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

D. King
A. Farrel
Old Dog Consulting

11 April 2026

Requirements for the Discovery of Agents, Workloads, and Named Entities (DAWN)
draft-king-dawn-requirements-00

Abstract

The proliferation of distributed systems, Artificial Intelligence (AI) agents, cloud workloads, and network services has created a need for interoperable mechanisms to discover entities across administrative and network boundaries. Entities may include AI agents, software services, compute workloads, and other named resources that need to be found and characterised before interaction can begin.

This document defines the requirements for Discovery of Agents, Workloads, and Named Entities (DAWN) and sets out the objectives that a discovery mechanism for such entities must satisfy. It describes what information must be discoverable, what properties a discovery mechanism needs to support, and what constraints apply to discovery in decentralised environments.

This document does not specify any particular discovery protocol or solution.

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.

Table of Contents

1. Introduction

Modern distributed systems increasingly rely on the dynamic composition of services, agents, and workloads that may not have pre-configured relationships. For example, an AI agent may need to find another agent with specific capabilities, a workload orchestrator may need to locate compute resources in a particular jurisdiction, or a service consumer may need to discover providers that support a required protocol version. Further use cases and scenarios are expected to be documented separately.

In each case, an entity needs knowledge of remote entities before interaction can proceed: what they are, what they offer, and whether they can be trusted. Such knowledge could be obtained through static configuration, but this approach is impractical at scale and across organisational boundaries. Automated discovery mechanisms are therefore needed.

Today, where automated discovery exists, it is typically handled through proprietary directories or platform-specific mechanisms. These approaches do not scale across organisational boundaries and create fragmented ecosystems where entities cannot find entities managed by other organisations.

An interoperable discovery mechanism is needed that builds on existing protocols and tools, benefits from an established trust model, supports proven delegation and federation architectures, and allows organisations to independently publish discovery information.

This document defines requirements that any Discovery of Agents, Workloads, and Named Entities (DAWN) mechanism must satisfy. It is informed by:

* [I-D.akhavain-moussa-dawn-problem-statement] DAWN Problem Statement I-D.

1.1. Scope

The requirements in this document address what information must be discoverable about entities, what properties a discovery mechanism must support, and what architectural constraints apply. The detailed requirements are set out in Section 3.

The following topics are explicitly out of scope:

* Entity registration processes, including attestation and other security mechanisms for registration;

* Design, definition, and governance of naming systems for entities;

* Trust, authentication, and authorisation of entities themselves (as distinct from trust in discovery information);

* Capability exchange and negotiation between entities;

* Entity selection mechanisms and policies;

* Task management and orchestration;

* Agent-to-agent communication protocols.

2. Terminology

The following terms are used throughout this document. It is expected that these definitions will be consolidated into a common terminology document for the DAWN work.

Agent: An entity that acts autonomously or semi-autonomously on behalf of a user, organisation, or system. An agent may initiate interactions with other entities, make decisions, and perform tasks. AI agents are a specific class of agent that employ artificial intelligence techniques.

Capability: A description of the functions, services, or actions that an entity can perform. Capabilities may be described using structured schemas such as capability cards.

Capability Card: A structured, machine-readable description of an entity's capabilities and interface. Variants include agent cards, task cards, resource cards, tool cards, and skill cards depending on the type of entity.

Capability Exchange: The processes by which entities exchange details of what they can do, dynamic status information, and which particular features or functions they wish to engage. Capability exchange and negotiation are out of scope for this document.

Discovered Entity: An entity whose properties are returned as the result of a discovery query. A discovered entity may be a specific instance or a member of a class of entities that can perform a desired function.

Discovering Entity: An entity (or its operator) that initiates the discovery process in order to find other entities to interact with.

Discovery: The process by which an entity or its operator locates other entities that are capable of performing a desired function or providing a desired service, and obtains sufficient information to initiate interaction.

Discovery Mechanism: A protocol, system, or method used to perform discovery. Examples include DNS-based service discovery, directory services, and distributed registries.

Entity: A system component that communicates with other entities in

a peer-to-peer or client-server relationship. Entities include, but are not limited to, AI agents, software services, compute workloads, network functions, and application endpoints.

Function: The functional processing capability that an entity offers. Examples include task execution, data transformation, inference, routing, storage, and orchestration.

