INTENT-BASED INTER-AGENT COMMUNICATION PROTOCOL

IICP

An open protocol for choosing AI providers through public rules

Roble Mumin

IETF / DISPATCH introduction · 14 August 2026

THE PROBLEM

Every agent action includes a provider choice

Agents call tools, delegate work, and reach local or remote services. Before they act, an application or platform chooses a provider and a route.

That choice is usually hard-coded or platform-specific.

An open system needs a way to choose across compatible providers without rebuilding the same logic in every application.

Source: IICP Overview, August 2026; IICP directory and client architecture.

WHERE IICP FITS

A clear role beside other AI standards

- `MCP` connects AI tools and context
- `A2A` coordinates work between agents
- `APIs` connect applications to services

> IICP understands the need, applies capability and policy rules, and selects an eligible provider. The provider then uses a connection method both sides support.

Source: IICP standards crosswalk, 31 July 2026, refreshed against the August 2026 protocol landscape.

THE SELECTION FLOW

Describe the need before choosing the provider

HARD-CODED

Call a named provider before checking which services meet the need.

INTENT-BASED

Declare a registered capability and constraints; discover and select among eligible providers.

Intent → discovery → eligibility → selection → execution

Source: IICP core, directory, and semantics specifications.

HOW IT WORKS

A directory matches needs with providers

Public directory
Records provider capabilities and applies published eligibility and selection rules.

Providers
Perform the selected work through a route supported by both sides.

Client libraries
Help applications describe a need, receive a match, and connect to the selected provider.

The directory can make the match without receiving the task. The selected provider receives the work it performs.

Source: IICP directory, capability, and client specifications.

ILLUSTRATIVE USE CASE

Local policy and open selection can coexist

A clinic could keep an approved model on local hardware, advertise only its capability description, and allow authorized clients to select that route under local policy.

A remote route remains possible when policy allows it.

Illustrative, not customer proof.
Local execution can limit data movement; a chosen remote provider can still read the task.

Source: Illustrative application of current IICP selection and privacy boundaries.

PROPOSED ENTERPRISE USE

One policy across a mixed AI estate

An organization could register local models, commercial AI services, and internal services in a private directory, then apply one eligibility and selection policy.

For application teams
Describe the required capability instead of maintaining provider-specific choice logic.

For governance teams
Apply consistent policy checks and keep approved local routes available.

Source: Proposed use case; no customer or deployment claim.

WHAT EXISTS TODAY · 14 AUGUST 2026

A working public beta

PUBLIC RULES

The specification defines how needs, capabilities, and selection constraints are expressed.

WORKING SOFTWARE

Directory software and client libraries complete the matching and connection flow.

SHARED CHECKS

Automated tests help implementations follow the same behavior.

Anyone can inspect the specification, try the public reference, and review the implementation record.

Source: CURRENT_VERSIONS.md, SPEC_STATUS.md, IMPLEMENTATIONS.md; verified 14 August 2026.

THE PUBLIC REFERENCE

The complete journey works today

DESCRIBE

State the capability needed and the conditions that matter.

MATCH

Identify providers that satisfy the published requirements.

CONNECT

Send the task through a route the selected provider supports.

This turns provider choice into an inspectable process that applications can reuse.

Source: Public reference and live status: iicp.network. Technical detail follows in the appendix.

WHAT CROSSES THE WIRE

The selected client and provider exchange five message types



Start the session
`INIT` offers supported versions. `ACK` selects one version for the connection.

Request the work
`CALL` carries a task ID, Intent identifier, timeout, constraints, and the application payload.

Return the outcome
`RESPONSE` repeats the task ID and carries either a result or a structured error.

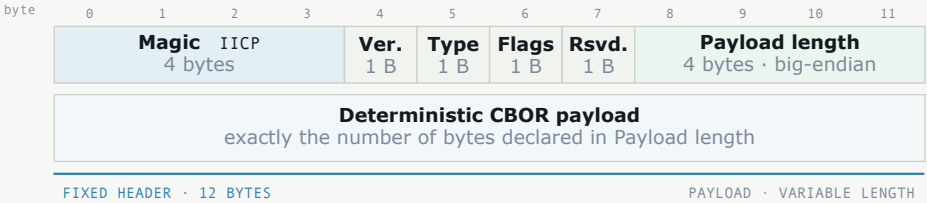
End or maintain
`CLOSE` ends the session; `PING/PONG` can check liveness.

TLS protects the connection. The selected provider still receives the plaintext task after decryption.

Source: IICP framing specification and unsubmitted peer-transport draft candidate.

HOW THE RECEIVER FINDS AND READS ONE MESSAGE

A native frame starts with a 12-byte header



Reading the frame
Check the marker, version, type, flags, and length before decoding the CBOR payload.

Safety and status
Check length before allocation; one frame is limited to 16 MiB. Magic identifies the format, not the sender.

Source: IICP framing §§1–2. Native framing is a project draft; HTTP/JSON and CBOR-over-HTTP remain supported mappings.

THE INTEROPERABILITY QUESTION

Some parts of this journey may need shared rules

ACROSS IMPLEMENTATIONS

How an Intent is identified
How a provider describes a capability
How eligibility is requested and returned
How errors and negotiation are represented

ACROSS TRUST BOUNDARIES

How a client authenticates a selected provider
Which evidence can be carried consistently
Which privacy and security requirements apply

The question is not whether every IICP feature should become an RFC. It is which exchanges must be consistent for independent implementations to interoperate.

