGTP. Receiving parties *MAY* elect to terminate the session on first scope violation rather than continuing to accept further invocations. Termination on scope violation is *RECOMMENDED* for bounded sessions where the violation suggests the session boundary has been compromised.

10.5. QUIC 0-RTT Resumption Considerations

Threat: When AGTP-SESSION runs over QUIC, 0-RTT connection resumption allows initial method invocations on a resumed connection to be processed before full handshake completion. Replay of 0-RTT data is a known QUIC consideration.

Mitigation: Method invocations transmitted in 0-RTT data *MUST NOT* be permitted to ESTABLISH a new session. Session establishment *MUST* occur after the QUIC handshake has completed. Invocations within an existing session *MAY* use 0-RTT but *MUST* be idempotent

or otherwise replay-safe at the application level. Implementations *SHOULD* consult QUIC 0-RTT replay guidance ([RFC9001] Section 9.2) when configuring session resumption behavior.

10.6. Graceful Termination and Revocation

Threat: A session may need to be terminated mid-flow due to revocation, scope violation, time expiry, or governance policy changes. Improper termination handling can leave participants in inconsistent state.

Mitigation: Implementations *MUST* define termination semantics that produce a final Attribution-Record signed by the terminating party identifying the termination reason. Mid-session revocation of either participant's Agent Genesis or Agent Certificate *MUST* result in session termination. Time-based expiry of bounded sessions *MUST* be enforced based on the establishing governance platform's authoritative time, not on participant-local clocks.

11. IANA Considerations

TBD.

12. References

12.1. Normative References

[AGTP] Hood, C., "Agent Transfer Protocol (AGTP)", Work in Progress, Internet-Draft, draft-hood-independent-agtp-06, 2026, <https://datatracker.ietf.org/doc/html/draft-hood-independent-agtp-06>.

[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/rfc/rfc2119>.

[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/rfc/rfc8174>.

12.2. Informative References

[AGTP-CERT] Hood, C., "AGTP Agent Certificate Extension", Work in Progress, Internet-Draft, draft-hood-agtp-agent-cert-00, 2026, <https://datatracker.ietf.org/doc/html/draft-hood-agtp-agent-cert-00>.

[AGTP-LOG] Hood, C., "AGTP Transparency Log Protocol", Work in Progress, Internet-Draft, draft-hood-agtp-log-00, 2026, <https://datatracker.ietf.org/doc/html/draft-hood-agtp-log-00>.

[RFC9000] Iyengar, J., Ed. and M. Thomson, Ed., "QUIC: A UDP-Based Multiplexed and Secure Transport", RFC 9000, DOI 10.17487/RFC9000, May 2021, <https://www.rfc-editor.org/rfc/rfc9000>.

[RFC9001] Thomson, M., Ed. and S. Turner, Ed., "Using TLS to Secure QUIC", RFC 9001, DOI 10.17487/RFC9001, May 2021, <https://www.rfc-editor.org/rfc/rfc9001>.

Appendix A. Open Questions

* Should sessions be established explicitly via an ESTABLISH method, or implicitly via an extended-context flag on a standard method?

* What is the canonical format and scope for session identifiers? (256-bit similar to Agent-ID, or shorter for performance?)

* How does session attribution interact with the base AGTP Attribution-Record? Are session-level attribution rollups defined?

* How are persistent sessions migrated across governance platform changes (e.g., merchant changes governance providers mid-session)?

* Should AGTP define an agent type system (Agent Content-Type) separately, or as part of session establishment?

* What is the right composition story with MOQT for real-time agent media use cases?

* Should multi-party sessions (more than two participants) be in scope for this draft, or deferred to a separate companion?

Appendix B. Contributors

This draft is informed by the IETF agent2agent charter formation discussion and by feedback on the architectural layering question raised in that thread.

Author’s Address

Chris Hood
Nomotic, Inc.

Email: chris@nomotic.ai
URI: https://nomotic.ai