

Network Working Group
Internet-Draft
Intended status: Informational
Expires: 13 October 2026

A. Akhavain
H. Moussa
Huawei Technologies Canada
D. King
Old Dog Consulting
11 April 2026

Problem Statement for the Discovery of Agents, Workloads, and Named
Entities (DAWN)
draft-akhavain-moussa-dawn-problem-statement-00

Abstract

Interacting entities such as agents, tasks, users, workloads, data, compute, etc., in AI ecosystem/network are proliferating, yet there is no standardised way to discover what entities exist, what attributes such as skills, capabilities, physical characteristics, etc., they possess, what services they offer, or how to reach them across organisational boundaries.

Discovery today relies on proprietary directories or manual configuration, creating fragmented ecosystems that prevent cross-domain collaboration.

This document describes the problem space that motivates Discovery of Agents, Workloads, and Named Entities (DAWN). It clarifies the scope of work within entity ecosystems, identifies why current approaches are insufficient, and outlines the challenges a standardised discovery mechanism must address. It does not propose a specific solution or protocol.

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 13 October 2026.

Copyright Notice

Copyright (c) 2026 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

Table of Contents

1. Introduction	3
2. Terminology {sec-terms}	4
3. Motivation	5
3.1. Example of discovery lifecycle in AI ecosystem	6
4. Functional Requirements {sec-func-req}	7
4.1. Discovering entities and query granularity	8
4.2. Discovering response and minimum discoverable information	8
4.3. Cross-Domain Collaboration	8
4.4. Discovery and dynamic attributes in discoverable objects	9
4.5. Broker and Aggregator Discovery	9
4.6. Human-Initiated Discovery	9
4.7. Discovery and OAM	9
5. Current Approaches and Their Limitations	10
5.1. Proprietary Directories	10
5.2. Static Configuration	10
5.3. DNS-SD and SRV Records	10
5.4. Well-Known URIs	10
5.5. Ad Hoc Agent Discovery Proposals	10
6. Core Challenges	10
6.1. Discovering Skills and Capabilities at Scale	10
6.2. Fragmented Discovery Ecosystem	11
6.3. Trust in Discovery Information	11
6.4. Scalability and Decentralisation	11
6.5. Static Versus Dynamic Properties	11
6.6. Extensibility	11
7. Relationship to Existing Work	11
8. Security Considerations	11
9. Privacy Considerations	12
10. Operational Consideration	12
11. IANA Considerations	12

12. Potential topics for the use case document	12
Acknowledgements	12
Authors' Addresses	12

1. Introduction

Entities in entity ecosystem collaborate to render services and follow the lifecycle shown in Figure 1.

