

2-1: Required Functions of an AI Agent Gateway

Chenguang Du (Beijing Zhongguancun Laboratory)

Xiaohui Xie (Tsinghua University)

Zian Wang (Beijing University of Posts and Telecommunications)

July, 2026

What the previous talks asked for

6G Networks

(3GPP TR 22.870)

- Agent identity
- Registration & discovery
- Cross-domain collaboration
- Network context (QoS / slicing)

Cloud Multi-Agent

(problem statement)

- Stop rebuilding comms plumbing
- Capability-based routing
- Multi-tenant isolation
- Protocol bridging

Deployment Scenarios

(UnionPay · embodied agents)

- Onboarding & lifecycle
- Policy-controlled exposure
- Gateway-to-gateway exchange
- Audit w/o content inspection

Grouping the functions

Many needs → three groups of functions. Grouped by two things: what kind of information · how fast it changes.

CONTROL PLANE (the agent's own information)

Registration · Discovery · Info sync · Policy setup

changes slowly — stays the same for hours, or days

SEMANTIC MEDIATION (different from traditional apps)

Task dispatch · Routing · Agent selection

runs per request — the per-request decisions

RUNTIME DATA PLANE (closest to protocols like A2A)

Protocol adaptation · Observability · Operations & management

fast — one user request can trigger it many times

Trust & security (→ 2-2)

Grouping, not architecture.

GROUP 1: Control plane

It holds the slow-changing information: who is here, what they can do, and what the rules are.

Function 1 · Registration & attachment

An agent joins the gateway, proves its identity, and gets a reachable binding.
The gateway tracks its lifecycle: online, offline, expired.

Function 2 · Capability directory

The gateway keeps a record of what each agent can do.
Each record has an origin, a version, and a freshness time.

Function 3 · Gateway-to-gateway sync

Gateways exchange capability and reachability information with each other, so an agent in another domain can be found before any request comes.

Function 4 · Policy (setup side)

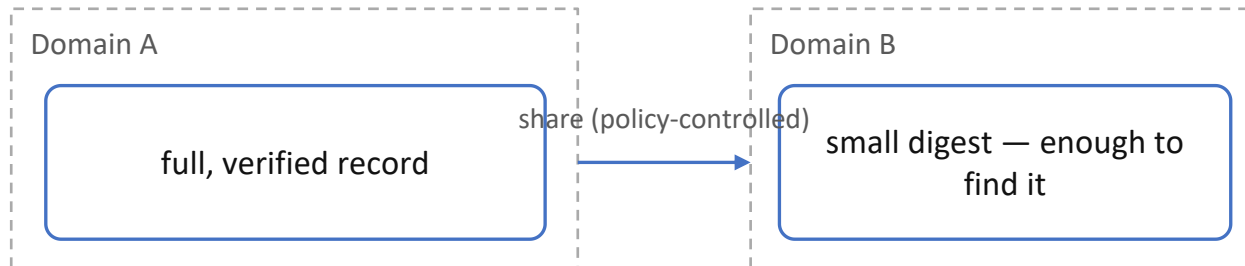
Operators write the rules here: who may call whom, with what quota.

Four functions can manage the phone book of the agent world.

GROUP 1 Control plane — drafts and running code

DRAFTS

- **draft-sz-dmsc-iaip-02:** Register, advertise capability, refresh with a time limit.
- **draft-zhang-dmsc-gateway-directory-sync-01:** a full record inside the domain, a small digest across domains.
- **draft-zhao-opsawg-agent-gateway-policy-01:** a YANG management model: subjects, bindings, applied state.



RUNNING CODE

- **agentgateway** (Linux Foundation) does registration, directory, and access control.
- **Higress** (CNCF sandbox, from Alibaba) does access, auth, and rate limits.
- **ANP** does DID-based identity and well-known discovery.

*Different teams
built the same functions.*

STATUS: Drafts and deployments point the same way. Standardization can start here.

GROUP 2: Semantic mediation

Function 5 · Routing / peer selection

A request comes in, described by its goal — not by an address.

The gateway filters the candidate agents, checks the constraints, and picks a peer.

Function 6 · Task context & handoff

A task crosses several agents, step by step.

The gateway carries the context from one step to the next, under policy control, so the next agent knows what happened before.

