Agent-GW
Internet-Draft
Intended status: Standards Track
Expires: 20 August 2026

Xiaohui. Xie
Tsinghua University
Zian. Wang
Beijing University of Posts and Telecommunications
Tianshuo. Hu
Tsinghua University

16 February 2026

Agent Communication Gateway for Semantic Routing and Working Memory
draft-agent-gw-00

Abstract

This document presents an architectural framework for an Intelligent Agent Communication Gateway (Agent-GW), designed to support large-scale, heterogeneous, and dynamic multi-agent collaboration. As agents evolve from isolated software entities to a collaborative digital workforce, the underlying infrastructure must transition from rigid, host-based connectivity to flexible, intent-based interaction.

This document outlines the requirements for such a transition and proposes the Agent-GW as a unified infrastructure hub. The gateway provides native primitives for Semantic Routing—dispatching tasks based on intent and capability—and Working Memory, which manages structured context for multi-step workflows. Furthermore, it defines mechanisms for automated protocol adaptation, oracle-free agent evaluation, and collaborative inference acceleration (KDN). The architecture aims to enable agents and legacy systems to interoperate through standardized protocols while ensuring observability, security, and operational scalability.

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 20 August 2026.

Table of Contents

1. Introduction

The rapid advancement of Large Language Models (LLMs) has catalyzed the emergence of the "Internet of Agents," a paradigm where autonomous software entities and tool-like services interconnect to form collaborative workflows. Unlike traditional microservices, these agents possess varying degrees of autonomy, reasoning capabilities, and diverse interface standards. Early agent deployments were typically siloed within proprietary frameworks, limiting their ability to collaborate across administrative domains

or heterogeneous platforms.

As these systems scale, the fundamental bottleneck shifts from basic network connectivity to context management and efficient orchestration. Delivering the right context to the right agent at the right time—while managing the high computational cost of inference—becomes a critical infrastructure challenge. Existing network/application gateways, designed for static endpoints and stateless packet forwarding, lack the semantic awareness required to interpret agent intents or manage the lifecycle of a collaborative task.

This document introduces the Agent Communication Gateway (Agent-GW), an architectural entity situated between agents and external tools or services. The Agent-GW elevates the network's role from a passive transport layer to an active semantic intermediary. It introduces two core primitives: Semantic Routing, which decouples task execution from physical endpoints by routing based on capabilities and runtime state; and Working Memory, which provides a shared, incrementally updated context layer to support multi-step reasoning.

2. Conventions used in this document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

3. Terminology

The following terms are defined in this draft:

Agent-GW (Agent-GW) Agent Communication Gateway; the infrastructure component coordinating multi-agent communication, responsible for protocol translation, semantic routing, and context management.

Semantic Routing The process of routing a request based on the semantic intent of the task and the capabilities of available agents.

Working Memory A structured, temporary storage mechanism within the gateway that maintains the context and state of a multi-turn agent interaction.

KDN (Knowledge Delivery Network) A mechanism that treats inference states (e.g., LLM KV caches) as reusable and distributable artifacts.

MCP Model Context Protocol; a reference standard for connecting AI

assistants to systems/data.

4. Network and Infrastructure Requirements

The proliferation of intelligent agents fundamentally reshapes interaction patterns in future networks. Agent interactions are typically short-lived, context-heavy, and driven by high-level goals rather than explicit commands. To support this, the infrastructure must satisfy the following requirements:

Intent-Based Addressing: The network must support addressing schemes based on what constitutes the service (Capability) rather than where it is located (Topology).

Stateful Context Management: Unlike stateless HTTP requests, agentic workflows often involve multi-turn reasoning where context accumulates.

Heterogeneous Interoperability: The ecosystem comprises diverse entities. The infrastructure must provide automated adaptation layers.

Dynamic Capability Discovery: The network requires a dynamic discovery mechanism that can match task needs with agent capabilities in real-time.

Inference Efficiency: Mechanisms to cache and share intermediate inference states (such as KV caches) are required.

5. Architecture Overview

This section describes the reference architecture of the Agent Communication Gateway (Agent-GW). It functions as a Semantic Intermediary operating at the application and cognitive layers.

