Independent Submission
Internet-Draft
Intended status: Informational
Expires: 24 September 2026

C. Hood
Nomotic, Inc.

23 March 2026

AGTP Composition with Agent Group Messaging Protocols
draft-hood-agtp-composition-00

Abstract

Agent Group Messaging Protocols (AGMPs) -- including the Model Context Protocol (MCP), the Agent-to-Agent Protocol (A2A), and the Agent Communication Protocol (ACP) -- define what AI agents say to each other. The Agent Transfer Protocol (AGTP) defines how those messages move across a network. This document specifies how AGMP messages are carried over AGTP as a transport substrate, providing normative mapping rules for identity, authority scope, delegation, and session fields between the AGMP and AGTP layers. AGTP headers take precedence over equivalent AGMP payload fields for all infrastructure-level routing, enforcement, and audit operations. Agents gain transport-level governance, observability, and identity without modification to the AGMP layer.

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 24 September 2026.

Table of Contents

1. Introduction

1.1. The Layering Problem

AI agent systems in 2026 are characterized by a proliferation of messaging protocols -- MCP [MCP], A2A [A2A], ACP [ACP], ANP [ANP] -- each defining rich semantics for agent-to-agent communication, tool invocation, task delegation, and capability advertisement. These protocols are well-designed for what they address: the content and structure of agent communication.

They share a common limitation: they all run over HTTP and inherit its constraints for agent traffic. At the infrastructure layer, an A2A Task message is indistinguishable from a human-initiated POST request. An MCP tool call carries no protocol-level signal of which agent sent it, under what authority, or within what resource budget. Governance, observability, and identity enforcement require application-layer instrumentation that is expensive, inconsistent, and invisible to routing infrastructure.

AGTP [AGTP] addresses this at the transport layer. When AGMP messages are carried over AGTP, every request carries the sending agent's verified identity, principal authorization, authority scope, delegation lineage, and resource budget in protocol headers. Infrastructure components -- load balancers, gateways, SEPs -- can route, filter, and enforce governance without parsing message payloads.

1.2. Relationship to AGTP Core

This document is a companion to [AGTP] and is normative for implementations that carry AGMP messages over AGTP. The core AGTP specification defines the protocol; this document defines how specific AGMP concepts map to AGTP constructs. Implementations that use AGTP as a transport for AGMP traffic *MUST* follow the mapping rules in this document.

1.3. Scope

This document covers composition profiles for MCP [MCP], A2A [A2A], and ACP [ACP]. It does not modify any AGMP specification. AGMPs retain their own schemas, semantics, and development paths. This document only specifies how AGMP messages are wrapped in AGTP requests and how AGMP identity and delegation fields relate to AGTP headers.

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.

AGMP (Agent Group Messaging Protocol): The collective term for higher-layer AI agent messaging standards that operate over AGTP as their transport substrate, including MCP, A2A, ACP, and ANP. AGMPs define what agents say. AGTP defines how those messages move.

Substrate: The transport layer on which an AGMP operates. In this document, AGTP is the substrate for all AGMPs in scope. The substrate is not visible to the AGMP layer; AGMPs require no modification to run over AGTP.

Composition Profile: The normative set of mapping rules specifying how a specific AGMP’s identity, delegation, session, and capability concepts correspond to AGTP header fields and method semantics.

3. The Narrow-Waist Architecture

AGTP functions as the narrow-waist layer of the AI agent protocol stack, analogous to IP in the internet architecture. IP does not understand TCP or UDP; it carries their packets. AGTP does not understand MCP or A2A; it carries their messages. The narrow-waist property means any AGMP can run over AGTP, and any AGTP-aware infrastructure can observe and govern any AGMP traffic, without coupling between the layers.

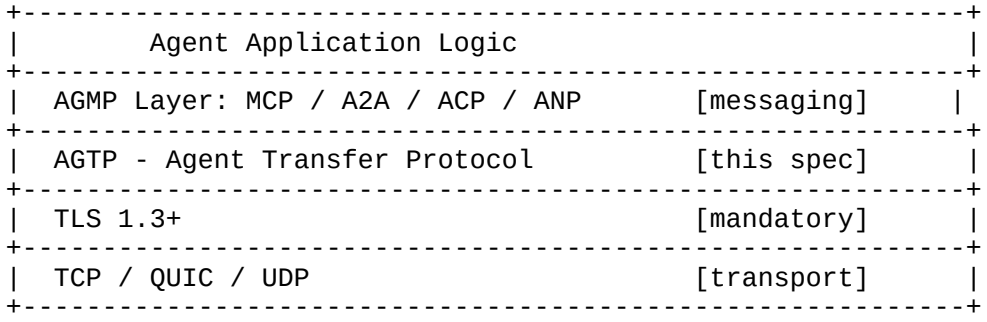

Figure 1: AGTP as Narrow-Waist Layer

The composition rules in this document specify how AGMP concepts map to the AGTP layer below them. The AGMP layer is unaware of AGTP internals; the AGTP layer carries AGMP payloads without interpreting their content.

