

Operations and Management Area Working Group
Internet-Draft
Intended status: Informational
Expires: 4 September 2025

J. Jeong, Ed.
Sungkyunkwan University
Y. Shen
Ajou University
Y. Ahn
Sungkyunkwan University
Y. Kim
Soongsil University
E. Duarte Jr.
Federal University of Parana
K. Yao
China Mobile
3 March 2025

Interface to In-Network Computing Functions (I2ICF): Problem Statement
draft-jeong-opsawg-i2icf-problem-statement-00

Abstract

This document specifies the problem statement for the Interface to In-Network Computing Functions (I2ICF) for user services both on the network-level and application-level. In-Network Computing Functions (ICF) include In-Network Network Functions (INF) which are defined in the context of Network Functions Virtualization (NFV) and Software-Defined Networking (SDN). ICFs also include In-Network Application Functions (IAF) which appear in the context of Internet-of-Things (IoT) Devices, Software-Defined Vehicles (SDV), and Unmanned Aerial Vehicles (UAV). Intent-Based Networking (IBN) can be used to compose user services and consist of a combination of ICFs in a target network. This document investigates the need for a standard framework with the interfaces for ICFs, in terms of applications with the need to run Artificial Intelligence (AI) in the network and interoperability among multi-vendor ICFs.

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 4 September 2025.

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	2
2. Terminology	5
3. Problem Statement for Interface to In-Network Computing Functions	6
3.1. In-Network Computing Functions	8
3.2. Intent-Based Networking	10
3.3. Problem Statement	11
4. IANA Considerations	12
5. Security Considerations	12
6. References	12
6.1. Normative References	12
6.2. Informative References	13
Acknowledgments	18
Contributors	18
Authors' Addresses	19

1. Introduction

Network softwarization has been widely adopted in multiple environments, such as in cloud and edge computing, as well as in the network infrastructure itself, facilitating the deployment of network services (e.g., 5G mobile networks [TS-23.501]). The multiple technologies behind network softwarization include Network Functions Virtualization (NFV) [ETSI-NFV][ETSI-NFV-Release-2] and Software-Defined Networking (SDN) [RFC7149]. Furthermore, there is also an

integration with Intent-Based Networking (IBN) [RFC9315][Survey-IBN-CST-2023], which can be used to define and deploy intelligent network services as well as intelligent application services. End-user devices such as smartphones and smartwatches are connected to various Internet-of-Things (IoT) devices for customer-tailored services. Recently, Software-Defined Vehicles (SDVs) [AUTOSAR-SDV][Eclipse-SDV][COVESA] have the potential to become as popular as smartphones. SDVs are intended to use network softwarization technologies such as NFV and SDN. System components and applications in the context of SDVs are usually executed on containers in a cloud native environment and can be orchestrated for instance with Kubernetes [Kubernetes].

There is an undeniable trend towards using Artificial Intelligence (AI) to improve multiple complex network operations. AI can, for instance, enable the creation of dynamic, adaptable security policies, which are particularly important in the cloud-edge-core-continuum, which is by definition a heterogeneous environment to which different parts can bring advantages, challenges and threats. AI has been successfully used in the context of intrusion detection, but it also improves troubleshooting, being able to predict problems before they occur. AI can learn from telemetry data collected from multiple networks and reach conclusions that can be applied globally or to individual networks. In all these cases, there are tremendous benefits to running AI processes within the network itself.

In this context, network automation and management have become critical. It is important to facilitate the construction of intelligent services and applications for both network operators and end users [I-D.jeong-nmrg-ibn-network-management-automation]. A user intent (who can be an end user or network operator) in the form of either text or voice needs to be understood and processed by the system. An intent is a declarative request for a specific goal rather than an imperative request having a series of configuration or commands for specific operations. Thus, an intent needs to be translated into a network policy or an application policy that satisfies the user request. A network policy consists of rules to execute a user intent, which can be for instance defined in terms of Quality of Service (QoS) requirements, defining targets for metrics such as throughput and delay. An application policy consists of rules to execute the service's application demands, for example in terms of functionality and timing. After network and application policies are translated, there is a need to invoke the appropriate Network Functions (NF) in the network infrastructure, edge, or cloud.

Thus, a user intent has to be translated either into a network policy executed as a network service on the network infrastructure or an application policy for an application service. For example, services

for user applications (e.g., video conference) need to be accurately configured and efficiently processed by not only Application Functions (AF) such as a client (e.g., a video conference client) and a server (e.g., a video conference server), but also Network Functions (NF) (e.g., video broadcast coordinator) defined in the context of Computing in the Network (COIN) [I-D.irtf-coinrg-use-cases][NFV-COIN].

In the context of Computing in the Network (COIN) terminology [I-D.irtf-coinrg-coin-terminology], a Programmable Network Device (PND) in an In-Network Computing (INC) environment can have multiple kinds of features and capabilities. A PND can also interact with other PNDs. PNDs from different product lines or vendors can provide different functionalities for INC functions. In order to compose a COIN system consisting of multiple PDNs that interact among themselves, it is necessary to define a standard interface for PNDs to expose so that they can learn about each other's capabilities and properly interact.