5.1. Architectural Model

Figure 1 illustrates the logical components and their interactions within the Agent-GW.

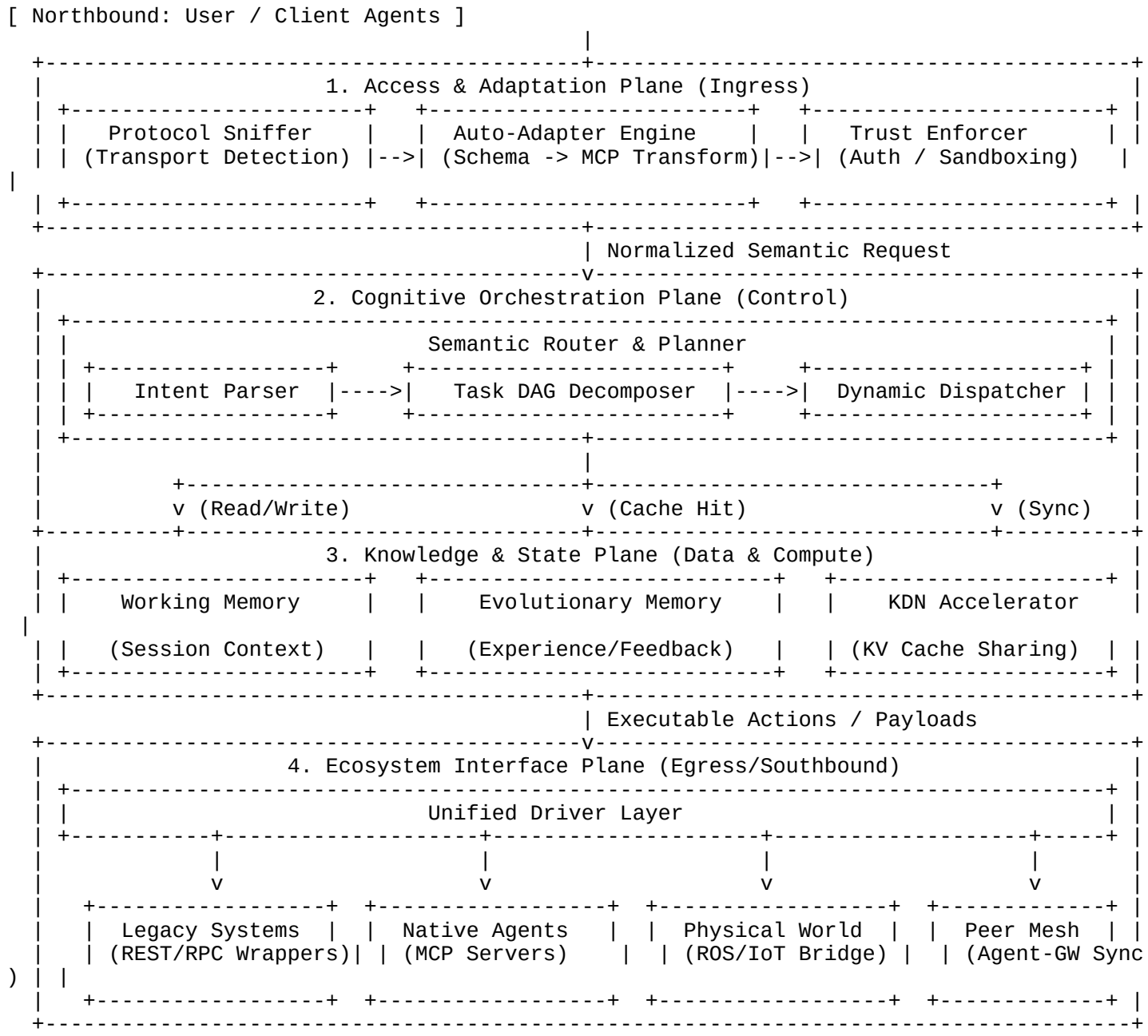

Figure 1: Agent-GW Reference Architecture

6. Infrastructure Functions Enabling Active Network Participation

6.1. Agent Identification and Capability Directory

This function establishes the "Root of Trust" for the agent network, shifting security from network-layer spoofing prevention to application-layer Capability Spoofing mitigation. The Agent-GW maintains a dynamic, verified directory where agent entries are not static records but active, verified states.

