

ANIMA  
Internet-Draft  
Intended status: Standards Track  
Expires: 3 September 2026

L. Zhu  
B. Wang  
S. Jiang  
BUPT  
2 March 2026

Definition of Service Intent in Autonomic Networks  
draft-zhu-anima-service-intent-00

## Abstract

While ANIMA Intent enables goal-oriented control within an Autonomic Domain, emerging services (e.g., AI inference) require a common, interoperable representation for expressing service-level objectives and constraints that span network, compute, and storage resources, rather than connection-centric descriptions. This document defines Service Intent for Autonomic Networks by specifying a structured semantic model and a concise format with identification, scope, versioning, and lifecycle semantics.

## 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 September 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.

This document may contain material from IETF Documents or IETF Contributions published or made publicly available before November 10, 2008. The person(s) controlling the copyright in some of this material may not have granted the IETF Trust the right to allow modifications of such material outside the IETF Standards Process. Without obtaining an adequate license from the person(s) controlling the copyright in such materials, this document may not be modified outside the IETF Standards Process, and derivative works of it may not be created outside the IETF Standards Process, except to format it for publication as an RFC or to translate it into languages other than English.

## Table of Contents

|  |    |
|--|----|
| 1. Introduction . . . . .                            | 2  |
| 2. Requirements . . . . .                            | 3  |
| 3. Definition of Service Intent . . . . .            | 4  |
| 3.1. Overview and Metadata . . . . .                 | 4  |
| 3.2. Resource Requirement Intents . . . . .          | 6  |
| 3.2.1. Network Resource Requirement Intent . . . . . | 6  |
| 3.2.2. Compute Resource Requirement Intent . . . . . | 7  |
| 3.2.3. Storage Resource Requirement Intent . . . . . | 8  |
| 4. IANA Considerations . . . . .                     | 9  |
| 5. Security Considerations . . . . .                 | 9  |
| 6. References . . . . .                              | 9  |
| 6.1. Normative References . . . . .                  | 9  |
| 6.2. Informative References . . . . .                | 9  |
| Authors' Addresses . . . . .                         | 10 |

## 1. Introduction

Within the ANIMA framework, intent is an abstract, declarative, high-level policy applicable to an Autonomic Domain. Expressed in terms of desired goals and constraints, intent steers network operation rather than acting as device-level configuration commands. It is distributed across the domain and interpreted by autonomic nodes, which translate it into target configurations while considering local state and environmental constraints. This approach enables closed-loop automation and reduces the operational burden on human operators by shifting from imperative configuration to goal-oriented management.

However, emerging services increasingly rely on the coordinated use of heterogeneous resources beyond simple connectivity. Traditional connection-centric service descriptions, which primarily specify endpoints and basic transport characteristics, are insufficient to represent user-perceived quality objectives for complex tasks such as AI inference, interactive media processing, and large-scale content rendering. These services are often constrained not only by network performance (e.g., bandwidth, latency, and jitter), but also by compute availability (e.g., processing capacity and placement-induced request latency) and storage capabilities (e.g., capacity and access throughput). Moreover, practical deployment decisions frequently require expressing cross-resource coordination constraints, such as coupling compute placement with transport path selection, or accounting for the time cost of data access along selected paths.

This document introduces and defines the concept of Service Intent for Autonomic Networks by analyzing the characteristics of user service demand expressed as intent. The aim is to provide a structured, declarative, semantic framework that enables service-level requirements and constraints across network, compute, and storage dimensions to be specified consistently. By enabling the autonomic interpretation of these requirements within an Autonomic Domain, the Service Intent model facilitates the automated discovery, selection, routing, and scheduling of resources, thereby supporting the automated deployment of services and reducing the complexity of using heterogeneous resources. The detailed mechanisms for realizing intent interpretation and automated service deployment are out of scope for this document and are described in [I-D.ietf-anima-network-service-auto-deployment] and [draft-du-anima-service-intent-auto-deployment].

In addition, this document specifies a concrete, machine-readable format for Service Intent to support interoperable intent exchange and processing within an Autonomic Domain, including essential fields for identification, scoping, versioning, and lifecycle management, and a structured representation of intent content aligned with network-, compute-, and storage-resource requirement intents.

## 2. Requirements

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. Definition of Service Intent

This section defines a Service Intent as a structured, machine-interpretable declaration of the desired service objectives and associated resource constraints for autonomic service deployment within Autonomic Networks.

#### 3.1. Overview and Metadata

Unlike traditional connectivity-oriented service descriptions that primarily specify endpoints and transport parameters, a Service Intent expresses coordinated, multi-dimensional requirements across heterogeneous infrastructure resources. These requirements MAY span compute resources, storage resources, transport resources, and cross-layer or cross-domain constraints. By abstracting the desired service outcome rather than specific configuration actions, a Service Intent enables automated resource discovery, path computation, compute scheduling, and storage placement, thereby supporting goal-driven service orchestration and deployment.

