



IICP Overall Architecture and Component Interfaces

Architecture reference for DMSC review and alignment

Roble Mumin

29 September 2026 / Version 0.1.3

Discussion draft

Informative interim reference. Not an IETF submission.

Components, interfaces and revision-pinned evidence for a first DMSC architectural review and alignment verification.

Published architectural foundation:

[IICP: Provider-Neutral Infrastructure for Intelligence Virtualization](#)

Roble Mumin, v1.0.1, 25 August 2026.

This interim document adds a revision-pinned component and interface account; it does not replace the published report.

Contents

1. Purpose and reading guide	2
2. Components and architectural view	4
3. Interface inventory and authority	7
4. One request through the architecture	8
5. State, policy, trust, and authority	11
6. Bindings, extensions, and deployment forms	13
7. Implementation and evidence scope	14
8. Relationship to DMSC	15
9. Two initial discussion proposals and reuse boundaries	16
10. Limitations and next steps	18
Appendix A. Source and evidence map	19
Appendix B. Release snapshot and publication delta	26
Appendix C. Gap report	29
Appendix D. Path to an IETF architecture draft	30
Appendix E. Versioned DMSC/IICP audit crosswalk	31
References and document provenance	46

1. Purpose and reading guide

This reference describes the Intent-based Inter-agent Communication Protocol (IICP) for Dynamic Multi-agent Secured Collaboration (DMSC) participants. It maps existing components and interfaces to DMSC concerns, distinguishes their authority and semantics, and proposes two bounded areas for technical discussion within the Internet Engineering Task Force (IETF).

Status: informative interim document, version 0.1.3, 29 September 2026; a discussion draft for DMSC architectural review and alignment verification. It is not an IETF submission. Neither IETF acceptance nor DMSC endorsement is claimed.

Executive summary

The **Intent-based Inter-agent Communication Protocol (IICP)** separates finding an eligible provider from invoking that provider. A **consumer** requests a named capability; a **provider** advertises capabilities and accepts or refuses execution; a **Directory** maintains discovery and route-selection metadata. **Management** handles domain-local policy and controlled state changes. An optional **relay** carries execution traffic when direct reachability is unavailable. These are logical roles, not requirements for separate machines. [Core](#); [Directory](#); [Management](#).

The main interfaces are provider–Directory registration and heartbeat, consumer–Directory discovery and route disclosure, consumer–provider execution, Directory–Directory synchronization, and Management–local authority planning and observation. The ordinary direct execution path does not send task payloads through the Directory. A relay, when used, is a separate data-path role even if colocated with Directory software. A route-disclosure ticket does not grant provider admission. [Directory](#); [ticket boundary](#).

IICP’s wire-level **Intent** is a stable capability identifier, not the user’s natural-language task. Capabilities describe the provider side of the same matching problem. Mandatory constraints and policy determine eligibility before preferences rank candidates. The provider still enforces its own authorization and availability checks. [Core](#); [selection semantics](#).

The published foundation is Roble Mumin’s *IICP: Provider-Neutral Infrastructure for Intelligence Virtualization*, v1.0.1, published 25 August 2026, DOI [10.5281/zenodo.22098223](https://doi.org/10.5281/zenodo.22098223). This interim reference preserves its separation of discovery, execution, management, and trust, while identifying later refinements and the difference between released specifications, working implementation code, optional drafts, and future work.

For Dynamic Multi-agent Secured Collaboration (DMSC), the useful comparison is functional rather than a contest between direct and gateway-mediated architectures. The DMSC architecture draft also permits peer-to-peer sessions after gateway coordination. IICP distributes related functions across the Directory, consumer, provider, and local Management authority; these roles do not map one-to-one to an Agent Gateway. [DMSC architecture](#), §7.1.

Two interfaces deserve first discussion: capability exposure with freshness and eligibility state (I1/I2), and the selection-to-execution handoff with distinct route and provider authority (I2/I3/I6). Synchronization and content-minimized operational evidence are secondary topics. Independent implementation evidence and coordinated pre-1.0 qualification remain open. The paper offers interfaces to review, not a request to standardize the whole IICP ecosystem. [qualification boundary](#); [independent record](#).

Reading paths. For a fast review, read this summary and the DMSC lens. For engineering review, follow Sections 2–9 and the five questions in Section 10. For audit review, use Appendices A–E and the matching JSON/CSV records. Section 9 proposes capability exposure and selection-to-execution handoff as the first discussion topics.

IICP through the DMSC lens

This view maps responsibilities, not interchangeable protocols. Audit IDs lead to the source-linked records in Appendix E; I1–I6 and O1 are defined in Section 3.

DMSC lens	IICP responsibilities and boundary	Audit
Gateway responsibility distribution	Directory coordinates metadata; Consumer dispatches; Provider admits and executes; Relay is optional. Directory alone is not an Agent Gateway.	D01
Management, control and forwarding planes	Discovery/control maps mainly to Directory; execution to bindings and endpoints; Management and trust cross role boundaries.	D02
Capability advertisement and inventory	Provider advertisements feed Directory discovery; complete variants and freshness qualify the claim. Inventory is not request eligibility.	D04
Selection and route disclosure	Directory filters/selects a route; Consumer validates and chooses dispatch. Selection does not grant Provider admission.	D07
Authentication and provider admission	Provider enforces caller authorization and capacity. Registration credentials and route tickets are not execution credentials; Agent Management Center (AMC) is not IICP Management.	D08
Execution paths and bindings	I3 invokes Provider directly; I6 binds the selected execution protocol or optional Relay. Peer execution is possible in both accounts.	D09
Telemetry and bounded evidence	O1 groups health, inventory and bounded evidence. Observation is not activation authority, and a signature does not establish truth.	D13

Boundaries that the mapping preserves

Intent is not task content. Directory is not automatically an Agent Gateway. Registration is not eligibility; eligibility is not selection. Route disclosure is not execution authorization; selection is not provider admission. Provider acceptance is not successful completion; successful completion is not semantic correctness. A signature establishes attribution and integrity of a bounded statement, not its truth.

Relationship to the published paper

The Zenodo report supplies the architectural foundation, not a September deployment inventory. Its Section 5 describes five planes; Sections 6 and 7 develop federation and Management; Section 11 discusses intelligence virtualization; Section 14 and Appendices D–I distinguish architecture, authority, conformance, and evidence. This document reorganizes those concerns around components and interfaces for a protocol-engineering audience. It does not reproduce the earlier paper’s full argument, experiment inventory, or research programme. [Published report](#).

Evidence cutoff and vocabulary. The frozen evidence account remains dated 29 September 2026, with sources pinned to revisions in Appendix B and the source manifest. The [refresh record](#) distinguishes unchanged repository/draft observations from newly checked adjacent references. “Released” refers to an immutable project release, not an IETF standard. “Working” means the frozen main revision; “proposed,” “pre-normative,” and “experimental” retain their source meanings. Implementation presence is not evidence of deployment, production suitability, or independent interoperability. The spec/v1.9/directory preserves a compatibility lineage; its name does not establish every contained document’s release or maturity. [Status policy](#); [version projection](#).

2. Components and architectural view

Logical roles

Consumer. The caller supplies an Intent, requirements, preferences, and a private task payload. It obtains candidates or a ticketed route, validates applicable trust and policy conditions, chooses whether to dispatch, invokes the selected binding, and handles the outcome. A consumer can itself be an agent or a provider in another interaction. [Core](#); [Python implementation](#).

Provider. The executing node advertises what its effective service can do, maintains Directory state, applies local admission and authorization, and returns a result or error. A model, tool, retrieval service, or agent can sit behind that interface; the advertised capability is not proof of output quality. Provider identity and the identity of its operator are distinct: optional operator identity mechanisms bind metadata but do not replace runtime authentication. [Capability format](#); [effective capability](#); [operator identity](#).

Directory. This control-plane role receives registration and capability metadata, tracks current state, filters discovery, and supports route selection/disclosure. It is not the task executor. The ticketed dispatch implementation can select a concrete route, so selection should not be attributed exclusively to the consumer. The consumer nevertheless decides whether to use that route, and the provider retains admission authority. [Directory](#); [dispatch controller](#).

Management. This role turns desired state into reviewable plans, accepts authorized local changes, and compares observations with intended state. It does not acquire execution authority merely by producing a plan. Current implementation contracts use owner-restricted local interfaces and bounded adapters, not a generally deployed remote administration service. [Management](#); [local authority](#).

Relay and browser participant. A relay forwards execution traffic for a provider that cannot accept a direct connection. A browser consumer is simply a caller using browser-compatible interfaces. A browser provider adds experimental execution and lifecycle constraints, including tab lifetime, WebGPU/WebAssembly availability, and relay connectivity. Directory replicas synchronize a root-authorized view. Local authority and bounded adapters accept and apply Management plans within an operator’s domain. These are implementation and topology choices, not extra Core planes. [Directory](#), [relay section](#); [browser node](#).

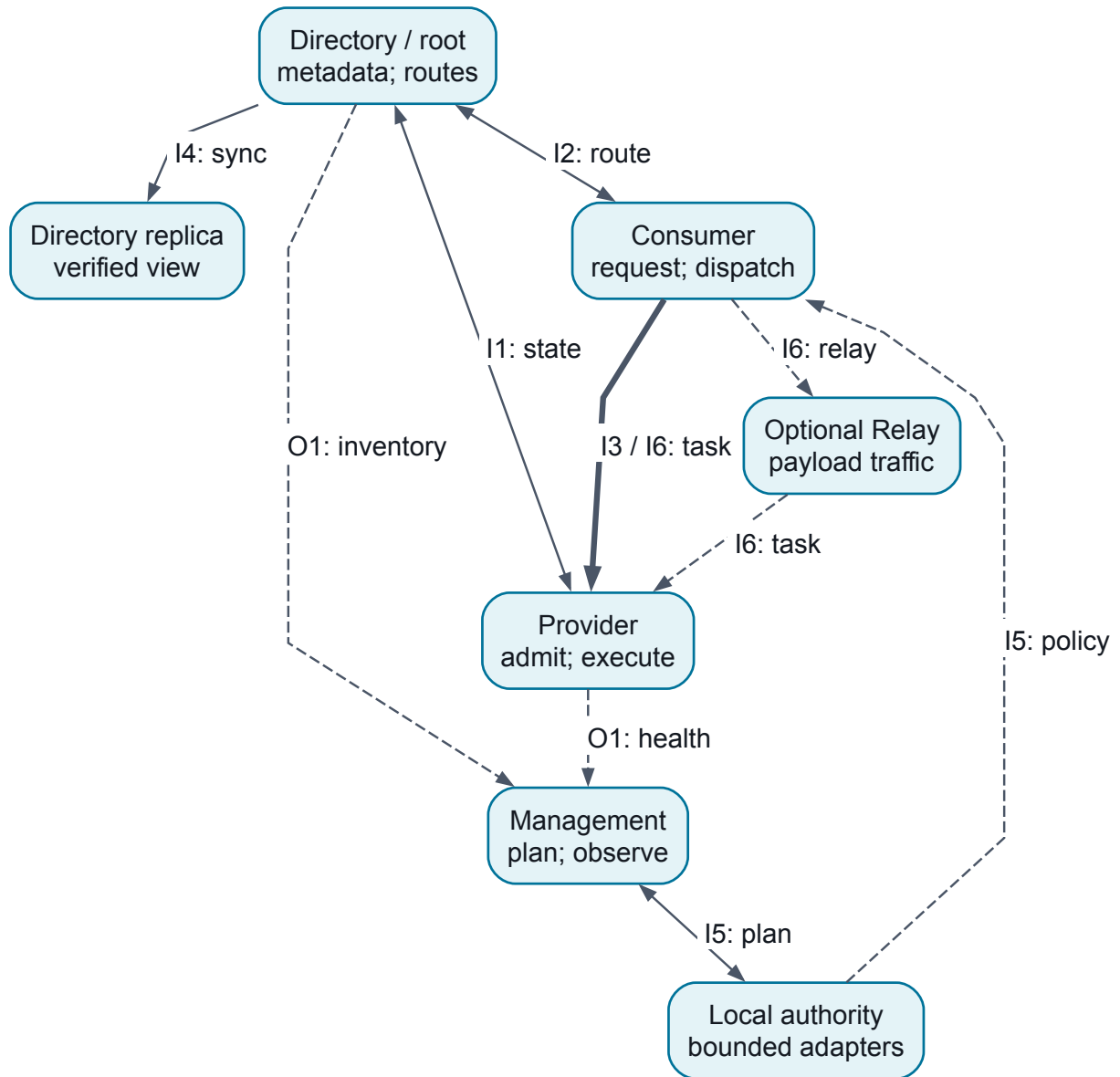


Figure 1: Component responsibilities and interface IDs. Solid thin arrows carry control metadata; the thick arrow is direct execution. Dashed arrows mark optional paths, bounded local policy or observation, as labelled; observation confers no authority. Role presence does not imply equal maturity: I4 is draft/preview, I5 local, and O1 an editorial grouping. No task-payload arrow terminates at the Directory.

Relationship to the five-plane model

The published paper's five planes remain a useful separation of concerns. The view below groups them for this discussion; it is explanatory, not a new normative layering rule. Semantic/protocol contracts govern discovery and execution. Discovery/federation maps to Directory functions. Execution maps to bindings, providers, and optional relays. Management/operations supplies local policy and observation. Trust/governance crosses all these boundaries. Providers are endpoints within this view, not a sixth protocol plane. [Published report, §5; extension architecture.](#)

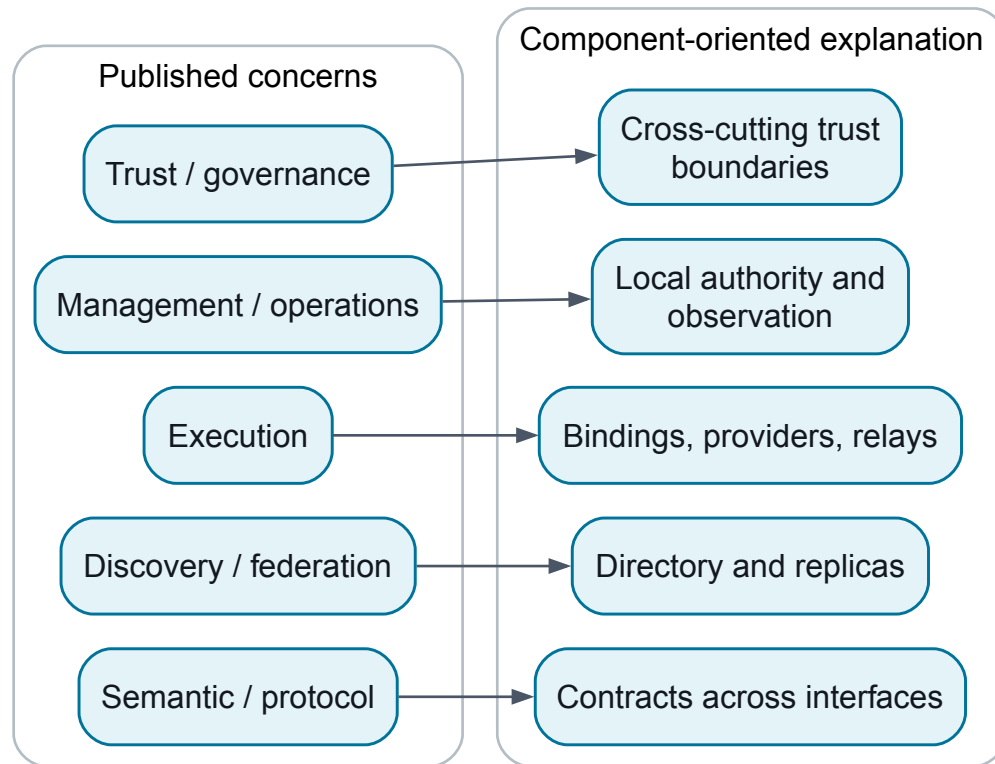


Figure 2: Five published concerns mapped to the component-oriented view. Trust and governance apply across roles; placement does not transfer authority.

