LISP Working Group
Internet-Draft
Intended status: Informational
Expires: 5 October 2026

W. Wang
C. Xie
China Telecom

3 April 2026

Using LISP as a Network Substrate for AI Agent Communication
draft-wang-lisp-ai-agent-01

Abstract

The emergence of distributed artificial intelligence (AI) systems, particularly those composed of autonomous agents operating across cloud, edge, and endpoint environments, introduces new networking requirements. These include location transparency, seamless mobility, multi-homing, and logical isolation at scale. This document explores how the Locator/ID Separation Protocol (LISP) can serve as a robust network substrate to meet these requirements. The document outlines use cases, design considerations, and minimal extensions to the existing LISP framework to support context-aware mapping and AI agent-centric communication.

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 5 October 2026.

Copyright Notice

Table of Contents

1. Introduction

Modern AI systems are increasingly distributed, comprising autonomous software entities referred to as AI agents that collaborate across heterogeneous infrastructure, including public clouds, private data centers, edge nodes, and end-user devices. These AI agents may migrate dynamically (e.g., due to resource constraints, latency optimization, or failure recovery), yet their communication sessions must remain uninterrupted.

Traditional IP networking binds identity and location into a single address, making seamless mobility and multi-homing challenging without application-layer intervention (e.g., session re-establishment or DNS updates). The Locator/ID Separation Protocol (LISP) [RFC9300], however, decouples identity from location, enabling transparent mobility and flexible traffic engineering. This document proposes using LISP as a network substrate for AI agent communication. We show how LISP' s existing architecture naturally

supports key requirements of AI agent communications, and we propose minimal, backward-compatible extensions to enable context-aware routing decisions driven by agent-level semantics.

The goal is not to redefine LISP, but to illustrate how it can be leveraged and slightly enhanced to serve as a foundational layer for next-generation intelligent systems.

1.1. Requirements Language

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.

2. Terminology

The following terms are used in this draft:

* Endpoint Identifier (EID) [RFC9299]: Addresses assigned topologically to network attachment points. Typically routed inter-domain. An AI-agent system will be assigned one or more EID addresses.

* xTR [RFC9299]: A router that implements both ITR and ETR functionalities. An xTR can be co-located with an AI-agent EID or be part of a LISP site where AI agents are assigned EID addresses. That is, an EID and an RLOC-set can be on the mobile agent or the mobile agent can move to new RLOC xTRs. xTR can be muli-homed , when underlay performance changes, the xTR can select better paths to other AI agents.

* Map-Server [RFC9301]: A Map-Server stores the mapping information and relationships published by xTRs and returns key-value pairs to the requester.

* Map-Resolver [RFC9301]: A network infrastructure component that accepts LISP Encapsulated Map-Requests, typically from an ITR, and determines whether or not the destination IP address is part of the EID namespace; if it is not, a Negative Map-Reply is returned. Otherwise, the Map-Resolver finds the appropriate EID-to-RLOC mapping by consulting a mapping database system. This mechanism can be used for agent-to-agent packet delivery, AI agent discovery, and AI agent capability inventory.

* Instance ID (IID) [RFC9299]: A 24-bit identifier used to create isolated VPN groups.

* AI agent: A software entity capable of perception, decision-making, and action, often operating autonomously or in coordination with other AI agents. An AI agent is discoverable via an Endpoint Identifier (EID) or a Distinguished Name, distinguished by the use of specific LISP Canonical Address Format (LCAF) encodings for its EID.

* Agent VPN Group: A logical collection of AI agents that share a common task, security policy, or privacy level. Each group is associated with a unique LISP IID, which serves to isolate the domain and facilitate mechanisms such as EID Anycast for discovering the topologically closest agent within the group.

3. Requirements from AI agent communication

3.1. Persistent identity across mobility

AI agents must maintain a consistent network identity when migrating across hosts or networks; if traditional IP addresses are used as identity identifiers, any change in address will disrupt existing communication sessions and require upper-layer applications to reestablish connections, thereby compromising communication continuity and the overall capability of AI systems.

3.2. Logical isolation of Agent VPN Groups

Even when multiple agent VPN groups operate on the same physical or virtual network infrastructure, they must be isolated from one another to prevent interference and ensure that their respective security policies are strictly enforced.

Agent VPN groups can be deployed across the same or different underly networks, relying on one or more mapping systems. This sharded deployment model presents specific trade-offs and advantages.

3.3. Context-aware routing

To facilitate dynamic path selection based on communication intent (such as the requirements for latency or security), AI agents should employ a multi-homed deployment equipped with multiple wireless interfaces. This architecture ensures path diversity across different network providers, allowing the network to select the optimal transmission route that satisfies the agent’s specific context.