The Service Intent model defined herein is outcome-oriented rather than configuration-oriented. It specifies the intended operational state of a service without mandating implementation procedures, device-level configurations, or specific realization mechanisms.

The Service Intent consists of the following fields:

```
service-intent = {  
  "intent-id"      : uint,  
  "intent-version" : uint,  
  "intent-timestamp" : uint,          ; Epoch time  
  "intent-scope"   : scope-obj,      ; in seconds  
  "intent-content" : content-obj  
}
```

- \* Intent Identifier(unsigned integer). The Intent Identifier uniquely identifies an intent instance within its applicable scope. It MUST be unique within the defined scope and MUST remain stable across version updates of the same intent.
- \* Intent Version(unsigned integer). The Intent Version indicates the revision of a Service Intent. It MUST increase monotonically upon modification and MUST enable determination of the most recent intent state.
- \* Intent Timestamp(unsigned integer, epoch time). The Intent Timestamp indicates the time at which a Service Intent instance was created, issued, or last updated.

- \* Intent Lifetime(unsigned integer, seconds). The Intent Lifetime defines the temporal validity period during which a Service Intent is considered active, observable, and eligible for evaluation or enforcement within its declared Intent Scope. A Service Intent instance MUST be considered expired when its defined lifetime condition is met.
- \* Intent Scope (object). The Intent Scope defines the applicability scope of a Service Intent. It MUST identify at least one Autonomic Domain in which the intent is valid, and MAY further restrict applicability to specific sub-domains, logical partitions, tenants, service slices, or topology segments. An autonomic node MUST only process the Service Intent applicable within its operational scope.

```
scope-obj = {  
  "domain"           : tstr,  
  ? "subdomain"      : tstr,  
  ? "tenant"         : tstr,  
  ? "topology-segment" : tstr  
}
```

- \* Intent Content (object). A Service Intent MUST be expressed as a structured object that MAY include one or more of the following logical components defined in Section 3.2: a) Network resource requirement intent; b) Compute resource requirement intent; c) Storage resource requirement intent. Each component declaratively specifies the desired target state of the corresponding resource dimension and MAY include inter-resource dependency constraints. The content-obj MAY include any subset of the three members, depending on the service scenario. Processing of content-obj MUST handle each present member independently according to the semantics defined in Section 3.2, and SHOULD combine multiple present members to derive a consistent multi-resource provisioning decision. If none of the three members is present, content-obj is invalid and MUST be ignored. Any unrecognized additional member MUST be ignored and MUST NOT cause failure in processing of recognized members.

```
content-obj = {  
  ? "network" : network-intent,  
  ? "compute" : compute-intent,  
  ? "storage" : storage-intent  
}
```

### 3.2. Resource Requirement Intents

#### 3.2.1. Network Resource Requirement Intent

The Network resource requirement intent specifies the required network resource capabilities necessary to satisfy the service. It expresses quantitative and qualitative requirements over transport resources within the declared Intent Scope. These requirements define the desired performance, availability, and forwarding characteristics of network resources allocated for the service.

The Network resource requirement intent includes the following fields:

```
network-intent = {  
  "Bandwidth-Requirement" : uint,      ; Bandwidth Requirement(bps)  
  "latency-bound"         : uint,      ; Latency Bound(ms)  
  "jitter-bound"          : uint,      ; Jitter Bound(ms)  
  "destinations"          : 1* tstr,   ; non-empty array of text strings  
  "multipath-permission"  : bool,      ; TRUE allows multipath  
}
```

- \* Bandwidth Requirement (unsigned integer, bps). The minimum or guaranteed bandwidth required. The value MUST be greater than or equal to 0; if it is specified as zero, no minimum bandwidth constraint is imposed.
- \* Latency Bound (unsigned integer, ms). The maximum acceptable end-to-end delay. The value MUST be greater than or equal to 0; if it is specified as zero, no latency constraint is imposed.
- \* Jitter Bound (unsigned integer, ms). The maximum acceptable delay variation. The value MUST be greater than or equal to 0; if it is specified as zero, no jitter constraint is imposed.
- \* Destinations (array of text strings, non-empty). Target endpoint or location identifiers; MUST contain at least one element.
- \* Multipath Permission (boolean). A boolean indicator specifying whether multipath transport MAY be used; TRUE allows multipath, FALSE forbids multipath.

### 3.2.2. Compute Resource Requirement Intent

The Compute resource requirement intent specifies the required computational resource capabilities necessary to satisfy the service. It expresses quantitative and qualitative requirements over compute resources within the declared Intent Scope. These requirements define the desired processing capacity, placement constraints, and latency characteristics of compute resources allocated for the service.

