

OPS Area Working Group
Internet-Draft
Intended status: Informational
Expires: January 6, 2017

Q. Wu
W. Liu
Huawei Technologies
A. Farrel
Juniper Networks
July 5, 2016

Service Models Explained
draft-wu-opsawg-service-model-explained-00

Abstract

The IETF has produced a considerable number of data models in the YANG modelling language. The majority of these are used to model devices and they allow access for configuration and to read operational status.

A small number of YANG models are used to model services (for example, the Layer Three Virtual Private Network Service Model produced by the L3SM working group).

This document briefly sets out the scope of and purpose of an IETF service model, and it shows where a service model might fit into a Software Defined Networking architecture or deployment.

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 <http://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 January 6, 2017.

Copyright Notice

Copyright (c) 2016 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 (<http://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 Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1.	Introduction	2
2.	Terms and Concepts	3
3.	Using Service Models	4
4.	Service Models in an SDN Context	5
5.	Possible Causes of Confusion	8
6.	Further Concepts	9
6.1.	Technology Agnostic	9
6.2.	Relationship to Policy	9
6.3.	Operator-Specific Features	10
6.4.	Supporting Multiple Services	10
7.	Security Considerations	10
8.	Manageability Considerations	10
9.	IANA Considerations	10
10.	Acknowledgements	11
11.	References	11
11.1.	Normative References	11
11.2.	Informative References	11
	Authors' Addresses	11

[1.](#) Introduction

In recent years the number of data models written in the YANG modelling language [[RFC6020](#)] for configuration and monitoring has blossomed. Many of these are used for device-level configuration (for example, [[RFC7223](#)]) or for control of protocols (for example, [[RFC7407](#)]).

Within the context of Software Defined Networking (SDN) [[RFC7426](#)] YANG data models may be used on Southbound Interfaces (SBIs) between a controller and network devices, and between network orchestrators and controllers.

Recently there has been interest in using YANG to define and document data models that describe services in a portable way that is independent of which network operator uses the model. These models may be used in manual and even paper-driven service request processes

moving to IT-based mechanisms. Ultimately they could be used in online, software-driven dynamic systems.

This document explains the scope and purpose of service models within the IETF and describes how a service model can be used by a network operator. Equally, this document clarifies what a service model is not, and dispells some common misconceptions.

2. Terms and Concepts

The following terms are used in this document:

Network Operator: This term is used interchangeably to refer to the company that owns a network that provides Internet connectivity and services, or the individual who performs operations and management on that network.

Customer: Someone who purchases connectivity and other services from a network operator. In the context of this document, a customer is usually the company that runs their own network or computing platforms and wishes to connect to the Internet or between sites. Such a customer may operate an enterprise network or a data center. Sometimes this term may also be used to refer to the individual in such a company who contracts to buy services from a network operator. A customer as described here is a separate commercial operation from the network operator, but some companies may operate with internal customers so that, for example, an IP/MPLS packet network is the customer of an optical transport network.

Service: A network operator delivers one or more services to a customer. A service is some form of connectivity between customer sites and the Internet or between customer sites across the network operator's network and across the Internet. A service may be limited to simple connectivity (such as IP-based Internet access), may be a tunnel (such as a virtual circuit), or may be a more complex connectivity model (such as a multi-site virtual private network). Services may be further enhanced by additional functions providing security, load-balancing, accounting, and so forth. Additionally, services usually include guarantees of quality, throughput, and fault reporting.

Data Model: The concepts of information models and data models are described in [[RFC3444](#)]. That document defines a data model by contrasting it with the definition of an information model, so it may be helpful to quote some text to give context within this document.

The main purpose of an information model is to model managed objects at a conceptual level, independent of any specific implementations or protocols used to transport the data. The degree of specificity (or detail) of the abstractions defined in the information model depends on the modeling needs of its designers. In order to make the overall design as clear as possible, an information model should hide all protocol and implementation details. Another important characteristic of an information model is that it defines relationships between managed objects.

Data models, conversely, are defined at a lower level of abstraction and include many details. They are intended for implementors and include protocol-specific constructs.

Service Model: A service model is a specific type of data model. It describes a service and all of the parameters of the service in a portable, operator-independent way. It can be used by a human or by software to configure or request a service and may equally be consumed by a human or by a software component.

It needs to be repeatedly clarified that a service model is not a data model used to directly configure network devices, protocols, or functions: it is not something that is sent to network devices (i.e., routers or switches) for processing. Equally, a service model is not a data model that describes how a network operator realizes and delivers the service described by the model. This issue is discussed further in later sections.