A standard framework to define the interfaces of Application Functions (AFs) and Network Functions (NFs) is required to allow the configuration and monitoring of applications and network services consisting of those functions. There is currently no standard data model to describe the capabilities of AFs and NFs. Furthermore, there is no standard data model defining an interface to register the capabilities of AFs and NFs on a controller-like device that would process service requests for those functions. In addition, there are no standard interfaces to configure and monitor those AFs and NFs according to user's intent. The Interface to Network Security Functions (I2NSF) was standardized for the control and management of Network Security Services with Network Security Functions (NSFs) [RFC8329][I-D.ietf-i2nsf-applicability]. The present document is defined taking into account the I2NSF document, but the purpose is beyond the scope of Security Functions, defining a more general control and management framework for intelligent services consisting of AFs and NFs.

This document specifies the problem statement and use cases for the Interface to In-Network Computing Functions (I2ICF) considering arbitrary In-Network Computing Functions (ICFs) presenting arbitrary features and capabilities. The ICFs consist of Network Functions (NFs) including PNDs and Application Functions (AFs) in order to compose a user's services. First of all, ICFs include In-Network Network Functions (INF) which are NFs defined within the context of NFV and SDN [I-D.irtf-coinrg-use-cases]. Secondly, they also include In-Network Application Functions (IAF) which are AFs employed by Internet-of-Things (IoT) Devices, Software-Defined Vehicles (SDV) [AUTOSAR-SDV][Eclipse-SDV][COVESA], and Unmanned Aerial Vehicles

(UAV). Finally, this document shows how Intent-Based Networking (IBN) can be realized with the proposed I2ICF framework and its interfaces for user services that consist of a combination of ICFs in a target network.

Note that a standard framework for In-Network Functions is the only way to provide the interoperability of diverse autonomous systems and networks on the service-level. The proposed I2ICF allows the integration of existing and new services implemented with different technologies, at different locations (e.g., cloud or edge) each with different requirements and functionalities. It can also have an impact on innovation, as it provides a new level of integration for various ICFs. The standard interfaces for the ICFs can also allow global-level decisions to be enforced, such as those related to elasticity and scalability.

2. Terminology

This document uses the terminology described in [RFC9315], [RFC8329], [I-D.irtf-coinrg-coin-terminology], [I-D.irtf-coinrg-use-cases], [I-D.jeong-opsawg-security-management-automation], [I-D.jeong-nmrg-ibn-network-management-automation], and [SPT]. In addition, the following terms are defined below:

- * Intent: the set of operational goals (that a network should meet) and outcomes (that a network is supposed to deliver) defined in a declarative manner without specifying how they are achieved or should be implemented [RFC9315].
- * Intent-Based System (IBS): the system that enforces an intent from a user (or administrator) into a target system (e.g., SDV). An intent can be expressed in Natural Language (e.g., English) and can be translated into a policy (i.e., network policy and application policy) using Natural Language Processing (NLP) [USENIX-ATC-Lumi] [BERT] [Deep-Learning]. In this document, the intent can be translated into a corresponding high-level policy by an intent translator [I-D.jeong-opsawg-security-management-automation]. The high-level policy can also be translated into the corresponding low-level policy by a Policy Translator (PT) [SPT]. The low-level policy is dispatched to appropriate Service Functions (SFs). Through the monitoring of the SFs, the activity and performance of the SFs is monitored and analyzed. If needed, the rules of the high-level or low-level network policy are augmented or new rules are generated and configured to appropriate SFs.

- * Mobile Object (MO): the object that is capable of moving with its own power source and wireless communication capability, e.g., in the context of 5G Vehicle-to-Everything (e.g., 5G V2X). An MO can be an Internet-of-Things (IoT) device, Software-Defined Vehicle (SDV) [AUTOSAR-SDV][Eclipse-SDV][COVESA], and Unmanned Aerial Vehicle (UAV). An MO is a Programmable Network Device (PND) [I-D.irtf-coinrg-coin-terminology] that can be reconfigured for different network requirements inside the MO.
- * In-Network Network Functions (INF): the service functions that do the computing in the network infrastructure. They are a group of COIN programs [I-D.irtf-coinrg-coin-terminology] to provide required computing tasks and functions.
- * In-Network Application Functions (IAF): the service functions that work for applications in Mobile Objects. They are a group of COIN programs [I-D.irtf-coinrg-coin-terminology] to provide the required application tasks and functions.
- * Interface to In-Network Computing Functions (I2ICF): the interfaces that are used between a pair of ICFs for the interaction, configuration and monitoring.
- * A Framework for the Interface to In-Network Computing Functions (I2ICF): the framework that consists of components and interfaces to configure and monitor ICFs that can be employed by applications and services in the network infrastructure and MOs.

3. Problem Statement for Interface to In-Network Computing Functions

This section starts with a description and examples of In-Network Computing Functions. Next, an overview of Intent-Based Networking (IBN) is presented, and finally the Problem Statement for the Interface to In-Network Computing Functions (I2ICF). Figure 1 shows Wireless and Wired Networks of a Central Cloud. The I2ICF framework includes network entities and Mobile Objects (MO). Figure 2 shows a VNF-Consensus Architecture that allows the I2ICF framework to synchronize flow table information of replicated SDN Controllers all in the same Edge Cloud [NFV-COIN]. These are example networks within the I2ICF problem space.

