

Onions Working Group
Internet-Draft
Intended status: Informational
Expires: 3 July 2026

C. Xie
Q. Sun
China Telecom
L. Dunbar
Futurewei
30 December 2025

Onions Problem Statement
draft-xie-onions-problem-statement-00

Abstract

YANG-based service APIs are widely used to expose network and service abstractions to external systems such as controllers, orchestration platforms, and OSS/BSS applications. Despite the availability of numerous YANG data models and YANG-to-API tools, operators continue to face significant challenges in operationalizing these APIs in a consistent, scalable, and interoperable manner. APIs derived from similar YANG models often differ in semantics, lifecycle behavior, observability, and consumption patterns, complicating automation and cross-vendor integration. This document describes the problem space associated with operationalizing YANG-based service APIs, drawing on operator experience, IAB workshop findings, and IETF applicability studies to highlight gaps in current practices and motivate requirements for improving API predictability and operational effectiveness, without defining specific solutions, protocols, or tools.

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 July 2026.

Copyright Notice

Copyright (c) 2025 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
1.1. Requirements Language	4
2. Terminology	4
3. Use Cases Highlighting Challenges	4
3.1. Inter-Data-Center Connectivity	4
3.2. Data Transmission for Data-Intensive Workloads	5
3.3. On-Orbit Networking with Dynamic Interconnection	5
3.4. Summary	6
4. Operational Challenges with Network and Service Abstractions	6
4.1. Fragmented Operational Lifecycles	6
4.2. Misalignment Between Abstraction Layers	7
4.3. Inconsistent Semantics and Operational Assumptions	7
4.4. Limited Integration with Automation and External Systems	8
4.5. Gaps Revealed by Applicability and Deployment Experience	8
4.6. Impact on Operational Efficiency and Interoperability	8
4.7. Limited Feedback and Observability for Abstraction Enforcement	9
5. Limitations of Existing Approaches	9
5.1. Fragmentation Across Working Groups and Technologies	9
5.2. Insufficient Alignment Between Abstractions and Realization	9
5.3. Lack of Consistent Operational Semantics	10
5.4. Limited Guidance for API-Based Consumption	10
5.5. Incomplete Support for End-to-End Automation	10
5.6. Summary	10
6. Operational Evidence from the IAB NEMOPS Workshop	11
7. Requirements to Network Operation	12
8. Manageability Considerations	13
9. Security Considerations	13

10. IANA Considerations	13
11. References	13
11.1. Normative References	13
11.2. Informative References	13
Authors' Addresses	14

1. Introduction

Despite the availability of YANG data models and numerous YANG-to-API tools, operators continue to face significant challenges in operationalizing YANG-based service APIs in a consistent, scalable, and interoperable manner. As highlighted by the IAB Next Era of Network Management Operations (NEMOPS) workshop, operational workflows that rely on these APIs remain fragmented and difficult to automate end to end. In practice, APIs generated from similar YANG models often differ in structure, semantics, lifecycle behavior, and feedback mechanisms, complicating integration across systems, vendors, and deployment environments.

These challenges are further evidenced by recent IETF applicability studies that examine the use of existing YANG models for emerging operational scenarios. Applicability drafts focusing on unequal-cost multipath, traffic-engineered services, and network service models for telco cloud environments (e.g., [I-D.dunbar-neotec-ac-pe2pe-ucmp-applicability], [I-D.dunbar-onions-ac-te-applicability], and [neotec-ns-models-telcocloud]) identify recurring issues such as ambiguous semantics, weak alignment between service intent and realizable network behavior, insufficient lifecycle handling, and limited observability when services are exposed through APIs. While these studies address specific technical contexts, they collectively demonstrate a broader problem: current YANG-based service APIs lack consistent operational semantics and guidance required for reliable automation and interoperability.

The Operationalizing Network and service abstractIONS (ONIONS) Working Group is chartered to address this problem space by focusing on the operational aspects of YANG-based service APIs, rather than defining new protocols or API technologies. The goal of ONIONS is to improve automation, operational efficiency, and interoperability by identifying common problems, clarifying requirements, and providing guidance on how YANG-based service APIs should be structured, exposed, and consumed by external systems in a predictable and interoperable manner.