Named Entity: An entity that is identified by a stable name within a naming system. The naming system may be hierarchical (e.g., the Domain Name System (DNS)) or flat.

Properties: Discoverable characteristics of an entity. Properties include, but are not limited to, communication protocols, capability descriptions, location, trust indicators, and operational status.

Registrar: An entity or system responsible for accepting and maintaining records about entities that wish to be discoverable. Registration itself is out of scope for this document, but the requirements herein may constrain how a registrar exposes information.

Selection: The mechanisms and policies by which an entity determines which discovered entities it will interact with. Selection is out of scope for this document but depends on information obtained through discovery.

Trust Indicator: Information associated with an entity that allows a discovering party to assess the trustworthiness or provenance of the entity and its advertised properties. Examples include digital signatures, certificates, and attestations.

Workload: A unit of compute or processing that is deployed and executed within a hosting environment. Workloads may be transient or long-lived and may move between hosting environments.

3. Requirements

The requirements are organised into the following categories: discovery actors and scenarios, entity classification, entity properties, trust and security, scalability and architecture, discovery protocol, and extensibility.

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.

3.1. Discovery Actors and Scenarios

REQ-DISC-1: A discovery mechanism MUST support discovery initiated by any type of entity, including agents, services, workloads, and human operators.

REQ-DISC-2: A discovery mechanism MUST support the discovery of both specific entity instances and classes of entities that can perform a desired function.

REQ-DISC-3: A discovery mechanism MUST identify the primary scenario groupings (categories) of entities where discovery is needed and address the specific discovery requirements of each category.

REQ-DISC-4: A discovery mechanism MUST define where discovery fits within the overall workflow of entity interaction, distinguishing discovery from registration, selection, and capability exchange.

REQ-DISC-5: A discovery mechanism SHOULD support discovery of intermediary aggregation points or brokers that can provide further dynamic information about entities.

3.2. Entity Classification

REQ-CLASS-1: A discovery mechanism MUST allow entities to be classified by type (e.g., AI agent, service, workload, network function).

REQ-CLASS-2: A discovery mechanism MUST allow entities to be classified by the function or capability they provide.

REQ-CLASS-3: A discovery mechanism SHOULD allow entities to be associated with a geographic or jurisdictional location.

REQ-CLASS-4: A discovery mechanism SHOULD allow entities to be associated with an owning or operating organisation.

REQ-CLASS-5: A discovery mechanism SHOULD allow entities to be associated with a source or origin (e.g., the standards body, vendor, or open-source project that defined the entity’s interface).

3.3. Entity Properties

REQ-PROP-1: A discovery mechanism MUST define a set of mandatory base properties that every discoverable entity provides.

REQ-PROP-2: A discovery mechanism MUST support the discovery of communication protocols and transport parameters needed to interact with an entity.

REQ-PROP-3: A discovery mechanism MUST support the discovery of capability descriptions for an entity.

REQ-PROP-4: A discovery mechanism MUST distinguish between static properties (e.g., entity type, supported protocols) and dynamic properties (e.g., operational status, current load).

REQ-PROP-5: A discovery mechanism SHOULD support the association of version information with entity properties and capability descriptions.

REQ-PROP-6: A discovery mechanism MUST support the identification of the party responsible for an entity. The mechanism SHOULD also allow entities to be registered anonymously where appropriate.

REQ-PROP-7: A discovery mechanism SHOULD support the discovery of an entity’s functional capacity, such as the scope or volume of work it can perform.

REQ-PROP-8: A discovery mechanism MUST support the discovery of security-related communication parameters needed to establish a secure connection with an entity, either directly (e.g., through protocol-level fields) or by reference to an external capability descriptor that contains details such as supported Transport Layer Security (TLS) versions and authentication methods.

REQ-PROP-9: A discovery mechanism MUST support the discovery of capability cards or equivalent structured, machine-readable descriptions of an entity’s interface and functions.

REQ-PROP-10: A discovery mechanism MUST categorise each discoverable property as mandatory or optional, so that consumers of discovery information can determine which properties are guaranteed to be present.

REQ-PROP-11: A discovery mechanism MUST indicate whether each property is static, mainly static, or dynamic, so that consumers can determine appropriate caching and refresh strategies.

3.4. Trust and Security

REQ-SEC-1: A discovery mechanism MUST provide a means to verify the authenticity and integrity of discovery information.