4. Precedence Rule

This rule applies universally across all AGMP composition profiles defined in this document.

AGTP headers (Agent-ID, Principal-ID, Authority-Scope, Delegation-Chain, Session-ID, Task-ID, Budget-Limit, and AGTP-Zone-ID) take precedence over equivalent fields in the AGMP message payload for all infrastructure-level operations, including routing, enforcement, rate limiting, audit, and attribution.

Specifically:

* Infrastructure components including SEPs, governance gateways, load balancers, and monitoring systems *MUST* use AGTP header values for all protocol-level decisions. They *MUST NOT* parse AGMP payload fields to make routing or enforcement decisions.

* AGMP payload fields *MAY* carry equivalent identity or delegation information for application-layer use by the receiving agent. These fields are not authoritative for infrastructure decisions.

* In the event of a conflict between an AGTP header value and an AGMP payload field carrying equivalent information, the AGTP header value *MUST* be treated as authoritative.

* An AGTP request carrying an AGMP payload *MUST* be rejected by infrastructure if the AGTP headers are absent or invalid, regardless of whether the AGMP payload itself is well-formed.

This rule enables infrastructure to govern agent traffic uniformly regardless of which AGMP is in use, without requiring AGMP-specific parsing.

5. A2A Composition Profile

5.1. Overview

The Agent-to-Agent Protocol (A2A) [A2A] defines semantics for agent-to-agent task delegation, capability advertisement, artifact exchange, and provenance tracking. A2A runs over HTTP in its base specification. When running over AGTP, A2A messages are carried in AGTP request and response bodies, with A2A task, capability, and identity concepts mapped to AGTP headers.

5.2. A2A Concept Mapping

A2A Concept	A2A Field	AGTP Mapping
Sending agent identity	agent.id	Agent-ID header
Accountable principal	agent.owner	Principal-ID header
Agent capabilities	Agent Card	AGTP DESCRIBE response
Agent identity document	Agent Card	AGTP Agent Manifest Document
Task delegation	Task object	AGTP DELEGATE method body
Task identifier	task.id	Task-ID header
Delegation provenance	Task lineage	Delegation-Chain header
Session context	Conversation ID	Session-ID header
Artifact delivery	Artifact object	AGTP NOTIFY body
Capability scope	Agent Card capabilities	Authority-Scope header

Table 1: A2A-to-AGTP Field Mapping

5.3. Mapping Rules

5.3.1. Agent Identity

The A2A agent.id field *MUST* be used to populate the AGTP Agent-ID header when the sending agent's A2A identity is known at the transport layer. If the sending agent has a canonical AGTP Agent-ID (agtp:// URI form), that value *MUST* be used in the Agent-ID header; the A2A agent.id *MAY* appear in the payload for application-layer use.

The A2A agent's declared owner or registrant *MUST* be used to populate the AGTP Principal-ID header. If no owner is declared, the Principal-ID *MUST* reflect the organizational identity of the deploying party.

5.3.2. Task Delegation

An A2A Task sent from one agent to another *MUST* be carried as an AGTP DELEGATE method invocation. The AGTP DELEGATE parameters *MUST* include:

* target_agent_id: the canonical AGTP Agent-ID of the receiving agent

* authority_scope: the subset of the sending agent's Authority-Scope applicable to this task; *MUST NOT* exceed the sending agent's declared scope

* delegation_token: a signed token from the governance platform authorizing this delegation

* task: the A2A Task object as the AGTP body payload

The A2A task.id *MUST* be used as the AGTP Task-ID header value, or if no A2A task ID is present, a new Task-ID *MUST* be generated by the AGTP layer.

5.3.3. Delegation Provenance

A2A's task provenance chain *MUST* be reflected in the AGTP Delegation-Chain header. Each agent in the A2A provenance chain *MUST* appear as a canonical AGTP Agent-ID in the Delegation-Chain header, in origination-to-current order.

5.3.4. Capability Advertisement

When an agent exposes an A2A Agent Card, the same capability information *SHOULD* be exposed via the AGTP DESCRIBE method. An AGTP DESCRIBE request to the agent's canonical URI (agtp://[domain]/agents/[label]) *MUST* return a Capability Document that reflects the agent's A2A capabilities translated into AGTP capability domain format.