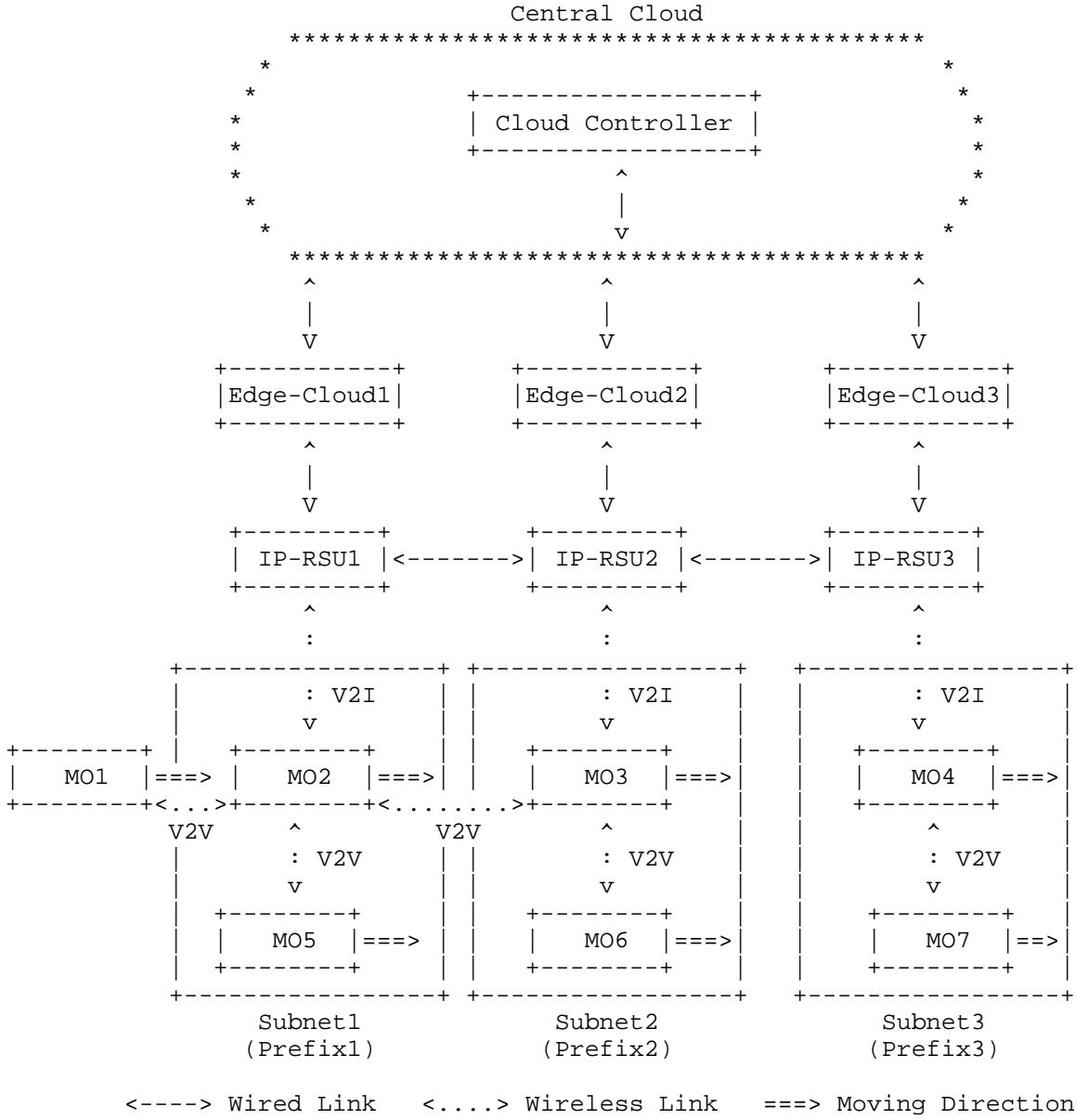


Figure 1: I2ICF Framework: Wireless and Wired Networks in a Central Cloud

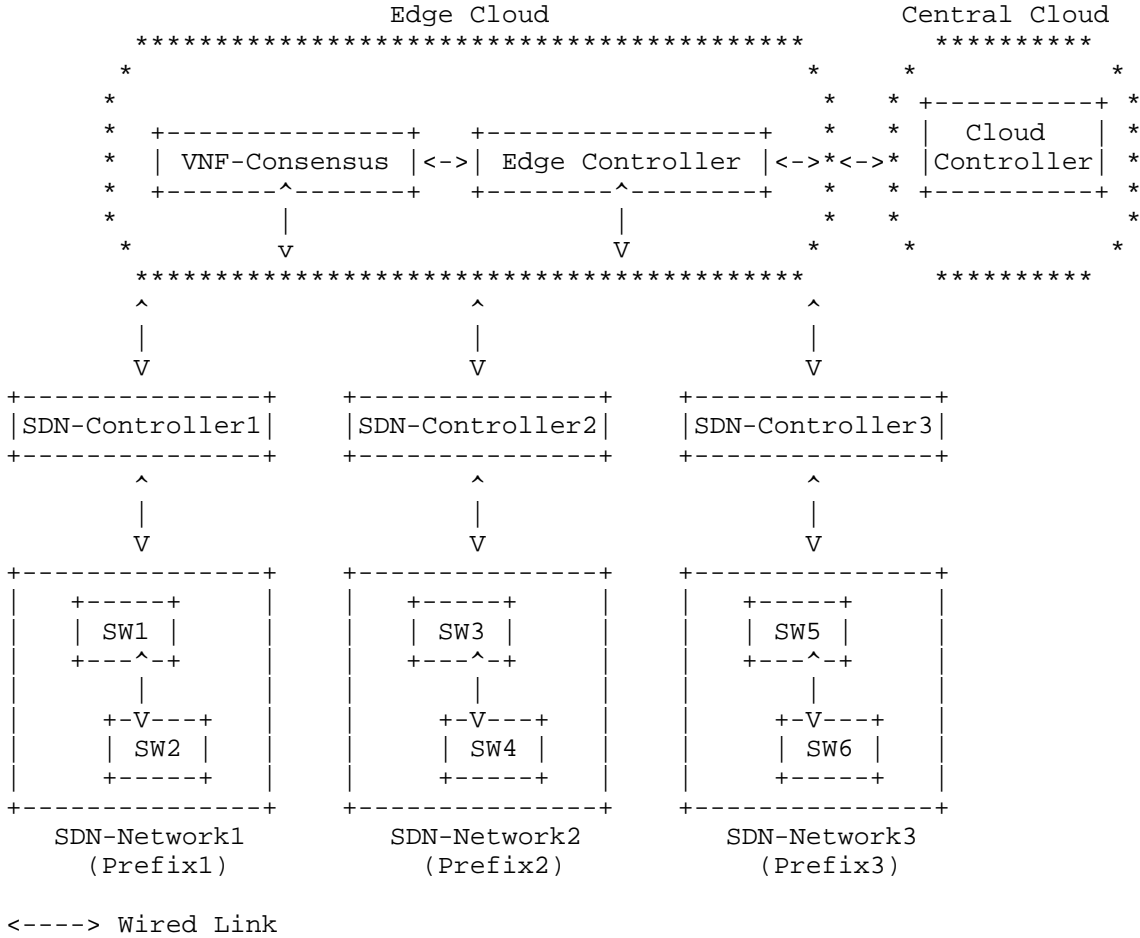


Figure 2: I2ICF Framework: VNF-Consensus in an Edge Cloud

3.1. In-Network Computing Functions

A large variety of In-Network Computing Functions (ICF) have been proposed for the implementation of various services implemented with COIN (COmputing In-the Network) which is based on network softwarization technologies, mainly NFV and SDN [I-D.irtf-coinrg-use-cases][NFV-COIN].

The COIN Use Cases Document [I-D.irtf-coinrg-use-cases] proposes four kinds of use cases for In-Network Computing. Those use cases are (i) Providing New COIN Experiences, (ii) Supporting New COIN Systems, (iii) Improving Existing COIN Capabilities, and (iv) Enabling New COIN Capabilities.

1. For Providing New COIN Experiences, the document describes mobile application offloading and Extended Reality (XR) and immersive media.
2. For Supporting New COIN Systems, the document describes In-Network Control, Time-Sensitive Applications, Large Volume Applications, and Industrial Safety.
3. For Improving Existing COIN Capabilities, the document describes Content Delivery Networks (CDN), Compute-Fabric-as-a-Service (CFaaS), and Virtual Network Programming (e.g., P4 programs and OpenFlow rules).
4. For Enabling New COIN Capabilities, the document describes Distributed AI Training among geographically dispersed endpoints for solving large-scale problems.

NFV-COIN [NFV-COIN] describes three use cases for In-Network Computing. Its use cases are (i) NFV Failure Detection, (ii) Virtual Network Function (VNF) Consensus, and (iii) NFV Reliable Broadcast.

