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

T. Hori

May 2026

Agent Quality Graph (AQG): A Protocol for Evaluating AI Agent Trustworthiness via Delegation Graphs
draft-hori-agent-quality-graph-00

Abstract

This document describes the Agent Quality Graph (AQG) protocol, a method for evaluating and ranking AI agent trustworthiness based on delegation transaction graphs. As the number of autonomous AI agents grows rapidly, there is no standardized mechanism for determining which agents reliably complete delegated tasks. AQG applies graph-based ranking algorithms, analogous to web page ranking via hyperlink analysis, to the domain of agent-to-agent delegation. Agents that are frequently delegated to by other highly-ranked agents receive higher trust scores. This document defines the delegation record format, the graph construction process, the scoring algorithm, and the API for querying trust scores.

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 2 November 2026.

Copyright Notice

Table of Contents

1. Introduction

As of 2026, over 100,000 AI agents are deployed across more than 15 registries and marketplaces. Protocols such as MCP (Model Context Protocol) and A2A (Agent-to-Agent) enable agents to communicate and delegate tasks. However, no standard mechanism exists for evaluating whether an agent will reliably complete a delegated task.

Current approaches to agent discovery rely on self-reported capabilities, download counts, or manual reviews. These signals are easily manipulated and do not reflect actual task completion quality.

AQG addresses this gap by building a directed graph of delegation transactions between agents. Each delegation creates a weighted edge from the delegating agent to the delegated agent. A graph-based ranking algorithm then computes trust scores that reflect the accumulated evidence of successful task completion.

1.1. Motivation

The design of AQG is inspired by the success of link-based ranking in web search (PageRank). In the web graph, a link from page A to page B is treated as a "vote" for page B's relevance. Similarly, in AQG, a delegation from agent A to agent B is treated as evidence of agent B's capability.

Key differences from web link analysis:

* Delegation edges carry outcome metadata (success, failure, quality score)

* Edges have recency weights (recent delegations matter more)

* The graph is partitioned by task category

* Anti-gaming mechanisms prevent Sybil attacks and score manipulation

1.2. Terminology

Agent An autonomous software entity capable of receiving and completing tasks

Delegation A transaction where one agent (delegator) assigns a task to another agent (delegatee)

Delegation Record A signed, immutable record of a delegation transaction including outcome

Trust Score A value between 0.0 and 1.0 representing an agent's accumulated reliability

AQG Node A vertex in the quality graph representing an agent

AQG Edge A directed, weighted edge representing accumulated delegation evidence between two agents

Trust Provider An entity that computes and publishes trust scores from delegation data

2. Delegation Record Format

Each delegation transaction produces a Delegation Record. The record is a JSON object with the following fields:

```
{
  "record_id": "uuid-v4",
  "delegator": "agent:travel-planner@example.com",
  "delegatee": "agent:hotel-booker@example.com",
  "task_category": "booking",
  "task_description": "Book hotel room for 2 nights",
  "timestamp": "2026-05-03T12:00:00Z",
  "outcome": {
    "status": "success|failure|partial|timeout",
    "quality_score": 0.95,
    "latency_ms": 450,
    "verifier": "agent:travel-planner@example.com",
    "verified_at": "2026-05-03T12:00:01Z"
  },
  "context_hash": "sha256:abcdef...",
  "signature": {
    "algorithm": "Ed25519",
    "value": "base64-encoded-signature",
    "public_key": "base64-encoded-public-key"
  }
}
```

2.1. Required Fields

* record_id: Unique identifier (UUID v4)

* delegator: Agent identifier of the task delegator

* delegatee: Agent identifier of the task executor

* timestamp: ISO 8601 datetime of delegation

* outcome: Result of the delegation (status required, others optional)

2.2. Outcome Status Values

success	Task completed satisfactorily
failure	Task could not be completed
partial	Task partially completed
timeout	Task did not complete within expected time

2.3. Signature

Delegation records SHOULD be signed by the delegator using Ed25519 or ECDSA-P256. The signature covers the canonical JSON of all fields except the signature object itself. This prevents tampering and enables verification of record authenticity.

3. Graph Construction

3.1. Node Creation

Each unique agent identifier becomes a node in the quality graph. Nodes are created on first appearance in any delegation record.

3.2. Edge Aggregation

For each (delegator, delegatee) pair, a single directed edge is maintained. The edge weight is computed from all delegation records between the pair:

Edge weight = sum(outcome_weight * recency_weight) for each record

Where:

* outcome_weight: success=1.0, partial=0.5, timeout=-0.2, failure=-0.5

* recency_weight: exponential decay with half-life of 90 days

3.3. Category Partitioning

The graph is partitioned by task_category. An agent may have different trust scores in different categories. The global trust score is the weighted average across all categories.