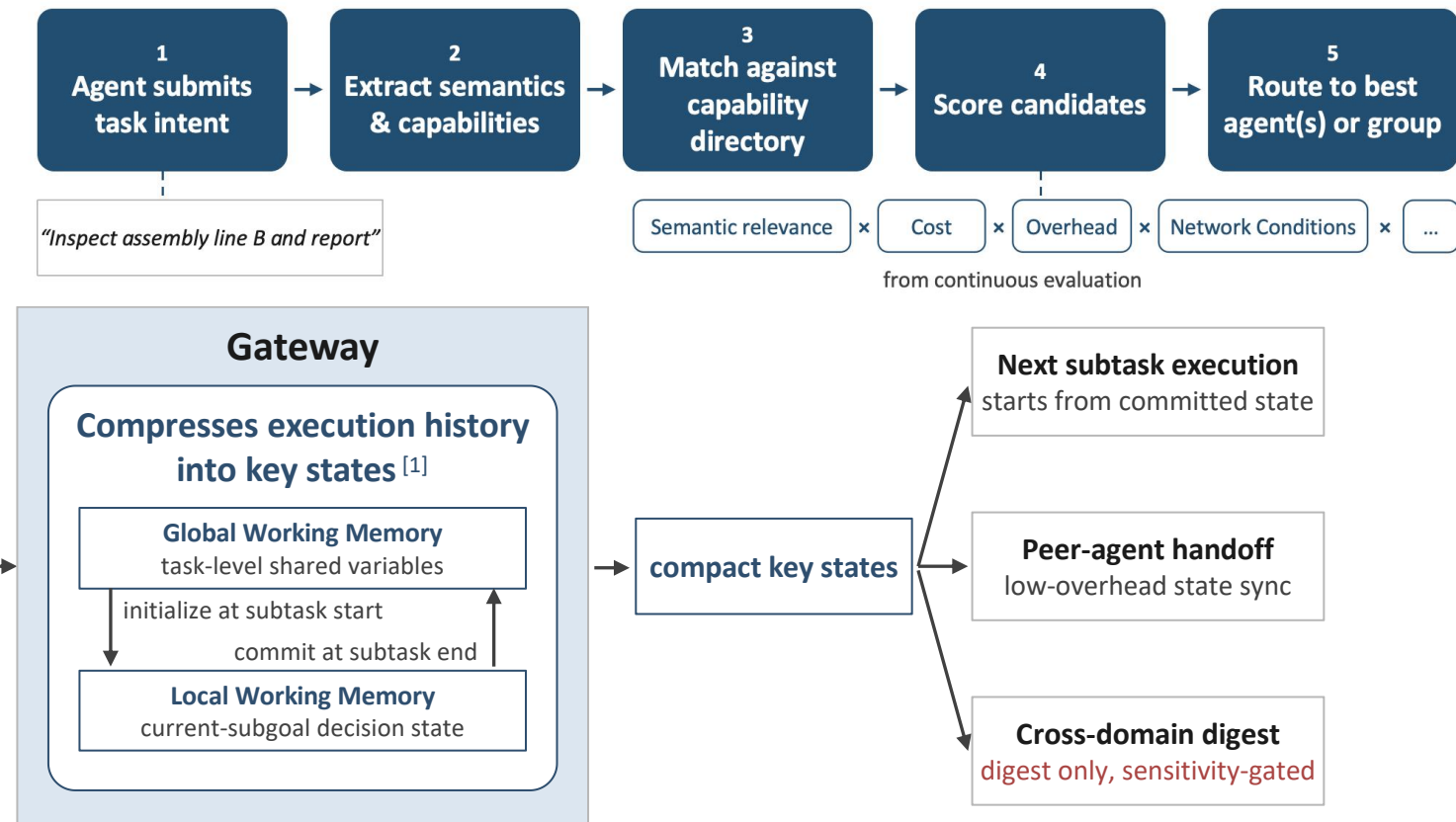
GROUP 2 Semantic mediation — drafts and running code

DRAFTS

- **draft-agent-gw:** semantic routing, working memory, knowledge (KVCache) transfer across domain gateways.
- **draft-cui-ai-agent-discovery-invocation:** task-agent matching, agent invocation.

RUNNING CODE

- OpenGateway
- LMCache+SGLang



STATUS: Active — real drafts, real systems.
But how much to define is still open.

GROUP 3: Runtime data plane

Function 7 · Protocol adaptation

Agents speak different protocols: A2A, MCP, ANP, HTTP API

The gateway translates once, at the edge — so agents do not need N×M adapters.

Function 8 · Policy enforcement

The rules written in the control plane are applied here, on every message:
access control, QoS, token quota.

Function 9 · Observability & audit

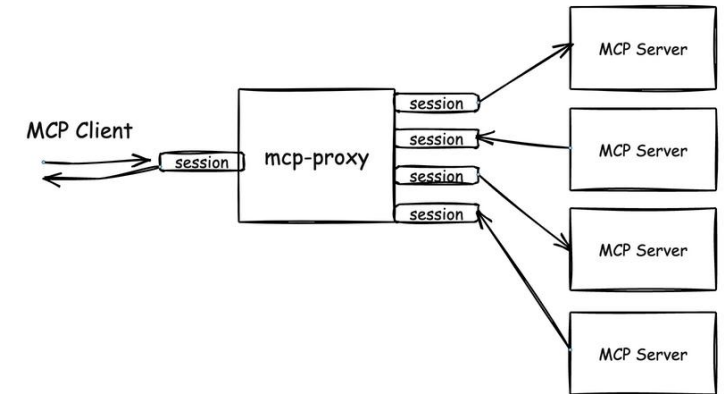
The gateway records who talked to whom, when, and at what cost.
Metadata only — no content inspection.

The daily work of a gateway.

GROUP 3 Runtime data plane — the market built it already

They all provide proxying, token metering, and access control.

	agentgateway	Kong	Higress
Proxy A2A	✓	✓	✓
Proxy MCP	✓	✓	✓
Token quota	✓	✓	✓
AuthN / Z	✓	✓	✓
OpenTelemetry	✓	✓	✓



- Different teams — and they built the same functions.
- The use cases stay inside one LAN, one organization — cross-domain cases are still few.
- The need for interoperation is low

Directions for standardization

Direction 1 · Control plane (Functions 1–4)

Start here. The draft text is concrete, and independent running code agrees.

Standardize the objects: registration, directory record, cross-domain digest, policy model.

Direction 2 · Semantic mediation (Functions 5–6)

Define the interfaces, not the intelligence.

Standardize what the two sides exchange: request, response, handoff, failure.

The choosing logic — matching, ranking — stays with implementations.

Direction 3 · Runtime data plane (Functions 7–9)

Watch — do not standardize yet.

The market converged on its own, but use stays inside one organization.

Reuse existing protocols. Revisit when cross-domain use grows.

Thanks