3. Using Service Models

As already indicated, service models are used on the interface between customers and network operators. This is simply shown in Figure 1

The language in which a service model is described is a choice for whoever specifies the model. The IETF uses the YANG data modeling language defined in [[RFC6020](#)]

The encoding and communication protocol used to exchange a service model between customer and network operator are deployment- and implementation-specific. The IETF recommends the use of the NETCONF Configuration Protocol [[RFC4741](#)] with data encoded in XML or JSON for interactions "on the wire" between software components. However, co-located software components might use an API, while systems with more direct human interactions might use web pages or even paper forms.

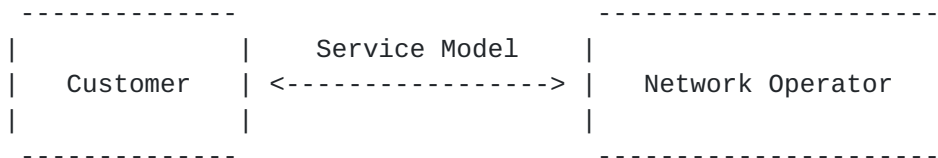


Figure 1: Service Models used on the Interface between Customers and Network Operators

How a network operator processes a service request described by a service model will depend on the commercial and operational tools, processes, and policies used by the operator. These may vary considerably from one network operator to another.

However, the intent is that the network operator maps the service request into configuration and operational parameters that control one or more network to deliver the requested services. That means that the network operator (or software run by the network operator) takes the information in the service model and determines how to deliver the service by enabling and configuring network protocols and devices.

[4.](#) Service Models in an SDN Context

In an SDN system, the control and configuration of network resources and protocols is performed by software systems that determine how best to utilize the network. Figure 2 shows common architectural view of an SDN system where network elements are programmed by a component called a Controller, and where Controllers are instructed by an Orchestrator that has a wider view of the whole of, or part of, a network.

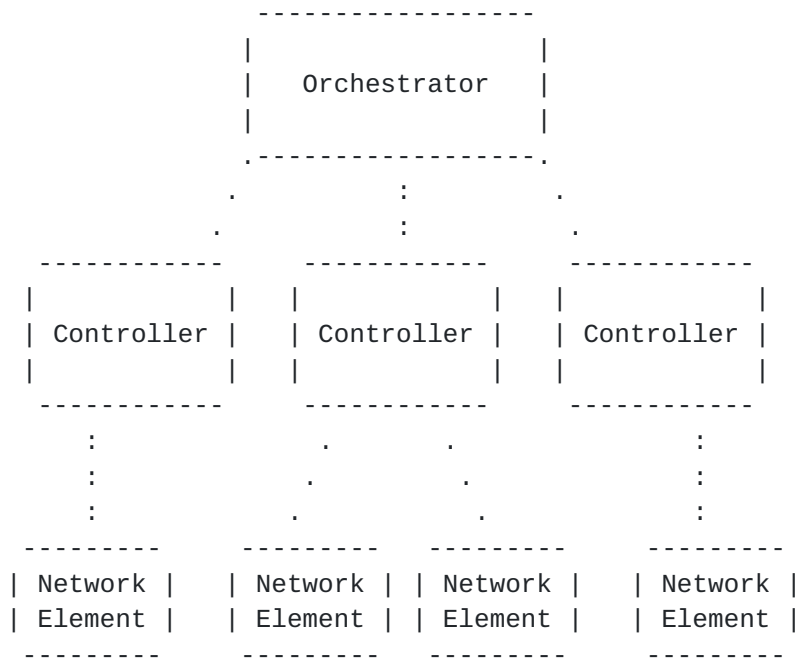


Figure 2: A Common SDN Architecture

But a service request is (or should be) network-agnostic. That is, there should be an independence between the behavior and functions that a customer requests and the technology that the network operator has available to deliver the service. This means that the service request must be mapped to the Orchestrator's view, and this mapping may include a choice of which networks to use depending on what technologies are available and which service features have been requested.

This mapping can be achieved by splitting the orchestration function between a "Service Orchestrator" and a "Network Orchestrator" as shown in Figure 3. In a system that is fully implemented in software, this could lead to agile service delivery or service automation.