This document describes the problem space addressed by the ONIONS Working Group. It consolidates operator-observed challenges related to YANG-based service APIs, explains why existing approaches and

tools are insufficient when considered in isolation, and frames the requirements that ONIONS is chartered to examine to improve the operationalization and consumption of YANG-based service APIs. This document does not propose specific solutions, protocols, or data models.

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 document:

- * Abstraction: refers to the definition of simplified, high-level constructs that represent network and service capabilities, while hiding the details of their underlying realization. Such abstractions enable interaction between management and automation systems without requiring direct exposure of device-specific configurations or protocol behaviors.
- * Cloud DC: Third party data centers that host applications and workloads owned by different organizations or tenants.

3. Use Cases Highlighting Challenges

The following use cases illustrate operational scenarios in which YANG-based service APIs are increasingly used to request, operate, and evolve network services. They highlight common challenges encountered by operators and automation systems when attempting to consume these APIs in a predictable, interoperable, and scalable manner.

3.1. Inter-Data-Center Connectivity

Enterprises and service providers frequently operate across multiple data centers and hybrid cloud environments. In such deployments, applications and services rely on reliable, high-speed, and secure connectivity between geographically distributed data centers or cloud resource pools. Typical scenarios include service data synchronization, cross-cloud disaster recovery, and dynamic scaling of application services across locations. In operational practice, orchestration and management systems expect to request inter-data-center connectivity as a service, specifying endpoints and desired

characteristics while relying on YANG-based service APIs to instantiate, modify, and monitor the service. However, existing APIs often expose network configuration constructs rather than a coherent service abstraction, making it difficult to express service intent, track service lifecycle, or confirm that connectivity objectives are being met. This use case highlights challenges related to lifecycle handling, semantic consistency, and observability when YANG-based service APIs are used to support inter-data-center services across vendors or administrative domains.

3.2. Data Transmission for Data-Intensive Workloads

As the industry enters the AI era, data has become a primary driver of productivity growth, leading to a sharp increase in data processing and transfer demands. To improve efficiency and control costs, computing resources are increasingly centralized in cloud or large-scale data center environments, while data sources may remain distributed.

In this context, massive datasets are routinely transferred to centralized computing and storage infrastructure for analysis and processing. These transfers may involve large volumes of data, require predictable completion times, and demand bandwidth that varies significantly over time. From an operational perspective, orchestration systems rely on YANG-based service APIs to request connectivity that can support these data transfers.

Operational experience shows that existing YANG-based service APIs often lack the semantics needed to express such data-driven connectivity needs in a service-oriented manner. As a result, automation systems must compensate with additional logic to manage bandwidth changes, track progress, or determine when a transfer-oriented service has completed, complicating integration and reducing interoperability.

3.3. On-Orbit Networking with Dynamic Interconnection

Emerging on-orbit networking environments introduce highly dynamic operational conditions. Networks with steerable beams may depend on infrastructure provided by other networks, such as third-party ground stations or additional on-orbit assets. Connectivity requirements can change rapidly based on orbital movement, coverage availability, and mission needs.

In these environments, network services must be instantiated, modified, and torn down on short timescales. Constructs such as bearers or attachment circuits may need to be dynamically established to interconnect infrastructure at orbital speeds. YANG-based service APIs are a natural interface for exposing such services to planning and control systems.

However, most existing YANG-based service APIs are designed for relatively static terrestrial networks and lack clear semantics for short-lived services, rapid reconfiguration, or coordination across heterogeneous infrastructure. This use case highlights the difficulty of operationalizing YANG-based service APIs in highly dynamic, multi-operator environments.

3.4. Summary

Across these use cases, YANG-based service APIs are expected to serve as the primary interface between network infrastructure and external automation systems. The scenarios demonstrate that while APIs and models exist, operators continue to face challenges related to lifecycle management, semantic clarity, observability, and interoperability. These challenges motivate the problem statements discussed in subsequent sections and inform the requirements outlined later in this document.