Source: IICP suite and individual Internet-Draft workspace; IETF new-work guidance.

RELATIONSHIP TO CURRENT WORK

IICP selects; existing protocols can execute

Protocol or work	What it provides	How IICP can relate
HTTP / QUIC	Transport and application substrates	Carry a supported execution binding
MCP	Context, tools, and agent interaction	A selected provider may expose MCP
A2A	Agent discovery and task lifecycle	A selected endpoint may execute through A2A
IAIP / AIDIP	Discovery and invocation proposals	Material overlap requiring boundary review
DNS-AID / agent URI	Publication or addressing	Possible inputs, not live eligibility by themselves

IICP does not need to replace adjacent protocols. Its proposed role is the decision between a described need and a compatible provider route.

Source: Primary sources listed in the IICP agent-protocol landscape, 8 August 2026. Individual drafts are not IETF endorsement.

WHY THE ARCHITECTURE SEPARATES CONCEPTS

The request should keep its meaning when machinery changes

Intent ≠ transport
The same requested work can use different supported connection methods.

Capability ≠ implementation
A local model and a hosted service can advertise the same kind of work.

Identity ≠ locator
A provider can retain its identity when its endpoint changes.

Task ≠ connection
Long-running work can survive one network session.

Execution ≠ delivery
A selected task can use direct, relayed, or delayed delivery.

Requirement ≠ mechanism
Security and service requirements can use mechanisms suited to the route.

Source: IICP identity, lifecycle, framing, environmental-independence, and binding material.

DELIBERATE LIMITS

Several project choices can remain outside a standard

Ranking policy Implementations can use different weights while exchanging compatible eligibility data.

Commercial terms Pricing, credits, and marketplace operation need not define the core request flow.

Project governance Repository stewardship is separate from protocol interoperability.

A narrow specification should standardize only what independent clients, directories, and providers must understand in the same way.

Source: IICP extension architecture and current individual-draft scope.

REQUEST TO THE COMMUNITY

We are asking where the interoperable work belongs

Does intent resolution and cross-provider eligibility form a useful, separable protocol function?

Where does IICP overlap current IETF work, and what terminology or mechanisms should it reuse?

Which parts should be documented together, and is DISPATCH or another venue appropriate?

The implementation provides something concrete to review. The request is for corrections, scope, venue guidance, and interested contributors.

Source: IETF "Bringing New Work" guidance and DISPATCH charter; nothing has been submitted.

FURTHER STUDY

Read the specifications and inspect the implementations

PROTOCOL AND RUNNING REFERENCE

Specification, registries, and research
Native framing specification
Peer-transport draft candidate
Public reference service

DIRECTORIES

PHP reference directory
Rust operator preview

CLIENT AND PROVIDER SDKS

Python SDK
TypeScript SDK
Rust SDK

PROJECT STATUS AND STANDARDS WORK

Specification status
Current component versions
Implementation index
IICP protocol-landscape research

Source: All linked repositories and resources were publicly reachable on 14 August 2026.

APPENDIX A · 14 AUGUST 2026

Current component status

Component	Status	Evidence
Intent registry	Project-normative	17 entries with lifecycle metadata
Wire baseline	Stable project line	v1.9.0 schemas and fixtures
Client SDKs	Active	Python, TypeScript, Rust v0.7.103
PHP directory	Active	v1.10.90
Rust directory	Operator preview	v0.1.11
Native framing	Project draft	Individual I-D candidate, not submitted
Federation	Groundwork / preview	No external ratification

Source: CURRENT_VERSIONS.md and SPEC_STATUS.md at project commit 9bda966.

APPENDIX B

Security and privacy boundaries

Directory
Can observe Intent and routing metadata; does not need the task payload.

Provider
Must be authenticated independently; receives the plaintext task it executes.

Relay
Can observe connection metadata and may enter the payload path, depending on the selected route.

Open review areas include authorization mapping, downgrade handling, Sybil resistance, metadata minimization, and cross-domain trust.

Source: IICP peer-transport candidate, confidentiality, identity, and directory specifications.

APPENDIX C

Explicit non-goals

IICP does not seek to replace IP, HTTP, QUIC, MCP, A2A, IEEE networking, or DTN/BPv7; prescribe model architecture; define AGI; centralize providers; require a blockchain or token; require one marketplace; or standardize payment networks.

Bindings reuse existing protocols. Profiles add optional behavior. Selection policy can remain an implementation or deployment choice.

Source: IICP protocol-scope and IANA-considerations statements.

APPENDIX D · FOR DISCUSSION

Possible document decomposition

1 · Problem and architecture Terminology, roles, requirements, privacy, and security.

2 · Intent and capability model Identifiers, advertisements, matching, and errors.

3 · Discovery and selection Eligibility exchange without fixing commercial ranking policy.

Later, if justified Lifecycle, bindings, federation, receipts, and registries as separate documents.

Source: A discussion structure, not an adopted IETF work plan.

APPENDIX E

Open architectural questions

- Is intent resolution separable from gateway implementation?
- Which eligibility evidence can interoperate without standardizing ranking?
- How should current IAIP and AIDIP overlap change the problem statement?
- Which existing identity and authorization mechanisms should be mandatory?
- Is a new peer transport justified, or should bindings remain primary?
- Which identifiers or parameters require IANA coordination?

Source: Questions intentionally left open for community review.