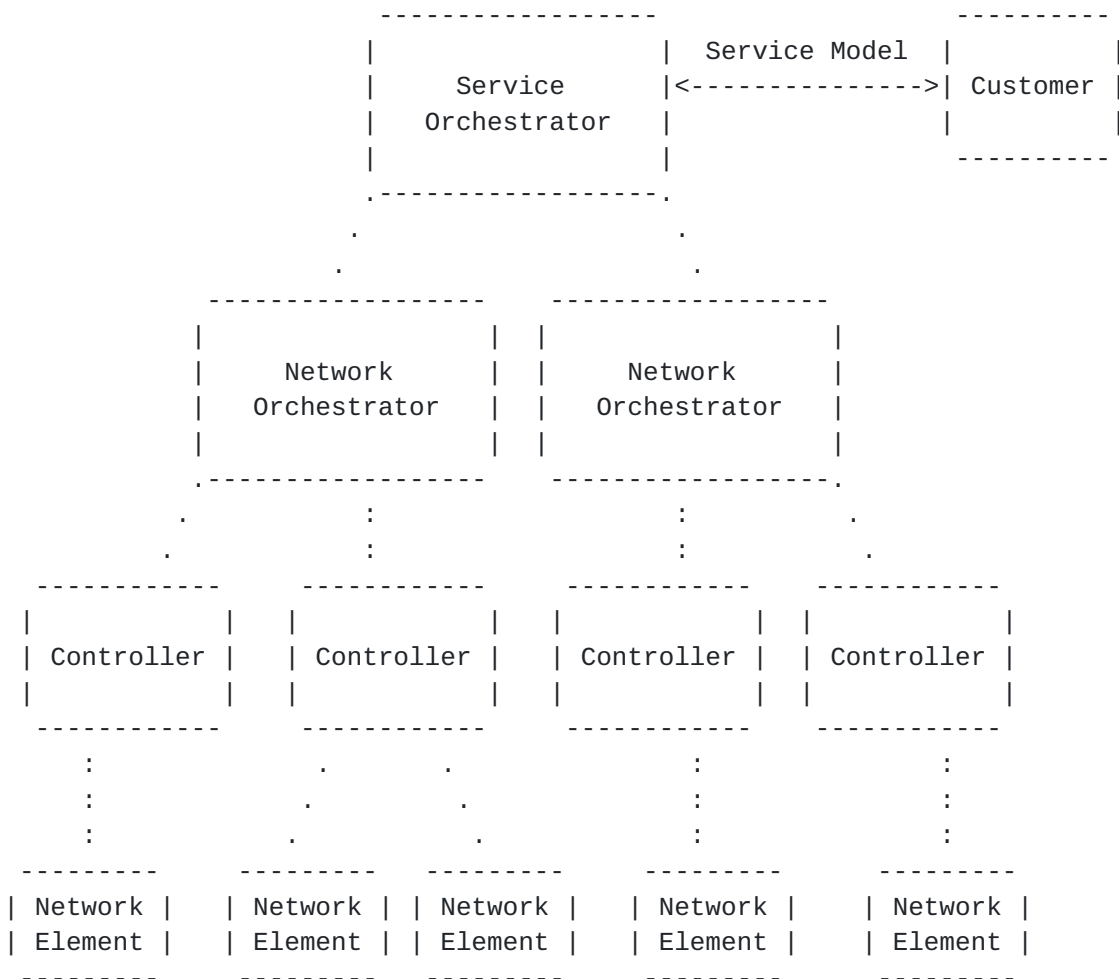


Figure 3: An SDN Architecture with a Service Orchestrator

The split between control components that exposes a "service interface" is present in many figures showing extended SDN architectures:

- o Figure 1 of [\[RFC7426\]](#) shows a separation of the "Application Plane", the "Network Services Abstraction Layer (NSAL)", and the "Control Plane". It marks the "Service Interface" as situated between the NSAL and the Control Plane.
- o [\[RFC7491\]](#) describes an interface between an "Application Service Coordinator" and an "Application-Based Network Operations Controller".

5. Possible Causes of Confusion

In discussing service models, there are several possible causes of confusion:

- o The services we are discussing are services provided by network operators to customers. This is a completely different thing to "Foo as a Service" (for example, Infrastructure as a Service (IaaS)) where a service provider offers a service at some location that is reached across a network. The confusion arises not only because of the use of the word "service", but also because network operators may also offer value-added services to their customers.
- o Network operation is completely out of scope in the discussion of service models. That means that the service model does not reveal to the customer anything about how the network operator delivers the service. The model does not expose details of technology or network resources used to provide the service. For example, in the simple case of point-to-point virtual link connectivity provided by a network tunnel (such as an MPLS pseudowire) the network operator does not expose the path through the network followed by the tunnel. Of course, this does not preclude the network operator from taking guidance from the customer (such as to avoid routing traffic through a particular country) or from disclosing specific details (such as might be revealed by a route trace), but these are not standard features of the service as described in the service model.
- o The network operator may use further data models that help to describe how the service is realized in the network. These models might be used on the interface between the Service Orchestrator and the Network Orchestrator as shown in Figure 3 and might include many of the pieces of information in the service model alongside protocol parameters and device configuration information. It is important that the Service Orchestrator should be able to map from a service model to these data models, but they are not the same things.
- o Commercial terms are generally not a good subject for standardization. It is possible that some network operators will enhance standard service models to include commercial information, but the way this is done is likely to vary widely between network operators.
- o Service Level Agreements (SLAs) have a high degree of overlap with the definition of services present in service models. Requests for specific bandwidth, for example, might be present in a service model, and agreement to deliver a service is a commitment to the

description of the service in the service model, however, SLAs typically include a number of fine-grained details about how services are allowed to vary, by how much, and how often. SLAs are also linked to commercial terms with penalties and so forth, and so are also not good topics for standardization.

