International Telecommunication Union

ITU-T

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

Y.3529

(02/2022)

SERIES Y: GLOBAL INFORMATION INFRASTRUCTURE, INTERNET PROTOCOL ASPECTS, NEXT-GENERATION NETWORKS, INTERNET OF THINGS AND SMART CITIES

Cloud Computing

Cloud computing – Data model framework for NaaS OSS virtualized network function

Recommendation ITU-T Y.3529

ITU-T Y-SERIES RECOMMENDATIONS

GLOBAL INFORMATION INFRASTRUCTURE, INTERNET PROTOCOL ASPECTS, NEXT-GENERATION NETWORKS, INTERNET OF THINGS AND SMART CITIES

GLOBAL INFORMATION INFRASTRUCTURE	
General	Y.100–Y.199
Services, applications and middleware	Y.200–Y.299
Network aspects	Y.300–Y.399
Interfaces and protocols	Y.400–Y.499
Numbering, addressing and naming	Y.500–Y.599
Operation, administration and maintenance	Y.600–Y.699
Security	Y.700–Y.799
Performances	Y.800–Y.899
INTERNET PROTOCOL ASPECTS	
General	Y.1000–Y.1099
Services and applications	Y.1100–Y.1199
Architecture, access, network capabilities and resource management	Y.1200–Y.1299
Transport	Y.1300–Y.1399
Interworking	Y.1400–Y.1499
Quality of service and network performance	Y.1500–Y.1599
Signalling	Y.1600–Y.1699
Operation, administration and maintenance	Y.1700–Y.1799
Charging	Y.1800–Y.1899
IPTV over NGN	Y.1900–Y.1999
NEXT GENERATION NETWORKS	
Frameworks and functional architecture models	Y.2000–Y.2099
Quality of Service and performance	Y.2100–Y.2199
Service aspects: Service capabilities and service architecture	Y.2200–Y.2249
Service aspects: Interoperability of services and networks in NGN	Y.2250–Y.2299
Enhancements to NGN	Y.2300–Y.2399
Network management	Y.2400–Y.2499
Computing power networks	Y.2500–Y.2599
Packet-based Networks	Y.2600–Y.2699
Security	Y.2700–Y.2799
Generalized mobility	Y.2800–Y.2899
Carrier grade open environment	Y.2900–Y.2999
FUTURE NETWORKS	Y.3000–Y.3499
CLOUD COMPUTING	**Y.3500–Y.3599**
BIG DATA	Y.3600–Y.3799
QUANTUM KEY DISTRIBUTION NETWORKS	Y.3800–Y.3999
INTERNET OF THINGS AND SMART CITIES AND COMMUNITIES	
General	Y.4000–Y.4049
Definitions and terminologies	Y.4050–Y.4099
Requirements and use cases	Y.4100–Y.4249
Infrastructure, connectivity and networks	Y.4250–Y.4399
Frameworks, architectures and protocols	Y.4400–Y.4549
Services, applications, computation and data processing	Y.4550–Y.4699
Management, control and performance	Y.4700–Y.4799
Identification and security	Y.4800–Y.4899
Evaluation and assessment	Y.4900–Y.4999

For further details, please refer to the list of ITU-T Recommendations.

Recommendation ITU-T Y.3529

Cloud computing – Data model framework for NaaS OSS virtualized network function

Summary

Recommendation ITU-T Y.3529 specifies the data model framework for network as a service (NaaS) OSS network function (OSS-NF), as a functional component of NaaS functional architecture defined in Recommendation ITU-T Y.3515, in the virtualized environment. It covers both the basic software-defined networking (SDN) and non-SDN functions of NaaS OSS-NF.

History

Edition	Recommendation	Approval	Study Group	Unique ID*
1.0	ITU-T Y.3529	2022-02-13	13	11.1002/1000/14859

Keywords

Data model, NaaS, OSS-NF, SDN, virtualized network function.

* To access the Recommendation, type the URL http://handle.itu.int/ in the address field of your web browser, followed by the Recommendation's unique ID. For example, http://handle.itu.int/11.1002/1000/11830-en.

FOREWORD

The International Telecommunication Union (ITU) is the United Nations specialized agency in the field of telecommunications, information and communication technologies (ICTs). The ITU Telecommunication Standardization Sector (ITU-T) is a permanent organ of ITU. ITU-T is responsible for studying technical, operating and tariff questions and issuing Recommendations on them with a view to standardizing telecommunications on a worldwide basis.

The World Telecommunication Standardization Assembly (WTSA), which meets every four years, establishes the topics for study by the ITU-T study groups which, in turn, produce Recommendations on these topics.

The approval of ITU-T Recommendations is covered by the procedure laid down in WTSA Resolution 1.

In some areas of information technology which fall within ITU-T's purview, the necessary standards are prepared on a collaborative basis with ISO and IEC.

NOTE

In this Recommendation, the expression "Administration" is used for conciseness to indicate both a telecommunication administration and a recognized operating agency.

Compliance with this Recommendation is voluntary. However, the Recommendation may contain certain mandatory provisions (to ensure, e.g., interoperability or applicability) and compliance with the Recommendation is achieved when all of these mandatory provisions are met. The words "shall" or some other obligatory language such as "must" and the negative equivalents are used to express requirements. The use of such words does not suggest that compliance with the Recommendation is required of any party.