Entities needed to read the interfaces

Entity	Meaning and boundary
Intent	Opaque, case-sensitive capability address, for example <code>urn:iicp:intent:llm:chat:v1</code> ; the private task is separate.
Capability / effective capability	Advertised service support; the effective form describes the complete model, template, runtime, interface, and adapter path.
Candidate / route	A possible provider versus concrete invocation information for a selected provider and binding. Neither guarantees admission.
Policy	Mandatory constraints and authority limits, distinct from preferences used to rank eligible candidates.
Task / attempt	Logical work versus an individual invocation attempt. Base Core uses <code>task_id</code> ; the optional lifecycle Profile distinguishes logical <code>task_id</code> from attempt <code>call_id</code> .
Ticket	A scoped signed statement whose authority depends on its specific contract, issuer, audience, and validity interval.
Result / error	Execution outcome under the chosen protocol or binding; neither is a proof of semantic correctness.
Evidence / receipt	A record of a bounded observation or claim. A signature identifies a signer and protects integrity; it does not make the claim true.

Sources: [Core](#), [identifiers](#), [effective capability](#), [lifecycle Profile](#), [tickets and receipts](#). The identifier decision treats `urn:iicp:` as deployed project syntax with external namespace registration pending; this paper does not claim an IANA registration.

3. Interface inventory and authority

The matrix is the component diagram's interface key. I1–I6 name logical boundaries, not six new network protocols. **O1 is an editorial grouping of observability surfaces**, not an additional IICP Core interface. Source status is part of each row; a working implementation does not make every optional Profile universally available.

Interface / exchanged information	Authority / task visibility	Status and direct sources
I1 — Provider ↔ Directory. Provider sends registration, capabilities, heartbeat, availability and withdrawal; Directory returns registration outcomes.	Directory verifies registration authority and controls its view; provider/operator controls its advertisement. Visibility: Metadata only; no task payload.	Working Directory contract and PHP implementation; effective-capability additions parity-gated. Directory protocol ; Directory state semantics ; PHP Directory .
I2 — Consumer ↔ Directory. Intent and constraints → candidates, or selected route plus disclosure ticket.	Directory filters and may select a route; consumer validates and decides whether to dispatch. Visibility: Metadata only; PHP rejects task-bearing fields.	Working discovery and v1 ticket path; opt-in v2 is separate. Directory protocol ; Dispatch route-ticket controller ; Python consumer implementation .
I3 — Consumer ↔ Provider. Task, binding-specific credentials → result, error or binding-owned lifecycle events.	Provider authenticates, authorizes and admits independently of route selection. Visibility: Execution payload visible to executor; link/binding protection varies.	Core CALL/RESPONSE; optional lifecycle and admission Profiles have separate status. Core wire format ; Service lifecycle Profile ; Provider admission Profile .
I4 — Directory root → replica; replica ↔ root synchronization. Signed snapshots, sequence/event-tail state and synchronization requests.	Replica verifies root authority and recalculates freshness; no sovereign trust negotiation implied. Visibility: Registration/state metadata, not task payload.	Draft root/replica contract; Rust Directory operator preview. Federated Directory ; Context and service event ownership ; Rust Directory .
I5 — Management ↔ local authority / bounded adapters. Desired proposal → exact plan and acceptance → bounded application and observations.	Domain owner grants scoped local authority; adapter limits actions; acceptance is not application. Visibility: Configuration and operational evidence; no general task-payload service.	Local implementation contracts; process-local Rust client enforcement. Management local authority ; Management local apply gate ; Management Rust client enforcement .
I6 — IICP ↔ execution binding; consumer ↔ relay ↔ provider when used. Selected provider, capability/interface context and binding-owned traffic; relay association has separate credentials.	Binding retains execution/authentication rules; relay gains no Directory or provider authority. Visibility: Payload crosses binding; relay visibility depends on actual protection.	MCP draft; A2A pre-normative; relay implementation-specific. MCP binding ; A2A execution binding v0 ; Directory protocol .
O1 — Operator / Management ← provider, Directory and local runtime. Health, authorized inventory, runtime snapshots and local event observations.	Producing surface owns observation; local policy controls access and use. No activation authority. Visibility: Content-minimized operational metadata; local logs may need redaction.	Editorial grouping of existing surfaces; observability architecture candidate, not a new protocol. Node observability interfaces ; Observability event scenarios .

What crosses the selection boundary. I2 returns invocation information, not permission to execute. The consumer checks the route's identity, Intent, validity and applicable policy, then supplies the credentials required at I3. A `dispatch-route-ticket:v1`, a consumer credential, a provider's registration credential,

and a relay-bind credential have different issuers, audiences and purposes. Registration credentials are not execution credentials. A route ticket is not automatically a bearer token. These artifacts cannot substitute for one another. Opt-in v2 trust and disabled admission prototypes are tracked separately in Appendix B. [Ticket boundary](#); [Core authentication](#); [Directory credentials](#).

O1 surface inventory. The observability candidate classifies provider GET /iicp/health for direct liveness/capacity/service information; Directory GET /v1/registry/stats, /v1/registry/intents and /v1/registry/nodes for authorized inventory; and local runtime-health snapshots and events.jsonl for operational observation. Inventory is not request-specific discovery. Direct health is preferred over a valid local snapshot; a log-based fallback must be labelled inferred. Logs are not a stable primary machine interface. Private/local-only configuration must not silently fall back to public Genesis. [Authority map and precedence](#); [event scenarios](#).

Exact application programming interface (API) prefixes, token behavior and implementation-specific limits belong to the source contracts and Appendix B. This inventory is sufficient to compare responsibilities, but is not a complete HTTP API specification.

4. One request through the architecture

Onboard a chat provider

Consider a provider advertising urn:iicp:intent:llm:chat:v1. Over **I1**, its operator authorizes registration and advertises the service path, endpoint and capabilities. The provider maintains heartbeat and availability information; the Directory combines these with its reachability evidence. If a replica supplies discovery, **I4** synchronization and freshness checks also affect the available view. Registration alone does not make the provider eligible for every request. [Directory](#); [state semantics](#); [federation](#).

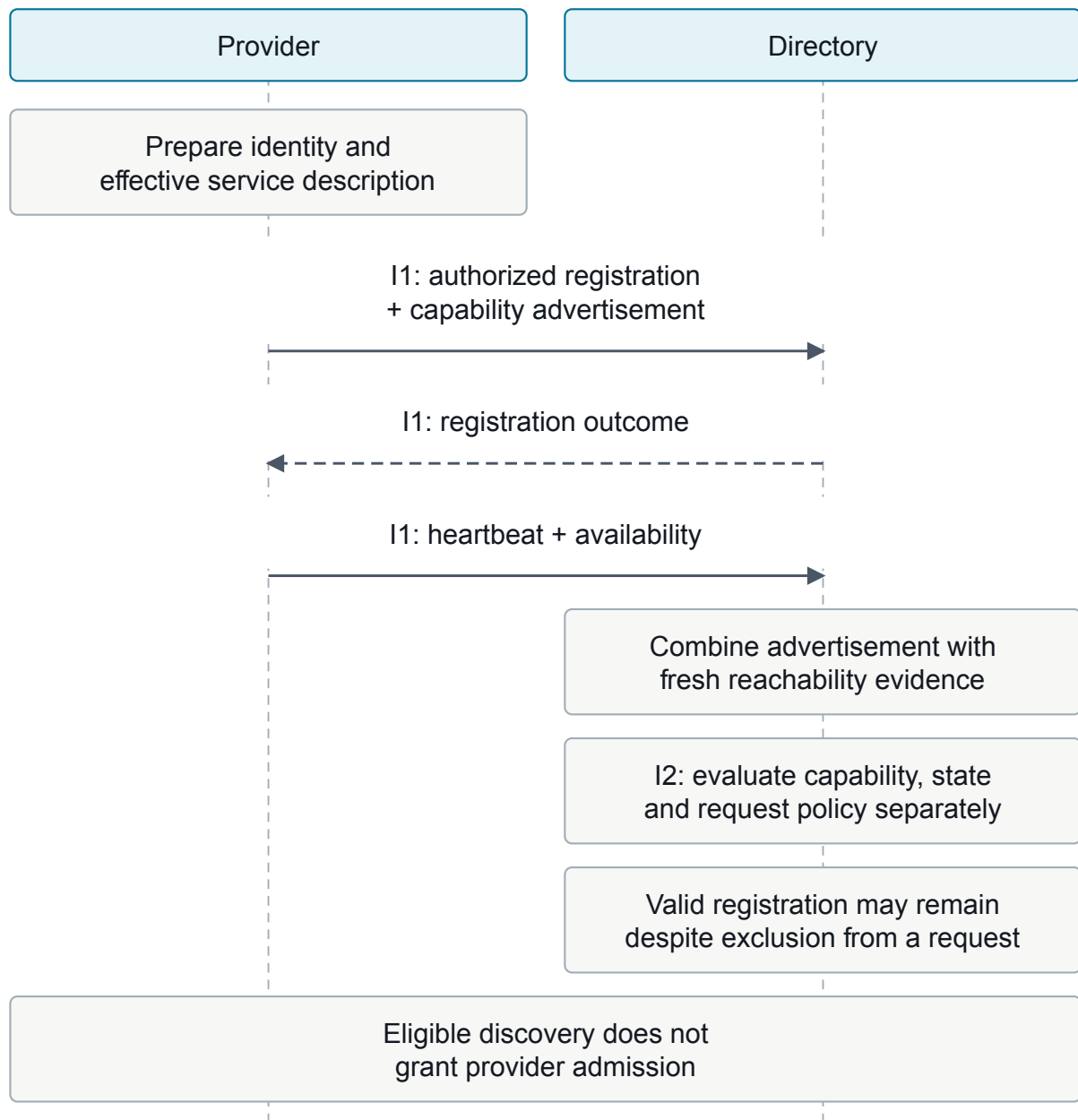


Figure 3: Provider onboarding over I1 and subsequent eligibility for I2. Notes describe local evaluation, not invented notification messages. The effective-capability additions remain parity-gated.

Select, authorize and execute

1. **Express the request.** The consumer names the chat Intent and a mandatory execution binding, applies its domain-local policy, and retains the private prompt. Intent identifies the requested capability; it does not contain the task text. [Core](#).
2. **Find an eligible path (I2).** Under the proposed effective-capability contract, all mandatory requirements must hold on one complete model/template/runtime/interface/adaptor variant. Features from different variants cannot be combined. Unknown required information fails closed; preferences rank only eligible candidates. This additive contract remains parity-gated and does not introduce a universal discovery endpoint. [Matching contract](#); [direct vectors](#).
3. **Validate the selected route (I2).** In the ticketed path, the Directory returns a concrete route and signed disclosure ticket. The consumer checks applicable issuer, audience, Intent, expiry and policy bindings and decides whether to dispatch. No prompt has crossed I2. [Dispatch implementation](#); [consumer](#).

4. **Invoke and admit (I3/I6).** The consumer sends the prompt through the selected execution binding with separate required credentials. The provider authenticates and authorizes the caller, checks its current admission conditions, then accepts or refuses. If a relay is needed, it carries execution traffic under I6 rather than turning the Directory into an executor. [Core](#); [Directory relay contract](#).
5. **Interpret the outcome (I3/O1).** A result or error returns under the chosen binding. Operational observations can help diagnose the attempt; they do not establish semantic correctness. An accepted task under another protocol retains that protocol's lifecycle and retry rules. [A2A binding](#); [observability](#).

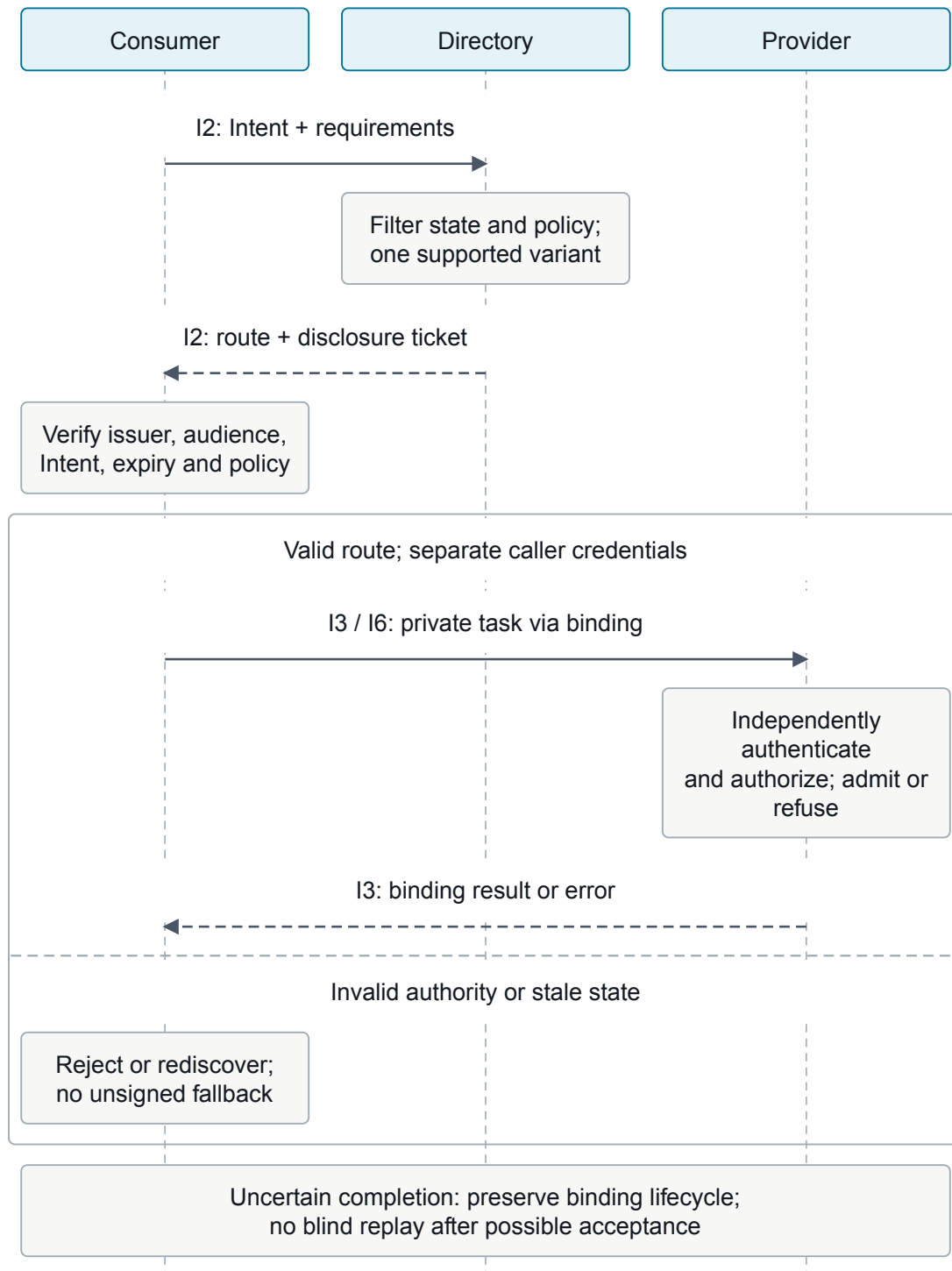


Figure 4: Ticketed chat-capability path across I2, I3 and I6. Checks are local decisions; messages describe existing logical exchanges, not a new wire format.