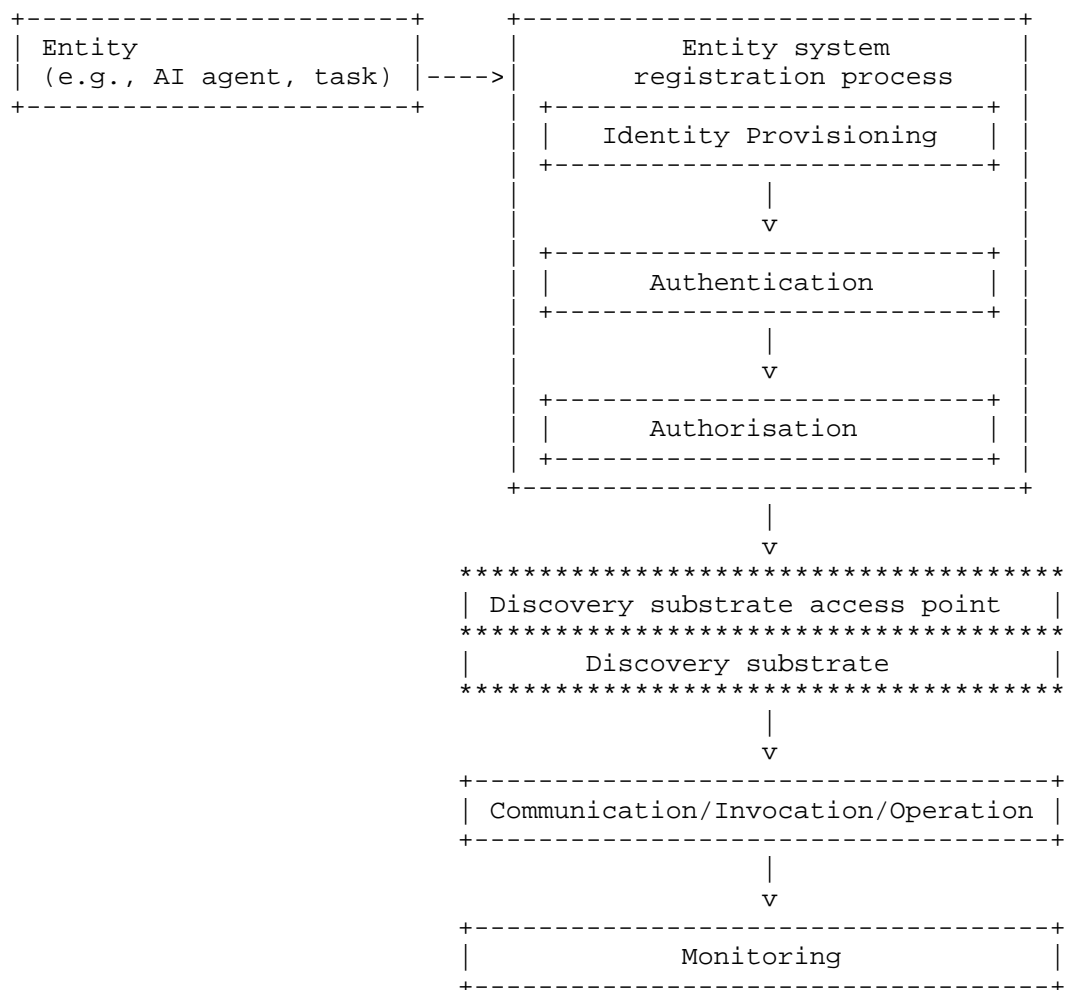


Figure 1: An example of Entity Lifecycle

As shown in Figure 1, an entity will pass through a set of important functional blocks before it becomes active and start interacting with other entities in the ecosystem. This document focuses on the discovery problem space in the above diagram namely: "Discovery substrate access point" and "Discovery substrate".

Entities increasingly need to discover, connect with, and collaborate with one another to deliver services. This discovery process is driven by the need to identify the most suitable set of entities that satisfy the requirements of a particular service. To achieve this, an entity must be able to find others based on attributes such as skills, capabilities, physical characteristics, names, and other relevant qualities they possess. Obviously, as static configuration is impractical at scale, an automated discovery of entities, their skills, and their capabilities becomes essential.

Discovery within an AI ecosystem can be multi-dimensional and complex. A discovery request may trigger a cascade of subsequent discovery requests by other AI entities, occurring either sequentially or in parallel and the process might become unbounded. In addition, the discovery step can be interactive. For example, an entity might be looking for another entity that might not be available at the time of request (e.g., the desired entity might be busy). Furthermore, entities might be looking for a variety of other entities with different cards/descriptors. Discovery might also be subjected to either a system wide or local policy and might span cross organisation. There also challenges w.r.t the nature of discovery request itself as will be explained later in this document.

Assuming that trust has already been established between entities and within the ecosystem in the steps prior to the discovery stage, the discovering entity must learn what the remote entity does, what attributes it possesses, how to communicate with it, etc.

This document describes the problem space and informs the development of requirements set out in [draft-king-dawn-requirements] and future solution proposals for Discovery of Agents, Workloads, and Named Entities (DAWN).