INTELLECTUAL PROPERTY RIGHTS

ITU draws attention to the possibility that the practice or implementation of this Recommendation may involve the use of a claimed Intellectual Property Right. ITU takes no position concerning the evidence, validity or applicability of claimed Intellectual Property Rights, whether asserted by ITU members or others outside of the Recommendation development process.

As of the date of approval of this Recommendation, ITU had not received notice of intellectual property, protected by patents/software copyrights, which may be required to implement this Recommendation. However, implementers are cautioned that this may not represent the latest information and are therefore strongly urged to consult the appropriate ITU-T databases available via the ITU-T website at http://www.itu.int/ITU-T/ipr/.

© ITU 2022

All rights reserved. No part of this publication may be reproduced, by any means whatsoever, without the prior written permission of ITU.

Table of Contents

Recommendation ITU-T Y.3529

Cloud computing – Data model framework for NaaS OSS virtualized network function

1 Scope

This Recommendation specifies the data model framework for NaaS OSS-NF in the virtualized environment. It covers the following aspects:

– General description of NaaS OSS-NF;

– Data model framework for the basic functions of NaaS OSS-NF;

– Data model framework for the extensible functions of NaaS OSS-NF.

This Recommendation also provides an appendix describing:

– Examples of typical NaaS service plugin implementation framework.

NOTE – NaaS OSS-NF is also referred to as OSS-NF in this Recommendation.

2 References

The following ITU-T Recommendations and other references contain provisions which, through reference in this text, constitute provisions of this Recommendation. At the time of publication, the editions indicated were valid. All Recommendations and other references are subject to revision; users of this Recommendation are therefore encouraged to investigate the possibility of applying the most recent edition of the Recommendations and other references listed below. A list of the currently valid ITU-T Recommendations is regularly published. The reference to a document within this Recommendation does not give it, as a stand-alone document, the status of a Recommendation.

[ITU-T X.1601] Recommendation ITU-T X.1601 (2015), *Security framework for cloud computing.*

[ITU-T X.1604] Recommendation ITU-T X.1604 (2020), *Security requirements of Network as a Service (NaaS) in cloud computing*.

[ITU-T Y.3502] Recommendation ITU-T Y.3502 (2014) | ISO/IEC 17789:2014, *Information technology – Cloud computing – Reference architecture*.

[ITU-T Y.3515] Recommendation ITU-T Y.3515 (2017), *Cloud computing – Functional architecture of Network as a Service*.

3 Definitions

3.1 Terms defined elsewhere

This Recommendation uses the following terms defined elsewhere:

3.1.1 network as a Service (NaaS) [b-ITU-T Y.3500]: Cloud service category in which the capability provided to the cloud service customer (CSC) is transport connectivity and related network capabilities.

3.1.2 network function [ITU-T Y.3515]: A function of a network infrastructure whose external interfaces and functional behaviour are well specified.

NOTE – Examples of network functions include network switches and network routers.

3.1.3 software-defined networking [b-ITU-T Y.3300]: A set of techniques that enables to directly program, orchestrate, control and manage network resources, which facilitates the design, delivery and operation of network services in a dynamic and scalable manner.

3.1.4 virtualized network function [ITU-T Y.3515]: A network function that can be deployed as a software on a NaaS cloud service provider (CSP) infrastructure.

NOTE – Examples of virtualized network functions include virtual switches and virtual routers.

3.2 Terms defined in this Recommendation

None.

4 Abbreviations and acronyms

This Recommendation uses the following abbreviations and acronyms:

API Application Programming Interface

BGP Border Gateway Protocol

CIDR Classless Inter-Domain Routing

CRUD Create, Read, Update, Delete

CSC Cloud Service Customer

CSN Cloud Service Partner

CSP Cloud Service Provider

DNS Domain Name System

FWaaS Firewall as a Service

LBaaS Load Balance as a Service

MTU Maximum Transmission Unit

NaaS Network as a Service

QoS Quality of Service

SDN Software-Defined Networking

VIM Virtualized Infrastructure Manager

VNF Virtualized Network Function

VNFM Virtualized Network Function Manager

VPN Virtual Private Network

5 Conventions

The keywords "**is required to**" indicate a requirement that must be strictly followed and from which no deviation is permitted if conformance to this Recommendation is to be claimed.

In the body of this Recommendation and its annexes, the words should and may sometimes appear, in which case they are to be interpreted, respectively, as is recommended and can optionally. The appearance of such phrases or keywords in an appendix or in material explicitly marked as informative is to be interpreted as having no normative intent.

6 General description of NaaS OSS-NF

Figure 6-1 presents the position of OSS-NF and its corresponding reference points, with the highlight on the architecturally newly added software-defined networking (SDN) controller and its reference points with OSS-NF and the virtualized network function (VNF) based on [ITU-T Y.3515]. Considering that VNF is always modelled in SDN or non-SDN scenarios as one kind of NaaS resource including the forwarding device or the non-forwarding device, it is required to standardize the interaction mechanism driven by data models including the interaction between OSS-NF, VNF and SDN controller in the virtualized environment.