The same request when something changes

Branch	Decision at the affected interface
Stale state or explicit unavailability	I2 excludes the candidate or refreshes its view. A valid signature on an old record does not restore freshness. If state changes after selection, I3 can still refuse.
No matching variant or policy denial	I2 returns no eligible route or a bounded refusal; it does not silently relax the requirement.
Invalid or expired authority	The consumer rejects the affected I2 route, or the provider refuses I3. Reacquisition may be possible; unsigned fallback is not a remedy for failed verification.
Connection fails before acceptance	Retry or rediscovery is considered only under the binding's retry policy.
Completion is uncertain after possible acceptance	I3 timeout does not establish that work never ran. Preserve task/attempt identity and binding rules; do not automatically replay accepted work at another provider.

Sources: [state](#), [selection revalidation](#), [ticket boundary](#), [Core retry rules](#), [A2A lifecycle](#).

Execution timeout, delivery lifetime, logical deadline, result validity and caller wait are separate time domains. Current `constraints.timeout_ms` is a provider-attempt budget. Stopping a wait is not cancellation. The proposed lifecycle Profile separates logical `task_id` from attempt `call_id`, but does not establish universal stream/cancel endpoints, resumable streaming or exactly-once execution. [Time contract](#); [lifecycle Profile](#).

5. State, policy, trust, and authority

Directory state is not a single agent status

Axis	What it establishes	What it does not establish
Identity validity	Whether the claimed registration identity remains valid	Capability or permission to execute
Advertisement state	What service metadata is currently advertised	Proven runtime behavior
Reachability evidence	Whether and how the provider was recently reachable	Present capacity for this task
Execution availability	Provider readiness/availability indication	Eligibility under every consumer policy
Dispatch eligibility	Whether this request can use this candidate now	Provider admission or successful execution

Probes and self-reports provide different evidence. A fresh heartbeat can restore the connected view, but an explicit unavailable state still matters. Signed metadata preserves attribution and integrity, not freshness or correctness. [Directory state semantics](#).

Management state is a different model

Desired state expresses a portable proposal. **Accepted** state records the generation authorized by the local domain. **Observed** state records measurements or reports. **Effective** state reflects evidence-backed convergence under the implementation's contract. These are not synonyms for the Directory's identity,

privacy analysis: predictable inputs can make hashes informative. A Directory-only record cannot prove what the provider executed. [Ticket and receipt proposal](#); [privacy model](#).

Confidentiality and observation

The Directory's exclusion from the task path reduces exposure but does not remove metadata: capability requests, identities, endpoints, timing, and operational reports can remain sensitive. Relays and network observers introduce different visibility. The ordinary executor sees the data needed to execute; transport encryption does not create executor blindness. [Privacy adversary model](#).

The IICP confidentiality extension (IICP-CX) must be described by tier. The draft's Tier 1 encrypts requests for the provider but does **not** provide end-to-end response encryption; that is a Tier 2 target. Tier 1 also does not establish forward secrecy against compromise of the provider's long-term key. Deployment and binding support must be checked before any confidentiality claim. Neither relay opacity nor bidirectional confidentiality follows merely from selecting an IICP route. [Confidentiality draft](#).

6. Bindings, extensions, and deployment forms

Where new behavior belongs

The accepted extension architecture separates **Core**, **Profile**, **Binding**, **Registry**, and **implementation extension**. Core holds universal semantics; a Profile specifies optional interoperable behavior; a Binding maps to an execution protocol or environment; a Registry allocates identifiers and descriptions; an implementation extension controls local choices such as ranking. A new domain capability normally belongs in a registry, not in a new universal Core operation. This avoids making every deployment implement every feature. [Extension architecture](#); [identifier architecture](#).

Effective capability, optional admission/lifecycle, and trust/evidence work must retain their individual maturity labels. Cooperative inference adds coordinator/worker responsibilities for a specialized distributed-execution use case. Billing and recognition-related mechanisms are optional ecosystem concerns, not prerequisites for the provider-discovery interface being discussed with DMSC. This paper does not propose their standardization. [Effective capability](#); [cooperative inference](#); [billing extension](#).

Binding status

Model Context Protocol (MCP) and Agent2Agent (A2A) retain their own execution contracts. Transport Layer Security (TLS) protects a transport connection; it is not provider admission or an assurance of task correctness.

Surface	Architectural role and current boundary
HTTP	Maintained stable-target execution surface. Core semantics remain separate from deployment URLs.
OpenAI-compatible HTTP	An adapter/proxy-facing interface, not a second IICP Core semantic model.
Native framed TCP	Explicitly enabled, plaintext experimental path. No maintained native TLS terminator; outside stable/production scope.
QUIC	Unsupported research direction, not a current portable binding guarantee.
MCP	Draft mapping in both directions; MCP retains tool/protocol semantics.

Surface	Architectural role and current boundary
A2A	Informative pre-normative execution binding. Binds the selected card/interface/skill/version; credentials and lifecycle remain A2A responsibilities.

Sources: [transport decision](#), [Python SDK](#), [MCP binding](#), [A2A binding](#). Port 9484 is provisional in the cited project decision, not an asserted IANA assignment. This paper proposes neither a fixed port nor a new ALPN identifier.

Deployment forms and their authority

The **public mesh** uses project-operated Directory authority and eligible public participation. That is a deployment choice, not the definition of Directory. A **private or local Directory** places inventory and trust policy under another operator; it does not need an automatic public fallback. The proposed **restricted trust domain or Closed User Group** would add membership and exposure policy; membership remains distinct from execution permission. That Profile is pre-normative and explicitly not implemented; selecting a private Directory alone does not establish its fail-closed membership boundary. These are architectural forms, not claims that every SDK implements every combination equally. [Directory](#); [restricted-trust-domain proposal](#); [observability candidate](#).

Replica deployments use the specified synchronization relationship and require their own freshness and operational checks. **Direct execution** connects consumer and provider without Directory payload transit. **Relay-assisted execution** adds a payload intermediary; **browser-provider participation** depends on that reachability arrangement and remains experimental. The same host can implement several roles without erasing their authority or privacy boundaries. [Federation](#); [Directory](#); [browser node](#).

Optional external assurance

An external verifier could assess an execution under its own assurance contract and return signed evidence associated with the attempt. The conceptual sequence is route validation → separately authorized execution → external assessment → evidence verification by the relying party. This is an explanatory extension point, not an implemented IICP Core flow or a claim of full attestation. The verifier's identity, claim scope, binding to the actual input/output, and trust basis would all need specification. No verified public experiment is claimed here (G5). Potential RATS and SCITT-related evidence reuse is considered in Section 9 without assuming an IICP integration. [Existing receipt boundary](#); [independent-evidence record](#).

7. Implementation and evidence scope

The frozen ecosystem contains Python, TypeScript and Rust software development kits (SDKs) for consumer/provider integration; a PHP Directory line; a Rust Directory operator preview; an experimental browser-provider implementation; and local Management contracts, including process-local Rust client policy enforcement. These are same-project implementations. Appendix B collects exact releases, working-version differences and commit identifiers. [SDKs](#); [Directory](#); [Rust preview](#); [browser node](#); [Management integration](#).

The evidence classes answer different questions:

- A specification or architecture decision states a contract or boundary.
- Implementation source shows a concrete interpretation, not whether it is running in a particular deployment.
- A fixture defines a bounded test case; a conformance runner provides a way to execute checks. Neither is a passing result. An attributable run supports only its stated versions, configuration and profile.

- Same-project cross-language parity can expose implementation differences but is not independent interoperability.
- Release integrity binds artifacts to a release; it does not prove their behavior or security.
- External implementation or operator evidence needs an actual attributable result, not a prepared record template.

Appendix A identifies the evidence supporting each material claim. Appendix B records the runner scope, existing local baseline results and release-integrity limits. Six component API reviews and independent implementation evidence remain open; this reference does not grant qualification credit. [Pre-1.0 boundary](#); [independent record](#).

8. Relationship to DMSC

The comparison uses *draft-li-dmsc-inf-architecture-07* (22 May 2026), the frozen DMSC charter proposal, and Intent-based Agent Interconnection Protocol (IAIP), *draft-sz-dmsc-iaip-02* (25 May 2026) only where semantic distinctions matter. These are discussion sources, not evidence of IETF adoption. A functional correspondence below does not establish wire compatibility. [DMSC](#); [charter](#); [IAIP](#).

Compact comparison

Audit concern / IICP interfaces	Relationship
D01 – Gateway responsibility distribution (I1/I2/I3/I5/I6)	Correspondence with different responsibility distribution
D02 – Management, control and forwarding planes (I1/I2/I3/I4/I5/I6/O1)	Correspondence with different responsibility distribution
D03 – Identity and registration (I1/I3)	Partial/conditional correspondence
D04 – Capability advertisement and inventory (I1/I2/O1)	Partial/conditional correspondence
D05 – Intent and task semantics (I1/I2/I3)	Semantic translation required
D06 – Discovery, freshness and eligibility (I1/I2)	Partial/conditional correspondence
D07 – Selection and route disclosure (I2/I3)	Correspondence with different responsibility distribution
D08 – Authentication and provider admission (I1/I2/I3)	Partial/conditional correspondence
D09 – Execution paths and bindings (I3/I6)	Partial/conditional correspondence
D10 – Lifecycle, failure and retry (I2/I3/I6)	Partial/conditional correspondence
D11 – Intra-domain synchronization (I4)	Partial/conditional correspondence
D12 – Inter-domain policy and trust (I2/I3/I4/I5/I6)	Partial/conditional correspondence
D13 – Telemetry and bounded evidence (O1/I5)	Partial/conditional correspondence
D14 – Management state and local activation (I5/O1)	Partial/conditional correspondence

Appendix E gives each entry’s DMSC sections, IICP contracts, authority owners, pinned evidence, maturity, unresolved gap and verification method. The [JSON audit](#) and [CSV audit](#) are generated from the same records. Functional correspondence never implies wire compatibility; no DMSC–IICP exchange was executed for this comparison.

The DMSC architecture’s Agent Gateway is a logical function with flexible placement; its §7.1 also permits peer-to-peer semantic sessions after gateway coordination. IICP’s Directory does not carry task payloads, while optional relay and provider roles do. Thus “DMSC always forwards every payload through a gateway” would be an incorrect contrast. [DMSC architecture](#).

Planes, authentication, and semantics

DMSC separates management, control, and forwarding planes. IICP's discovery/control concerns partly correspond to its control plane; IICP bindings and execution paths only partly correspond to its forwarding plane. The DMSC gateway can assemble functions that IICP describes across several roles. DMSC's Agent Management Center (AMC) handles authentication/credential functions; it is not equivalent merely by name to IICP Management's local desired-state and activation model. [DMSC architecture, §§4–6; Management](#).

DMSC §5.1 explicitly excludes network interpretation of agent task semantics; its capability-based routing must not be conflated with IAIP's richer normalized semantic/vector Intent processing. Neither terminology is silently substituted for IICP's capability-address field. IAIP identifies capability-registry, authentication/admission, and intent-forwarding functions, and applies hard constraints before ranking (§8.4.2). Constraint-first selection is therefore a shared pattern, not a unique IICP feature. Mapping richer descriptions to an IICP Intent would require an explicit, versioned interpretation contract, with uncertainty and authorization kept separate from a match score. [IAIP; Core; effective capability](#).

DMSC §6.5 also distinguishes telemetry used for observation/longer-term management from real-time forwarding, with policy and privacy controls. That offers a useful comparison with IICP's metadata-only Directory, direct health/inventory surfaces, and local Management authority. DMSC §8 identifies identity spoofing, capability misrepresentation, routing manipulation, trust, and metadata exposure as security concerns while leaving detailed mechanisms outside that architecture's scope. Neither document alone closes those security questions. [DMSC; security model](#).

9. Two initial discussion proposals and reuse boundaries

These are candidate problem statements for discussion, not a request to adopt IICP or its formats. The intended outcome is agreement on one or two narrowly scoped interfaces, their authorities, and the evidence needed to compare implementations.

First: capability exposure, freshness and eligibility (I1/I2)

Problem and contributions. Determine the minimum assertion needed to expose a service without confusing advertised support with current reachability, request eligibility or provider admission. IICP contributes its complete effective-variant and state models; DMSC contributes gateway-oriented capability exposure and policy context. [Effective capability; state; DMSC charter](#).

Authority and exclusions. The advertiser owns its claim; the gateway or Directory applies exposure policy; the consumer applies request constraints; the provider retains admission. Exclude task content, ranking algorithms, a new agent-card format and any inference that advertisement grants permission or proves capacity.

First deliverable and verification. Produce a shared vocabulary and refusal table before choosing a representation. Project one pinned description from each approach into that table, then compare decisions for stale, withdrawn, unavailable and incomplete-variant cases. These are proposed checks, not completed experiments.

Decision for reviewers: which assertions, freshness bounds and refusal cases belong in a small common discovery contract, and which must remain local?

Second: selection-to-execution handoff (I2/I3/I6)

Problem and contributions. Identify the context needed to invoke a selected provider without transferring authority implicitly. IICP contributes its route-ticket, revalidation and binding boundaries; DMSC contributes gateway coordination and peer-session context. [Ticket semantics](#); [selection revalidation](#); [DMSC architecture](#), §7.1.

Authority and exclusions. Keep the route issuer's disclosure, consumer dispatch choice and provider authentication, authorization and admission separate. Retain the selected protocol's lifecycle after acceptance. Exclude a common ranking algorithm, credential substitution, universal cancellation and cross-provider replay of accepted work. [MCP binding](#); [A2A binding](#).

First deliverable and verification. Produce an authority/verification matrix covering provider and capability/interface identity, version, validity, policy, credential expectations and attempt correlation. For one agreed binding, trace a normal handoff and refusals for stale routes or invalid authority, plus uncertain completion. Record each verifier's decision without claiming wire interoperability.

Decision for reviewers: which party verifies each assertion, and what bounded result may the next party consume without inheriting authority that was never granted?

Consume existing work where the interface fits

The table is a reuse assessment, not an implemented integration or a wire-compatibility claim. Supplemental primary sources were retrieved separately from the frozen IICP baseline. A charter proposal, an individual Internet-Draft and an RFC have different standing.

