AI Agent Discovery and Invocation Protocol

Intent-based agent selection as application-layer semantic routing

Yong Cui, Yihan Chao, Chenguang Du

The problem is cross-platform agent interoperability

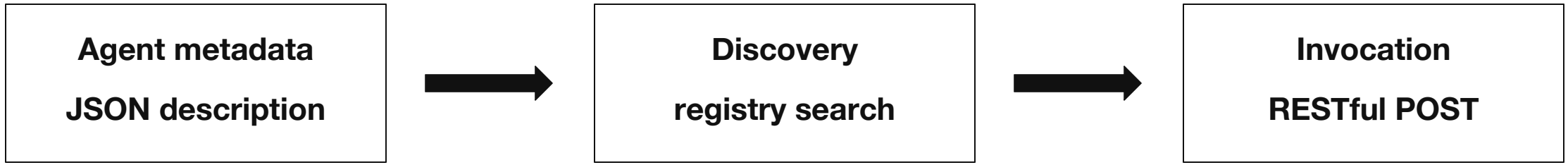
AI agents are increasingly exposed as network services, but platform-specific APIs make multi-agent integration harder.

- Agent frameworks often operate inside isolated ecosystems.
- Clients that use agents from multiple sources must adapt to different description and invocation interfaces.
- The draft targets developers of AI agent platforms, service providers, and architects of AI-enabled or multi-agent systems.

The protocol response is a common way to describe agents, discover suitable agents, and invoke them consistently.

AIDIP standardizes the path from metadata to invocation

The draft defines three protocol surfaces, with security and existing web standards around them.



Security considerations cover authentication, authorization, TLS, trust measures, rate limits, and validation of outputs.

Metadata makes an agent discoverable and callable

Agents provide a structured metadata document used for registration, discovery, and later invocation.

CORE FIELDS

- id, name, description, version, publisher
- capabilities, tags, endpoint, status
- location, supported_languages, authentication

OPERATION FIELDS

- operation name and description
- input schema and output schema
- optional example input and output pairs

The operation level lets one agent expose multiple callable actions without treating each action as a separate agent.

Discovery is registry-based and attribute-aware

An Agent Registry stores metadata and provides endpoints for registration, search, and retrieval.

POST /agents

register metadata

POST /agents/search

filter or intent query

GET /agents/{id}

retrieve full metadata

- Attribute filters include capabilities, tags, supported languages, and location.
- For complex queries, POST /agents/search is recommended.
- Multiple registries can coexist; formal registry federation is out of scope.

Semantic routing means intent-based agent selection here

The draft uses the term descriptively for selecting suitable AI agents at the application layer.

(Task Intent, Execution Context, Selection Constraints)



Ranked Candidate Agent Set

It helps a client, host agent, or coordinator describe the task first, then receive candidate agents before invoking any specific agent.

Selection combines intent, context, and constraints

The intent should describe the desired outcome, while constraints are binding when understood.

REQUEST


```
{
  "intent": "Summarize a long Chinese legal contract
and identify unusual risk clauses.",
  "context": { "language": "zh", "domain": "legal",
"data_location": "cn" },
  "constraints": { "required_trust_level": "verified",
"data_residency": "cn" },
  "top": 3
}
```


RESPONSE

Ranked candidates MAY include:

- agent_id and name
- operation_id and endpoint
- match_score
- selection_reason
- constraint_result
- authentication metadata
- expires_at

Selection does not replace discovery or invocation

It is a capability of discovery that produces candidates before normal invocation.



A client may invoke one returned agent, retrieve more metadata, or decline all candidates.

Regardless of whether selection runs in a registry, host agent, or coordinator, it does not change the metadata format or invocation interface.

Invocation stays RESTful and schema-driven

Clients invoke the selected agent endpoint with JSON that conforms to the agent's published operation schema.

- Method: POST to the endpoint from metadata, or to a gateway-provided path.
- Headers include Content-Type: application/json and required authentication.
- A shared gateway MAY use target_agent_id and operation_id to dispatch the request.
- 2xx, 4xx, and 5xx status codes are used to reflect the semantic outcome.

Streaming, batch requests, and idempotency safeguards are noted as additional implementation considerations.

The draft draws a clear boundary around semantic routing

Intent-based selection answers which agent is suitable for a task, not how network traffic reaches a destination.

EXPLICIT NON-GOALS

- IP routing or packet forwarding
- path computation or next-hop selection
- reachability advertisement
- DNS or name-to-address resolution
- transport-layer load balancing
- service mesh traffic routing
- agent invocation semantics
- global or persistent agent identifiers

Security extends into the pre-invocation phase

The registry, gateway, and client all have security responsibilities before and during invocation.

- Discovery and invocation traffic MUST use TLS.
- OAuth 2.0 bearer tokens, API keys, or mutual TLS are required except for public discovery endpoints.
- Registries MUST enforce per-client entitlements across search results and invocation access.
- A gateway MUST NOT treat target_agent_id as sufficient authorization.
- Intent-based selection SHOULD validate capability claims, apply constraints, and exclude unsafe agents.
- Clients must treat agent outputs as untrusted until verified.

The revision is backward-compatible and incremental

Existing discovery and invocation mechanisms remain valid and unchanged.

- Intent-based agent selection is optional for baseline protocol conformance.
- Registries can add intent-based selection without affecting existing clients.
- The draft recommends it for dynamic agent selection, multi-agent orchestration, and intent-driven task delegation.
- The document has no IANA actions.

Discussion focus: whether the intent/context/constraints model is the right protocol surface for application-layer semantic routing among AI agents.