4. LISP as a Network Substrate

4.1. AI agent identity as EID

Each AI agent is assigned a stable EID that serves as its permanent network identity, remaining invariant regardless of execution location or mobility events. The identifier may be implemented as an auto-generated random number or a structured prefix to support aggregation, depending on routing flexibility requirements.

The discovery of an AI agent is not predicated on the bit-pattern of the address itself, but rather on rich metadata within the mapping system. This includes records such as Distinguished Names, JSON-encoded capabilities, geo-location data, and traffic engineering constraints, allowing for context-aware resolution beyond simple address matching.

4.2. Attachment points as RLOCs

When an AI agent runs on a host connected to the network, the local xTR registers the AI agent's EID along with one or more RLOCs. Multiple RLOCs enable multi-homing, with each RLOC annotated with capabilities.

When an AI agent operates on a host, the registration of its Endpoint Identifier (EID) with one or more Routing Locators (RLOCs) depends on the deployment architecture:

* xTR co-located with AI agent: In this scenario, the xTR resides within the AI agent's system. The AI agent is assigned a stable EID, while the RLOC is assigned by the current network provider. As the AI agent roams across different locations and network providers, the EID remains constant, but the underlying RLOC changes. The local xTR updates the mapping system with the new EID-to-RLOC binding.

* AI agent behind stationary xTR: In this scenario, the xTR is a fixed infrastructure component (e.g., a router in a data center or site). The xTR typically registers a covering EID-prefix (e.g., /16) representing the entire site. When AI agents move within this local domain, their mobility is handled by the underlay routing within the site, and no updates to the global mapping system are required. Only when an agent moves outside this domain to a different set of xTRs does it need to register its specific EID with the new infrastructure.

4.3. Instance ID for agent VPN groups

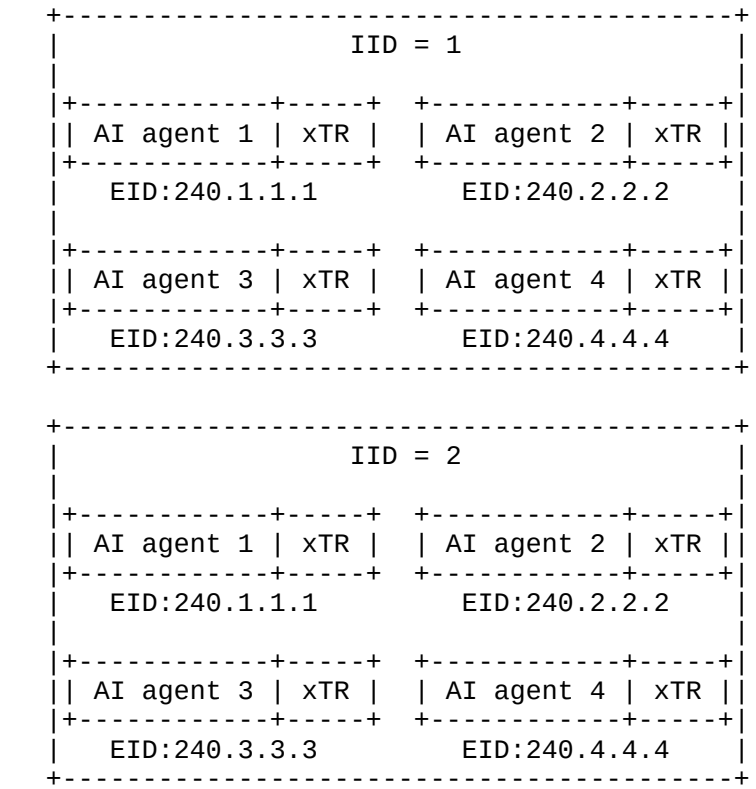

Figure 1 Address overlap by using IID

As shown in Figure 1, LISP Instance IDs (IIDs) [RFC9299] enable multiple virtual networks to operate over the same physical infrastructure, providing scalable and secure multi-tenancy for heterogeneous AI agent workloads. To ensure isolation between different agent VPN groups, especially when EID addressing schemes might overlap. Each agent VPN group is assigned a unique IID.

This mechanism allows for efficient address reuse across isolated domains. For example, a GPU cluster in IID 1 could assign EIDs 240.1.1.1, 240.1.1.2, etc., to its agents. A completely different cluster in IID 2 could reuse those exact same EID prefixes without conflict, as the distinct IID scopes the addressing.

5. Architecture Overview

5.1. The architecture of LISP for AI agent communication.

The LISP provides the network substrate that enables stable identity, mobility, multi-homing, and policy-aware routing for AI agents. It consists of several logically distinct but tightly coordinated components, as illustrated in Figure 2.