1. NFV Failure Detection is an NFV-based failure detector that obtains monitoring data from SDN Switches via an SDN Controller and also detects the failure of communication links. This failure detector is a standalone NF and is thus separated from the SDN Controller and thus it does not sacrifice SDN Controller performance (e.g., CPU usage).
2. VNF Consensus is a consensus service that performs the synchronization of the control planes of replicated SDN Controllers. This consensus service does not require any modification of both the data plane and control plane of SDN switches and controllers. Through the consensus service, if a new rule is configured by an SDN Controller, this rule is reliably distributed to all the other SDN Controllers through the VNF-Consensus service.
3. NFV Reliable Broadcast is an NFV-based broadcast service (NFV-RBCast) that provides both reliable and ordered delivery of messages. This ordered broadcast is implemented by NFV-RBCast using a VNF-Sequencer. A flow to be broadcast the NFV-RBCast service causes an SDN Controller to install a forwarding rule on the necessary SDN Switches. All the packets of the flow are forwarded to the VNF-Sequencer. The VNF-Sequencer inserts a sequence number into each of those forwarded packets, and sends them to the destination.

Functionalities of each service need to be decomposed into AFs and NFs in edge computing. The management and configuration of those AFs and NFs is a functionality that must be provided by a service coordinator in the context of COIN-based network services. There is currently no framework or interfaces defined as standards specifying the life cycle of COIN-based services.