4. Operational Challenges with Network and Service Abstractions

To support the use cases described in previous section, operators increasingly rely on service and network abstractions exposed through YANG-based service APIs. While these abstractions are widely deployed, operators report persistent challenges in operationalizing them in a consistent, scalable, and automatable manner. As highlighted by the IAB Next Era of Network Management Operations (NEMOPS) workshop [NEMOPS], these challenges are systemic and operational in nature, arising from fragmented tooling, inconsistent abstraction semantics, and limited end-to-end coordination. They are not confined to a specific technology or service type, but recur across abstraction domains and deployment environments.

4.1. Fragmented Operational Lifecycles

Operational workflows associated with service abstractions, such as service instantiation, monitoring, troubleshooting, modification, and decommissioning, are often fragmented and inconsistently handled. Even when abstractions coexist within the same network or service offering, they frequently rely on different tools, data models, and interfaces. NEMOPS discussions highlighted that operators commonly depend on a heterogeneous mix of management protocols, vendor-

specific APIs, and legacy mechanisms to complete these workflows, significantly increasing operational complexity and cost.

In practice, lifecycle actions initiated through YANG-based service APIs often require coordination across orchestration systems, controllers, and device configurations. However, these components are rarely aligned in terms of lifecycle semantics, data models, or operational assumptions. This fragmentation limits end-to-end automation, complicates validation and rollback, and increases the likelihood of configuration drift and operational errors. Existing service and network abstractions generally lack native constructs to express lifecycle attributes such as activation time, duration, expiration, or rollback behavior. As a result, transient service intents must be tracked and enforced outside the abstraction framework, increasing operational complexity and the risk of inconsistency.

4.2. Misalignment Between Abstraction Layers

Service abstractions are typically realized through a combination of service-level models, network-level models, control-plane behavior, and management interfaces. These layers are often developed independently, with limited coordination across working groups or operational domains.

This misalignment can manifest as:

- Service abstractions that do not cleanly map to underlying network capabilities.
- Network models that expose parameters without clear service-level semantics.
- Control-plane behaviors that are difficult to correlate with service-level intent.

As a result, operators face challenges ensuring that a service behaves as intended throughout its lifecycle, particularly when changes occur at one layer without corresponding visibility or coordination at others.

4.3. Inconsistent Semantics and Operational Assumptions

Service and network abstractions frequently rely on metrics, attributes, or parameters whose semantics vary across models, implementations, or consumption contexts. Concepts such as cost, availability, or performance may be represented using different definitions, units, scopes, or update models.

These inconsistencies complicate integration between systems and undermine the reliability of automation. Consumers of YANG-based service APIs cannot assume uniform behavior or interpretation, forcing operators to introduce custom logic, static assumptions, or manual intervention. Over time, this erodes interoperability and limits scalability.

4.4. Limited Integration with Automation and External Systems

Service abstractions are commonly exposed to external systems, such as orchestration platforms and OSS/BSS applications, through APIs derived from YANG data models. However, the lack of consistent guidance on how abstractions should be modeled, exposed, and consumed results in APIs that vary significantly across vendors and deployments. This variability makes it difficult for external systems to consume YANG-based service APIs in a predictable and interoperable manner. Integration often requires bespoke adaptations and vendor-specific knowledge, limiting reuse and slowing the adoption of automation across domains and administrative boundaries.

4.5. Gaps Revealed by Applicability and Deployment Experience

Recent IETF applicability studies and deployment experience further highlight these operational challenges across different abstraction contexts. Recurring issues include unclear semantics, difficulty aligning service intent with realizable network behavior, limited lifecycle handling, and challenges integrating abstractions across technologies and domains.

Although these efforts focus on specific use cases or technical areas, they expose common operational gaps that extend beyond any single abstraction or working group. Collectively, they reinforce the need for improved consistency, coordination, and operational guidance when service abstractions are exposed and consumed through YANG-based service APIs.

4.6. Impact on Operational Efficiency and Interoperability

The challenges described above directly impact operational efficiency, automation, and interoperability. Operators are required to invest significant effort in integration, validation, and troubleshooting, reducing the benefits that abstractions are intended to provide. Without a more coordinated approach to abstraction modeling and operational usage, these issues are likely to persist as networks continue to evolve.

4.7. Limited Feedback and Observability for Abstraction Enforcement

Existing abstractions primarily focus on configuration and offer limited standardized mechanisms for reporting whether requested behaviors have been successfully applied or remain valid over time. This lack of feedback impedes closed-loop automation and increases reliance on manual monitoring and intervention.

