Network Management Research Group
Internet-Draft
Intended status: Informational
Expires: 21 October 2026

C. Feng
Ruijie Networks

April 2026

Agentic Intent Network (AIN): Applicability and Deployment Scenarios
draft-feng-nmrg-ain-deployment-00

Abstract

The Agentic Intent Network (AIN) architecture defines a routing-based coordination substrate for open, heterogeneous, Internet-scale multi-agent systems. This document describes applicability and deployment scenarios for AIN, targeting decision-makers in carrier networks, enterprises, and network equipment vendors. It maps the technical mechanisms defined in [AIN-ARCH] to concrete operational contexts, describes migration paths from existing infrastructure, and discusses the cold-start bootstrapping challenge inherent to network-effect-dependent systems. This document does not define new protocol mechanisms; it is intended to inform deployment planning and expand the AIN reader community beyond protocol engineers.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 3 October 2026.

Copyright Notice

Table of Contents

1. Introduction

The AIN architecture, defined in [AIN-ARCH], is written primarily for protocol engineers and researchers. That text defines problem drivers, architectural components, design invariants, and a research agenda with the rigor required for interoperable implementations. Deployment decisions, however, are often made by a different audience: operators, CTO organizations, enterprise architects, and product managers.

These stakeholders usually ask a different first question. They do not ask, "What is the exact on-wire encoding?" They ask, "Why should we do this now in our environment, and what is the lowest-risk first step?" This document answers that question. It maps mechanisms from [AIN-ARCH] to concrete operational contexts and migration paths that begin from current infrastructure.

This style has precedent in IETF/IRTF Informational publications. [RFC7454] and [RFC8799] translate protocol concepts into deployment and operational guidance for decision-makers. AIN benefits from the same approach because adoption is shaped by migration cost, role incentives, sequencing, and governance timing, not only by protocol design.

The scenarios therefore use narrative language intentionally. Decision makers evaluate trajectories, not just mechanisms. They need to compare before/after positions, identify where value appears in Months 1-6, and understand where unresolved research still exists. This format makes those trade-offs explicit without changing protocol semantics.

This document does not define new AIN packet fields, new capability advertisement semantics, or new management objects. Normative protocol behavior remains in the architecture document [AIN-ARCH]. Normative language here is used only for critical operational constraints where architecture compliance matters directly; the rest is descriptive guidance for planning.

Four scenarios are included: carrier operator, enterprise, equipment vendor, and AI-native developer. Together they cover likely early adopters and ecosystem enablers. Final sections address cold-start bootstrapping and clarify compatibility expectations with existing standards, including explicit inter-domain cautions.

2. Terminology and Document Conventions

All AIN-specific terms, including Intent Router, Handler, Originator, IC-OID, CAP, CRT, Agent Domain, Mode A through Mode E, and the Foreman Pattern, are defined in [AIN-ARCH] and are not redefined here.

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

This document uses SHOULD where architecture compliance is important. Otherwise, it uses descriptive language.

The following terms are defined for use in this document. They describe deployment roles, execution modes, and coordination patterns that are relevant to the scenarios presented here. They are consistent with the component model defined in [AIN-ARCH] Section 5.5 and are introduced here to support deployment planning without requiring additional companion documents.

Routing and Infrastructure Operator (RIO): An entity that operates Intent Routers and Agent Domain infrastructure, analogous to an Internet Service Provider in IP networking.

Intent-Aware Platform (IAP): A platform operator that combines RIO-level infrastructure with value-added services above the routing substrate, such as capability marketplaces or managed Handler hosting.

Capability Provider (CP): An entity that operates one or more Intent Handlers and registers their capabilities via CAP into one or more Agent Domains.

Mode A Handler (AI-Native Handler): A Handler that natively processes Intent Datagrams and implements AIN interfaces directly, with no wrapping layer between the AIN control plane and the executing agent.

Mode C Handler (Gateway Handler): A Handler that wraps a legacy or non-AIN system, translating Intent Datagrams to and from proprietary interfaces. The wrapped system requires no modification; the Mode C adapter presents a single AIN-facing interface regardless of internal complexity.

Capability Description Language (CDL): The vocabulary and schema

used to describe Handler capabilities in CAP messages and the IC-OID namespace. CDL semantics determine how IC-OID prefixes are aggregated and how capability matching is performed. CDL specification is identified as a Phase 1 research problem in [AIN-ARCH] Section 7.1.

Foreman Pattern: A deployment pattern in which a single registered Handler (the "foreman") accepts intents on behalf of an internal worker pool. From the AIN routing perspective, the entire pool appears as one CRT entry, preventing route-state growth proportional to worker count. This pattern is especially useful when workers are numerous, ephemeral, or dynamically scaled.