3.2. Intent-Based Networking

According to [RFC9315] the intent life cycle of an Intent-Based System (IBS) is shown in Figure 3. The life cycle involves intent management for network entities and MOs. [RFC9315] divides the IBS life cycle into three spaces, namely MO User Space, Translation & IBS Space, and Network Operations (Ops) & Application (App) Space. Each space is further subdivided into two sections, fulfillment and assurance. The fulfillment section pipelines the steps (i.e., intent input, translation/refinement, learning/planning/rendering, and configuration/provisioning) toward the final SFs such as Network Functions (NFs) and Application Functions (AFs) in MOs. The assurance section monitors final results of the intent fulfillment to validate and analyze the resulted NFs and applications for MOs.

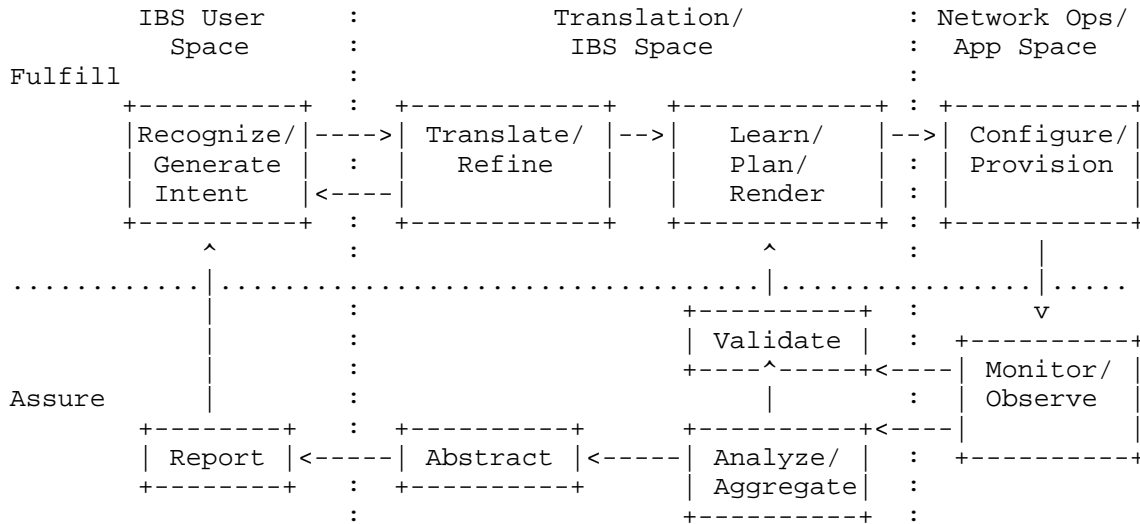


Figure 3: Intent Management: IBS Intent Life Cycle

The life cycle in Figure 3 is presented as a conceptual view and needs to be made concrete in the form of a framework with interfaces among components in the framework. The data models of an intent, a network policy, and an application policy should be specified using

either YANG [RFC6020][RFC7950] or YAML [YAML]. Messages are to be delivered to target components via some message delivery protocol, such as NETCONF [RFC6241], RESTCONF [RFC8040], or REST API [REST].

3.3. Problem Statement

The goal of an Intent-Based System (IBS) is to enforce the service corresponding to a user's intent with an appropriate application in a target network in terms of functionality and quality [RFC9315][RFC8329] [I-D.jeong-opsawg-security-management-automation] [I-D.jeong-nmrg-ibn-network-management-automation]. To achieve this goal, first of all, an intent needs to be translated into either a network policy or an application policy by an intent translator [I-D.jeong-nmrg-ibn-network-management-automation] [SPT]. Then those network policies and application policies need to be delivered to a network controller and an application controller, respectively. The network controller further translates the network policy into the network rules to be sent to the network entities (i.e., NFs). In the same way, the application controller further translates the application policy into the application rules to be sent to the application entities (i.e., AFs).

For the translation of either an intent or a policy, the capabilities of NFs and AFs should be registered with databases (e.g., NF database and AF database). Thus, a capability data model for those NFs and AFs should be specified [I-D.ietf-i2nsf-capability-data-model]. Also, a registration interface is required for an NF or AF vendor to register its NF or AF with the corresponding database such as the NF database and the AF database, respectively [I-D.ietf-i2nsf-registration-interface-dm]. Therefore, a data model for this registration interface should be specified to make a registration message for the Vendor's Management System (VMS) [RFC8329].

An IBS user needs an interface to send an intent to an IBS controller (e.g., Cloud Controller in Figure 1), it must have an intent translator, which translates the intent into a network policy or an application policy, and a dispatcher, which dispatches the policies to appropriate destinations (e.g., NF controller and AF controller). This interface is called a Customer-Facing Interface (CFI) for the IBS user [I-D.ietf-i2nsf-consumer-facing-interface-dm]. A data model for the Customer-Facing Interface should also be specified.

Both an NF controller and an AF controller need an interface to deliver the network rules and the application rules to the appropriate NFs and the appropriate AFs, respectively. This interface is called a Service Function-Facing Interface (SFI) for both the NF controller and the AF controller [I-D.ietf-i2nsf-nsf-facing-interface-dm].