5. Limitations of Existing Approaches

A wide range of mechanisms, models, and frameworks already exist within the IETF to support service and network abstractions. These include protocol-specific control-plane mechanisms, device- and network-level YANG data models, and management frameworks for service configuration and monitoring. While these efforts address important aspects of abstraction definition and realization, operators report that they are insufficient when applied independently to support consistent, end-to-end operational workflows.

5.1. Fragmentation Across Working Groups and Technologies

Service and network abstractions are defined and evolved across multiple IETF working groups, each focusing on a specific technology, protocol, or layer. Although this separation is appropriate for protocol development, it has resulted in abstraction models and operational assumptions that are not well coordinated across domains. As a result, operators must integrate abstractions that were designed with different scopes, semantics, and lifecycle assumptions. This fragmentation increases integration effort and complicates automation, particularly when a service abstraction spans multiple technologies or administrative domains.

5.2. Insufficient Alignment Between Abstractions and Realization

Existing abstraction models often focus on configuration or control-plane aspects without fully considering how abstractions are realized operationally across networks. In practice, operators must reconcile service-level intent with network-level capabilities, control-plane behaviors, and device-specific constraints. When abstraction definitions do not align cleanly with realizable behaviors, operators encounter difficulties validating service behavior, correlating faults, or evolving services over time. These gaps are typically addressed through custom logic or manual processes, reducing portability and interoperability.

5.3. Lack of Consistent Operational Semantics

Many abstraction models expose parameters or metrics that are syntactically similar but semantically inconsistent across technologies or implementations. Differences in interpretation, update frequency, or scope can lead to unpredictable behavior when abstractions are consumed by automation systems.

Without consistent operational semantics, it is difficult for management and orchestration systems to reliably interpret abstraction state, compare information across domains, or make automated decisions based on abstraction models alone.

5.4. Limited Guidance for API-Based Consumption

YANG data models are commonly used as the basis for APIs that expose service abstractions to external systems. However, existing work provides limited guidance on how these abstractions should be exposed, versioned, or consumed in a predictable and interoperable manner. As a result, APIs derived from similar abstraction models may differ significantly across vendors or deployments, requiring bespoke integration by operators and OSS/BSS systems. This variability undermines the portability and reuse that abstractions are intended to provide.

5.5. Incomplete Support for End-to-End Automation

While individual mechanisms support automation at specific layers or points in the service lifecycle, they do not collectively provide a coherent framework for abstraction-driven automation across systems. Automation workflows frequently span service modeling, network configuration, monitoring, and fault management, yet existing approaches treat these aspects in isolation.

This lack of coordination limits the effectiveness of automation and makes it difficult to implement closed-loop operational workflows that adapt to changes in service requirements or network conditions.

5.6. Summary

Taken together, these limitations demonstrate that existing mechanisms, while necessary, are not sufficient to address the operational challenges associated with service and network abstractions. The gaps are not primarily due to missing protocols or data models, but to the lack of coordination, consistency, and operational guidance across abstraction efforts. Addressing these issues requires a holistic examination of abstraction modeling and operational usage, which is the focus of the ONIONS WG.

6. Operational Evidence from the IAB NEMOPS Workshop

The operational challenges described in this document are consistent with, and reinforced by, the findings of the IAB Next Era of Network Management Operations (NEMOPS)[NEMOPS] workshop, which examined the state of network management and automation based on operator experience across diverse deployment environments.

The NEMOPS workshop identified that, despite significant progress in protocol development and data modeling, operational workflows remain fragmented and difficult to automate end-to-end. Operators reported continued reliance on a heterogeneous mix of tools, protocols, and interfaces, including YANG-based management protocols, vendor-specific APIs, legacy mechanisms such as CLI and SNMP, and bespoke orchestration logic to deploy and operate services. This fragmentation increases operational complexity and limits the effectiveness of abstraction-driven automation.

A key observation from the workshop is that model-driven network management is generally successful, yet insufficient on its own to address higher-level operational needs. In particular, the workshop highlighted gaps between device-level and service-level abstractions, noting that existing models often lack the semantic alignment and contextual information required by orchestration and OSS/BSS systems. As a result, operators must perform extensive model mapping, data transformation, and system-specific integration outside the scope of standardized abstractions.