3. Deployment Scenarios Overview

Scenario 1 models a carrier operator that already runs large routing infrastructure and seeks monetization beyond commodity transport. This scenario is included because carriers can start AIN at one PoP with incremental operational risk.

Scenario 2 models an enterprise that is both Originator and Handler provider. This scenario is included because internal integration simplification can produce immediate value before any external peering.

Scenario 3 models a network equipment vendor. This scenario is included because deployment speed depends on software enablement of existing hardware fleets.

Scenario 4 models an AI-native developer. This scenario is included because many teams have application protocols but still lack scalable discovery and routing across domains.

Scenario	Primary Role	AIN Entry Point
Carrier Operator	RIO / IAP	Single Agent Domain at one PoP
Enterprise	Originator + Handler Provider	Internal Intent Router deployment
Equipment Vendor	Enabler	Software module on existing HW
AI-Native Dev.	CP / Originator	Handler registration via CAP

Table 1: Scenario Overview

4. Scenario 1: Carrier Network Operator

4.1. Current State and Pain Points

Consider a carrier with PoPs in multiple regions, a mature BGP full-mesh underlay, and enterprise customers requesting API integration services. Traffic grows while revenue per user declines.

The pain point is that customers increasingly want agent-to-agent collaboration, while the carrier can monetize only bandwidth. The intent layer between enterprise systems remains outside the carrier's service model.

4.2. AIN Entry Point

The operator upgrades edge routers at one PoP to support CAP as a software module and deploys the first Agent Domain. AIN does not require new hardware categories; Intent Router forwarding and CAP support are implemented as software modules on existing router platforms [AIN-ARCH]. Enterprise Handlers then register to their nearest PoP.

4.3. Migration Path

Phase	Plan
Phase 1 (Months 1-6)	Single-domain pilot; static CRT; Mode C Gateway Execution for legacy enterprise APIs

Phase 2 (Months 6-18)	OSPF-inspired intra-domain CAP propagation; dynamic CRT; multiple PoPs connected
Phase 3 (Months 18-36)	Inter-domain CAP exchange with peer carriers (BGP-inspired inter-domain protocol, subject to ongoing research; see Section 7.2 of [AIN-ARCH])
Phase 4 (36+ months)	Handler capability marketplace portal; per-query billing analogous to DNS resolution combined with CDN delivery

Table 2: Carrier Migration Phases

IMPORTANT: Phase 3 is BGP-inspired in structure (capability prefix aggregation, border policy filtering, and per-domain administrative autonomy) but is NOT a literal BGP wire-protocol extension. It requires new protocol design currently under research. Operators SHOULD NOT plan Phase 3 based on [RFC4271] wire compatibility.

4.4. Before and After Comparison

Dimension	Before AIN	After P1/P2	After P3/P4
Revenue model	Bandwidth only	Managed domain fee	Intent routing + marketplace
Enterprise value proposition	Connectivity only	Internal discovery and routing	Cross-domain capability exchange
Technical complexity	Familiar IP operations	Added CAP and CRT ops	Inter-domain policy and governance
Inter-carrier coordination	Transit peering	Optional	Required for ecosystem scale

Table 3: Carrier Before/After

5. Scenario 2: Enterprise (Dual Role as Originator and Handler Provider)

5.1. Current State and Pain Points

Consider a multinational manufacturer in five countries with SAP, Salesforce, ServiceNow, and proprietary MES systems. Each new integration requires custom adapters and bilateral maintenance.

A procurement AI assistant may need to trigger supply-chain replanning in another business unit where API documentation is not available. This reproduces the N(N-1)/2 integration pattern. At N=20 systems, that is 190 pairwise integrations.

5.2. AIN Entry Point

Deploy a lightweight Intent Router on x86 servers running CAP software. Existing SAP and Salesforce interfaces are wrapped as Mode C Handlers using Execution Runtime adapters. Internal AI assistants become Originators.

Integration complexity collapses from O(N^2) pairwise links to O(N) registrations.

5.3. Migration Path

Phase 1: Single Intent Router, internal network only. Existing systems wrapped as Mode C Handlers. Internal AI assistants as Originators. IC-OID namespace governed by internal IT policy.

Phase 2: Connect to carrier AIN domain. Cross-BU collaboration. Introduce Mode A Handlers for AI-native services. Adopt Foreman Pattern for dynamic worker pools.

Phase 3: Selectively expose Handlers to external partners. Join AIN ecosystem as Capability Provider. Participate in inter-domain routing.

5.4. Before and After Comparison

Metric	Before AIN	After Phase1	After Phase3
Integration complexity	O(N^2) custom links	O(N) inside one domain	O(N) with governed exposure
Discovery overhead	Manual docs and contacts	Internal CRT lookup	Cross-domain lookup via policy
Cross-BU latency	Human relay and tickets	Programmatic resolution	Routed with external partners

Governance overhead	Distributed and opaque	Central IT namespace control	Expanded trust and contracts

Table 4: Enterprise Before/After

6. Scenario 3: Network Equipment Vendor

6.1. Current State and Pain Points

Consider a mid-size vendor with routers deployed in more than 200 carriers. BGP/OSPF business is stable but growth is slow. Carriers now ask about AIN readiness.