For the assurance of the intent in the target network and application, the collection and analysis of monitoring data from the NFs and AFs is required. A Monitoring Interface [I-D.ietf-i2nsf-nsf-monitoring-data-model] is an interface to collect monitoring data from either an NF or an AF to a data collector (e.g., IBS analyzer [I-D.lingga-opsawg-analytics-interface-dm] [TS-23.288][TS-29.520]). For further actions, the analysis results of the NF and the AF should be reported to the NF controller and the AF controller, respectively. An Analytics Interface is an interface to deliver analysis results to either an NF controller or an AF controller [I-D.lingga-opsawg-analytics-interface-dm].

The required data models can be constructed by either YANG [RFC6020][RFC7950] or YAML [YAML]. The message delivery protocol for the interfaces can be one among NETCONF [RFC6241], RESTCONF [RFC8040], or REST API [REST].

4. IANA Considerations

This document does not require any IANA actions.

5. Security Considerations

The same security considerations for the Interface to Network Security Functions (I2NSF) Framework [RFC8329] are applicable to the Intent-Based System this document.

6. References

6.1. Normative References

- [RFC6020] Bjorklund, M., Ed., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", RFC 6020, DOI 10.17487/RFC6020, October 2010, <<https://www.rfc-editor.org/info/rfc6020>>.
- [RFC6241] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed., and A. Bierman, Ed., "Network Configuration Protocol (NETCONF)", RFC 6241, DOI 10.17487/RFC6241, June 2011, <<https://www.rfc-editor.org/info/rfc6241>>.

- [RFC7149] Boucadair, M. and C. Jacquenet, "Software-Defined Networking: A Perspective from within a Service Provider Environment", RFC 7149, DOI 10.17487/RFC7149, March 2014, <<https://www.rfc-editor.org/info/rfc7149>>.
- [RFC7950] Bjorklund, M., Ed., "The YANG 1.1 Data Modeling Language", RFC 7950, DOI 10.17487/RFC7950, August 2016, <<https://www.rfc-editor.org/info/rfc7950>>.
- [RFC8040] Bierman, A., Bjorklund, M., and K. Watsen, "RESTCONF Protocol", RFC 8040, DOI 10.17487/RFC8040, January 2017, <<https://www.rfc-editor.org/info/rfc8040>>.
- [RFC8329] Lopez, D., Lopez, E., Dunbar, L., Strassner, J., and R. Kumar, "Framework for Interface to Network Security Functions", RFC 8329, DOI 10.17487/RFC8329, February 2018, <<https://www.rfc-editor.org/info/rfc8329>>.
- [RFC9315] Clemm, A., Ciavaglia, L., Granville, L. Z., and J. Tantsura, "Intent-Based Networking - Concepts and Definitions", RFC 9315, DOI 10.17487/RFC9315, October 2022, <<https://www.rfc-editor.org/info/rfc9315>>.
- [RFC9365] Jeong, J., Ed., "IPv6 Wireless Access in Vehicular Environments (IPWAVE): Problem Statement and Use Cases", RFC 9365, DOI 10.17487/RFC9365, March 2023, <<https://www.rfc-editor.org/info/rfc9365>>.

6.2. Informative References

- [I-D.jeong-nmrg-ibn-network-management-automation]
Jeong, J. P., Ahn, Y., Gu, M., Kim, Y., and J. Jung-Soo, "Intent-Based Network Management Automation in 5G Networks", Work in Progress, Internet-Draft, draft-jeong-nmrg-ibn-network-management-automation-05, 4 November 2024, <<https://datatracker.ietf.org/doc/html/draft-jeong-nmrg-ibn-network-management-automation-05>>.
- [I-D.irtf-coinrg-coin-terminology]
Hong, J., Kunze, I., Wehrle, K., Trossen, D., Montpetit, M., de Foy, X., Griffin, D., and M. Rio, "Terminology for Computing in the Network", Work in Progress, Internet-Draft, draft-irtf-coinrg-coin-terminology-01, 10 July 2023, <<https://datatracker.ietf.org/doc/html/draft-irtf-coinrg-coin-terminology-01>>.

`[I-D.irtf-coinrg-use-cases]`

Kunze, I., Wehrle, K., Trossen, D., Montpetit, M., de Foy, X., Griffin, D., and M. Rio, "Use Cases for In-Network Computing", Work in Progress, Internet-Draft, draft-irtf-coinrg-use-cases-07, 4 December 2024, <<https://datatracker.ietf.org/doc/html/draft-irtf-coinrg-use-cases-07>>.

`[I-D.ietf-i2nsf-applicability]`

Jeong, J. P., Hyun, S., Ahn, T., Hares, S., and D. Lopez, "Applicability of Interfaces to Network Security Functions to Network-Based Security Services", Work in Progress, Internet-Draft, draft-ietf-i2nsf-applicability-18, 16 September 2019, <<https://datatracker.ietf.org/doc/html/draft-ietf-i2nsf-applicability-18>>.

`[I-D.ietf-i2nsf-capability-data-model]`

Hares, S., Jeong, J. P., Kim, J. T., Moskowitz, R., and Q. Lin, "I2NSF Capability YANG Data Model", Work in Progress, Internet-Draft, draft-ietf-i2nsf-capability-data-model-32, 23 May 2022, <<https://datatracker.ietf.org/doc/html/draft-ietf-i2nsf-capability-data-model-32>>.