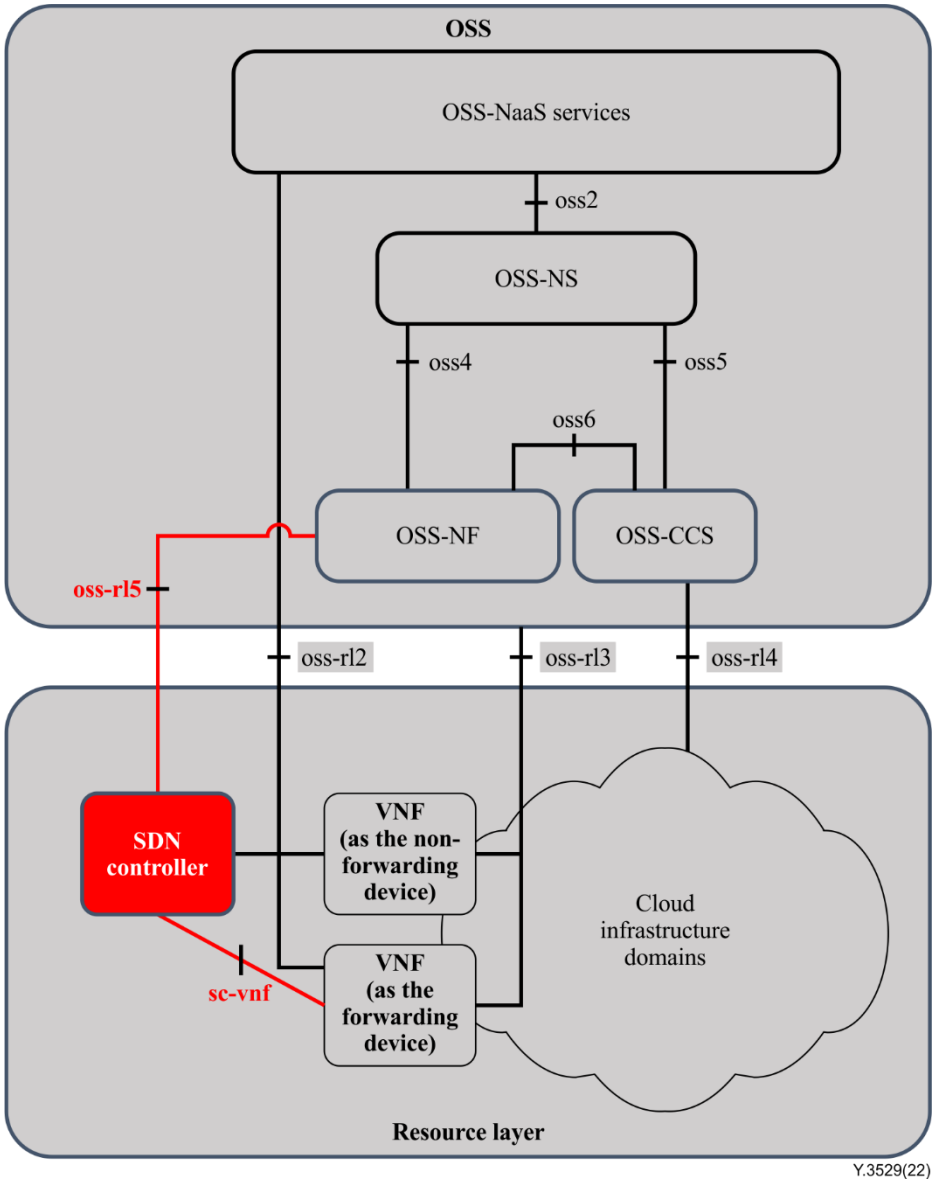

Figure 6-1 – OSS-NF and its related reference points

6.1 OSS-NF functional components in the virtualized environment

As defined in [ITU-T Y.3515], OSS-NF functional components are responsible for the management of network functions such as instantiation, update, query, scaling, and termination of network functions. In the virtualized environment, on the one hand, it can perform the function of virtualized network function manager (VNFM) see Appendix IV of [ITU-T Y.3515], and trigger the VNF; on the other hand, it can trigger the different SDN controllers as the dedicated plugins.

To perform the function of VNFM, OSS-NF functional components in the virtualized environment are required to provide the basic resource data models, such as network, port, subnet, etc., and the

extensible agent function data models, linking to the dedicated VNFs, such as layer 2, layer 3 connectivity, etc. To activate different SDN controllers, OSS-NF functional components in the virtualized environment are required to provide the SDN extension data models as the plugins to match the corresponding SDN controllers.

6.2 oss-rl5

This reference point covers the interactions between the OSS-NF functional components and an SDN controller functional component. It includes the interactions related to matching the SDN controller and its corresponding plugin.

6.3 sc-vnf

This reference point covers the interactions between a SDN controller functional component and a VNF functional component (as the forwarding device). It includes the interactions related to control of the virtualized network forwarding devices.

6.4 oss-rl2

This reference point covers interactions between the OSS-NaaS functional components and a VNF functional component (as the forwarding device or the non-forwarding device). It includes interactions related to the general VNF management on configuration, fault and performance from a NaaS service perspective.