The workshop further emphasized challenges related to observability, verification, and feedback. While configuration mechanisms are relatively mature, operators reported limited ability to validate whether service intent is being met over time or to correlate operational state across abstraction layers. This lack of consistent feedback undermines closed-loop automation and complicates troubleshooting, particularly in multi-vendor and multi-domain environments.

Another recurring theme from the NEMOPS discussions is the lack of architectural documentation and operational guidance explaining how existing abstractions, models, protocols, and tools are intended to work together as a system. Operators expressed difficulty understanding which abstractions to use, how they should be combined, and how responsibilities are divided across layers and working groups. This absence of cohesive guidance leads to divergent interpretations and inconsistent deployments.

These findings closely align with the limitations identified in the applicability studies discussed earlier and reinforce a broader operational problem: while many of the necessary technical components for service and network abstractions exist, they are not sufficiently coordinated, aligned, or documented to support consistent, interoperable, and automatable operations. Addressing these systemic issues requires a focus on abstraction coherence, lifecycle semantics, and operational consumption concerns that fall squarely within the scope of the ONIONS Working Group.

7. Requirements to Network Operation

Based on the illustrations above, the following requirements have been identified.

- Application and Network Coordination

The network operation must be aware of the requirements for supporting services, including: establishing Overlay connections for specific service needs, providing on-demand underlay physical resource guarantees, and achieving coordination between the network and services.

- On-Demand Setup, Teardown, and Flexible Networking

The network operation and maintenance system must be capable of rapidly establishing real-time service connections between the user-specified start and end points based on user requirements. Upon service completion, connections should be dismantled and resources released, providing users with on-demand connectivity and billing. This meets users' sudden service demands and reduces their expenses.

- Elastic Provision of Super high-Bandwidth Services

The network must possess the capability to elastically provide super high bandwidth, meeting users' elastic resource requirements and enabling full utilization of network resources.

- Ubiquitous Access Provisioning

To facilitate user convenience in accessing computational resources, the network must support ubiquitous access and wide coverage, allowing users to flexibly connect through various methods and ensuring computational resources are readily available on demand.

- Trustworthiness and Reliability

The network must be trustworthy and reliable, strictly ensuring the security and dependability of user data transmission.

- Cross-Domain Coordination

For user data transmission needs across domains or even across different operators, the network must possess cross-domain coordination capabilities, enabling flexible end-to-end scheduling of network resources and services.

The above requirements need the network operation system needs to dynamically coordinate behavior across multiple network segments, expose the YANG-based network and service capabilities through open APIs, driven by service-level events, workload changes, or short-lived operational needs.

8. Manageability Considerations

TBD.

9. Security Considerations

TBD.

10. IANA Considerations

No Action is needed.

11. References

11.1. 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>>.

11.2. Informative References

[I-D.dunbar-neotec-ac-pe2pe-ucmp-applicability]

Dunbar, L., Qiong, S., Wu, B., Contreras, L. M., and C. Xie, "Applying Attachmet Circuit and PE2PE YANG Data Model to dynamic policies Use Case", Work in Progress, Internet-Draft, draft-dunbar-neotec-ac-pe2pe-ucmp-applicability-01, 22 June 2025, <<https://datatracker.ietf.org/doc/html/draft-dunbar-neotec-ac-pe2pe-ucmp-applicability-01>>.

[I-D.dunbar-onions-ac-te-applicability]

Dunbar, L., Qiong, S., Wu, B., Contreras, L. M., and C. Xie, "Applying Attachmet Circuit and Traffic Engineering YANG Data Model to Edge AI Use Case", Work in Progress, Internet-Draft, draft-dunbar-onions-ac-te-applicability-00, 3 October 2025, <<https://datatracker.ietf.org/doc/html/draft-dunbar-onions-ac-te-applicability-00>>.

[NEMOPS] "NEMOPS",
<<https://datatracker.ietf.org/group/nemopsws/about/>>.

Authors' Addresses

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

Qiong Sun
China Telecom
Beiqijia Town, Changping District
Beijing
102209
China
Email: sunqiong@chinatelecom.cn

Linda Dunbar
Futurewei
Email: ldunbar@futurewei.com