2. Terminology {sec-terms}

The following terms are used in this document. Further definitions are provided in [draft-king-dawn-requirements].

- * **Attributes:** The term attributes refers to properties, features, capabilities, skills, etc., that an entity possesses or may have access to such as skill type, communication language, capacity, task description, contact information, ID, etc.

- * Capability exchange / negotiation: The processes by which agents can exchange details of what they can do, dynamic information about statuses (including agents, tasks, and workloads), and which particular features/functions they want to engage. [Hesham]{This is a step outside the basic discovery step and may be carried out after entities discover the MDI that allow them to connect later on. Hence it is outside the scope of the problem space}
- * Discoverable object: An informative object that is discoverable and captures necessary information that defines what an entity is, what attributes it possess, how to reach to the associated entity, etc. Examples include agent cards, task cards, resource cards, tool cards, and skill cards.
- * Discoverable object validation: The process that verifies a discoverable object, ensures its compliance to standards, and make it available to the discovery substrate.
- * Discovery: A process by which an entity can find another entity or a set of entities of a type that can perform a desired function. The purpose of this work effort is limited to just this element. It is described in more detail in the Problem Statement and the Functional Requirements, below.
- * Entity: A system component that communicates at a peer-to-peer or client-server level with another entity within the AI ecosystem. Examples include, AI agents, tasks, tools, skills, task owners, workloads, and services.
- * Function: The functional processing capability that an entity offers. Examples include, tasks, workloads, endpoints, jobs, services, tools.
- * Minimum Discoverable Information (MDI): The minimum amount of information an entity needs to provide to become discoverable. Think of it as common header of a data structure.

3. Motivation

The main motivation behind DAWN is to tackle the discovery problem space within the entity ecosystem. It is driven by a few factors:

- * Discovery use cases in real-world
 - Many practical scenarios require discovery, not only for agent-to-agent, but also agent-to-tools, agent-to-task, task-to-agent, and other forms of entity interaction.

- * Limitations of traditional discovery methods
 - Existing discovery mechanisms are not designed to natively handle scenarios where entities are dynamic, mobile, cross-domain, or when they have complex attributes.
- * Current approaches are ad-hoc, entity specific, and do not scale:
 - Even in today's implementations (e.g., MCP-based systems or A2A-based systems), discovery tends to be contained and handled through simple mechanisms such as name lookup, search engines, or static agent cards/tool cards. These approaches work only in small, closed environments. They do not address challenges such as inter-domain discovery, dynamic endpoint association, chained discovery queries, blind or exploratory search sessions, or large-scale environments. In addition, they do not address the need of other discoverable entities such as task, workloads, etc.
- * Emergence of discoverable entities, discoverable objects, and discovery mechanism:
 - Entities may have associated MDIs (e.g., task , capabilities, endpoints, policies), and that a discovery substrate/mechanism/vehicle is needed. The discovery substrate may implement unified mechanism or may support multiple discovery strategies depending on the scenario.

3.1. Example of discovery lifecycle in AI ecosystem

Consider a task owner (e.g., an entity such as an end user, AI agent, model, data owner, resource/compute owner) which intends to submit a task to the ecosystem and, as shown in Figure 1, has already been processed and accepted by the entity registration block. The following describes the steps after which the entity becomes available for discovery.

1. Discovery substrate access point validates the task owner's credentials and verifies that its associated discoverable object meets compliance requirements. The discoverable object is what the discovery substrate makes available/visible to system participants. It contains entity's different attributes and information that others need to initiate an interaction session with it once they discover the entity.