6.5 oss-rl3

This reference point covers interactions between the OSS-NF functional components and a VNF functional component (as the forwarding device or the non-forwarding device). It includes interactions related to the general VNF management on configuration, fault and performance. In addition, it also includes interactions related to matching the customized VNF (as the non-forwarding device) and its corresponding plugin.

6.6 oss6

This reference point covers interactions between the OSS-NF functional components and OSS-CCS functional components, taking the role of a virtualized infrastructure manager (VIM). It includes interactions between a computing function and a networking function.

6.7 oss4

This reference point covers interactions between the OSS-NF functional components and OSS-NS functional components. It includes interactions related to VNF lifecycle management, VNF fault and performance management, VNF usage management and VNF inventory management.

7 Data model framework for the basic functions of NaaS OSS-NF

7.1 Resource data models

The core elements of resource data models include network, port and subnet. The relationship among network, port, and subnet is shown in Figure 7-1.

Figure 7-1 – Relationship between a network, port and subnet

A network is a virtual domain that is typically reserved to the tenant who created it unless the network has been explicitly configured to be shared. A subnet represents an IP address block that can be used for assigning IP addresses to virtual instances. A network may have many subnets and a subnet only belongs to one network, therefore, the relationship between network and subnet is one to many.

A port represents a virtual switch port on a logical network switch. A network may have many ports and a port only belongs to one network, therefore, the relationship between network and port is one to many. When an IP address is associated with a port, it also implies that the port is associated with a subnet, as the IP address is taken from the allocation pool for a specific subnet. A port may have many IP addresses that belong to different subnets, therefore, the relationship between a subnet and port is many to many.

Apart from core elements, in order to connect different networks, it is also required to define the router and border gateway protocol (BGP) virtual private network (VPN) as the resource data models.

7.1.1 Network

The mandatory data model description for a network is specified in Table 7-1.

Table 7-1 – Data model description for a network

Element	Type	Description
NAME	STRING	Name of the network
CREATION_TIME	STRING	Creation time of the network
UPDATED_TIME	STRING	Latest updated time of the network
NETWORK_ID	STRING	Unique identification of the network
MTU	INTEGER	Maximum transmission unit
TENANT_ID	STRING	Unique identification of the tenant
PROVIDER_NETWORK_TYPE	STRING	Physical network type that this network is mapped to. For example, flat, vlan, vxlan or gre
PROVIDER_PHYSICAL_NETWORK	STRING	Physical network where this network is implemented
PROVIDER_SEGMENTATION_ID	INTEGER	Unique identification of the network segmentation on the physical network. The PROVIDER_NETWORK_TYPE attribute defines the segmentation model. For example, if the PROVIDER_NETWORK_TYPE value is vlan, this ID is a vlan ID.
QOS_POLICY_ID	STRING	Unique identification of the quality of service (QoS) policy associated with the network
ROUTER:EXTERNAL	BOOLEAN	Whether this network is associated with the external network via the router
SHARED	BOOLEAN	Whether this network could be shared with all the tenants
STATUS	STRING	Status of the network
SUBNETS	ARRAY	List of the subnets associated with this network
DESCRIPTION	STRING	Description of the network

7.1.2 Subnet

The mandatory data model description for a subnet is specified in Table 7-2.

Table 7-2 – Data model description for a subnet

Element	Type	Description
NAME	STRING	Name of the subnet
SUBNET_ID	STRING	Unique identification of the subnet
TENANT_ID	STRING	Unique identification of the tenant
CREATION_TIME	STRING	Creation time of the subnet
UPDATED_TIME	STRING	Latest updated time of the subnet
NETWORK_ID	STRING	Unique identification of the network associated with this subnet
DNS_NAMESERVERS	ARRAY	List of the domain name system (DNS) associated with this subnet
ALLOCATION_POOLS	ARRAY	Allocated IP address pools
HOST_ROUTES	ARRAY	Route table associated with this subnet
IP_VERSION	INTEGER	IP Version
GATEWAY-IP	STRING	Gateway IP of this subnet
CIDR	STRING	Classless inter-domain routing of this subnet
DESCRIPTION	STRING	Description of the subnet
SERVICE_TYPE	STRING	Service type associated with this subnet

7.1.3 Port

The mandatory data model description for a port is specified in Table 7-3.

Table 7-3 – Data model description for a port

Element	Type	Description
NAME	STRING	Name of the port
PORT_ID	STRING	Unique identification of the port
MAC_ADDRESS	STRING	Mac address of the port
ADMIN_STATUS	BOOLEAN	Whether the administration state of this port is active
BINDING_HOST_ID	STRING	Unique identification of the host to which the port belongs
ALLOWED_ADDRESS_PAIRS	ARRAY	Allocated address pairs (IP address and MAC address) which can be taken as a source address to receive messages
BINDING_VNIC_TYPE	STRING	The type of virtual network interface cards binding the port. What type of virtual network interface card depends on for deployments
DESCRIPTION	STRING	Description of the port

Table 7-3 – Data model description for a port

Element	Type	Description
DEVICE_ID	STRING	Unique identification of the device using this port
DEVICE_TYPE	STRING	Type of the device that uses this port
FIXED_IPS	ARRAY	IP address group of this port
NETWORK_ID	STRING	Unique identification of the network associated with this port
PORT_SECURITY_ENABLED	BOOLEAN	Security status of this port. If port security is enabled for the port, security group rules are applied to the traffic on the port.
SECURITY_GROUPS	ARRAY	Security groups associated with this port
STATUS	STRING	Status of the port
TENANT_ID	STRING	Unique identification of the tenant

7.1.4 Router

The mandatory data model description for a router is specified in Table 7-4.