The pain point is perceived hardware commoditization risk if AIN functionality is treated as software only.

6.2. Two Strategic Paths

Path A (Defensive)	Path B (Offensive)
Implement CAP support as optional software module	Package edge routers as "AIN-ready" line and provide turnkey first Agent Domain deployment
Allow existing routers to join AIN control plane	Bundle hardware + software + deployment services
Do not proactively market readiness	Proactively market readiness before standards freeze
Timeline: 6-12 months software work	Timeline: 12-18 months to first reference deployment
Risk: Low, Cost: Low, Upside: preserves current position	Risk: Higher, Cost: Higher; Upside: ecosystem positioning before standards freeze
Downside: misses first-mover position	Downside: larger execution and market risk

Table 5: Strategic Paths

6.3. Migration Path

Regardless of path, Phase 1 is identical: implement Intent Router forwarding and CAP support as software modules on existing hardware. [AIN-ARCH] confirms no new hardware category is needed; Intent Router forwarding and CAP processing are software functions.

A vendor that completes Phase 1 can switch from Path A to Path B at any time.

Because [AIN-ARCH] Section 5.6 Invariant 1 separates forwarding from execution, forwarding hardware retains differentiated value and the commodity risk is lower than initially feared.

7. Scenario 4: AI-Native Application Developer

7.1. Current State and Pain Points

Teams building with LangGraph or A2A often have good in-domain agent collaboration but no shared discovery mechanism across organizations. They still negotiate bilateral APIs and maintain static endpoint files.

A2A defines interaction format (analogous to HTTP) but not discovery or routing. Endpoint knowledge must still be known in advance. At scale this reproduces O(N^2) coupling.

7.2. AIN Entry Point

Existing agents register as Handlers via CAP and become discoverable through IC-OID without requiring callers to know concrete endpoints.

A2A agent cards are candidate inputs to AIN CDL. MCP tool integrations can be wrapped as Mode A Handlers.

7.3. Migration Path

Step 1: Register existing agents as Handlers. Assign IC-OIDs. No change to core agent logic; add CAP registration and Intent Datagram ingestion adapter.

Step 2: Replace static endpoint config with Intent Router lookup. Originators emit Intent Datagrams; routing resolves Handler location dynamically.

Step 3: Use Foreman Pattern for dynamic worker pools. Worker lifecycle events do not force CRT churn.

Step 4: Enable cross-domain discovery when inter-domain routing is available (Phase 3 prerequisite; see Section 4.3).

Key insight: AIN is not a replacement for A2A or MCP. A2A defines handshake semantics; AIN provides discovery and routing so handshake can happen without prior endpoint knowledge.

8. The Cold-Start Problem and Bootstrap Strategies

8.1. The Network Effect Dependency

AIN value grows with network effects; more Handlers make routing useful to more Originators. First deployers therefore begin with a thin ecosystem.

This is not a design flaw. It is the same bootstrap challenge faced by routing infrastructures in their early phases, including the commercial Internet.

8.2. Recommended Bootstrap Sequence

1. Start closed. Deploy first within one enterprise or one PoP. Prove internal value before external connectivity. Stage 1 value (O(N^2) to O(N) integration) is reachable with zero external participants.

2. Use Mode C as the entry ramp. Mode C (Gateway Execution), as defined in Section 2 of this document and consistent with the Application Entities model in [AIN-ARCH] Section 5.5, lets legacy systems participate with minimal code change. A full SAP environment can be exposed via a single Mode C adapter without changes to SAP itself.

3. Build internal proof before seeking peers. Collect 6-12 months of latency, reliability, and integration-cost evidence before external onboarding.

4. Seed the capability directory. Pre-populate IC-OID classes for internal capabilities before opening the domain externally.

8.3. Role of Mode C (Gateway Execution)

Mode C is central to cold-start because it turns legacy complexity into routable capability while keeping rewrites out of the critical path.

A Mode C Handler presents one AIN-facing interface while it orchestrates many legacy internals. From routing perspective, a large SAP estate with many transaction types can appear as one registered Handler under well-defined IC-OIDs.

This is the Foreman Pattern at the integration boundary: worker complexity stays local while routing-state growth remains controlled.

9. Relationship to Existing Standards and Protocols

9.1. Protocol Reuse and New Design

AIN reuses protocol structures where semantics transfer exactly. It introduces new mechanisms where IC-OID identifier semantics diverge from topological IP addressing.

Operators can use the compatibility table below to plan tool-chain investment and identify low-friction reuse points.