2. Task owner submits its tasks to the system. Submitted tasks are entities themselves. They have their own discoverable object (task card in this case) which the discovery substrates makes available/visible to other entities in the ecosystem once submitted tasks pass through the entity registration block.
3. A registered entity (e.g., an AI agent) then issues a discovery query to identify and/or locate suitable tasks it can perform, or to find other agents, resources, etc., it must interact with to complete a given task.
4. The discovery substrate processes the above query and returns the relevant discoverable objects such as tasks, agents, resources, etc., to the entity that issued the query. It must be noted that the nature and structure of the query, the format of the discoverable objects (e.g., standardised object cards), and the discovery mechanism employed (e.g., simple name lookup or semantic matching) are key factors influencing the accuracy, volume, timeliness, etc., of the results.

For example, the querying entity may need to provide details about its skills, capabilities, pricing, or other relevant attributes so the discovery substrate can match its request with an appropriate subset of registered entities in the system.

5. Upon receiving the discovery results (e.g., a list of suitable entities), the querying entity (e.g., an AI agent) might need additional information before initiating its interaction with the discovered entities. For example, it might need to know more about the parent entity of the discovered entity whose name/ID can be potentially found in the discovered entity's discoverable object.

The example above illustrates the broader concept of discovery within an ecosystem. Other factors such as entity's mobility can further complicate the problem space. The example, underscore the significance and complexity of the problem space that DAWN aims to address. It highlights why a structured problem definition, clear requirements, and well-designed solutions are essential for enabling robust, scalable, and interoperable discovery across diverse entities and use cases.

4. Functional Requirements {sec-func-req}

4.1. Discovering entities and query granularity

Discovery in ecosystem should support different levels of granularity. Queries may range from broad capability-based searches (such as identifying all models with mathematical abilities) to more specific lookups. The discovery system should also enable entities to be found through the attributes reflected in their discoverable objects that capture aspects like their skill sets, functionality, name/ID, ratings, regional associations, and more.

4.2. Discovering response and minimum discoverable information

Information an entity discovers about another entity must be meaningful and useful for delivering the required service. Accordingly, a response to a discovery query should include attributes that describe the discovered entity: such as what it can do, the skills it possesses, the protocols it supports, the security guarantees it claims to offer, the policies it can potentially enforce, its pricing for services, its current operational status (e.g., available, busy, or offline), communication means, etc.

Such information can be either embedded within the entity's discoverable object or retrieved through a subsequent interaction outside the discovery substrate (for example, after discovery, an interview-style exchange may be conducted using the communication method indicated by the entity).

In either case, there is a need for a standardized structure for discoverable objects that provides the minimum set of information needed for the discovery substrate to return results that meaningfully support service delivery within the AI ecosystem.

4.3. Cross-Domain Collaboration

Entities operating across organisational boundaries need to discover counterparts without depending on a shared infrastructure. For example, a customer-service agent in one organisation may need to find a logistics-tracking agent in another. Models in one administrative domain may need to find compute resources in another administrative domain for training. Similarly, a model or agent in one domain might need to use data in another domain for retrieval-augmented generation (RAG) based inference. Current platform-specific mechanisms do not interoperate, so entities remain invisible outside their own ecosystem.

Administrative domains are typically unwilling to disclose their internal structures or detailed operational information to one another. In traditional networking, for instance, they use

abstraction and aggregation techniques to share only high-level insights about their operations. A standards-based mechanism to support controlled information sharing while ensuring administrative domain interoperability without exposing sensitive internal details is potentially desirable.

4.4. Discovery and dynamic attributes in discoverable objects

Entities whose discoverable objects contain dynamic attributes introduce distinct challenges for discovery. Dynamic attributes such as location information, dataset samples, compute capacity, etc., can change at different rates. These dynamics introduce variability that static discovery systems are not designed to handle. Such dynamic attributes complicate the assumptions in traditional discovery approaches and demand careful consideration when defining the problem space.

4.5. Broker and Aggregator Discovery

In large-scale networks, entities may need to discover intermediary broker nodes that operate across multiple administrative domains and provide dynamic operational information such as availability, capabilities, or decision guidance via the use of mechanisms that support interoperable and standards-compliant discovery procedures.