Table 7-4 – Data model description for a router

Element	Type	Description
ROUTER_ID	STRING	Unique identification of the router
DESCRIPTION	STRING	Description of the router
ADMIN_STATE	BOOLEAN	Whether the administration state of this router is active
STATUS	STRING	Status of the router
NETWORK_ID	STRING	Unique identification of the network
ROUTES	ARRAY	External route information. The format of the tuple is [DESTINATION, NEXTHOP].
DESTINATION	STRING	The CIDR of the destination IP address
NEXTHOP	STRING	The next hop for reaching the destination

7.1.5 BGP VPN

The mandatory data model description for BGP VPN (defined in [b-IETF RFC 4364]) is specified in Table 7-5.

Table 7-5 – Data model description for BGP VPN

Element	Type	Description
BGP_VPN_ID	STRING	Unique identification of the BGP VPN
TYPE	STRING	The type of the VPN, including layer 2 VPN and layer 3 VPN
RD_LIST	ARRAY	The list of RD (Route distinguisher)
RT_LIST	ARRAY	The list of RT (Route target)

Table 7-5 – Data model description for BGP VPN

Element	Type	Description
IMPORT_TARGETS	ARRAY	The import of the RT_LIST
EXPORT_TARGETS	ARRAY	The export of the RT_LIST
NETWORKS	ARRAY	The network list associated with this VPN
ROUTERS	ARRAY	The router list associated with this VPN
PORTS	ARRAY	The port list associated with this VPN
LOCAL_PREF	INTEGER	The default property of BGP local-preference

7.2 Operations of the resource data models

The NaaS OSS-NF resource data models have similar operations. Table 7-6 provides the description of their operations, taking the network as an example.

Table 7-6 – Operations of the network data model

No.	Operation	HTTP Action	Description
1	List networks	GET	The information of all the networks is listed
2	Show network	GET	The information of the specific network is shown
3	Create network	POST	One network is created
4	Bulk-create networks	POST	One group of networks is created
5	Update network	PUT	The specific network is modified
6	Delete network	DELETE	The specific network is deleted

CSP provides functions of NaaS OSS-NF through an application programming interface (API). An API is a RESTful HTTP service that uses all aspects of the HTTP protocol including methods, URIs, media types, response codes, etc. The elements listed in Table 7-1 could be used as the request and response parameters of API. Combining with the operations listed in Table 7-6, they constitute the core APIs of NaaS OSS-NF.

8 Data models framework for the extensible functions of NaaS OSS-NF

8.1 Categories of the extensible mechanism

Based on the resource data models defined in Table 7-1, only the NaaS service with limited functions can be provided by CSP. However, in the real complex networking environment, it is required for CSP to provide the NaaS service with extensible function.

NaaS OSS-NF provides the extensible functions based on the basic resource data models in the following two mechanisms.

– Extending functions through the core APIs which are defined by NaaS OSS-NF and linked to the corresponding resource data models. The API could be extended by adding new operations or extra attributes for the resource data model. For example, to make it possible to associate a subnet with a specific segment on the network, instead of spanning all the segments in the network, the SEGMENT_ID attribute could be added to the data model of the subnet to indicate which segment on the network it associates, and the implementation of this API should be modified accordingly to support this extension.

– Extending functions by adding a new API, which is not originally defined by the NaaS OSS-NF. The newly defined API links to the dedicated NaaS service plugin and its implementation developed by the cloud service partner (CSN). By this extensible mechanism, SDN controllers of different vendors are called by OSS-NF as the plugins via oss-rl5 and the customized VNFs (as the non-forwarding devices) such as virtualized firewall and virtualized load balancer are activated by OSS-NF as the plugins via oss-rl3. The examples of a typical NaaS service plugin implementation framework are provided in Appendix I.

8.2 Procedures for the extensible mechanism by new API definition

The procedures for extending functions by adding a new API are described as follows:

1) CSP defines a new API for supporting the NaaS service, which includes the data models of the necessary resources and the operations for these data models.
2) CSN develops the dedicated NaaS service plugin, NaaS service agent for OSS-NF and the implementation of the VNF for the NaaS service.
3) CSN deploys the plugin, agent and the implementation in CSP's environment, which means the new API for supporting the NaaS service is ready.
4) CSP publishes the NaaS service to CSC.