`[I-D.ietf-i2nsf-registration-interface-dm]`

Hyun, S., Jeong, J. P., Roh, T., Wi, S., and J. Jung-Soo, "I2NSF Registration Interface YANG Data Model for NSF Capability Registration", Work in Progress, Internet-Draft, draft-ietf-i2nsf-registration-interface-dm-26, 10 May 2023, <<https://datatracker.ietf.org/doc/html/draft-ietf-i2nsf-registration-interface-dm-26>>.

`[I-D.ietf-i2nsf-consumer-facing-interface-dm]`

Jeong, J. P., Chung, C., Ahn, T., Kumar, R., and S. Hares, "I2NSF Consumer-Facing Interface YANG Data Model", Work in Progress, Internet-Draft, draft-ietf-i2nsf-consumer-facing-interface-dm-31, 15 May 2023, <<https://datatracker.ietf.org/doc/html/draft-ietf-i2nsf-consumer-facing-interface-dm-31>>.

`[I-D.ietf-i2nsf-nsf-facing-interface-dm]`

Kim, J. T., Jeong, J. P., Jung-Soo, J., Hares, S., and Q. Lin, "I2NSF Network Security Function-Facing Interface YANG Data Model", Work in Progress, Internet-Draft, draft-ietf-i2nsf-nsf-facing-interface-dm-29, 1 June 2022, <<https://datatracker.ietf.org/doc/html/draft-ietf-i2nsf-nsf-facing-interface-dm-29>>.

[I-D.ietf-i2nsf-nsf-monitoring-data-model]

Jeong, J. P., Lingga, P., Hares, S., Xia, L., and H. Birkholz, "I2NSF NSF Monitoring Interface YANG Data Model", Work in Progress, Internet-Draft, draft-ietf-i2nsf-nsf-monitoring-data-model-20, 1 June 2022, <<https://datatracker.ietf.org/doc/html/draft-ietf-i2nsf-nsf-monitoring-data-model-20>>.

[I-D.lingga-opsawg-analytics-interface-dm]

Lingga, P., Jeong, J. P., and Y. Choi, "I2NSF Analytics Interface YANG Data Model for Closed-Loop Security Control in the I2NSF Framework", Work in Progress, Internet-Draft, draft-lingga-opsawg-analytics-interface-dm-01, 13 February 2025, <<https://datatracker.ietf.org/doc/html/draft-lingga-opsawg-analytics-interface-dm-01>>.

[I-D.jeong-opsawg-security-management-automation]

Jeong, J. P., Lingga, P., Jung-Soo, J., Lopez, D., and S. Hares, "An I2NSF Framework for Security Management Automation in Cloud-Based Security Systems", Work in Progress, Internet-Draft, draft-jeong-opsawg-security-management-automation-01, 13 February 2025, <<https://datatracker.ietf.org/doc/html/draft-jeong-opsawg-security-management-automation-01>>.

[SPT]

Lingga, P., Jeong, J., Yang, J., and J. Kim, "SPT: Security Policy Translator for Network Security Functions in Cloud-Based Security Services", IEEE Transactions on Dependable and Secure Computing, Volume 21, Issue 6, DOI <https://doi.org/10.1109/TDSC.2024.3371788>, November 2024, <<https://doi.org/https://doi.org/10.1109/TDSC.2024.3371788>>.

[YAML]

Ingerson, B., Evans, C., and O. Ben-Kiki, "Yet Another Markup Language (YAML) 1.0", Available: <https://yaml.org/spec/history/2001-05-26.html>, October 2023.

[TS-23.501]

"System Architecture for the 5G System (5GS)", Available: <https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3144>, September 2023.

[TS-28.312]

"Intent Driven Management Services for Mobile Networks", Available:
<https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3554>, September 2023.

[TR-28.812]

"Study on Scenarios for Intent Driven Management Services for Mobile Networks", Available:
<https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3553>, December 2020.

[TS-23.288]

"Architecture Enhancements for 5G System (5GS) to Support Network Data Analytics Services", Available:
<https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3579>, September 2023.

[TS-29.520]

"Network Data Analytics Services", Available:
<https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3355>, September 2023.

[ETSI-NFV] "Network Functions Virtualisation (NFV); Architectural Framework", Available:

https://www.etsi.org/deliver/etsi_gs/nfv/001_099/002/01.02.01_60/gs_nfv002v010201p.pdf, December 2014.

[ETSI-NFV-Release-2]

"Network Functions Virtualisation (NFV) Release 2; Management and Orchestration; Architectural Framework Specification", Available:
https://www.etsi.org/deliver/etsi_gs/nfv/001_099/006/02.01.01_60/gs_nfv006v020101p.pdf, January 2021.

[NFV-COIN] Venancio, G., Turchetti, R., and E. Duarte Jr., "NFV-COIN: Unleashing The Power of In-Network Computing with Virtualization Technologies", SBC Journal of Internet Services and Applications, Available: <https://journals-sol.sbc.org.br/index.php/jisa/article/view/2342>, December 2022.