6. Further Concepts

This section introduces a few further, more advanced concepts

6.1. Technology Agnostic

Service models should generally be technology agnostic. That is to say, the customer should not care how the service is provided so long as the service is delivered.

However, some technologies reach the customer site and make a definition to the type of service delivered. Such features do need to be described in the service model.

Two examples are:

- o The data passed between customer equipment and network operator equipment will be encapsulated in a specific way, and that data plane type forms part of the service.
- o Protocols that are run between customer equipment and network operator equipment (for example, Operations, Administration, and Maintenance protocols, or protocols for exchanging routing information) need to be selected and configured as part of the service description.

6.2. Relationship to Policy

Policy appears as a crucial function in many places during network orchestration. A service orchestrator will, for example, apply the network operator's policies to determine how to provide a service for a particular customer (possibly considering commercial terms). However, the policies within a service model are limited to those over which a customer has direct influence, but which are acted on by the network operator.

The policies that express desired behavior of services on occurrence of specific events are close to SLA definitions: they should only be included in the base service model where they are common to all network operators' offerings. Policies that describe who at a customer may request or modify services (that is, authorization) are close to commercial terms: they, too, should only be included in the

base service model where they are common to all network operators' offerings.

Nevertheless, policy is so important that all service models should be designed to be easily extensible to allow policy components to be added and associated with services as needed.

6.3. Operator-Specific Features

When work in Layer Three Virtual Private Network Service Model (L3SM) was started, there was some doubt as to whether network operators would be able to agree on a common description of the services that they offer to their customers because, in a competitive environment, each markets the services in a different way with different additional features. Thus, when a basic description of the core service is agreed and documented in a service model, it is important that that model should be easily extended or augmented by each network operator so that the standardized model can be used in a common way and only the operator- specific features vary from one environment to another.

6.4. Supporting Multiple Services

Network operators will, in general, offer many different services to their customers. Each would normally be the subject of a separate service model.

It is an implementation and deployment choice whether all service models are processed by a single Service Orchestrator that can coordinate between the different services, or whether each service model is handled by a specialized Service Orchestrator able to provide tuned behavior for a specific service.

7. Security Considerations

TBD

8. Manageability Considerations

TBD

9. IANA Considerations

This document makes no requests for IANA action

10. Acknowledgements

Thanks to Daniel King for review comments.

11. References

11.1. Normative References

- [RFC3444] Pras, A. and J. Schoenwaelder, "On the Difference between Information Models and Data Models", [RFC 3444](#), DOI 10.17487/RFC3444, January 2003, <<http://www.rfc-editor.org/info/rfc3444>>.
- [RFC7426] Haleplidis, E., Ed., Pentikousis, K., Ed., Denazis, S., Hadi Salim, J., Meyer, D., and O. Koufopavlou, "Software-Defined Networking (SDN): Layers and Architecture Terminology", [RFC 7426](#), DOI 10.17487/RFC7426, January 2015, <<http://www.rfc-editor.org/info/rfc7426>>.

11.2. Informative References

- [RFC4741] Enns, R., Ed., "NETCONF Configuration Protocol", [RFC 4741](#), DOI 10.17487/RFC4741, December 2006, <<http://www.rfc-editor.org/info/rfc4741>>.
- [RFC6020] Bjorklund, M., Ed., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", [RFC 6020](#), DOI 10.17487/RFC6020, October 2010, <<http://www.rfc-editor.org/info/rfc6020>>.
- [RFC7223] Bjorklund, M., "A YANG Data Model for Interface Management", [RFC 7223](#), DOI 10.17487/RFC7223, May 2014, <<http://www.rfc-editor.org/info/rfc7223>>.
- [RFC7407] Bjorklund, M. and J. Schoenwaelder, "A YANG Data Model for SNMP Configuration", [RFC 7407](#), DOI 10.17487/RFC7407, December 2014, <<http://www.rfc-editor.org/info/rfc7407>>.
- [RFC7491] King, D. and A. Farrel, "A PCE-Based Architecture for Application-Based Network Operations", [RFC 7491](#), DOI 10.17487/RFC7491, March 2015, <<http://www.rfc-editor.org/info/rfc7491>>.

Authors' Addresses

Qin Wu
Huawei Technologies

Email: bill.wu@huawei.com

Will Liu
Huawei Technologies

Email: liushucheng@huawei.com

Adrian Farrel
Juniper Networks

Email: adrian@olddog.co.uk