5.3.5. Artifact Delivery

A2A Artifact objects *MUST* be carried as AGTP NOTIFY method invocations when delivered asynchronously. The NOTIFY body *MUST* include the A2A Artifact object and a content_type field with value a2a-artifact.

5.4. Wire Example: A2A Task over AGTP

```
AGTP/1.0 DELEGATE
Agent-ID: agtp://agtp.acme.tld/agents/orchestrator
Principal-ID: usr-chris-hood
Authority-Scope: agents:delegate documents:query
Delegation-Chain: agtp://agtp.acme.tld/agents/orchestrator
Session-ID: sess-a1b2c3d4
Task-ID: task-0099
Budget-Limit: tokens=50000 ttl=300
Content-Schema: https://a2aprotocol.ai/schema/task/v1
Content-Type: application/agtp+json

{
  "method": "DELEGATE",
  "task_id": "task-0099",
  "parameters": {
    "target_agent_id": "agtp://agtp.acme.tld/agents/analyst",
    "authority_scope": "documents:query",
    "delegation_token": "[signed token]",
    "task": {
      "a2a_task_id": "a2a-task-7f3a",
      "message": {
        "role": "user",
        "parts": [{"type": "text",
          "text": "Summarize Q1 financial reports"}]
      },
      "artifacts": []
    }
  }
}
```

```
AGTP/1.0 200 OK
Task-ID: task-0099
Server-Agent-ID: agtp://agtp.acme.tld/agents/analyst
Attribution-Record: [signed attribution token]
Cost-Estimate: tokens=12450
Content-Type: application/agtp+json

{
  "status": 200,
  "task_id": "task-0099",
  "result": {
    "a2a_task_id": "a2a-task-7f3a",
    "status": "completed",
    "artifacts": [
      {
        "name": "Q1 Summary",
        "mimeType": "text/plain",
        "parts": [{"type": "text", "text": "..."}]
      }
    ]
  }
}
```

6. MCP Composition Profile

6.1. Overview

The Model Context Protocol (MCP) [MCP] defines structured communication between AI models and tools, resources, and context providers. MCP distinguishes between clients (AI models) and servers (tool/resource providers). When running over AGTP, MCP messages are carried in AGTP request and response bodies, with MCP session, context, and tool-call semantics mapped to AGTP constructs.

6.2. MCP Concept Mapping

MCP Concept	MCP Field	AGTP Mapping
Client identity	client_id	Agent-ID header
Server identity	server_id	Server-Agent-ID response header
Session context	context	Session-ID header + LEARN method
Conversation	Message	AGTP Session

state	history	persistent state
Tool call	tools/call	AGTP QUERY method body
Resource request	resources/ read	AGTP QUERY with scope: documents:*
Sampling request	sampling/ createMessage	AGTP QUERY with modality: inference
Prompt	prompts/get	AGTP QUERY with scope: knowledge:query
Capability list	capabilities	AGTP DESCRIBE response
Tool result	Tool response	AGTP QUERY response body

Table 2: MCP-to-AGTP Field Mapping

6.3. Mapping Rules

6.3.1. Client and Server Identity

The MCP client *MUST* be identified in the AGTP Agent-ID header using its canonical AGTP Agent-ID. The MCP client_id field *MAY* appear in the payload for application-layer use.

When the MCP server has a canonical AGTP Agent-ID, it *MUST* return that value in the Server-Agent-ID response header. The MCP server's capability list *SHOULD* be accessible via AGTP DESCRIBE at the server's canonical AGTP URI.

6.3.2. Session and Context Management

MCP's context window (the accumulated message history and state for a conversation) *MUST* be associated with an AGTP Session-ID. The AGTP session persists across multiple MCP message exchanges within a single conversation.

Context that should persist beyond the current AGTP session *SHOULD* be written using the AGTP LEARN method with scope principal or global as appropriate.

6.3.3. Tool Calls

An MCP tools/call operation *MUST* be carried as an AGTP QUERY method invocation. The AGTP QUERY intent parameter *MUST* describe the tool being called. The modality parameter *SHOULD* be set to tool to distinguish tool calls from information retrieval queries.

6.3.4. Resource Access

An MCP resources/read operation *MUST* be carried as an AGTP QUERY method invocation. The AGTP Authority-Scope header *MUST* include the appropriate resource scope (e.g., documents:query for document resources).

6.3.5. Sampling

An MCP sampling/createMessage operation *MUST* be carried as an AGTP QUERY method invocation with modality: inference. This signals to AGTP infrastructure that the request involves LLM inference, enabling differentiated routing and budget enforcement.