4. Scoring Algorithm

4.1. Base Score Computation

The base trust score for each agent is computed using a modified PageRank algorithm applied to the AQG graph:

Score(agent_i) = (1 - d) / N + d * sum(Score(agent_j) * w(j->i) / out_degree(j)) for all agents j that delegate to agent_i

Where:

* d = damping factor (0.85)

* N = total number of agents

* w(j->i) = normalized edge weight from j to i

4.2. Score Normalization

Raw scores are normalized to the range [0.0, 1.0] using min-max normalization across all agents. A minimum of 10 delegation records are required before a score is published (cold-start threshold).

4.3. Anti-Gaming Mechanisms

* Sybil Resistance: Newly created agents have no score until they receive delegations from established agents (bootstrap problem).

* Collusion Detection: If a cluster of agents only delegate among themselves with uniformly positive outcomes, their mutual edge weights are discounted.

* Temporal Decay: Scores naturally decay without ongoing positive delegations, preventing legacy agents from maintaining high scores indefinitely.

* Verification Requirement: Outcomes signed by both delegator and delegatee carry higher weight than single-signed outcomes.

5. API Specification

5.1. Submit Delegation Record

POST /aqg/v1/records

Accepts a signed delegation record. Validates signature, indexes the record, and triggers asynchronous score recomputation.

5.2. Query Trust Score

GET /aqg/v1/scores/{agent_id}

Returns the current trust score for an agent:


```
{
  "agent_id": "agent:hotel-booker@example.com",
  "global_score": 0.87,
  "categories": {
    "booking": { "score": 0.92, "records": 156 },
    "scheduling": { "score": 0.78, "records": 23 }
  },
  "computed_at": "2026-05-03T12:00:00Z",
  "provider": "aqg.example.com",
  "signature": { "algorithm": "Ed25519", "value": "..." }
}
```


5.3. Query Delegation Graph

GET /aqg/v1/graph/{agent_id}?depth=2

Returns the subgraph of delegation relationships for the specified agent, up to the requested depth.

6. Integration with Existing Protocols

6.1. A2A Integration

AQG trust scores can be included in A2A Agent Cards as an extension:


```
{
  "name": "Hotel Booker",
  "...": "...(standard A2A Agent Card fields)...",
  "extensions": {
    "aqg": {
      "trust_score": 0.87,
      "score_provider": "https://aqg.example.com",
      "score_url": "https://aqg.example.com/aqg/v1/scores/agent:hotel-booker@example.com"
    }
  }
}
```


6.2. MCP Integration

MCP servers can expose their AQG trust score via the agent.json well-known URI:

```
{
  "name": "hotel-booker",
  "description": "Books hotel rooms",
  "trust": {
    "verified": true,
    "score": 0.87,
    "source": "aqg.example.com"
  }
}
```

7. Security Considerations

* Delegation records MUST be transmitted over TLS 1.2 or higher

* Record signatures prevent tampering with delegation history

* Trust score responses from providers SHOULD be signed to prevent spoofing

* The Sybil resistance mechanism prevents creation of fake agents to inflate scores

* The collusion detection mechanism prevents ring-boosting of scores

* Privacy: Delegation records contain only agent identifiers, not user data

* Score manipulation: The recency decay ensures that historical manipulation becomes less effective over time

8. IANA Considerations

This document requests registration of the Well-Known URI "aqg" in the IANA Well-Known URIs registry for discovering AQG endpoints.

URI suffix: aqg Change controller: IETF Specification document: this document Related information: Agent Quality Graph endpoint discovery

9. References

10. References

10.1. Normative References

[RFC8615] Nottingham, M., "Well-Known Uniform Resource Identifiers (URIs)", May 2019.

[A2A] LLC, G., "Agent-to-Agent Protocol", 2025.

[MCP] Anthropic, "Model Context Protocol", November 2024.

10.2. Informative References

[PAGERANK] Page, L., Brin, S., Motwani, R., and T. Winograd, "The PageRank Citation Ranking: Bringing Order to the Web", 1999.

[ARDP] Pioli, R., "Agent Registration and Discovery Protocol (ARDP)", February 2026.

Appendix A. Acknowledgements

The design of AQG is inspired by the PageRank algorithm (Page et al., 1999) and the Agent Registration and Discovery Protocol (Pioli, 2026).

Author’s Address

Takayuki Hori
Japan
Email: 0xoyabun@gmail.com