Work and maturity	Potential IICP boundary	Unresolved mapping
Naming/discovery: Discovery of Agents With Names (DAWN), proposed charter 00-07; DNS for AI Discovery (DNS-AID), individual draft -02. DAWN scopes initial discovery; DNS-AID uses DNS service-binding information. DAWN scope ; DNS-AID .	Bootstrap a named service/catalogue or endpoint for I2, before request-specific eligibility.	Map identifiers, provenance, cache freshness and privacy. Initial discovery does not supply IICP's eligibility, ranking or admission semantics.
Authorization/session/security: OAuth 2.0 roles (RFC 6749) with security BCP 240 (RFC 9700); TLS 1.3 (RFC 9846, which obsoletes RFC 8446), Standards Track; AgentProto principal-binding -07 and session-requirements -02, individual drafts. OAuth roles ; OAuth security BCP ; current TLS specification ; verifier matrix ; session scope .	Assess OAuth roles/scoped access and TLS protection at I3/I6; use the principal matrix to review I2→I3 claims and session requirements for binding-owned interactions.	Define actual principals, token audience/scope, endpoint binding, freshness and failure behavior. A route ticket is not automatically an OAuth token. Session work starts after peers are located; no blanket conformance follows.
Assurance: Remote ATtestation procedureS (RATS), Informational RFC 9334; Supply Chain Integrity, Transparency, and Trust (SCITT)-related agent-action receipts, individual draft -01. RATS roles ; receipt claim limits .	Optional assurance beside I3 and evidence at O1: RATS separates attester, verifier and relying party; the receipt draft distinguishes signed action records from transparency registration.	Bind claims to the actual attempt and trust policy; specify privacy/retention. Neither automatically proves task correctness, complete inputs, prior authorization or an IICP end-to-end attestation.

Work and maturity	Potential IICP boundary	Unresolved mapping
Execution: IICP's MCP draft and A2A pre-normative binding. MCP mapping ; A2A mapping .	I6 carries selected capability/interface context into the existing execution protocol.	Pin protocol/card/skill/interface versions, obtain proper credentials, negotiate optional behavior and test lifecycle/refusal cases. Do not replay accepted work across providers by default.

I4 synchronization and O1/I5 operational evidence are secondary discussion areas. Gateway exchange could inform future root/replica or cross-domain work, but record ownership, deletion, lag and trust must be scoped first. Operational work should distinguish inventory, routing decisions, outcomes and effective policy state while minimizing content exposure. These do not require changing IICP's capability-address Intent, metadata-only Directory, or domain-local activation authority. [Federation](#); [observability](#); [local authority](#).

10. Limitations and next steps

This is an evidence-backed architecture consolidation, not a fresh implementation qualification. It does not establish independent interoperability, production readiness across deployment combinations, sovereign federation, executor blindness, universal streaming, or full execution attestation. The specific gaps and closure conditions appear once, in Appendix C; they should be resolved in their owning projects rather than silently “fixed” by this paper.

Software Defined Intelligence (SDI) remains a future, non-normative direction. The published paper's intelligence-virtualization discussion motivates describing desired capabilities and policies independently of a particular provider. Current Management planning and observation can support parts of that direction, but they do not establish a deployed general SDI control system. Future optimization, autonomous orchestration, or cross-domain activation must preserve local authority and require separate contracts and evidence. [Published report](#), §11; [Management](#).

Questions for DMSC review

1. Which responsibilities in D01–D02 are misplaced, and which role should own them?
2. Where does D05 leave ambiguity between capability-address Intent, task interpretation and IAIP intent processing?
3. Which capability and state assertions in D04–D06 could be shared, with what provenance and freshness bounds?
4. What should each party verify at the route-disclosure and provider-admission boundary in D07–D08?
5. Which of the two Section 9 experiments should come first, with what binding scope, refusal cases and observable result?

These questions seek an agreed problem boundary, not adoption of IICP formats. Section 9 and Appendix E describe candidate verification methods; they are not reports of completed experiments. Appendix D outlines a possible later architecture draft.

Appendix A. Source and evidence map

Claim-level traceability

Each identifier below denotes a bounded statement, not a conformance badge. P = published-paper foundation; S = specification/decision; I = implementation; E = fixture, test source or result. Missing roles remain explicit. Fixture presence and source inspection are not test execution; only C21 cites local checker results, retained from the v0.1.1 review. No independent result is claimed. Claim records also identify the manuscript sections and figures that use them.

Claim, scope and maturity	Supporting locations by role	Evidence limit
C01 — Discovery and execution are separate: the Directory has no task-payload role. <i>Architecture with working paths</i> . Sections 1, 2, 3; figures: components, planes.	P: Mumin (2026), IICP: Provider-Neutral Infrastructure for Intelligence Virtualization, v1.0.1: §5; Appendix D . S: Core wire format: CALL/RESPONSE ; Directory protocol: discovery and dispatch. I: Dispatch route-ticket controller: issue method (L38) ; Python consumer implementation: discovery and task submission . E: PHP route ticket tests: payload-field rejection test (L181) .	No new run result. Optional relay is a separate payload role, even if colocated.
C02 — Wire Intent is an opaque capability address, separate from the private task. <i>Working contract</i> . Sections 1, 2, 4, 8; figures: execution.	P: Mumin (2026), IICP: Provider-Neutral Infrastructure for Intelligence Virtualization, v1.0.1: §5; semantic/protocol concern . S: Core wire format: Intent identifier and payload ; Identifiers and registries: identifier roles . I: Python consumer implementation: task request construction . E: none asserted.	No new run result. No semantic/vector interoperability with IAIP is implied.
C03 — Effective capability requirements apply to one complete service variant, not a union. <i>Accepted additive pre-normative Profile; parity-gated</i> . Sections 2, 3, 4, 9; figures: onboarding, execution.	P: N/A. S: Effective service capability: Capability variants; Matching; Effective capability contract: machine-readable contract . I: Python effective capability implementation: variant matcher (L324) . E: Effective capability vectors: matching_ scenarios ; Python capability matching tests: complete-variant matching test (L61) .	No new run result. No universal new discovery endpoint or ecosystem-wide adoption claim.
C04 — Identity, advertisement, reachability, availability and dispatch eligibility are distinct axes. <i>Accepted state boundary</i> . Sections 3, 4, 5, 8, 9; figures: onboarding.	P: N/A. S: Directory state semantics: state axes and freshness; Directory state contract and scenarios: axes; views; scenarios . I: none asserted. E: Directory state contract and scenarios: successful_registration; freshness scenarios .	No new run result. Scenario presence does not demonstrate every implementation or live freshness.
C05 — Route ticket v1 permits disclosure, not provider admission or global redemption. <i>Pre-normative profile with working v1 paths</i> . Sections 1, 3, 4, 5, 9; figures: components, execution, trust.	P: N/A. S: Route tickets and receipts: v1 disclosure and recovery boundaries . I: Dispatch route-ticket controller: issue method (L38) ; Python consumer implementation: ticket verification before dispatch . E: Route ticket v1 vectors: v1 vectors ; Python ticket verification tests: fail-closed ticket test (L32) .	No new run result. Same-origin signing keys are not independent trust anchors.

Claim, scope and maturity	Supporting locations by role	Evidence limit
C06 — Trust v2 is opt-in; provider-local admission v2 is a disabled prototype. <i>Pre-normative / prototype</i> . Sections 3, 5, 7.	P: N/A. S: Dispatch ticket trust v2: runtime adoption boundary ; Dispatch admission v2: disabled provider-local prototype . I: none asserted. E: none asserted.	No new run result. No ordinary-discovery default, global redemption or whole-store rollback guarantee.
C07 — Provider execution requires its own authorization and admission decision. <i>Core contract; optional admission Profile</i> . Sections 3, 4, 5; figures: execution, trust.	P: Mumin (2026), IICP: Provider-Neutral Infrastructure for Intelligence Virtualization, v1.0.1: §5, execution ; Appendix F authority . S: Core wire format: §3.1; authentication ; Directory protocol: §3.12 consumer credentials ; Provider admission Profile: provider refusal . I: Python consumer implementation: consumer credential handling . E: none asserted.	No new run result. Route tickets, provider tokens, consumer tokens and binding credentials differ.
C08 — Federation uses a Genesis root, signed snapshots and replica event-tail synchronization. <i>Draft; Rust operator preview</i> . Sections 3, 6, 8, 9; figures: components.	P: Mumin (2026), IICP: Provider-Neutral Infrastructure for Intelligence Virtualization, v1.0.1: §6 . S: Federated Directory: root/replica synchronization ; Context and service event ownership: event ownership . I: Rust Directory: synchronization surfaces . E: none asserted.	No new run result. Not sovereign multi-root federation, automatic trust or Genesis cutover authority.
C09 — Caller wait, execution timeout, delivery lifetime, deadline and result validity differ. <i>Accepted time boundary; optional lifecycle proposed</i> . Sections 4; figures: execution.	P: N/A. S: Task time semantics: five time domains ; Task time contract: contract ; Service lifecycle Profile: task and attempt lifecycle . I: none asserted. E: none asserted.	No new run result. Stopping a wait does not prove cancellation; no universal replay or exactly-once guarantee.
C10 — Management keeps desired, accepted, observed and effective state separate under local authority. <i>Working local implementation contracts</i> . Sections 2, 3, 5, 7, 8, 9; figures: components, trust.	P: Mumin (2026), IICP: Provider-Neutral Infrastructure for Intelligence Virtualization, v1.0.1: §7; Appendix F . S: Management local authority: exact-plan acceptance and bounded application ; Management architecture context map: authority map . I: Management local apply gate: exact-plan gate (L88) ; Management Rust client enforcement: ManagedIicpClient (L125) . E: Management routing enforcement tests: process-local enforcement tests .	No new run result. Acceptance is not application; not a deployed general remote service.
C11 — Bindings retain their execution semantics; relays are optional payload intermediaries. <i>Draft bindings; topology choice</i> . Sections 2, 3, 4, 6, 9; figures: components, execution.	P: Mumin (2026), IICP: Provider-Neutral Infrastructure for Intelligence Virtualization, v1.0.1: §5, execution concern . S: Environmental independence and extensions: Binding role ; Directory protocol: relay and relay-bind tickets . I: Browser node: experimental browser-provider relay path . E: none asserted.	No new run result. Relay-bind credentials are not route-disclosure tickets.

Claim, scope and maturity	Supporting locations by role	Evidence limit
C12 — MCP and A2A bindings do not replace those protocols' task, credential or lifecycle rules. <i>MCP draft; A2A pre-normative</i> . Sections 3, 6, 9.	P: N/A. S: MCP binding: two mapping directions; A2A execution binding v0: card/interface/skill binding; authorization and retry. I: none asserted. E: none asserted.	No new run result. No wire interoperability or safe cross-provider replay inferred from a mapping document.
C13 — Health, inventory and local observations are different from request-specific discovery. <i>Observability architecture candidate</i> . Sections 3, 4, 8, 9; figures: components.	P: N/A. S: Node observability interfaces: Authority map; Source and precedence. I: none asserted. E: Observability event scenarios: event classification and no-public-fallback scenarios.	No new run result. O1 is an editorial inventory, not a new endpoint, wire Profile or uniform implementation claim.
C14 — Native framed TCP is explicitly enabled plaintext experimentation; QUIC is unsupported research. <i>Current transport decision</i> . Sections 6.	P: N/A. S: Transport, binding and port decision: amended August 28 boundary. I: Python SDK: native transport status. E: none asserted.	No new run result. No maintained native TLS terminator or stable/production claim; port provisional.
C15 — Receipts are bounded signer claims, not correctness or full execution attestation. <i>Pre-normative evidence proposals</i> . Sections 2, 5, 6, 9; figures: trust.	P: Mumin (2026), IICP: Provider-Neutral Infrastructure for Intelligence Virtualization, v1.0.1: Appendices E–G. S: Route tickets and receipts: split decision/consumer/provider evidence; Privacy adversary and trust model: evidence and privacy boundaries. I: none asserted. E: none asserted.	No new run result. No verified Correctover experiment located in the frozen search; G5 remains open.
C16 — CX Tier 1 encrypts requests, not end-to-end responses or against the executor. <i>Draft confidentiality tiers</i> . Sections 5, 6.	P: Mumin (2026), IICP: Provider-Neutral Infrastructure for Intelligence Virtualization, v1.0.1: §5 trust concern only; not tier-specific. S: IICP confidentiality: Tier 1 limits; Tier 2 target; Privacy adversary and trust model: executor and relay visibility. I: none asserted. E: none asserted.	No new run result. No forward secrecy against provider long-term-key compromise asserted.
C17 — Five planes are architectural concerns; Core, Profile, Binding and Registry have distinct jobs. <i>Published concerns; accepted extension boundary</i> . Sections 2, 6; figures: planes.	P: Mumin (2026), IICP: Provider-Neutral Infrastructure for Intelligence Virtualization, v1.0.1: §5. S: Environmental independence and extensions: extension responsibilities; Identifiers and registries: registry authority. I: none asserted. E: none asserted.	No new run result. Diagrams are explanatory, not new normative layers or maturity claims.
C18 — Private Directory selection is not the unimplemented restricted-trust-domain membership contract. <i>Deployment choice; pre-normative Profile unimplemented</i> . Sections 6.	P: N/A. S: Restricted trust domain v0: status and membership boundary; Node observability interfaces: private/local-only configuration. I: none asserted. E: none asserted.	No new run result. No automatic public fallback or full membership isolation can be inferred.

Claim, scope and maturity	Supporting locations by role	Evidence limit
C19 — Published releases, working manifests and deployment are separate evidence axes. <i>Frozen source/release observations</i> . Sections 1, 7.	P: Mumin (2026), IICP: Provider-Neutral Infrastructure for Intelligence Virtualization, v1.0.1: §14 historical evidence. S: Specification status and authority: authority rules; Current-version projection: release projection. I: Python SDK: working SDK; TypeScript SDK: working SDK; Rust SDK: working SDK; PHP Directory: working Directory; Rust Directory: preview; Browser node: experimental provider; Management: local contracts. E: none asserted.	No new run result. Exact revisions and releases in Appendix B; no fresh deployment attestation.
C20 — Standalone conformance and release-integrity tools have bounded, different purposes. <i>Tools and tests present; not run here</i> . Sections 7.	P: N/A. S: Standalone conformance runner scope: profiles and loopback-only restrictions; Conformance test suite: profile-scoped requirements. I: Standalone runner implementation: runner implementation; Release integrity checker: manifest/hash gate. E: Standalone runner tests: runner tests; Release integrity manifest: artifact-integrity inventory.	No new run result. Fixtures and integrity manifests are not passing behavioral or security results.
C21 — Baseline check is valid, with 28 families, six open reviews and strict freeze OPEN. <i>Observed local structural-check result</i> . Sections 7.	P: N/A. S: Pre-1.0 qualification boundary: strict freeze boundary; Pre-1.0 feature baseline: feature baseline. I: Pre-1.0 baseline checker: main validation. E: local run record.	Local structural results: pre1-default exit 0 ; pre1-strict exit 1 . Default exit 0 and strict exit 1 are expected; no qualification credit.
C22 — The frozen clean-room interoperability record is a blank external-implementer template. <i>Result absent</i> . Sections 7.	P: Mumin (2026), IICP: Provider-Neutral Infrastructure for Intelligence Virtualization, v1.0.1: §14; Appendix G, evidence distinction. S: SDK quality evidence: same-project versus independent evidence. I: none asserted. E: Clean-room interoperability record: <code>result_present: false</code> .	No new run result. Same-project SDK languages and self-review are not independent interoperability.
C23 — DMSC has management/control/forwarding concerns and permits peer-to-peer sessions after coordination. <i>Individual draft -07; editorial correspondence</i> . Sections 1, 8.	P: N/A. S: DMSC infrastructure architecture, draft-li-dmsc-inf-architecture-07: §§3, 5.1, 6.5, 7.1, 8 ; DMSC charter proposal: five proposed concerns. I: none asserted. E: none asserted.	No new run result. Agent Gateway is not a Directory alias; AMC is not IICP Management.
C24 — IAIP uses richer intent processing and also applies hard constraints before ranking. <i>Individual draft -02</i> . Sections 8.	P: N/A. S: IAIP, draft-sz-dm-sc-iaip-02: §8.4.2; intent processing . I: none asserted. E: none asserted.	No new run result. No unique constraint-first claim or automatic semantic mapping to IICP.
C25 — Discovery work may supply bootstrap metadata without replacing IICP eligibility. <i>Reuse hypothesis, not an implemented binding</i> . Sections 9.	P: N/A. S: DAWN charter, charter-ietf-dawn-00-07: scope and exclusions, revision 00-07 ; DNS-AID, draft-mozleywilliams-dnsop-dnsaid-02: Abstract; discovery model . I: none asserted. E: none asserted.	No new run result. Mapping, freshness and policy rules unresolved.