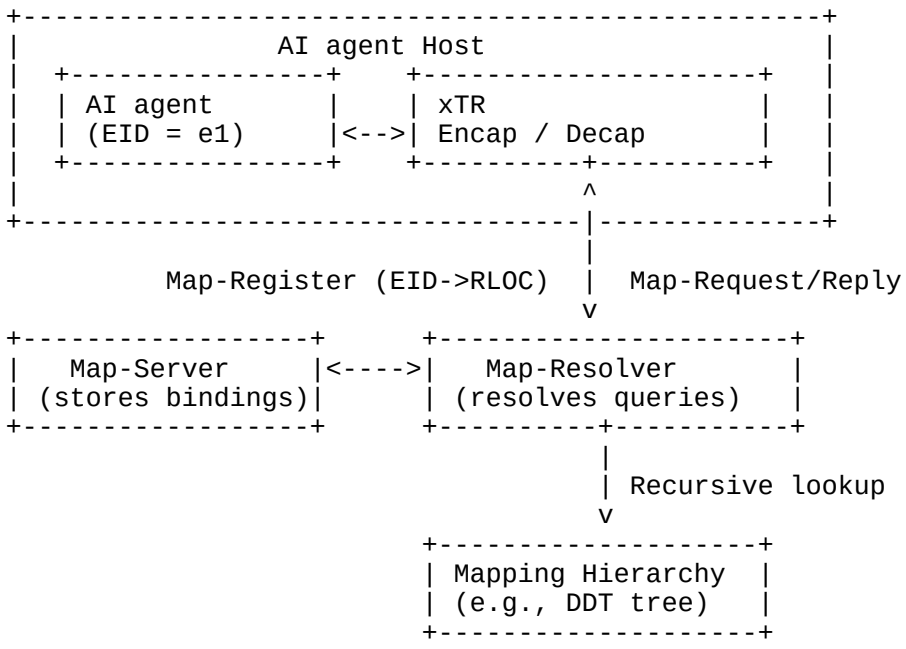

Figure 2 The architecture of LISP for AI agent communication.

5.2. Data Flow Example

The normal data-flow has been described in [RFC9300] and mobility movement has been described in both [I-D.ietf-lisp-mn] and [I-D.ietf-lisp-eid-mobility].

Consider agent A (EID_A) sending a message to agent B (EID_B):

1. Agent A sends a standard IP packet to EID_B.

2. The local xTR (acting as ITR) intercepts the packet.

3. ITR queries the mapping system via a Map-Resolver for EID_B.

4. The mapping system returns a Map-Reply containing one or more RLOCs for EID_B, possibly filtered by context.

5. ITR encapsulates the original packet in a LISP header (with optional IID) and forwards it to the selected RLOC_B.

6. The destination xTR (ETR) decapsulates and delivers the packet to agent B.

If agent B migrates to a new host, it registers its EID with a new RLOC. Subsequent Map-Requests return the updated mapping, and communication resumes transparently.

6. Extending LCAF for capability-aware mapping in AI agent communication

The LISP Canonical Address Format (LCAF) [RFC8060] further extends LISP by allowing EIDs and RLOCs to carry structured, non-IP identifiers such as Instance IDs, application-layer ports, or geographic coordinates, within a type-length-value (TLV) framework.

However, emerging use cases involving autonomous AI agents such as personal assistants, industrial digital twins, and multi-agent collaboration systems introduce new requirements that go beyond traditional network-layer addressing:

* Semantic identities: AI agent identifiers are often URIs or Decentralized Identifiers (DIDs), not IP addresses.

* Dynamic capabilities: An AI agent should support the ability to perform tasks (for example, medical-image-analysis) is context-dependent and must be discoverable.

* Conditional discovery: A caller may wish to discover AI agents that satisfy constraints on latency, location, or security policy, not just a specific EID.

Current LISP mapping mechanisms only support exact-match queries on flat EID spaces. To enable capability-aware service discovery in AI agent communication, we propose an extension to LCAF that allows Map-Request messages to express structured query predicates, and Map-Reply messages to return enriched, filtered results.

6.1. Query Expression LCAF (QE-LCAF)

To encapsulate structured discovery requests, this draft defines a new LCAF type: Query Expression LCAF (QE-LCAF). Its format is shown in Figure 3.