The Compute resource requirement intent includes the following fields.

```
compute-intent = {  
  ; Request Latency Bound(ms)  
  "request-latency"      : uint,  
  
  ; TRUE allows concurrent transport/compute  
  "transport-coordination" : bool,  
  
  ; Compute Capacity Requirement  
  "compute-capacity"      : 1* compute-capa  
}
```

```
compute-capa = {  
  ; "cpu" or "gpu"  
  "type"      : ("cpu" / "gpu"),  
  
  ; Compute capacity value(GFLOPS)  
  "capacity"  : uint  
}
```

- \* Request Latency Bound (unsigned integer, ms). The maximum acceptable request-to-response latency between the service and the compute resource. The value MUST be greater than or equal to 0; if it is specified as zero, no latency constraint is imposed.
- \* Transport Coordination (boolean). A boolean indicator that specifies whether compute processing and data transport are permitted to proceed concurrently.
  - TRUE: Compute processing and data transport MAY proceed concurrently (i.e., streaming or pipelined execution). Data MAY be transmitted incrementally during processing.
  - FALSE: Compute processing and data transport MUST be executed in separate phases. Data MUST be fully processed before transmission, or fully transmitted before processing begins.

- \* Compute Capacity Requirement (array). Contains at least one compute-capacity element. Each compute-capacity element is a structured requirement that specifies the minimum compute capacity needed to satisfy the service for a given type of computing resource. The compute-capacity is as follows.
  - Type (text). Indicates the type of compute resource. The values "cpu" and "gpu" are defined by this document.
  - Capacity (unsigned integer, GFLOPS). The minimum compute capacity required for the specified type. The value MUST be greater than or equal to 0; if it is specified as zero, no minimum compute capacity constraint is imposed for that type.

### 3.2.3. Storage Resource Requirement Intent

The Storage resource requirement intent specifies the required storage resource capabilities necessary to satisfy the service. It expresses quantitative and qualitative requirements over storage resources within the declared Intent Scope. These requirements define the desired capacity, performance, locality, and persistence characteristics of storage resources allocated for the service.

The Storage resource requirement intent includes the following fields:

```
storage-intent = {  
  ; Storage Capacity Requirement(GB)  
  "storage-capacity"      : uint,  
  
  ; Throughput numeric value  
  "storage-throughput"    : uint,  
  
  ? "storage-throughput-unit" : ("MB/s" / "GB/s"),  
}
```

- \* Storage Capacity Requirement (unsigned integer, GB). The required storage capacity representing the temporary or persistent space necessary to satisfy the service. The value MUST be greater than or equal to 0; if it is specified as zero, no minimum storage capacity constraint is imposed.
- \* Storage Throughput Requirement (unsigned integer, MB/s or GB/s). The required data access throughput representing the read and write rate necessary to satisfy the workload. The value MUST be greater than or equal to 0; if it is specified as zero, no minimum storage throughput constraint is imposed.



#### 4. IANA Considerations

There is currently no IANA consideration.

#### 5. Security Considerations

A strong security environment is required for the Service Intent, since malicious injection or tampering would pose a significant risk to automated resource selection and service deployment. If a Service Intent is disseminated and synchronized across an autonomic domain, the signaling channel must provide mutual authentication and integrity protection consistent with the Autonomic Domain security environment. In addition, an intent instance should employ an explicit signature to provide object-level authentication, integrity, and non-repudiation.

TODO more security considerations.

#### 6. References

##### 6.1. 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/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>>.
- [RFC8990] Bormann, C., Carpenter, B., Ed., and B. Liu, Ed., "GeneRic Autonomic Signaling Protocol (GRASP)", RFC 8990, DOI 10.17487/RFC8990, May 2021, <<https://www.rfc-editor.org/info/rfc8990>>.

##### 6.2. Informative References

- [I-D.ietf-anima-network-service-auto-deployment] Jiang, S. and Z. Du, "An Intent-based Framework of Network Services Autonomic Deployment and Management", Work in Progress, Internet-Draft, draft-ietf-anima-network-service-auto-deployment-07, 1 March 2026, <<https://datatracker.ietf.org/doc/html/draft-ietf-anima-network-service-auto-deployment-07>>.

- [RFC7575] Behringer, M., Pritikin, M., Bjarnason, S., Clemm, A., Carpenter, B., Jiang, S., and L. Ciavaglia, "Autonomic Networking: Definitions and Design Goals", RFC 7575, DOI 10.17487/RFC7575, June 2015, <<https://www.rfc-editor.org/info/rfc7575>>.
- [RFC8993] Behringer, M., Ed., Carpenter, B., Eckert, T., Ciavaglia, L., and J. Nobre, "A Reference Model for Autonomic Networking", RFC 8993, DOI 10.17487/RFC8993, May 2021, <<https://www.rfc-editor.org/info/rfc8993>>.

#### Authors' Addresses

Longwei Zhu  
Beijing University of Posts and Telecommunications  
No. 10 Xitucheng Road  
Haidian District, Beijing  
China  
Email: lwzhu@bupt.edu.cn

Bizhu Wang  
Beijing University of Posts and Telecommunications  
No. 10 Xitucheng Road  
Haidian District, Beijing  
China  
Email: wangbizhu\_7@bupt.edu.cn

Sheng Jiang  
Beijing University of Posts and Telecommunications  
No. 10 Xitucheng Road  
Haidian District, Beijing  
China  
Email: shengjiang@bupt.edu.cn