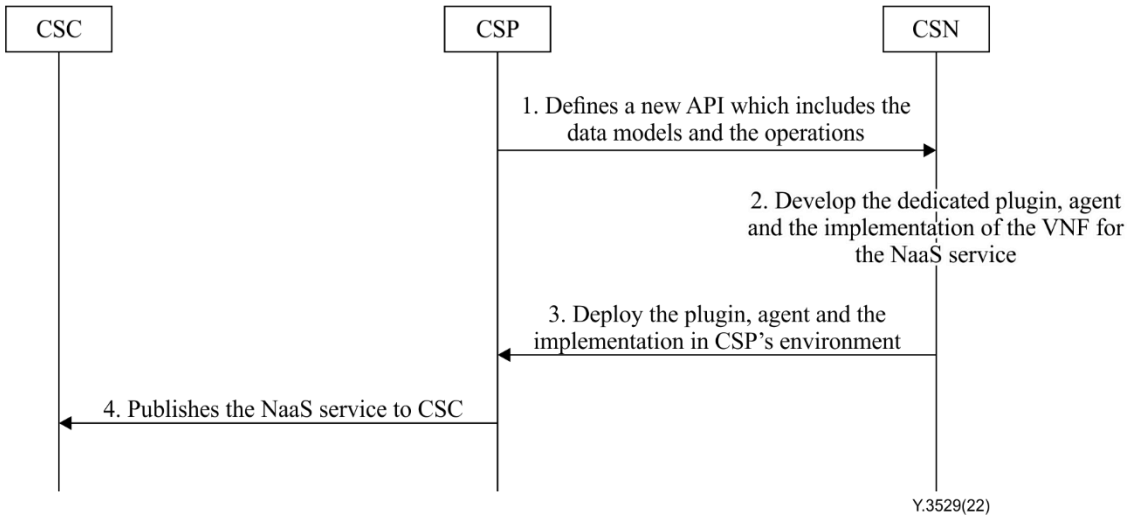

Figure 8-1 – Procedures for the extensible mechanism by new API definition

9 Security consideration

Consideration for security aspects within the cloud computing environment, including NaaS, are addressed for NaaS CSPs, as described in [ITU-T X.1601] and [ITU-T X.1604]. In particular, [ITU-T X.1601] and [ITU-T X.1604] analyse security threats and challenges and describes security capabilities that could mitigate these threats and meet the security challenges.

The functional components for security systems defined in clause 9.2.5.2 of [ITU-T Y.3502] are applicable in the context of the NaaS functional architecture. These components are responsible for applying security related controls to mitigate the security threats in cloud computing environments and encompass all the security facilities required to support cloud services of the NaaS cloud service category.

Appendix I

Examples of typical NaaS service plugin implementation framework

(This appendix does not form an integral part of this Recommendation.)

This appendix provides examples of a NaaS service plugin implementation framework as the extensible functions of NaaS OSS-NF by adding a new API.

I.1 FWaaS plugin

In order to implement the firewall as a service (FwaaS), the data models of firewall, policy and rule need to be defined. The NaaS CSP provides FwaaS service instances based on the CSC's request. A firewall is associated with a group of policies and each policy is an ordered list of rules. Policy templates are created by the NaaS CSP and can be shared by different CSC. Rules cannot be applied to the firewall directly and are required to be added to the dedicated policy.

The firewall plugin implementation framework is shown in Figure I.1. The firewall plugin and firewall agent are located in OSS-NF functional components and firewall VNF is located in VNF functional components, which are all developed by CSN. The firewall plugin sends the create, read, update, delete (CRUD) operation request and the firewall agent analyses the request, checks the validation, and sends back the status report to the firewall plugin. If the validation succeeds, the firewall agent activates the corresponding operation of the firewall VNF via the interface oss-rl3.

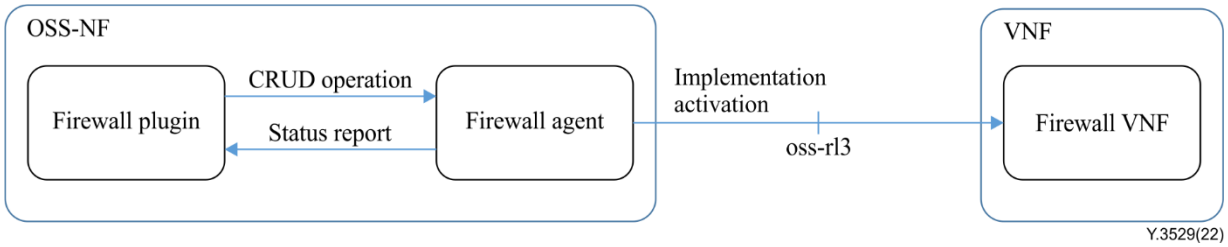

Figure I.1 – FWaaS plugin implementation framework

The procedures for activating firewall VNF are described as follows.

Step 1: The firewall plugin receives the CSC's firewall creation request message via the newly defined API.

Step 2: The firewall plugin sends the creation operation request message to the firewall agent.

Step 3: The firewall agent analyzes the creation operation request and checks its validation. If the validation succeeds, the firewall agent sends back the confirmation message to the firewall plugin and activates the firewall VNF via the interface oss-rl3. If the validation fails, the firewall agent sends back a failure message to the firewall plugin.

Step 4: The firewall VNF receives the activation message and creates the FWaaS service instance.