In large-scale networks, entities may need to discover intermediary broker nodes. These brokers often operate across multiple administrative domains with different jurisdictions. They also provide dynamic operational information, such as availability, capabilities, or decision guidance. In these scenarios, the intermediary brokers might need to discover other brokers. This makes the broker nodes another type of entity with its own discoverable object in an ecosystem. Discovery substrate needs to provide support for this capability via standards-compliant procedures.

4.6. Human-Initiated Discovery

Operators need to discover and inspect entities for operational purposes: auditing deployed agents, verifying capability claims, or troubleshooting failures. Discovery must be usable by humans through standard tooling, not only by automated systems.

4.7. Discovery and OAM

TBD

5. Current Approaches and Their Limitations

5.1. Proprietary Directories

Cloud providers, AI platforms, and other entity systems maintain their own registries, tightly coupled to their ecosystem. Entities registered in one platform are invisible to another, creating walled gardens.

5.2. Static Configuration

Manually configured endpoint lists cannot scale, cannot adapt to dynamic environments, and cannot convey the capability and trust metadata needed for cross-domain discovery.

5.3. DNS-SD and SRV Records

TBD

5.4. Well-Known URIs

TBD

5.5. Ad Hoc Agent Discovery Proposals

TBD

6. Core Challenges

6.1. Discovering Skills and Capabilities at Scale

The central challenge is enabling entities to discover other entities based on what they can do, such as:

- * Agents
- * Skills
- * Capabilities
- * TBA

A discovery mechanism that supports structured, scalable discovery of an entity's capabilities across organisational boundaries is therefore required.

6.2. Fragmented Discovery Ecosystem

Each platform develops its own discovery approach. This fragmentation prevents entities from being discoverable across boundaries and limits the value of interoperable protocols such as A2A and MCP.

6.3. Trust in Discovery Information

When discovery crosses organisational boundaries, the discovering entity must verify that the information is authentic. Without authenticated discovery, entities are vulnerable to poisoning attacks directing them to malicious endpoints. DNSSEC provides a foundation, but discovery mechanisms must be designed to use it.

6.4. Scalability and Decentralisation

Discovery must operate at Internet scale without a single centralised registry. Each organisation must be able to publish its entities' capabilities independently, mirroring the DNS delegation model.

6.5. Static Versus Dynamic Properties

Entity properties range from static (type, supported protocols, skills) to dynamic (availability, load, capacity). A discovery mechanism must handle both without causing stale results or excessive query load.

6.6. Extensibility

New agent types, skill taxonomies, and capability formats will emerge. Discovery must accommodate them without changes to the core mechanism.

7. Relationship to Existing Work

TBD

8. Security Considerations

This document describes a problem space, not a protocol.

Discovery information is a high-value target. Poisoned responses could direct entities to malicious endpoints. Any mechanism must provide integrity and authenticity guarantees.

Cross-domain discovery raises two distinct trust questions: whether the discovery source is authoritative, and whether the registered entity is what it claims to be.

Discovery may expose sensitive information about an organisation's entities and capabilities. Selective visibility mechanisms are needed.

9. Privacy Considerations

Querying for entities may reveal the discovering entity's intentions or interests. Discovery should minimise information leakage through the query process.

Published entity properties, such as skills, capabilities, and organisational affiliations, may be sensitive. Entities and their operators should control the granularity and audience of published information.

10. Operational Consideration

TBD

11. IANA Considerations

This document makes no requests of IANA.

12. Potential topics for the use case document

TBD

Acknowledgements

Thanks to Adrian Farrel for review comments.

Authors' Addresses

Arashmid Akhavain
Huawei Technologies Canada
Canada
Email: arashmid.akhavain@huawei.com

Hesham Moussa
Huawei Technologies Canada
Canada
Email: hesham.moussa@huawei.com

Daniel King
Old Dog Consulting
United Kingdom
Email: daniel@olddog.co.uk