9.2. Compatibility Table

Existing Technology	AIN Usage	Reuse	Notes
OSPF Opaque LSA ([RFC5250])	Intra-domain CAP propagation	Structural analogy	New convergence theory required
BGP MP Extensions ([RFC4760])	Inter-domain CAP exchange	Structural analogy	Not wire-compat.; New protocol
NETCONF/YANG ([RFC6241]/[RFC7950])	AIN management plane	Direct reuse	Management plane
SNMP OID format	IC-OID structure	Structural analogy	Different governance model
DNS resolution	IC-OID Registry governance	Structural analogy	DA not in forwarding path
A2A agent cards ([A2A2025])	CDL capability description	Candidate reuse	Under evaluation

MCP tool spec ([MCP2024])	Handler execution runtime	Candidate wrapper	Wrappable as Mode A Handler
IP TTL / hop limit	Datagram hop count	Exact reuse	Identical semantics

Table 6: Existing Technology Compatibility

Operators with existing NETCONF/YANG infrastructure can reuse it directly for AIN management-plane functions. This is likely the lowest-cost integration point for experienced operators.

10. Security Considerations

This document does not define new protocol mechanisms and therefore introduces no security considerations beyond those in [AIN-ARCH].

Deployment scenarios that use Mode C (Gateway Execution) should account for access-control boundaries when legacy systems are wrapped as AIN Handlers. Existing ACL and authentication controls on wrapped systems remain in effect and are not bypassed by AIN routing.

Operators should validate that exposing a legacy capability through a Mode C interface does not unintentionally broaden privilege scope. Least-privilege role mappings, credential hygiene, and auditable logs should be part of initial deployment readiness checks.

In multi-tenant deployments, operators should also ensure that tenant boundaries in legacy back-end systems are preserved at the Handler boundary, especially when one Mode C gateway fronts multiple internal services. Admission control, request attribution, and replay-resistant authentication are important companion controls for production rollout. None of these controls are replaced by AIN routing; they remain local obligations of the wrapped execution environment.

11. IANA Considerations

This document has no IANA actions.

12. Normative References

[AIN-ARCH] C., F., "Agentic Intent Network (AIN) Architecture", Work in Progress, Internet-Draft, draft-feng-nmrg-ain-architecture-00, Work in Progress , 2026, <https://datatracker.ietf.org/doc/html/draft-feng-nmrg-ain-architecture-00>.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

13. Informative References

[A2A2025] DeepMind, G., "Agent-to-Agent Protocol Specification v1.0", 2025.

[MCP2024] Anthropic, "Model Context Protocol", 2024.

[RFC2328] Moy, J., "OSPF Version 2", RFC 2328, DOI 10.17487/RFC2328, April 1998, <https://www.rfc-editor.org/info/rfc2328>.

[RFC4271] Rekhter, Y., Li, T., and S. Hares, "A Border Gateway Protocol 4 (BGP-4)", RFC 4271, DOI 10.17487/RFC4271, January 2006, <https://www.rfc-editor.org/info/rfc4271>.

[RFC4760] Bates, T., Chandra, R., Katz, D., and Y. Rekhter, "Multiprotocol Extensions for BGP-4", RFC 4760, DOI 10.17487/RFC4760, January 2007, <https://www.rfc-editor.org/info/rfc4760>.

[RFC5250] Berger, L., Bryskin, I., Zinin, A., and R. Coltun, "The OSPF Opaque LSA Option", RFC 5250, DOI 10.17487/RFC5250, July 2008, <https://www.rfc-editor.org/info/rfc5250>.

[RFC6241] Enns, R., Bjorklund, M., Schoenwaelder, J., and A. Bierman, "Network Configuration Protocol (NETCONF)", RFC 6241, DOI 10.17487/RFC6241, June 2011, <https://www.rfc-editor.org/info/rfc6241>.

[RFC6459] Korhonen, J., Soininen, J., and B. Patil, "IPv6 in 3GPP Releases", RFC 6459, DOI 10.17487/RFC6459, January 2012, <https://www.rfc-editor.org/info/rfc6459>.

[RFC7454] Durand, A., Dhamdhere, A., Donley, C., and W. George, "BGP Operations and Security", RFC 7454, DOI 10.17487/RFC7454, February 2015, <https://www.rfc-editor.org/info/rfc7454>.

[RFC7950] Bjorklund, M., "The YANG 1.1 Data Modeling Language", RFC 7950, DOI 10.17487/RFC7950, August 2016, <https://www.rfc-editor.org/info/rfc7950>.

[RFC8799] Carpenter, B. and B. Liu, "Limited Domains and Internet Protocols", RFC 8799, DOI 10.17487/RFC8799, July 2020, <https://www.rfc-editor.org/info/rfc8799>.

Author’s Address

Chong Feng
Ruijie Networks
Email: fengchonglly@gmail.com