I.2 LBaaS plugin

The NaaS CSP provides a load balance as a service (LBaaS) service instance in order to balance the CSC's workload among different VMs. The concrete functionalities of LBaaS service include, but are not limited to,

– providing load balance for different protocols, e.g., TCP, HTTP, etc.;
– monitoring the status of the NaaS service;
– restricting inbound access for attack prevention;

– persisting session via source IP or route cookie for sending the CSC's request to the dedicated VM.

The load balance plugin implementation framework is shown in Figure I.2. The load balance plugin and load balance agent are located in OSS-NF functional components and load balance VNFs are located in VNF functional components, which are all developed by CSN. The load balance plugin is responsible for the creation, monitoring, and scheduling of the computing pool which is composed of VNFs. The computing pool is created for workload balance and the dedicated load balance agent is allocated for each computing pool.

The load balance plugin sends the CRUD operation request to the selected load balance agent based on the monitoring and scheduling results, and the load balance agent analyses the request, checks the validation and sends back the status report to the load balance plugin. If the validation succeeds, the load balance agent activates the corresponding operation of the load balance VNFs via the interface oss-rl3.

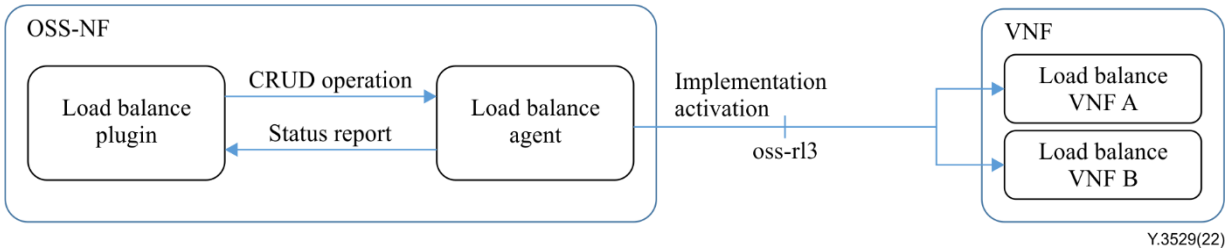

Figure I.2 – LBaaS plugin implementation framework

The procedures for activating load balance VNFs are described as follows.

Step 1: The load balance plugin receives the CSC's workload balance request message via the newly defined API.

Step 2: The load balance plugin sends the load balance service creation operation request message to the selected load balance agent based on the monitoring and scheduling results.

Step 3: The load balance agent analyses the creation operation request and check its validation. If the validation succeeds, the load balance agent sends back the confirmation message to the load balance plugin and activates multiple associated load balance VNFs in the computing pool via the interface oss-rl3. If the validation fails, the load balance agent sends back a failure message to the load balance plugin.

Step 4: The load balance VNFs in the computing pool receives the activation message and creates the LBaaS service instance.

I.3 Abnormal traffic cleaning plugin

In order to clean the abnormal traffic flexibly according to the CSC's on-demand service request without the re-deployment, re-configuration and replacement of the physical network devices, the NaaS CSP provides the software-defined abnormal traffic cleaning service based on the OpenFlow protocol [b-ONF TS-025]. The implementation mechanism provides a unified traffic traction and reinjection by separating the network control plane and the network forwarding plane.

The abnormal traffic cleaning plugin implementation framework is shown in Figure I.3. The traffic cleaning plugin and traffic cleaning agent are located in the OSS-NF functional components and the OpenFlow traffic cleaning switch VNF and OpenFlow service router VNF are located in the VNF functional components, which are controlled by the OpenFlow traffic cleaning controller via the OpenFlow protocol. The above mentioned OpenFlow-based controller, switch and router are all developed by the CSN.