6.4. Wire Example: MCP Tool Call over AGTP

```
AGTP/1.0 QUERY
Agent-ID: agtp://agtp.acme.tld/agents/assistant
Principal-ID: usr-chris-hood
Authority-Scope: documents:query knowledge:query
Session-ID: sess-mcp-b2c3d4
Task-ID: task-0100
Budget-Limit: tokens=5000 ttl=60
Content-Schema: https://modelcontextprotocol.io/schema/tool-call/v1
Content-Type: application/agtp+json

{
  "method": "QUERY",
  "task_id": "task-0100",
  "parameters": {
    "intent": "Search for recent IETF agent protocol drafts",
    "modality": "tool",
    "mcp_tool_name": "web_search",
    "mcp_tool_input": {
      "query": "IETF agent protocol drafts 2026"
    },
    "format": "structured",
    "confidence_threshold": 0.75
  }
}

AGTP/1.0 200 OK
Task-ID: task-0100
Server-Agent-ID: agtp://agtp.acme.tld/agents/search-tool
Attribution-Record: [signed attribution token]
Cost-Estimate: calls=1
Content-Type: application/agtp+json

{
  "status": 200,
  "task_id": "task-0100",
  "result": {
    "mcp_tool_name": "web_search",
    "content": [
      {
        "type": "text",
        "text": "draft-hood-independent-agtp-02 ..."
      }
    ]
  }
}
```

6.5. Wire Example: MCP Sampling over AGTP

```
AGTP/1.0 QUERY
Agent-ID: agtp://agtp.acme.tld/agents/assistant
Principal-ID: usr-chris-hood
Authority-Scope: knowledge:query
Session-ID: sess-mcp-b2c3d4
Task-ID: task-0101
Budget-Limit: tokens=10000 ttl=120
Content-Type: application/agtp+json

{
  "method": "QUERY",
  "task_id": "task-0101",
  "parameters": {
    "intent": "Generate a summary of the search results",
    "modality": "inference",
    "mcp_messages": [
      {
        "role": "user",
        "content": {
          "type": "text",
          "text": "Summarize these IETF drafts for me."
        }
      }
    ],
    "mcp_model_preferences": {
      "hints": [{"name": "claude"}],
      "maxTokens": 1000
    }
  }
}
```

7. ACP Composition Profile

7.1. Overview

The Agent Communication Protocol (ACP) [ACP] defines messaging semantics for agent-to-agent communication, including message routing, broadcast, and capability discovery. When running over AGTP, ACP messages are carried in AGTP request and response bodies.

7.2. ACP Concept Mapping

ACP Concept	ACP Field	AGTP Mapping
Sender identity	Sender agent ID	Agent-ID header
Recipient	Recipient agent ID	AGTP NOTIFY

		recipient parameter
Broadcast	Multi-recipient message	AGTP NOTIFY with broadcast group
Capability discovery	Capability query	AGTP DESCRIBE method
Message	Message object	AGTP NOTIFY or COLLABORATE body
Multi-agent task	Coordinated task	AGTP COLLABORATE method

Table 3: ACP-to-AGTP Field Mapping

7.3. Mapping Rules

7.3.1. Unicast Messages

ACP unicast messages sent from one agent to a specific recipient *MUST* be carried as AGTP NOTIFY method invocations. The NOTIFY recipient parameter *MUST* carry the canonical AGTP Agent-ID of the intended recipient.

7.3.2. Broadcast Messages

ACP broadcast messages *MUST* be carried as AGTP NOTIFY method invocations with a broadcast group identifier in the recipient parameter. The broadcast group *MUST* be registered in the governance platform before use.

7.3.3. Multi-Agent Coordination

ACP multi-agent coordination workflows *SHOULD* be carried as AGTP COLLABORATE method invocations where two or more agents work in parallel or defined roles toward a shared objective. AGTP COLLABORATE is peer-to-peer, consistent with ACP’s coordination model.

7.3.4. Capability Discovery

ACP capability queries *MUST* be carried as AGTP DESCRIBE method invocations. The AGTP DESCRIBE response *MUST* include all capability domains relevant to the querying agent’s task.