Claim, scope and maturity	Supporting locations by role	Evidence limit
C26 — Existing authorization/transport work and AgentProto drafts can inform the execution handoff. <i>Reuse hypothesis; RFCs versus individual drafts.</i> Sections 9.	P: N/A. S: OAuth 2.0 authorization framework , RFC 6749: §§1.1, 1.4; TLS 1.3 , RFC 8446: §1 <i>security properties</i> ; Agent principal binding , draft-bu-agentproto-security-principal-binding-07: §§6, 10, 14–17; Agent session requirements , draft-feng-agentproto-session-requirements-02: §§2, 5.1–5.11; Best Current Practice for OAuth 2.0 Security , RFC 9700: §2 and update scope; The Transport Layer Security (TLS) Protocol Version 1.3 , RFC 9846: §1 and RFC 8446 replacement. I: none asserted. E: none asserted.	No new run result. No automatic OAuth-token, principal-matrix or session conformance asserted.
C27 — RATS and SCITT-related receipts offer different optional evidence functions. <i>Reuse hypothesis; informational RFC and individual draft.</i> Sections 6, 9.	P: N/A. S: Remote ATtestation procedureS architecture , RFC 9334: §§4–5 <i>roles and artifacts</i> ; AI Agent Action Receipts , draft-noa-scitt-ai-agent-receipt-01: Abstract; security/claim boundaries . I: none asserted. E: none asserted.	No new run result. No IICP attestation/receipt binding, correctness guarantee or deployed integration established.
C28 — Software Defined Intelligence is a future direction, not a deployed general IICP system. <i>Future / non-normative.</i> Sections 10.	P: Mumin (2026) , IICP: Provider-Neutral Infrastructure for Intelligence Virtualization , v1.0.1: §11 <i>intelligence virtualization</i> . S: Management: planning and authority limits . I: none asserted. E: none asserted.	No new run result. Broader orchestration and activation need separate contracts and evidence.

Source catalogue

All repository links below are pinned to the frozen commit, except the explicitly identified release-tag source. Status labels describe the cited item; they do not promote adjacent documents. The machine-readable registry also records content hashes and source paths.

ID / source	Status
S01 — Core wire format	working draft 1.3.5
S02 — Intent and selection semantics	working specification
S03 — Directory protocol	working specification
S04 — Capability envelope	working specification
S05 — Effective service capability	accepted pre-normative Profile; parity-gated
S06 — Directory state semantics	accepted architecture boundary
S07 — Task time semantics	accepted boundary; additive delivery deferred
S08 — Environmental independence and extensions	accepted architecture boundary
S09 — Identifiers and registries	accepted project boundary; external registration pending
S10 — Node observability interfaces	architecture candidate
S11 — Context and service event ownership	architecture decision
S12 — Privacy adversary and trust model	project security analysis
S13 — IICP confidentiality	draft; tier-specific
S14 — Service lifecycle Profile	proposed optional Profile
S15 — Provider admission Profile	proposed optional Profile

ID / source	Status
S16 – Federated Directory	draft; Genesis and replicas
S17 – MCP binding	draft binding
S18 – A2A execution binding v0	informative pre-normative binding
S19 – Route tickets and receipts	fixture-gated pre-normative
S20 – Dispatch ticket trust v2	pre-normative opt-in
S21 – Dispatch admission v2	disabled prototype
S22 – Transport, binding and port decision	decision amended August 28
S23 – Protocol positioning	informative
S24 – Protocol comparison, September 25	informative
S25 – Selection trust and revalidation	informative candidate boundary
S26 – Pre-1.0 qualification boundary	strict freeze OPEN
S27 – Pre-1.0 feature baseline	non-authorizing qualification crosswalk
S28 – Clean-room interoperability record	blank external template; no result
S29 – Conformance test suite	test specification; profile-scoped
S30 – SDK quality evidence	evidence methodology
S31 – Specification status and authority	project status policy
S32 – Current-version projection	published version projection
S33 – Existing IETF candidate and boundaries	project candidate; not this submission
S34 – Standards readiness	readiness assessment
S35 – Cooperative inference	optional specialist specification
S36 – Billing extension	optional extension
S37 – Restricted trust domain v0	pre-normative Profile; not implemented
S38 – Operator identity slot	optional identity mechanism
S39 – Core at published suite v1.10.17	immutable release content; check document status
S40 – Python SDK	working implementation documentation; not a deployment attestation
S41 – TypeScript SDK	working implementation documentation; not a deployment attestation
S42 – Rust SDK	working implementation documentation; not a deployment attestation
S43 – PHP Directory	working implementation documentation; not a deployment attestation
S44 – Rust Directory	working implementation documentation; not a deployment attestation
S45 – Browser node	working implementation documentation; not a deployment attestation
S46 – Management	working implementation documentation; not a deployment attestation
S47 – Dispatch route-ticket controller	working implementation
S48 – Python consumer implementation	working implementation
S49 – Management local authority	working implementation contract
S50 – Management architecture context map	working architecture
S51 – DMSC charter proposal	repository proposal; not an adoption assertion
S52 – Mumin (2026), IICP: Provider-Neutral Infrastructure for Intelligence Virtualization, v1.0.1	published report, 25 August 2026
S53 – DMSC infrastructure architecture, draft-li-dmsc-inf-architecture-07	individual Internet-Draft, 22 May 2026
S54 – IAIP, draft-sz-dmsc-iaip-02	individual Internet-Draft, 25 May 2026
S55 – Standalone conformance runner scope	preview runner; bounded profiles
S56 – Standalone runner implementation	working implementation; not executed for this revision
S57 – Standalone runner tests	test source inspected; not executed

ID / source	Status
S58 – Release integrity checker	artifact-integrity gate; not behavior qualification
S59 – Release integrity manifest	working manifest; not a new release
S60 – Effective capability contract	accepted additive pre-normative contract
S61 – Effective capability vectors	fixture present; not a result
S62 – Directory state contract and scenarios	accepted semantic boundary; scenarios not a run
S63 – Task time contract	accepted semantic boundary
S64 – Observability event scenarios	candidate fixture; not a result
S65 – Route ticket v1 vectors	v1 fixture; not admission or revocation proof
S66 – Pre-1.0 baseline checker	structural checker; no qualification promotion
S67 – Python effective capability implementation	working matching implementation
S68 – Python capability matching tests	test source inspected; not executed
S69 – Python ticket verification tests	test source inspected; not executed
S70 – PHP route ticket tests	test source inspected; not executed
S71 – Management local apply gate	working validation and preview implementation
S72 – Management Rust client enforcement	working process-local integration
S73 – Management routing enforcement tests	test source inspected; not executed
S74 – DNS-AID, draft-mozleywilliams-dnsop-dnsaid-02	individual Internet-Draft, 27 May 2026
S75 – AI Agent Action Receipts, draft-noa-scitt-ai-agent-receipt-01	individual Internet-Draft, 15 August 2026
S76 – Agent principal binding, draft-bu-agentproto-security-principal-binding-07	individual Internet-Draft, 15 September 2026
S77 – Agent session requirements, draft-feng-agentproto-session-requirements-02	individual Internet-Draft, 20 August 2026
S78 – DAWN charter, charter-ietf-dawn-00-07	charter revision 00-07; scope proposal, not a protocol
S79 – Remote ATtestation procedureS architecture, RFC 9334	Informational RFC, January 2023
S80 – OAuth 2.0 authorization framework, RFC 6749	Standards Track RFC, October 2012; framework, not an IICP profile
S81 – TLS 1.3, RFC 8446	Standards Track RFC, August 2018
S82 – The Transport Layer Security (TLS) Protocol Version 1.3, RFC 9846	PROPOSED STANDARD, July 2026
S83 – Best Current Practice for OAuth 2.0 Security, RFC 9700	BEST CURRENT PRACTICE, January 2025

Appendix B. Release snapshot and publication delta

Frozen repositories and published releases

Repository	Frozen commit	Published release
IICP	e1f0c69ac844	v1.10.17
iicp-client-python	508b607dd542	v0.7.109
iicp-client-typescript	4a7f3198f2bb	v0.7.109
iicp-client-rust	0b8b52926297	v0.7.109
iicp-directory-php	5a84dc2e0d30	v1.10.93
iicp-directory-rust	47277fad5dad	v0.1.15
iicp-web-node	96b5a8786ff2	v0.2.6
iicp-management	9b9c910fe644	v0.11.0 (prerelease)
dmisc-charter	cabbe6ba8321	Not a release inventory

The refresh completed at 11:04 UTC on 29 September 2026 found all nine repository HEADs, published release observations, seven referenced draft/charter revisions and the latest Zenodo edition unchanged. [Timestamped source refresh](#). The adjacent-reference check identified RFC 9846 as the successor to RFC 8446 and retained RFC 9700 alongside the OAuth role model; neither changes the IICP implementation baseline. Full commit identifiers and retrieval timestamps are supplied in the [source freeze](#); the public Markdown links resolve to those same revisions. Release numbers, wire-compatibility lineage, internal document versions, and implementation manifests are separate axes. Management v0.11.0 is a prerelease; an API that returns only non-prerelease “latest” releases does not report it.

Implementation version projection

Role	Implementation at the source cutoff	Authority and maturity boundary
Protocol contracts	IICP suite v1.10.17 published; working Core text 1.3.5	Immutable suite authority is separate from later working drafts.
Consumer/provider libraries	Python, TypeScript, Rust SDKs: published v0.7.109; working manifests 0.7.110	Same-project implementations; not independent implementers.
Directory	PHP: published v1.10.93; working runtime documentation 1.10.94	Current Genesis implementation line; no new live-deployment attestation here.
Directory	Rust v0.1.15	Operator preview; implementation is not a Genesis authority transfer.
Browser node	v0.2.6	Optional browser implementation; provider path experimental.
Management	v0.11.0 prerelease	Domain-local implementation/preview contracts, not a deployed remote administration service.

Sources: [published-version projection](#); [Python](#), [TypeScript](#), [Rust SDK](#), [PHP](#), [Rust Directory](#), [browser node](#), [Management](#). The frozen-repositories table gives exact revisions and release links. A manifest ahead of a published release is not itself an inconsistency.

Implementation-specific interface notes

Boundary	Current detail; not a universal interface guarantee
I2 ticket issuance	PHP POST <code>/api/v1/dispatch/ticket</code> rejects prompt, messages, payload, input, chat, content and response. Client base URLs may already contain <code>/api</code> ; Rust exposes documented aliases. Controller ; Python ; Rust Directory .
I2 trust / I3 admission	V1 grants route disclosure, without stateful single-use redemption or signer revocation. V2 trust has caller-pinned material and explicit opt-in verification; provider-local atomic admission v2 is disabled. A local replay cache is not global redemption, and whole-store rollback detection needs a protected reference or recovery process. Ticket ; trust v2 ; admission v2 .
I3 credential split	Legacy <code>auth.node_token</code> concerns the provider's configured token. <code>X-IICP-Consumer-Token</code> is separately signed and bound to provider/audience, Intent and expiry; current issuance requires authenticated node-token authority. Binding-specific credentials remain separate. Core §3.1 ; Directory §3.12 .
I4 state freshness	The current state decision uses a 90-second heartbeat threshold for the connected view; explicit unavailability still matters. That threshold is not an Internet-wide constant. State contract .
I5 local enforcement	Exact-plan authorization binds scope, generation, audience, expiry and replay controls. Owner-restricted local interfaces and bounded adapters do not provide arbitrary shell or general restart. <code>ManagedIicpClient</code> narrows policy within one process with generation checks and five-minute authority expiry. Local authority ; apply gate ; integration .
I6 execution binding	A2A selection binds the exact card digest, interface, skill and version. The route ticket is not an A2A bearer credential; acceptance retains A2A lifecycle rules. Binding .

Qualification and result scope

The [standalone runner](#), [runner implementation](#) and [runner tests](#) provide direct entry points to bounded profiles. Ticket [vectors](#), Python [verifier tests](#) and PHP [payload-exclusion tests](#) are source evidence, not newly observed passing results. The dispatch and lifecycle black-box profiles require disposable loopback targets because they can issue routes or alter registration state; they were not run for this paper. The [release-integrity checker](#) and [manifest](#) address artifact consistency, not behavioral qualification. Appendix A records these distinctions for each claim.

The v0.1.1 review ran the frozen pre-1.0 baseline checker on 29 September and recorded “valid,” 28 capability families, six open component reviews, and strict freeze **OPEN**. Default mode exited 0; strict mode exited 1, as the source documentation expects. This validates the pre-1.0 feature-baseline structure while confirming that its strict gate remains open; it does not validate the DMSC audit crosswalk. It grants no qualification credit or coordinated stable designation. The frozen clean-room record says `result_present: false`; it remains a blank external-implementer template. [Pre-1.0 boundary](#); [baseline](#); [clean-room record](#).

What changed in the account since August

Published foundation	Current refinement or limit	Editorial treatment
Five-plane architecture and provider-neutral invocation	Component/interface view needed for DMSC discussion	Retain the concern model; add the explicit mapping, not a new normative layer stack.
Capability discovery	Effective-service variants and state/time boundaries now explicit in decisions	Cite working accepted/pre-normative contracts without claiming universal adoption.
Direct execution and confidentiality	Current CX tier and native-transport restrictions need precise wording	State request-only Tier 1 encryption and experimental plaintext native path.
Management authority framework	Exact-plan local contracts and working process-local routing enforcement	Separate implementation presence, release status, and remote-deployment claims.
Federation architecture	Current root/replica ownership and freshness limits	Avoid equating replicas with sovereign multi-root trust.
Validation and ecosystem evidence	September baseline still has six open API reviews; independent record blank	Keep historical evidence historical; report current gate without promotion.