Cryptographic Identity and Verification: Participating agents MUST possess a Cryptographic Agent ID (AID) derived from an X.509v3 digital certificate. Upon registration, the agent submits an AgentCard binding its identity to a specific capability hash. To prevent the registration of malicious entities, the Agent-GW implements a Capability Claim and Verification (CCV) mechanism. Utilizing Metamorphic Testing principles, the gateway issues "Challenge-Response" queries (e.g., semantic variants of a task) to verify the agent’s functional consistency without accessing its internal model weights (Zero-Knowledge verification).

Semantic Heartbeat and Dynamic Pruning: To maintain directory freshness, the Agent-GW enforces a Semantic Heartbeat. Unlike traditional Layer 3 keep-alives that only confirm network reachability, this mechanism periodically verifies Layer 7 functional integrity. Agents that fail these semantic challenges (indicating they are "Zombie Agents" or functionally impaired) are dynamically pruned from the directory.

6.2. Automated Protocol Adaptation and Interface Normalization

Residing within the Access & Adaptation Plane, this function serves as the "Semantic Edge" that normalizes heterogeneous external protocols (e.g., HTTP, MQTT, gRPC) into the unified Model Context Protocol (MCP) used by the internal Orchestration Plane.

To handle unstructured or poorly documented interfaces, the Agent-GW implements a Generative Adaptation Mechanism with Active Probing. Instead of relying on static drivers, an LLM-based engine ingests raw interface descriptions to generate preliminary bindings. These bindings are iteratively refined through a self correcting feedback loop.

6.3. Infrastructure-Level Agent Evaluation and Compliance

Agents are often deployed as "black-box" entities where internal logic is opaque. The Agent-GW introduces an infrastructure-level evaluation mechanism to ensure reliability and compliance without requiring access to model weights.

This function employs Metamorphic Testing protocols: the gateway generates semantic variations of task instructions (e.g., rewriting the prompt or injecting noise) and evaluates the consistency of the agent's responses. This "Oracle-free" approach allows the gateway to assign a dynamic reliability score to each agent.

6.4. Dynamic Orchestration and Semantic Routing Mechanism

Static routing tables are insufficient for dynamic multi-agent collaboration. The Agent-GW implements Semantic Routing, a mechanism that dispatches tasks based on high-level intent, real-time capability matching, and operational constraints.

The Cognitive Orchestration Plane decomposes complex user intents into a Directed Acyclic Graph (DAG) of sub-tasks. The Dynamic Dispatcher then assigns these sub-tasks to the most suitable agents based on Capability Match, Trust Score, and Operational Metrics.

6.5. Evolutionary Knowledge Management

To improve collaboration efficiency over time, the Agent-GW incorporates Evolutionary Memory. This function transforms the gateway from a stateless forwarder into a learning infrastructure.

The gateway captures execution traces, success/failure feedback, and user corrections from passing traffic. In local or edge deployments, this allows the Agent-GW to build a localized knowledge base to refine routing policies and provide "Feedback Guidance" to terminal agents.

6.6. Collaborative Inference Acceleration (KDN)

Multi-agent workflows frequently involve redundant reasoning over shared contexts. To address the computational inefficiency, the architecture proposes a Knowledge Delivery Network (KDN).

The KDN function enables the sharing of intermediate inference states, specifically the Key-Value (KV) cache of the LLM, across co-located agents or peer gateways. This significantly reduces the Time-to-First-Token (TTFT) and overall computational load.

7. Security Considerations

The introduction of an active Agent-GW introduces specific security challenges: Agent Identity Spoofing, Capability Poisoning, Context Leakage, and Inference Artifact Security.

8. IANA Considerations

This document has no IANA actions at this time.

9. Acknowledgement

TBD

10. References

10.1. 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>.

Authors' Addresses

Xiaohui Xie
Tsinghua University
Email: xiexiaohui@tsinghua.edu.cn

Zian Wang
Beijing University of Posts and Telecommunications
Email: zianwang@bupt.edu.cn

Tianshuo Hu
Tsinghua University
Email: huts22@mails.tsinghua.edu.cn