- [REST] Fielding, R. and R. Taylor, "Principled Design of the Modern Web Architecture", ACM Transactions on Internet Technology, Vol. 2, Issue 2,, Available: <https://dl.acm.org/doi/10.1145/514183.514185>, May 2002.
- [USENIX-ATC-Lumi] Jacobs, A., Pfitscher, R., Ribeiro, R., Ferreira, R., Granville, L., Willinger, W., and S. Rao, "Hey, Lumi! Using Natural Language for Intent-Based Network Management", USENIX Annual Technical Conference, Available: <https://www.usenix.org/conference/atc21/presentation/jacobs>, July 2021.
- [BERT] Devlin, J., Chang, M., Lee, K., and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding", NAACL-HLT Conference, Available: <https://aclanthology.org/N19-1423.pdf>, June 2019.
- [Deep-Learning] Goodfellow, I., Bengio, Y., and A. Courville, "Deep Learning", Publisher: The MIT Press, Available: <https://www.deeplearningbook.org/>, November 2016.
- [AUTOSAR-SDV] "AUTOSAR Adaptive Platform", Available: <https://www.autosar.org/standards/adaptive-platform>, March 2024.
- [Eclipse-SDV] "Eclipse Software Defined Vehicle Working Group Charter", Available: <https://www.eclipse.org/org/workinggroups/sdv-charter.php>, March 2024.
- [COVESA] "Connected Vehicle Systems Alliance", Available: <https://covesa.global/>, March 2024.
- [Kubernetes] "Kubernetes: Cloud Native Computing Platform", Available: <https://kubernetes.io/>, March 2024.

[Survey-IBN-CST-2023]

Leivadeas, A. and M. Falkner, "A Survey on Intent-Based Networking",
Available: <https://ieeexplore.ieee.org/document/9925251>,
March 2023.

[ClickINC] Xu, W., Zhang, Z., Feng, Y., Song, H., Chen, Z., Wu, W.,
Liu, G., Zhang, Y., Liu, S., Tian, Z., and B. Liu,
"ClickINC: In-network Computing as a Service in
Heterogeneous Programmable Data-center Networks",
Publisher: ACM SIGCOMM,
Available: <https://dl.acm.org/doi/10.1145/3603269.3604835>,
September 2023.

Acknowledgments

This work was supported by Institute of Information & Communications
Technology Planning & Evaluation (IITP) grant funded by the Korea
Ministry of Science and ICT (MSIT) (No. RS-2024-00398199 and RS-
2022-II221015).

Contributors

This document is made by the group effort of OPWAWG, greatly
benefiting from inputs and texts by Linda Dunbar (Futurewei), Yong-
Geun Hong (Daejeon University), and Joo-Sang Youn (Dong-Eui
University). The authors sincerely appreciate their contributions.

The following are coauthors of this document:

Mose Gu
Department of Computer Science & Engineering
Sungkyunkwan University
2066 Seobu-Ro, Jangan-Gu
Suwon
Gyeonggi-Do
16419
Republic of Korea
Phone: +82 31 299 4106
Email: rna0415@skku.edu
URI: <http://iotlab.skku.edu/people-Moses-Gu.php>

Juwon Hong
Department of Computer Science & Engineering
Sungkyunkwan University
2066 Seobu-Ro, Jangan-Gu
Suwon

Gyeonggi-Do
16419
Republic of Korea
Phone: +82 31 299 4106
Email: hongju2024@skku.edu
URI: <http://iotlab.skku.edu/people-Joo-Won-Hong.php>

Jiwon Suh
Department of Computer Science & Engineering
Sungkyunkwan University
2066 Seobu-Ro, Jangan-Gu
Suwon
Gyeonggi-Do
16419
Republic of Korea
Phone: +82 31 299 4106
Email: sjw6136@skku.edu
URI: <http://iotlab.skku.edu/people-Jiwon-Suh.php>

Giovanni Venancio
Department of Informatics
Federal University of Parana
Brazil
Email: giovanni@inf.ufpr.br

Authors' Addresses

Jaehoon Paul Jeong (editor)
Department of Computer Science & Engineering
Sungkyunkwan University
2066 Seobu-Ro, Jangan-Gu
Suwon
Gyeonggi-Do
16419
Republic of Korea
Phone: +82 31 299 4957
Email: pauljeong@skku.edu
URI: <http://iotlab.skku.edu/people-jaehoon-jeong.php>

Yiwen Shen
Department of Software
Ajou University
206 Worldcup-Ro, Yeongtong-Gu
Suwon

Gyeonggi-Do
16499
Republic of Korea
Email: chrisshen@ajou.ac.kr
URI: <https://chrisshen.github.io>

Yoseop Ahn
Department of Computer Science & Engineering
Sungkyunkwan University
2066 Seobu-Ro, Jangan-Gu
Suwon
Gyeonggi-Do
16419
Republic of Korea
Phone: +82 31 299 4106
Email: ahnjs124@skku.edu
URI: <http://iotlab.skku.edu/people-Ahn-Yoseop.php>

Younghan Kim
School of Electronic Engineering
Soongsil University
369, Sangdo-ro, Dongjak-gu
Seoul
06978
Republic of Korea
Email: younghak@ssu.ac.kr

Elias P. Duarte Jr.
Department of Informatics
Federal University of Parana
Brazil
Email: elias@inf.ufpr.br

Kehan Yao
China Mobile
Beijing
100053
China
Email: yaokehan@chinamobile.com