Sources: [published report](#), [effective capability](#), [state](#), [time](#), [CX](#), [transport](#), [Management](#), [federation](#), [pre1](#), [independent record](#). This is a reconciliation of scope and maturity, not a claim that every refinement was first introduced after publication.

Appendix C. Gap report

The gaps below are review items. No protocol code, support matrix, candidate, deployment, or qualification credit was changed to close them.

ID / gap	Current finding	Evidence required for closure
G1 – Consolidated interface specification	Responsibilities are now mapped, but this paper is not a complete wire/API definition. Deployment prefixes and optional Profiles differ.	A defined release/Profile scope with exact schemas, errors and fixtures. Directory ; status .
G2 – Release versus working behavior	Working Core and SDK/PHP manifests are ahead of published versions; accepted architecture decisions can still be pre-normative.	A release-scoped behavior inventory; frozen main behavior remains labelled working until released. Versions ; effective capability .
G3 – Coordinated qualification	Baseline is structurally valid; six component API reviews and strict freeze remain open.	Completion of the existing qualification process with attributable results; this document grants no credit. Pre1 ; baseline .
G4 – Independent implementation	Frozen clean-room record is a blank template, not a result. Same-project SDK parity is insufficient.	A public bounded, reproducible independent result with versions and participant role. Record .
G5 – Correctover/CCS evidence	No public IICP experiment record was located in the frozen repositories, related searches, or inspected external-evidence page. This is a search limitation, not proof no experiment exists.	A public version-pinned result with its assurance boundary; the unsupported experiment remains omitted.
G6 – Trust/admission maturity	v1 is route disclosure; v2 trust is opt-in; provider admission v2 is a disabled prototype.	Evidence for keys, replay, rollback/recovery, provider integration and parity before stronger claims. Tickets ; trust v2 ; admission v2 .
G7 – Federation scope	Root/replica synchronization is described; sovereign multi-root policy/trust exchange is not demonstrated.	An agreed domain model with ownership, freshness, revocation and failure behavior. Federation .
G8 – Lifecycle and execution bindings	Proposed Profiles do not imply universal stream/resume/cancel or safe cross-provider replay.	Specified mappings and adverse-case results at pinned binding/Profile versions. Lifecycle ; A2A .
G9 – Confidentiality and transport	Tier 1 response confidentiality and native production TLS cannot be inferred.	Separate tier/binding claims supported by configuration and deployment evidence. CX ; transport .
G10 – Management and SDI	Local planning/activation and working routing enforcement do not prove a remote general SDI service.	Separate qualification of any broader service; current adapter and authority limits remain. Management ; local authority .
G11 – External registration and submission	Project identifiers/port conventions and candidate drafts are not IANA allocation or IETF acceptance.	A documented registration need, change authority, succession and submission scope. Identifiers ; transport ; standards readiness .
G12 – DMSC scope agreement	Functional correspondences are editorial analysis; no joint interface or endorsement has been agreed.	Agreement on one or two interfaces from Section 9 and a recorded problem boundary. DMSC ; charter .

Appendix D. Path to an IETF architecture draft

1. **Review this interim reference.** Record DMSC feedback on the component/interface mapping and correct document defects; preserve project-maturity gaps as explicit limitations. G5 need not block discussion if the unsupported experiment claim remains omitted.
2. **Choose the document's problem boundary.** Decide whether the future draft describes the overall architecture or a narrower selection/handoff interface. Do not bundle billing, cooperative inference, SDI, every execution protocol, and operational implementation detail into one submission.
3. **Align terminology without erasing differences.** Define Intent, capability, gateway, Directory, candidate, route, authorization, and evidence. Preserve the difference between capability identifiers and semantic intent processing. Record correspondences as mappings, not synonyms.
4. **Separate architecture from protocol requirements.** Keep diagrams and role responsibilities informative. Use BCP 14 only for actual interoperable requirements in the selected scope, backed by formal object/interface definitions, versioning rules, error handling, and tests. Reference the existing project candidate rather than silently replacing it. [Candidate draft](#).
5. **Write security and privacy considerations around trust boundaries.** Cover spoofing, capability misrepresentation, stale state, route substitution, credential/ticket scope, replay, rollback, relay visibility, metadata leakage, and local authority. Identify assumptions and residual risks. Do not turn a receipt into a correctness claim.
6. **Determine IANA needs.** Evaluate namespace/registry requirements only after scope is fixed. Do not request a port or ALPN identifier without a demonstrated binding requirement. Resolve change control and succession. [Identifier decision](#); [transport decision](#); [standards readiness](#).
7. **Move implementation snapshots out of normative text.** Retain a short implementation-status section and pinned evidence references. Remove project-specific deployment inventory, build workflow, and internal review details from the architecture specification.
8. **Secure external review and follow the submission process.** Obtain the relevant author/editor approvals, DMSC/IETF participant review, and additional independent evidence appropriate to the claims. Submission, adoption, and eventual standardization remain different events.

A candidate outline is: Introduction and scope; Terminology; Architectural roles and assumptions; Component interfaces and information flows; Capability exposure and selection; Authorization and trust boundaries; Execution bindings; State and synchronization; Operational considerations; Security considerations; Privacy considerations; IANA considerations if needed; Implementation status; References. A conformance appendix belongs only where a defined protocol surface can actually be tested.

The immediate discussion can start before every implementation matures, provided unresolved behavior remains explicit and no qualification or adoption claim is implied.

Appendix E. Versioned DMSC/IICP audit crosswalk

This informative audit uses D01–D14 as stable review identifiers. Section 8 is its compact index. Each record maps a bounded concern rather than scoring an entire architecture. **Aligned** means correspondence only within the stated dimension; **partial** means overlap with unresolved differences; **distinct** means materially different semantics or authority; **absent** requires a bounded absence finding; **not evaluated** makes no assessment. No aggregate alignment percentage is calculated.

Relationship categories are: Direct functional correspondence; Correspondence with different responsibility distribution; Partial/conditional correspondence; Semantic translation required; No demonstrated correspondence; Out of scope. Unused categories remain available for later revisions. A direct functional correspondence is not a protocol-equivalence claim.

All DMSC architecture references are to the individual infrastructure draft -07, not the similarly named content-semantic architecture draft. IAIP -02 is cited only where relevant. Source hashes and full revisions are resolved against the [source registry](#); the [JSON](#) and [CSV](#) exports carry the same audit records. Dimension evidence keys below resolve through the source registry; repeated keys within a record are listed once where their evidence sets coincide. Full revisions and content hashes are retained in the exports. Verification remains proposed, not executed. A test file is not a passing result. No entry reports a joint or independent interoperability run.

D01. Gateway responsibility distribution

DMSC basis: [draft-li-dmsc-inf-architecture-07 §3](#); [draft-li-dmsc-inf-architecture-07 §5.1](#); [draft-li-dmsc-inf-architecture-07 §7.1](#).

IICP scope: Directory, Consumer, Provider, Management, optional Relay; I1/I2/I3/I5/I6. Objects/contracts: registration, route disclosure, execution binding, local policy. Claims: C01, C05, C10, C11, C23.

Relationship: Correspondence with different responsibility distribution. An IICP Directory owns discovery metadata, not the complete gateway function. Payload transport is a separate provider/binding/optional Relay responsibility.

Authority/trust owner: Directory: discovery and route statement; consumer: dispatch decision; provider: admission; local operator: activation.

Maturity: Core/Directory contracts with working PHP controller; relay optional; Management local. Role mapping is informative.

Dimension	Assessment and rationale
Functional responsibility	aligned — Both accounts separate coordination responsibilities from the agent's task logic.
Semantics	partial — The gateway concept spans more responsibilities than a Directory.
Authority/trust	distinct — IICP separates route issuer, dispatching consumer and admitting provider.
State/lifecycle	partial — Role correspondence does not establish a shared lifecycle.
Execution/binding	partial — Direct and mediated paths need a selected binding.
Evidence/maturity	partial — Support is limited to the listed contracts and source artifacts; no joint implementation result is registered.
Wire compatibility	not evaluated — No common wire contract or joint exchange result is supplied; functional correspondence is not wire compatibility.

Dimension evidence: functional responsibility, semantics, authority/trust, state/lifecycle, execution/binding: [dmsc](#), [directory](#), [tickets](#), [extension](#); evidence/maturity: [directory](#), [tickets](#), [extension](#), [dispatch](#), [dispatchtest](#), [independent](#); wire compatibility: [dmsc](#), [independent](#).

Specification: Directory protocol: REGISTER; Route tickets and receipts: [bounded evidence](#) and [v1/v2 split](#); Environmental independence and extensions: Core invariants and extension roles.

Implementation: Dispatch route-ticket controller: [ticket controller](#) and [payload exclusion](#).

Test source: PHP route ticket tests: [dispatch test source](#).

Frozen revisions: IICP: [e1f0c69ac844](#); iicp-directory-php: [5a84dc2e0d30](#). Full identifiers and content hashes are in the JSON audit and source registry.

Interoperability: No DMSC–IICP interoperability run. Same-project SDK parity is not an independent result. [Independent-record boundary](#).

Unresolved gap: G1, G12: no jointly agreed gateway-to-role allocation.

Closure condition: Agree a responsibility table and payload visibility for one deployment, without equating role names.

Verification method: Trace a request across I2 and I3; inspect whether any task payload enters the Directory. Test sources are pointers, not a run.

Execution status: proposed; not executed; no result reference. The comparison is source review, not an executed joint experiment.

D02. Management, control and forwarding planes

DMSC basis: [draft-li-dmsc-inf-architecture-07 §5.1](#).

IICP scope: Directory, Consumer, Provider, Management, optional Relay; I1/I2/I3/I4/I5/I6/O1. Objects/contracts: five concerns, logical interface families. Claims: C01, C10, C17, C23.

Relationship: Correspondence with different responsibility distribution. The published five concerns are not five sequential wire layers. Trust cuts across all roles; execution bindings are not a universal forwarding protocol.

Authority/trust owner: Authority remains attached to each component; placement in a plane grants no additional permission.

Maturity: Architectural classification supported by published and working contracts, not a separately deployable plane stack.

Dimension	Assessment and rationale
Functional responsibility	aligned — Both separate policy, coordination and traffic concerns.
Semantics	partial — Five IICP concerns and three DMSC planes are different decompositions.
Authority/trust	distinct — Local authority cannot be inferred from a plane name.
State/lifecycle	not evaluated — A plane classification does not define common state transitions.
Execution/binding	partial — Bindings implement execution; no common forwarding format is defined.
Evidence/maturity	partial — Support is limited to the listed contracts and source artifacts; no joint implementation result is registered.
Wire compatibility	not evaluated — No common wire contract or joint exchange result is supplied; functional correspondence is not wire compatibility.

Dimension evidence: functional responsibility, semantics, authority/trust, state/lifecycle, execution/binding: [dmsc](#), [extension](#), [authority](#); evidence/maturity: [extension](#), [authority](#), [independent](#); wire compatibility: [dmsc](#), [independent](#).

Specification: [Environmental independence and extensions: Core invariants and extension roles](#); [Management local authority: exact-plan acceptance and application boundary](#).

Implementation: No implementation is asserted for this complete mapping; the cited contracts alone do not establish runtime support.

Frozen revisions: IICP: [e1f0c69ac844](#); iicp-management: [9b9c910fe644](#). Full identifiers and content hashes are in the JSON audit and source registry.

Interoperability: No DMSC–IICP interoperability run. Same-project SDK parity is not an independent result. [Independent-record boundary](#).

Unresolved gap: G1, G12: a common plane vocabulary is not a shared interface contract.

Closure condition: Agree which plane owns each interface and which concerns are cross-cutting.

Verification method: Review I1–I6/O1 against the five-concern account and identify any authority transferred only by diagram placement.

Execution status: proposed; not executed; no result reference. The comparison is source review, not an executed joint experiment.

D03. Identity and registration

DMSC basis: [draft-li-dmsec-inf-architecture-07 §3](#); [draft-li-dmsec-inf-architecture-07 §6.1](#).

IICP scope: Provider, Directory; I1/I3. Objects/contracts: provider registration, identity validity, consumer credential. Claims: C04, C07, C23.

Relationship: Partial/conditional correspondence. Registered provider identity, operator identity and caller credentials are different IICP objects. Registration is not request eligibility or provider admission.

Authority/trust owner: Registration authority validates provider identity; the provider retains execution authorization. IICP Management is not an AMC identity issuer.

Maturity: Directory/Core contracts specify distinct registration and caller-authentication concerns; no cross-system identity mapping is implemented here.

Dimension	Assessment and rationale
Functional responsibility	aligned — Both need identifiable service participants.
Semantics	partial — Identity and credential objects are not specified as equivalent.
Authority/trust	distinct — AMC issuance is not IICP Management activation authority.
State/lifecycle	partial — Validity and withdrawal need explicit mapping.
Execution/binding	not evaluated — No credential translation or exchange has been tested.
Evidence/maturity	partial — Support is limited to the listed contracts and source artifacts; no joint implementation result is registered.
Wire compatibility	not evaluated — No common wire contract or joint exchange result is supplied; functional correspondence is not wire compatibility.

Dimension evidence: functional responsibility, semantics, authority/trust, state/lifecycle, execution/binding: [dmsec](#), [directory](#), [core](#), [state](#); evidence/maturity: [directory](#), [core](#), [state](#), [independent](#); wire compatibility: [dmsec](#), [independent](#).

Specification: Directory protocol: REGISTER; Core wire format: CALL and provider authentication; Directory state semantics: independent state axes.

Implementation: No implementation is asserted for this complete mapping; the cited contracts alone do not establish runtime support.

Frozen revisions: IICP: [e1f0c69ac844](#). Full identifiers and content hashes are in the JSON audit and source registry.

Interoperability: No DMSC–IICP interoperability run. Same-project SDK parity is not an independent result. [Independent-record boundary](#).

Unresolved gap: G1, G6: issuer, namespace, audience and revocation mappings remain undefined.

Closure condition: Define principals and credential issuers for one shared scenario; retain separate registration and execution checks.

Verification method: Compare registered, expired-identity and wrong-audience cases; verify that successful registration cannot bypass provider admission.

Execution status: proposed; not executed; no result reference. The comparison is source review, not an executed joint experiment.

D04. Capability advertisement and inventory

DMSC basis: [draft-li-dmsec-inf-architecture-07 §6.1](#); [draft-li-dmsec-inf-architecture-07 §6.2](#).

IICP scope: Provider, Directory; I1/I2/O1. Objects/contracts: capability envelope, complete effective variant, authorized inventory. Claims: C03, C13, C23.

Relationship: Partial/conditional correspondence. An advertisement describes claimed support. IICP's effective capability evaluates a complete model/template/runtime/interface/adaptor variant, not a union of features.

Authority/trust owner: Provider advertises; Directory applies matching and visibility policy; consumer applies its requirements.

Maturity: Effective-capability contract is additive, accepted/pre-normative and parity-gated; Python matching code and test sources are present.

Dimension	Assessment and rationale
Functional responsibility	aligned — Both expose service capability information.
Semantics	partial — A complete effective variant is more constrained than a generic capability inventory.
Authority/trust	partial — Advertisement provenance and policy ownership must be carried explicitly.
State/lifecycle	partial — Withdrawal and freshness need common assertions.
Execution/binding	partial — Capability support must bind to a concrete interface variant.
Evidence/maturity	partial — Support is limited to the listed contracts and source artifacts; no joint implementation result is registered.
Wire compatibility	not evaluated — No common wire contract or joint exchange result is supplied; functional correspondence is not wire compatibility.