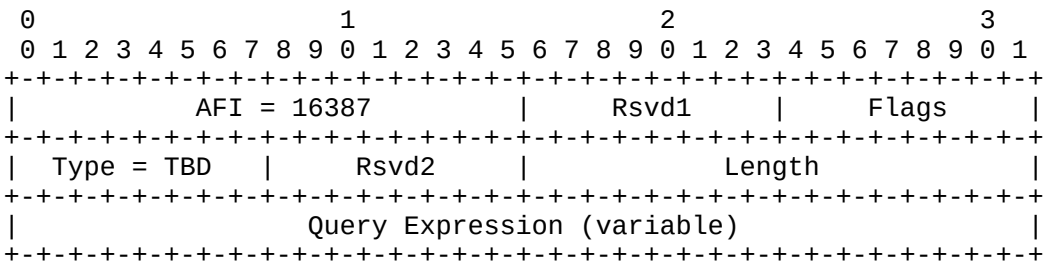

Figure 3 The format of Query Expression LCAF

Where:

* Type: TBD (to be assigned by IANA).

* Flags: Currently unused; set to zero.

* Length: Length of the Query Expression field in bytes.

* Query Expression: A self-describing, serialized query object that may include the target EID, required capabilities, constraints (e.g., maximum latency, service price range), and return fields (e.g., RLOC providing the service, latency, TTL).

Upon receiving such a request, a Map-Resolver or Map-Server:

1. Parses the QE-LCAF;

2. Matches against its local or federated mapping database;

3. Applies filtering based on target EID, required capabilities, and constraints;

4. Constructs a Map-Reply containing one or more matching entries.

The Map-Reply uses AFI-List LCAF to return multiple <EID, RLOC, Metadata> tuples. Each RLOC may itself be encoded in a protocol-specific LCAF (for example, a URI-LCAF, if defined).

To limit response size, the mapping system MAY:

* Return only the top-k results;

* Omit metadata fields not listed in return_fields.

6.2. Protocol Operation

A querier (acting as an ITR) constructs a Map-Request where the requested EID field contains a QE-LCAF instead of a conventional AFI plus EID.

7. Security Considerations

LISP inherits security considerations from [RFC9300]. For AI agent communication, logical isolation via IIDs provides strong tenant separation, reducing cross-domain attack surface.

8. IANA Considerations

This document defines a new LCAF type under the "LISP Canonical Address Format (LCAF) Types" registry group, entitled "Query Expression LCAF (LISP Canonical Address Format)". IANA needs to assign a value to it.

Value	Description	Reference
TBA	Query Expression	This document

9. Normative References

[I-D.ietf-lisp-eid-mobility]
Portoles-Comeras, M., Ashtaputre, V., Maino, F., Moreno, V., and D. Farinacci, "LISP L2/L3 EID Mobility Using a Unified Control Plane", Work in Progress, Internet-Draft, draft-ietf-lisp-eid-mobility-17, 20 October 2025, <https://datatracker.ietf.org/doc/html/draft-ietf-lisp-eid-mobility-17>.

[I-D.ietf-lisp-mn]
Farinacci, D., Lewis, D., Meyer, D., and C. White, "LISP Mobile Node", Work in Progress, Internet-Draft, draft-ietf-lisp-mn-15, 14 January 2024, <https://datatracker.ietf.org/doc/html/draft-ietf-lisp-mn-15>.

[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>.

[RFC8060] Farinacci, D., Meyer, D., and J. Snijders, "LISP Canonical Address Format (LCAF)", RFC 8060, DOI 10.17487/RFC8060, February 2017, <https://www.rfc-editor.org/info/rfc8060>.

[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>.

[RFC9299] Cabellos, A. and D. Saucez, Ed., "An Architectural Introduction to the Locator/ID Separation Protocol (LISP)", RFC 9299, DOI 10.17487/RFC9299, October 2022, <https://www.rfc-editor.org/info/rfc9299>.

[RFC9300] Farinacci, D., Fuller, V., Meyer, D., Lewis, D., and A. Cabellos, Ed., "The Locator/ID Separation Protocol (LISP)", RFC 9300, DOI 10.17487/RFC9300, October 2022, <https://www.rfc-editor.org/info/rfc9300>.

[RFC9301] Farinacci, D., Maino, F., Fuller, V., and A. Cabellos, Ed., "Locator/ID Separation Protocol (LISP) Control Plane", RFC 9301, DOI 10.17487/RFC9301, October 2022, <https://www.rfc-editor.org/info/rfc9301>.

Authors' Addresses

Wei Wang
China Telecom
Beiqijia Town, Changping District
Beijing
Beijing, 102209
China
Email: weiwang94@foxmail.com

Chongfeng Xie
China Telecom
Beiqijia Town, Changping District
Beijing
Beijing, 102209
China
Email: xiechf@chinatelecom.cn