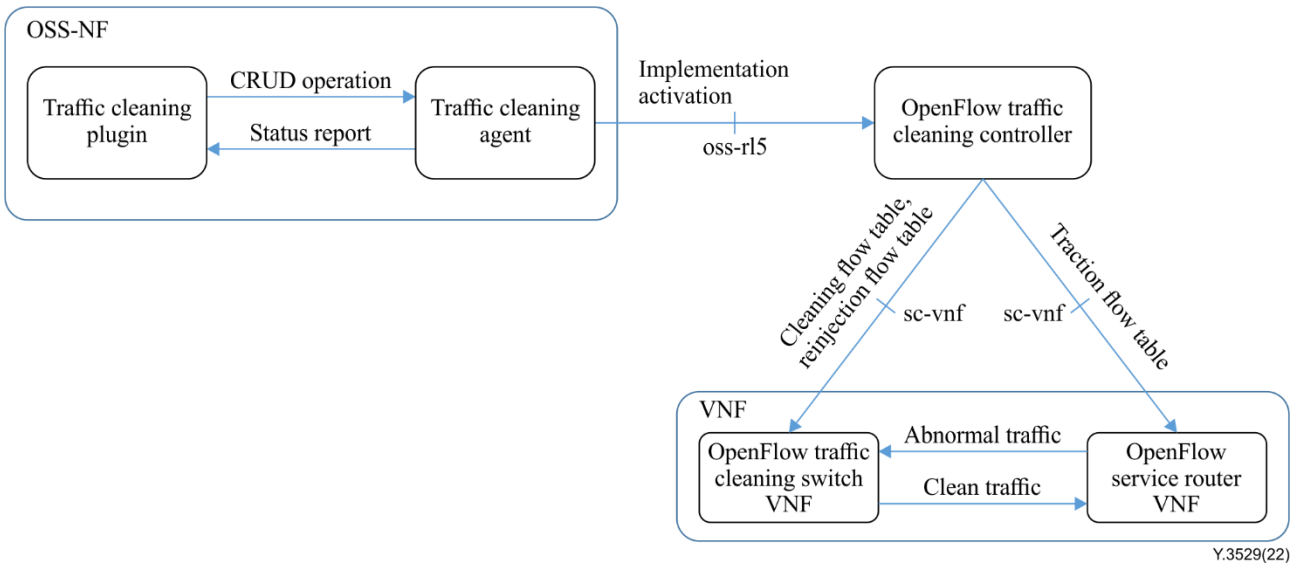

Figure I.3 – Abnormal traffic cleaning plugin implementation framework

The traffic cleaning plugin sends the CRUD operation request, and the traffic cleaning agent analyses the request, checks the validation, and sends back the status report to the traffic cleaning plugin. If the validation succeeds, the traffic cleaning agent activates the corresponding operation of the OpenFlow traffic cleaning controller via the interface oss-rl5.

The procedures for activating the software-defined abnormal traffic cleaning operation are described as follows.

Step 1: The traffic cleaning plugin receives the CSC's abnormal traffic cleaning request message via the newly defined API.

Step 2: The traffic cleaning plugin sends the abnormal traffic cleaning service creation operation request message to the traffic cleaning agent.

Step 3: The traffic cleaning agent analyzes the creation operation request and checks its validation. If the validation succeeds, the traffic cleaning agent sends back the confirmation message to the traffic cleaning plugin and activates the OpenFlow traffic cleaning controller via the interface oss-rl5 by providing the pre-defined abnormal traffic characteristics. If the validation fails, the traffic cleaning agent sends back the failure message to the traffic cleaning plugin.

Step 4: The OpenFlow traffic cleaning controller creates the traffic cleaning flow table based on the received abnormal traffic characteristics and send it to the OpenFlow traffic cleaning switch VNF via the interface sc-vnf.

Step 5: When the traffic attack happens, the OpenFlow traffic cleaning controller generates the traffic traction flow table based on the information of the attached server and the OpenFlow traffic cleaning switch VNF and sends it to the OpenFlow service router VNF via the interface sc-vnf.

Step 6: The OpenFlow traffic cleaning controller generates the traffic reinjection flow table based on the information of the attached server and sends it to the OpenFlow traffic cleaning switch VNF via the interface sc-vnf.

Step 7: The OpenFlow service router VNF forwards the abnormal traffic to the OpenFlow traffic cleaning switch VNF based on the traffic traction flow table.

Step 8: The OpenFlow traffic cleaning switch VNF matches the tractive traffic based on the traffic cleaning flow table, discards the matched traffic, and sends the clean traffic to the OpenFlow service router VNF based on the traffic reinjection flow table.

Bibliography

[b-ITU-T Y.3300] Recommendation ITU-T Y.3300 (2014), *Framework of software-defined networking.*

[b-ITU-T Y.3500] Recommendation ITU-T Y.3500 (2014) | ISO/IEC 17788:2014, *Information technology – Cloud computing – Overview and vocabulary.*

[b-IETF RFC 4364] IETF RFC 4364 (2006), *BGP/MPLS IP Virtual Private Networks (VPNs).* <https://datatracker.ietf.org/doc/rfc4364/>

[b-ONF TS-025] ONF TS-025 (2015), *OpenFlow Switch Specification (Version 1.5.1).* <https://opennetworking.org/wp-content/uploads/2014/10/openflow-switch-v1.5.1.pdf>

SERIES OF ITU-T RECOMMENDATIONS

Series A	Organization of the work of ITU-T
Series D	Tariff and accounting principles and international telecommunication/ICT economic and policy issues
Series E	Overall network operation, telephone service, service operation and human factors
Series F	Non-telephone telecommunication services
Series G	Transmission systems and media, digital systems and networks
Series H	Audiovisual and multimedia systems
Series I	Integrated services digital network
Series J	Cable networks and transmission of television, sound programme and other multimedia signals
Series K	Protection against interference
Series L	Environment and ICTs, climate change, e-waste, energy efficiency; construction, installation and protection of cables and other elements of outside plant
Series M	Telecommunication management, including TMN and network maintenance
Series N	Maintenance: international sound programme and television transmission circuits
Series O	Specifications of measuring equipment
Series P	Telephone transmission quality, telephone installations, local line networks
Series Q	Switching and signalling, and associated measurements and tests
Series R	Telegraph transmission
Series S	Telegraph services terminal equipment
Series T	Terminals for telematic services
Series U	Telegraph switching
Series V	Data communication over the telephone network
Series X	Data networks, open system communications and security
Series Y	**Global information infrastructure, Internet protocol aspects, next-generation networks, Internet of Things and smart cities**
Series Z	Languages and general software aspects for telecommunication systems

Printed in Switzerland
Geneva, 2022