Dimension evidence: functional responsibility, semantics, authority/trust, state/lifecycle, execution/binding: [dmsec](#), [effective](#), [observe](#); evidence/maturity: [effective](#), [observe](#), [effectcode](#), [effectivefixture](#), [effecttest](#), [independent](#); wire compatibility: [dmsec](#), [independent](#).

Specification: Effective service capability: [complete variants and matching](#); Node observability interfaces: [authority map and precedence](#).

Implementation: Python effective capability implementation: [effective variant matching](#).

Fixture: Effective capability vectors: [Effective capability vectors](#).

Test source: Python capability matching tests: [matching test source](#).

Frozen revisions: IICP: [e1f0c69ac844](#); iicp-client-python: [508b607dd542](#). Full identifiers and content hashes are in the JSON audit and source registry.

Interoperability: No DMSC–IICP interoperability run. Same-project SDK parity is not an independent result. [Independent-record boundary](#).

Unresolved gap: G1, G3: taxonomy, projection loss and parity are unresolved for a shared record.

Closure condition: Define a versioned minimal exposure record and publish refusal cases for unsupported or unknown required fields.

Verification method: Project two incompatible variants; request features split across them and require no match. Distinguish inventory access from request discovery.

Execution status: proposed; not executed; no result reference. The comparison is source review, not an executed joint experiment.

D05. Intent and task semantics

DMSC basis: [draft-li-dm-sc-inf-architecture-07 §5.1](#); [draft-li-dm-sc-inf-architecture-07 §6.3](#); [draft-sz-dm-sc-iaip-02 §8.4.2](#).

IICP scope: Consumer, Directory, Provider; I1/I2/I3. Objects/contracts: Intent capability identifier, private task payload, mandatory requirements, preferences. Claims: C02, C03, C24.

Relationship: Semantic translation required. IICP Intent is an opaque capability address. IAIP richer intent processing needs an explicit translation. DMSC infrastructure §5.1 does not interpret task semantics; its routing terminology does not establish IAIP vector processing as a universal gateway behavior.

Authority/trust owner: Application interprets the task; Directory matches the declared capability and policy; provider decides admission independently of match scores.

Maturity: IICP identifier/matching contracts and Python code are present; no IAIP-to-IICP interpreter is demonstrated.

Dimension	Assessment and rationale
Functional responsibility	partial — Capability selection is related, but semantic interpretation sits at a different boundary.
Semantics	distinct — A stable address is not natural-language or vector intent.
Authority/trust	distinct — Interpretation and ranking do not grant provider authorization.
State/lifecycle	not evaluated — Translation lifecycle and version invalidation are unspecified.
Execution/binding	partial — A translated identifier would still need a selected execution binding.
Evidence/maturity	partial — Support is limited to the listed contracts and source artifacts; no joint implementation result is registered.
Wire compatibility	not evaluated — No common wire contract or joint exchange result is supplied; functional correspondence is not wire compatibility.

Dimension evidence: functional responsibility, semantics, authority/trust, state/lifecycle, execution/binding: [dm-sc](#), [iaip](#), [semantics](#), [effective](#); evidence/maturity: [semantics](#), [effective](#), [effectcode](#), [effecttest](#), [independent](#); wire compatibility: [dm-sc](#), [independent](#).

Specification: [Intent and selection semantics](#): [Intent identifier semantics](#); [Effective service capability](#): [complete variants and matching](#).

Implementation: [Python effective capability implementation](#): [effective variant matching](#).

Test source: [Python capability matching tests](#): [matching test source](#).

Frozen revisions: IICP: [e1f0c69ac844](#); iicp-client-python: [508b607dd542](#). Full identifiers and content hashes are in the JSON audit and source registry.

Interoperability: No DMSC–IICP interoperability run. Same-project SDK parity is not an independent result. [Independent-record boundary](#).

Unresolved gap: G1, G12: no agreed semantic translation or uncertainty contract.

Closure condition: Specify input meaning, stable capability address, constraint preservation and explicit translation failure; do not embed private task text into discovery by default.

Verification method: Use semantically ambiguous inputs and unsupported mandatory constraints; require explicit no-match/translation failure, not relaxed authorization.

Execution status: proposed; not executed; no result reference. The comparison is source review, not an executed joint experiment.

D06. Discovery, freshness and eligibility

DMSC basis: [draft-li-dmsc-inf-architecture-07 §6.2](#); [draft-li-dmsc-inf-architecture-07 §6.3](#).

IICP scope: Directory, Consumer, Provider; I1/I2. Objects/contracts: identity, advertisement, reachability, availability, eligibility. Claims: C03, C04.

Relationship: Partial/conditional correspondence. IICP distinguishes five state axes. Signed advertisements can be stale; a heartbeat and explicit unavailable state do not collapse into one health bit.

Authority/trust owner: Provider reports availability; Directory combines policy and reachability evidence; consumer validates its selected route.

Maturity: State decision and scenarios are specified; effective matching has Python code. Full cross-implementation state parity is not asserted.

Dimension	Assessment and rationale
Functional responsibility	aligned — Both need a current usable candidate view.
Semantics	partial — IICP state axes need explicit projection into DMSC objects.
Authority/trust	partial — A peer report and local verification have different authority.
State/lifecycle	partial — Freshness windows and withdrawal behavior are not common constants.
Execution/binding	not evaluated — No discovery exchange has been jointly run.
Evidence/maturity	partial — Support is limited to the listed contracts and source artifacts; no joint implementation result is registered.
Wire compatibility	not evaluated — No common wire contract or joint exchange result is supplied; functional correspondence is not wire compatibility.

Dimension evidence: functional responsibility, semantics, authority/trust, state/lifecycle, execution/binding: [dmsc](#), [state](#), [effective](#); evidence/maturity: [state](#), [effective](#), [effectcode](#), [statejson](#), [effecttest](#), [independent](#); wire compatibility: [dmsc](#), [independent](#).

Specification: Directory state semantics: [independent state axes](#); Effective service capability: [complete variants and matching](#).

Implementation: Python effective capability implementation: [effective variant matching](#).

Fixture: Directory state contract and scenarios: [state contract and scenarios](#).

Test source: Python capability matching tests: [matching test source](#).

Frozen revisions: IICP: [e1f0c69ac844](#); iicp-client-python: [508b607dd542](#). Full identifiers and content hashes are in the JSON audit and source registry.

Interoperability: No DMSC–IICP interoperability run. Same-project SDK parity is not an independent result. [Independent-record boundary](#).

Unresolved gap: G1, G3: shared freshness bounds, probe provenance and refusal semantics are not defined.

Closure condition: Agree separate assertions for freshness and eligibility, including unknown required state.

Verification method: Test expired advertisements, fresh heartbeat with explicit unavailability, and state changing after selection. Expect exclusion or revalidation, not automatic admission.

Execution status: proposed; not executed; no result reference. The comparison is source review, not an executed joint experiment.

D07. Selection and route disclosure

DMSC basis: [draft-li-dmsec-inf-architecture-07 §5.1](#); [draft-li-dmsec-inf-architecture-07 §6.3](#).

IICP scope: Directory, Consumer; I2/I3. Objects/contracts: eligible candidate, selected route, dispatch-route-ticket:v1. Claims: C05, C06.

Relationship: Correspondence with different responsibility distribution. The Directory may select a concrete route; the consumer chooses whether to invoke it. A route-disclosure ticket is not execution authorization or global single-use redemption.

Authority/trust owner: Route issuer signs a bounded disclosure; consumer validates issuer, audience, Intent, expiry and policy; provider authority is unchanged.

Maturity: V1 ticket controller and consumer code are present; v2 verification is explicitly opt-in, not the universal default.

Dimension	Assessment and rationale
Functional responsibility	aligned — Both select a service path.
Semantics	partial — Route selection and disclosure are distinguished from invocation.
Authority/trust	distinct — The route issuer cannot grant arbitrary provider admission.
State/lifecycle	partial — Expiry and revalidation need a shared model.
Execution/binding	partial — The chosen route must identify its execution binding.
Evidence/maturity	partial — Support is limited to the listed contracts and source artifacts; no joint implementation result is registered.
Wire compatibility	not evaluated — No common wire contract or joint exchange result is supplied; functional correspondence is not wire compatibility.

Dimension evidence: functional responsibility, semantics, authority/trust, state/lifecycle, execution/binding: [dmsec](#), [tickets](#), [trustv2](#); evidence/maturity: [tickets](#), [trustv2](#), [dispatch](#), [clientcode](#), [ticketfixture](#), [tickettest](#), [dispatchtest](#), [independent](#); wire compatibility: [dmsec](#), [independent](#).

Specification: Route tickets and receipts: bounded evidence and v1/v2 split; Dispatch ticket trust v2: [opt-in trust Profile](#).

Implementation: Dispatch route-ticket controller: [ticket controller and payload exclusion](#); Python consumer implementation: [ticketed candidate validation](#).

Fixture: Route ticket v1 vectors: [Route ticket v1 vectors](#).

Test source: Python ticket verification tests: [canonical ticket verification tests](#); PHP route ticket tests: [dispatch test source](#).

Frozen revisions: IICP: [e1f0c69ac844](#); iicp-directory-php: [5a84dc2e0d30](#); iicp-client-python: [508b607dd542](#). Full identifiers and content hashes are in the JSON audit and source registry.

Interoperability: No DMSC–IICP interoperability run. Same-project SDK parity is not an independent result. [Independent-record boundary](#).

Unresolved gap: G6, G12: common route assertions, verifier responsibilities and recovery semantics remain open.

Closure condition: Define the minimum route handoff and an authority matrix before selecting formats.

Verification method: Present an expired route and a valid route without provider credentials; reject stale authority and retain provider refusal. Inspect sources separately from execution results.

Execution status: proposed; not executed; no result reference. The comparison is source review, not an executed joint experiment.

D08. Authentication and provider admission

DMSC basis: [draft-li-dmsc-inf-architecture-07 §3](#); [draft-li-dmsc-inf-architecture-07 §5.1](#); [draft-li-dmsc-inf-architecture-07 §8](#).

IICP scope: Provider, Consumer, Directory; I1/I2/I3. Objects/contracts: registration credential, X-IICP-Consumer-Token, binding credential, provider admission. Claims: C06, C07.

Relationship: Partial/conditional correspondence. Registration, route disclosure and caller authorization require different credentials. The disabled admission-v2 prototype is not ordinary mounted provider admission.

Authority/trust owner: Provider verifies its caller/audience/Intent conditions and local capacity. Directory token issuance has its own authenticated authority.

Maturity: Core caller credential boundary is specified; admission v2 remains a disabled provider-local prototype.

Dimension	Assessment and rationale
Functional responsibility	aligned — Both require authorized participants and protected execution.
Semantics	partial — Authentication does not by itself establish admission or capacity.
Authority/trust	distinct — AMC functions and IICP local Management have different authority.
State/lifecycle	partial — Expiry, replay and revocation need a pinned contract.
Execution/binding	not evaluated — No credential bridge or joint admission test exists.
Evidence/maturity	partial — Support is limited to the listed contracts and source artifacts; no joint implementation result is registered.
Wire compatibility	not evaluated — No common wire contract or joint exchange result is supplied; functional correspondence is not wire compatibility.

Dimension evidence: functional responsibility, semantics, authority/trust, state/lifecycle, execution/binding: [dmsc](#), [core](#), [admission](#), [admissionv2](#); evidence/maturity: [core](#), [admission](#), [admissionv2](#), [clientcode](#), [independent](#); wire compatibility: [dmsc](#), [independent](#).

Specification: Core wire format: [CALL](#) and provider authentication; Provider admission Profile: provider admission Profile; Dispatch admission v2: disabled admission prototype.

Implementation: [Python consumer implementation](#): [separate consumer-token acquisition](#).

Frozen revisions: IICP: [e1f0c69ac844](#); iicp-client-python: [508b607dd542](#). Full identifiers and content hashes are in the JSON audit and source registry.

Interoperability: No DMSC–IICP interoperability run. Same-project SDK parity is not an independent result. [Independent-record boundary](#).

Unresolved gap: G6: issuer trust, replay, rollback and deployed provider integration are not established jointly.

Closure condition: Choose one credential mechanism, define issuer/audience/scope and independently exercise refusal paths.

Verification method: Try registration credentials at execution, wrong-audience caller tokens and a valid route with no admission credentials. Require refusal without changing route meaning.

Execution status: proposed; not executed; no result reference. The comparison is source review, not an executed joint experiment.

D09. Execution paths and bindings

DMSC basis: [draft-li-dmsec-inf-architecture-07 §5.1](#); [draft-li-dmsec-inf-architecture-07 §7.1](#).

IICP scope: Consumer, Provider, optional Relay; I3/I6. Objects/contracts: CALL/RESPONSE, MCP mapping, A2A card/interface/skill, relay-bind credential. Claims: C01, C11, C12.

Relationship: Partial/conditional correspondence. Direct and relay-assisted paths are possible. IICP I6 names a logical binding boundary, not a new universal transport; A2A and MCP retain their own contracts.

Authority/trust owner: Consumer selects binding and credentials; provider admits; Relay forwards under a separate role and visibility boundary.

Maturity: Direct client and browser code exist; MCP mapping is draft, A2A binding pre-normative, browser provider path experimental.

Dimension	Assessment and rationale
Functional responsibility	aligned — Both permit coordination followed by peer execution.
Semantics	partial — Execution payload and capability selection are separate.
Authority/trust	partial — Relay visibility and provider admission must remain explicit.
State/lifecycle	partial — Binding-specific lifecycle remains authoritative.
Execution/binding	distinct — No common DMSC/IICP execution wire format is established.
Evidence/maturity	partial — Support is limited to the listed contracts and source artifacts; no joint implementation result is registered.
Wire compatibility	not evaluated — No common wire contract or joint exchange result is supplied; functional correspondence is not wire compatibility.

Dimension evidence: functional responsibility, semantics, authority/trust, state/lifecycle, execution/binding: [dmsec](#), [extension](#), [mcp](#), [a2a](#); evidence/maturity: [extension](#), [mcp](#), [a2a](#), [clientcode](#), [web](#), [independent](#); wire compatibility: [dmsec](#), [independent](#).

Specification: [Environmental independence and extensions](#); [Core invariants and extension roles](#); MCP binding: [two binding directions](#); A2A execution binding v0: [consumer validation](#), [authorization and lifecycle](#).

Implementation: [Python consumer implementation](#); [ticketed candidate validation](#); [Browser node: browser and relay scope](#).

Frozen revisions: IICP: [e1f0c69ac844](#); iicp-client-python: [508b607dd542](#); iicp-web-node: [96b5a8786ff2](#). Full identifiers and content hashes are in the JSON audit and source registry.

Interoperability: No DMSC–IICP interoperability run. Same-project SDK parity is not an independent result. [Independent-record boundary](#).

Unresolved gap: G8, G9: no universal stream/resume/cancel guarantee or common transport contract.

Closure condition: Pin one binding, its version and visibility, then test handoff without changing that protocol’s lifecycle.