7.4. Wire Example: ACP Unicast Message over AGTP

```
AGTP/1.0 NOTIFY
Agent-ID: agtp://agtp.acme.tld/agents/coordinator
Principal-ID: usr-ops-team
Authority-Scope: agents:notify
Session-ID: sess-acp-c3d4e5
Task-ID: task-0201
Content-Type: application/agtp+json

{
  "method": "NOTIFY",
  "task_id": "task-0201",
  "parameters": {
    "recipient": "agtp://agtp.acme.tld/agents/worker-01",
    "content": {
      "acp_message_type": "task_assignment",
      "payload": {
        "task": "Process batch-2026-Q1",
        "deadline": "2026-04-01T00:00:00Z"
      }
    },
    "urgency": "normal",
    "delivery_guarantee": "at_least_once"
  }
}
```

8. Cross-AGMP Interoperability

8.1. Agent Discovery Across AGMPs

An agent registered with an AGTP governance platform is discoverable regardless of which AGMP it uses. The AGTP Agent Manifest Document exposes the agent's supported methods and modalities; the AGTP DESCRIBE method exposes its capabilities. AGMP-specific capability formats (A2A Agent Cards, MCP capability lists) *MAY* be included as extensions to the AGTP Capability Document but are not required.

8.2. Session Continuity Across AGMPs

A single AGTP session *MAY* carry messages from multiple AGMPs within the same workflow. For example, an orchestrator agent might use A2A to delegate a task, then use MCP to invoke a tool within that task, all within a single AGTP session identified by the same Session-ID. AGTP session semantics persist across AGMP message types within the session.

8.3. Identity Consistency

An agent operating across multiple AGMPs *MUST* use the same canonical AGTP Agent-ID in all AGTP headers, regardless of which AGMP is carrying the message. AGMP-layer identity fields (A2A agent.id, MCP client_id) *MAY* differ from the canonical AGTP Agent-ID for backward compatibility with non-AGTP deployments, but the AGTP Agent-ID is authoritative for all governance and audit purposes.

9. Security Considerations

9.1. AGMP Payload Injection

Threat: A malicious actor embeds forged identity or scope fields in an AGMP message payload, attempting to override or supplement the AGTP header values with elevated permissions.

Mitigation: Infrastructure components *MUST NOT* parse AGMP payloads for identity or scope information. The precedence rule in Section 4 is absolute: AGTP headers govern all infrastructure decisions. Application code *SHOULD* validate that AGMP payload identity fields are consistent with the AGTP headers and *MUST* reject requests where they conflict in ways that suggest an escalation attempt.

9.2. Cross-AGMP Delegation Elevation

Threat: An agent uses AGMP composition to cross protocol boundaries in a way that circumvents AGTP scope enforcement. For example, receiving a task via A2A with a delegated scope, then re-delegating it via MCP with an elevated scope.

Mitigation: The AGTP Delegation-Chain header *MUST* be maintained across AGMP boundaries. Each re-delegation *MUST* result in a new AGTP DELEGATE invocation with a scope that is a strict subset of the delegating agent’s Authority-Scope. Scope *MUST NOT* increase across AGMP composition boundaries.

9.3. Payload Schema Validation

Implementers *SHOULD* use the AGTP Content-Schema header to declare the schema of the AGMP payload. Recipients *SHOULD* validate the payload against the declared schema before processing. This mitigates injection attacks that rely on schema ambiguity at the AGMP layer.

10. IANA Considerations

This document defines no new IANA registrations. All AGTP method, header, and status code registrations are defined in [AGTP]. AGMP specifications retain their own registrations. The Content-Schema URIs used in this document reference external schema registries maintained by the respective AGMP specification bodies.

11. References

11.1. Normative References

[AGTP] Hood, C., "Agent Transfer Protocol (AGTP)", Work in Progress, Internet-Draft, draft-hood-independent-agtp-02, 2026, <https://datatracker.ietf.org/doc/html/draft-hood-independent-agtp-02>.

[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>.

11.2. Informative References

[A2A] Linux Foundation, "Agent-to-Agent Protocol Specification", 2025, <https://a2aprotocol.ai>.

[ACP] IBM Research, "Agent Communication Protocol", 2025.

[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-DISCOVER] Hood, C., "AGTP Agent Discovery and Name Service", Work in Progress, Internet-Draft, draft-hood-agtp-discovery-00, 2026, <https://datatracker.ietf.org/doc/html/draft-hood-agtp-discovery-00>.

[ANP] "Agent Network Protocol", 2025.

[MCP] Anthropic, "Model Context Protocol", 2024, <https://modelcontextprotocol.io>.

Appendix A. Acknowledgments

The AGMP composition profiles in this document build on the foundational work of the MCP, A2A, and ACP specification teams. AGTP is designed to be additive to and compatible with their work, not competitive with it. Thank you to Anthropic, Linux Foundation A2A team, and IBM ACP.

Author’s Address

Chris Hood
Nomotic, Inc.
Email: chris@nomotic.ai
URI: https://nomotic.ai