REQ-SEC-2: A discovery mechanism MUST support the use of cryptographic trust indicators (e.g., digital signatures, certificates) to establish the provenance of entity information.

REQ-SEC-3: A discovery mechanism MUST be resilient to attacks that could poison or corrupt discovery information.

REQ-SEC-4: A discovery mechanism SHOULD allow an entity to control the visibility of its properties to different audiences (e.g., public versus organisation-internal discovery).

REQ-SEC-5: A discovery mechanism SHOULD support operation across trust boundaries without requiring a single global trust anchor.

3.5. Scalability and Architecture

REQ-ARCH-1: A discovery mechanism MUST be capable of operating in decentralised architectures.

REQ-ARCH-2: A discovery mechanism MUST NOT require a single centralised registry as a prerequisite for operation.

REQ-ARCH-3: A discovery mechanism MUST scale to support discovery across a large number of entities and administrative domains.

REQ-ARCH-4: A discovery mechanism SHOULD allow organisations to independently publish discovery information without depending on a third-party directory.

REQ-ARCH-5: A discovery mechanism SHOULD support discovery across heterogeneous network environments, including cloud, edge, and enterprise networks.

3.6. Discovery Protocol

REQ-PROTO-1: A discovery mechanism MUST define the protocol(s) used by discovering entities to communicate with discovery enablers (e.g., discovery servers, directories, or DNS resolvers).

REQ-PROTO-2: A discovery mechanism SHOULD be built in a modular way using existing Internet Engineering Task Force (IETF) protocols where possible, filling gaps only where existing protocols are insufficient.

REQ-PROTO-3: A discovery mechanism SHOULD support different protocols for different discovery scenarios where a single protocol cannot efficiently serve all use cases.

REQ-PROTO-4: A discovery mechanism MUST define a predictable entry point for discovery that is based on ubiquitous and interoperable mechanisms.

3.7. Extensibility

REQ-EXT-1: A discovery mechanism MUST support the addition of new entity types and property definitions without requiring changes to the core mechanism.

REQ-EXT-2: A discovery mechanism MUST support structured, versioned schemas for entity properties to enable backward-compatible evolution.

REQ-EXT-3: A discovery mechanism SHOULD allow domain-specific or industry-specific extensions to entity properties.

4. IANA Considerations

This document does not make any requests of IANA.

5. Security Considerations

This document defines requirements for entity discovery mechanisms. It does not define a protocol and therefore does not introduce specific security vulnerabilities. However, the requirements in Section 3.4 place security constraints on any solution that satisfies these requirements.

Implementers of discovery mechanisms that satisfy these requirements should pay particular attention to the following concerns:

* The integrity and authenticity of discovery information must be protected to prevent poisoning attacks that could direct entities to malicious endpoints.

* Access control mechanisms should be considered to prevent unauthorised disclosure of entity properties, particularly in environments where entity metadata may be sensitive.

* The discovery mechanism itself must not become a vector for denial-of-service attacks against the infrastructure on which it is built.

6. Privacy Considerations

Discovery mechanisms inherently involve the publication of information about entities. Implementers should consider the privacy implications of exposing entity properties, capabilities, and organisational associations. In particular:

* Entities should be able to control what information is made publicly discoverable versus restricted to specific audiences.

* The discovery mechanism should not require the disclosure of information beyond what is necessary for a discovering entity to determine whether interaction is appropriate.

* Where entities represent individuals or process personal data, compliance with applicable data protection regulations should be considered.

7. Acknowledgements

The authors wish to acknowledge the contributions of participants in athe DAWN discussions that shaped this document.

8. References

8.1. Normative References

[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/rfc/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/rfc/rfc8174>.

8.2. Informative References

[I-D.akhavain-moussa-dawn-problem-statement] Akhavain, A., Moussa, H., and D. King, "Problem Statement for the Discovery of Agents, Workloads, and Named Entities (DAWN)", Work in Progress, Internet-Draft, draft-akhavain-moussa-dawn-problem-statement-00, 11 April 2026, <https://datatracker.ietf.org/doc/html/draft-akhavain-moussa-dawn-problem-statement-00>.

Authors' Addresses

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

Adrian Farrel
Old Dog Consulting
United Kingdom
Email: adrian@olddog.co.uk