Verification method: Exercise direct and relay reachability using the chosen binding; verify credential separation and preserve an accepted task’s protocol identity.

Execution status: proposed; not executed; no result reference. The comparison is source review, not an executed joint experiment.

D10. Lifecycle, failure and retry

DMSC basis: [draft-li-dmsh-inf-architecture-07 §5.1](#); [draft-li-dmsh-inf-architecture-07 §6.3](#).

IICP scope: Consumer, Provider; I2/I3/I6. Objects/contracts: task_id, call_id, attempt timeout, caller wait, uncertain completion. Claims: C09, C12.

Relationship: Partial/conditional correspondence. IICP distinguishes five time domains and, in an optional Profile, task from attempt. Stopping a wait is not cancellation; possible acceptance prevents blind cross-provider replay.

Authority/trust owner: Provider owns acceptance/execution state; consumer owns its wait and retry choice subject to binding rules.

Maturity: Time contract/scenarios exist; richer lifecycle and A2A mapping are proposed/pre-normative, not universal endpoints.

Dimension	Assessment and rationale
Functional responsibility	partial — Failure recovery is a shared concern, not a shared state machine.
Semantics	distinct — Task, attempt, timeout and wait are separate IICP concepts.
Authority/trust	partial — A caller timeout cannot declare provider execution absent.
State/lifecycle	distinct — No common lifecycle/duplicate model is defined.
Execution/binding	partial — Retry and cancellation are binding-owned.
Evidence/maturity	partial — Support is limited to the listed contracts and source artifacts; no joint implementation result is registered.
Wire compatibility	not evaluated — No common wire contract or joint exchange result is supplied; functional correspondence is not wire compatibility.

Dimension evidence: functional responsibility, semantics, authority/trust, state/lifecycle, execution/binding: [dmsh](#), [time](#), [lifecycle](#), [a2a](#); evidence/maturity: [time](#), [lifecycle](#), [a2a](#), [timejson](#), [independent](#); wire compatibility: [dmsh](#), [independent](#).

Specification: Task time semantics: [five time domains](#); Service lifecycle Profile: [states and terminality](#); A2A execution binding v0: [consumer validation](#), [authorization and lifecycle](#).

Fixture: Task time contract: [Task time contract](#).

Implementation: No implementation is asserted for this complete mapping; the cited contracts alone do not establish runtime support.

Frozen revisions: IICP: [e1f0c69ac844](#). Full identifiers and content hashes are in the JSON audit and source registry.

Interoperability: No DMSC–IICP interoperability run. Same-project SDK parity is not an independent result. [Independent-record boundary](#).

Unresolved gap: G8: common failure outcomes, resumption and duplicate-suppression semantics are unproven.

Closure condition: Define pre-acceptance versus uncertain/post-acceptance failures for one binding.

Verification method: Disconnect before acceptance and after possible acceptance; verify that retry decisions differ and no exactly-once or universal cancellation property is inferred.

Execution status: proposed; not executed; no result reference. The comparison is source review, not an executed joint experiment.

D11. Intra-domain synchronization

DMSC basis: [draft-li-dm-sc-inf-architecture-07 §5.1](#); [draft-li-dm-sc-inf-architecture-07 §6.4](#).

IICP scope: Directory root, Directory replica; I4. Objects/contracts: signed snapshot, event tail, record ownership, freshness. Claims: C04, C08.

Relationship: Partial/conditional correspondence. Current IICP synchronization uses root-authorized views. Frequent availability state and durable registration events have distinct ownership and freshness rules.

Authority/trust owner: Genesis/root authorizes records; replicas verify and serve a bounded current view, not sovereign write authority.

Maturity: Federation is draft; Rust Directory is operator preview and not a transfer of Genesis authority.

Dimension	Assessment and rationale
Functional responsibility	aligned — Both distribute discovery-related information.
Semantics	partial — Replicated records and live state are different objects.
Authority/trust	distinct — IICP's current root authority is narrower than sovereign peer authority.
State/lifecycle	partial — Deletion and lag semantics require mapping.
Execution/binding	not evaluated — No shared synchronization exchange is implemented.
Evidence/maturity	partial — Support is limited to the listed contracts and source artifacts; no joint implementation result is registered.
Wire compatibility	not evaluated — No common wire contract or joint exchange result is supplied; functional correspondence is not wire compatibility.

Dimension evidence: functional responsibility, semantics, authority/trust, state/lifecycle, execution/binding: [dm-sc](#), [federation](#), [events](#); evidence/maturity: [federation](#), [events](#), [rustdir](#), [independent](#); wire compatibility: [dm-sc](#), [independent](#).

Specification: [Federated Directory: roles and root trust](#); [Context and service event ownership: event ownership](#).

Implementation: [Rust Directory: preview implementation scope](#).

Frozen revisions: IICP: [e1f0c69ac844](#); iicp-directory-rust: [47277fad5dad](#). Full identifiers and content hashes are in the JSON audit and source registry.

Interoperability: No DMSC–IICP interoperability run. Same-project SDK parity is not an independent result. [Independent-record boundary](#).

Unresolved gap: G7: gateway peer synchronization is not demonstrated as the same topology or schema.

Closure condition: Fix root/replica or peer assumptions and agree ownership, withdrawal, lag and recovery behavior.

Verification method: Compare a snapshot plus event tail with deletion and lag; reject unauthorized writes and distinguish valid signature from fresh state.

Execution status: proposed; not executed; no result reference. The comparison is source review, not an executed joint experiment.

D12. Inter-domain policy and trust

DMSC basis: [draft-li-dmsec-inf-architecture-07 §6.4](#); [draft-li-dmsec-inf-architecture-07 §8](#).

IICP scope: Directory, Consumer, Provider, Management; I2/I3/I4/I5/I6. Objects/contracts: root trust, local policy, restricted trust-domain proposal. Claims: C08, C18.

Relationship: Partial/conditional correspondence. A private Directory or a signed root replica is not a sovereign multi-root federation. Restricted trust-domain membership is an unimplemented pre-normative proposal.

Authority/trust owner: Each consumer/provider/operator retains local trust and policy. No common cross-domain credential or policy authority is supplied.

Maturity: Bounded root/replica and local policy contracts; broader membership/policy exchange remains proposed and unimplemented.

Dimension	Assessment and rationale
Functional responsibility	partial — Cross-domain interaction is a related problem with narrower IICP evidence.
Semantics	partial — Policy and membership vocabularies are not shared.
Authority/trust	distinct — Local enforcement and root trust do not implement sovereign multi-root trust.
State/lifecycle	not evaluated — Common revocation/withdrawal propagation has not been defined.
Execution/binding	not evaluated — No cross-domain exchange experiment is registered.
Evidence/maturity	partial — Support is limited to the listed contracts and source artifacts; no joint implementation result is registered.
Wire compatibility	not evaluated — No common wire contract or joint exchange result is supplied; functional correspondence is not wire compatibility.

Dimension evidence: functional responsibility, semantics, authority/trust, state/lifecycle, execution/binding: [dmsec](#), [federation](#), [restricted](#), [authority](#); evidence/maturity: [federation](#), [restricted](#), [authority](#), [independent](#); wire compatibility: [dmsec](#), [independent](#).

Specification: [Federated Directory: roles and root trust](#); [Restricted trust domain v0: pre-normative trust-domain proposal](#); [Management local authority: exact-plan acceptance and application boundary](#).

Implementation: No implementation is asserted for this complete mapping; the cited contracts alone do not establish runtime support.

Frozen revisions: IICP: [e1f0c69ac844](#); iicp-management: [9b9c910fe644](#). Full identifiers and content hashes are in the JSON audit and source registry.

Interoperability: No DMSC–IICP interoperability run. Same-project SDK parity is not an independent result. [Independent-record boundary](#).

Unresolved gap: G7: sovereign federation and interoperable cross-domain negotiation have no demonstrated mapping.

Closure condition: Define domain roots, trust anchors, policy conflict, membership withdrawal and failure behavior without assuming universal trust.

Verification method: Document two domains with conflicting policy and revoked membership; require explicit refusal and provenance rather than automatic federation.

Execution status: proposed; not executed; no result reference. The comparison is source review, not an executed joint experiment.

D13. Telemetry and bounded evidence

DMSC basis: [draft-li-dmsec-inf-architecture-07 §6.5](#); [draft-li-dmsec-inf-architecture-07 §8](#).

IICP scope: Provider, Directory, Consumer, Management; O1/I5. Objects/contracts: health, authorized inventory, local events, bounded receipt. Claims: C13, C15.

Relationship: Partial/conditional correspondence. Functional correspondence concerns policy-controlled operational observation, not a common telemetry schema. O1 is an editorial grouping of health, inventory and local observation surfaces. The broader record includes bounded receipts: signatures establish attribution/integrity, not semantic correctness.

Authority/trust owner: Source reports observations; relying party evaluates signer, scope and freshness; local operator controls retention and activation.

Maturity: Observability is an architecture candidate with machine-readable scenarios; receipts are bounded proposals, not full execution attestation.

Dimension	Assessment and rationale
Functional responsibility	aligned — Both provide operational observation subject to policy; this correspondence does not assign every IICP health signal to DMSC's long-term telemetry role.
Semantics	partial — IICP sources have an explicit precedence, not a shared telemetry schema.
Authority/trust	aligned — Observation remains subject to policy and is not an automatic authority grant.
State/lifecycle	partial — Timestamps, freshness and retention need mapping.
Execution/binding	not evaluated — Telemetry correspondence says nothing about execution-wire compatibility.
Evidence/maturity	partial — Support is limited to the listed contracts and source artifacts; no joint implementation result is registered.
Wire compatibility	not evaluated — No common wire contract or joint exchange result is supplied; functional correspondence is not wire compatibility.

Dimension evidence: functional responsibility, semantics, authority/trust, state/lifecycle, execution/binding: [dmsec](#), [observe](#), [tickets](#), [security](#); evidence/maturity: [observe](#), [tickets](#), [security](#), [observefixture](#), [independent](#); wire compatibility: [dmsec](#), [independent](#).

Specification: Node observability interfaces: [authority map and precedence](#); Route tickets and receipts: [bounded evidence](#) and [v1/v2 split](#); Privacy adversary and trust model: [adversary and evidence limits](#).

Fixture: [Observability event scenarios](#): [observation scenarios](#).

Implementation: No implementation is asserted for this complete mapping; the cited contracts alone do not establish runtime support.

Frozen revisions: IICP: [e1f0c69ac844](#). Full identifiers and content hashes are in the JSON audit and source registry.

Interoperability: No DMSC–IICP interoperability run. Same-project SDK parity is not an independent result. [Independent-record boundary](#).

Unresolved gap: G1, G12: shared event fields, privacy, retention and observation provenance need agreement.

Closure condition: Choose content-minimized event fields and label direct, snapshot and inferred observations separately.

Verification method: Compare direct health, a valid snapshot and inferred log fallback; verify that failure evidence cannot become an execution authorization or correctness assertion.

Execution status: proposed; not executed; no result reference. The comparison is source review, not an executed joint experiment.

D14. Management state and local activation

DMSC basis: [draft-li-dmsc-inf-architecture-07 §5.1](#); [draft-li-dmsc-inf-architecture-07 §6.4](#); [draft-li-dmsc-inf-architecture-07 §6.5](#).

IICP scope: Management, local authority, bounded adapters, Consumer; I5/O1. Objects/contracts: desired, accepted, observed, effective, exact-plan authorization. Claims: C10, C28.

Relationship: Partial/conditional correspondence. IICP distinguishes desired, accepted, observed and effective state. Plan acceptance is not application; Management is not a general remote administrative or SDI service.

Authority/trust owner: Local authority accepts a scoped generation; bounded adapter applies it; process-local client narrows routing within a short authority lifetime.

Maturity: Management prerelease/local preview with working apply gate and process-local Rust client; no generally deployed remote control service.

Dimension	Assessment and rationale
Functional responsibility	partial — Policy control is related, but local activation has a narrower scope.
Semantics	distinct — Four Management states are not Directory availability states.
Authority/trust	distinct — Only the local authority can accept and apply its bounded plan.
State/lifecycle	partial — Acceptance and effective convergence require different evidence.
Execution/binding	not evaluated — No common remote management protocol is specified.
Evidence/maturity	partial — Support is limited to the listed contracts and source artifacts; no joint implementation result is registered.
Wire compatibility	not evaluated — No common wire contract or joint exchange result is supplied; functional correspondence is not wire compatibility.

Dimension evidence: functional responsibility, semantics, authority/trust, state/lifecycle, execution/binding: [dmsc](#), [authority](#), [mgmtmap](#); evidence/maturity: [authority](#), [mgmtmap](#), [applygate](#), [routingcode](#), [routingtest](#), [independent](#); wire compatibility: [dmsc](#), [independent](#).

Specification: Management local authority: exact-plan acceptance and application boundary; Management architecture context map: component and authority map.

Implementation: Management local apply gate: local apply validation; Management Rust client enforcement: process-local enforcement.

Test source: Management routing enforcement tests: [enforcement test source](#).

Frozen revisions: iicp-management: [9b9c910fe644](#). Full identifiers and content hashes are in the JSON audit and source registry.

Interoperability: No DMSC–IICP interoperability run. Same-project SDK parity is not an independent result. [Independent-record boundary](#).

Unresolved gap: G10: interoperable remote activation and evidence-backed general SDI are not established.

Closure condition: Agree authority owner, exact scope, generation, expiry and observed convergence for one local change.

Verification method: Reject a changed or expired plan; distinguish accepted-but-unapplied from effective. Never infer arbitrary shell or restart authority from observation.

Execution status: proposed; not executed; no result reference. The comparison is source review, not an executed joint experiment.

References and document provenance

The primary published and IETF references are listed below. Revision-pinned repository sources are catalogued in Appendix A; the version snapshot is in Appendix B.

(Mumin 2026; Li et al. 2026; Sun et al. 2026)

This discussion draft used AI-assisted source inspection, consolidation, diagram generation and editorial checking. These checks are not an independent technical review. The Markdown reference, vector figures, audit exports, evidence registry and validation report accompany the draft. No new DOI or external submission has been created.

Published and IETF references

Li, Xueting, Aijun Wang, Bing Liu, and Changwang Lin (May 22, 2026). *Dynamic Multi-agent Secured Collaboration Infrastructure Architecture*. Internet-Draft draft-li-dmsc-inf-architecture-07. Work in progress; individual draft. IETF. URL:

<https://www.ietf.org/archive/id/draft-li-dmsc-inf-architecture-07.txt>.

Mumin, Roble (Aug. 25, 2026). *IICP: Provider-Neutral Infrastructure for Intelligence Virtualization*.

Version 1.0.1. Zenodo. DOI: [10.5281/zenodo.22098223](https://doi.org/10.5281/zenodo.22098223). URL:

<https://zenodo.org/records/22098223>.

Sun, Sheng, Xinyi Zhang, Qiangzhou Gao, Min Liu, and Yuwei Wang (May 25, 2026). *Intent-based Agent Interconnection Protocol at Agent Gateway*. Internet-Draft draft-sz-dmsc-iaip-02. Work in progress; individual draft. IETF. URL:

<https://www.ietf.org/archive/id/draft-sz-dmsc-iaip-02.txt>.