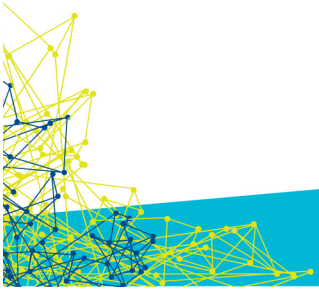
OPEN NETWORKING
FOUNDATION

# SDN Architecture Overview

Version 1.1

November, 2014

ONF TR-504

ONF Document Type: TR
ONF Document Name: TR_SDN ARCH Overview 1.1 11112014

## Disclaimer



Open Networking Foundation
2275 E. Bayshore Road, Suite 103, Palo Alto, CA 94303
www.opennetworking.org

# Table of Contents



# List of Figures



# List of Tables



# 1 Introduction

This document provides a brief overview of the main scope, requirements and principles of the SDN architecture as seen by ONF. It presents a high-level view of the major architectural components, functions and interfaces; and summarizes the more detailed definitions and examples contained in the SDN Architecture document [SDN ARCH].

The aim of SDN is to provide open interfaces that enable the development of software that can control the connectivity provided by a set of network resources and the flow of network traffic though them, along with possible inspection and modification of traffic that may be performed in the network.

In the SDN architecture, the control and data planes are decoupled, network intelligence and state are logically centralized, and the underlying network infrastructure is abstracted from the applications. As a result, enterprises and carriers gain unprecedented programmability, automation, and network control, enabling them to build highly scalable, flexible networks that readily adapt to changing business needs.

# 2 SDN Architecture Requirements and Scope

The SDN Architecture addresses the following requirements:

- Support for interoperability based upon open SDN controller plane interfaces
- Independence from the characteristics of SDN controller distribution
- Scalability and support for recursion to encompass all feasible SDN controller architectures
- Applicability to, and simplified and unified configuration of, a wide range of data plane resources
- Policy and security boundaries related to information sharing and trust
- Support for management interfaces, across which resources and policy may be established, as well as other more traditional management functions
- Co-existence with existing business and operations support systems, and other administrative or control technology domains

To avoid over-specification, the architecture only describes functions that are required, but does not preclude additional functions, allowing a wide range of scenarios and compliant implementations.

The SDN architecture specifies, at a high level, the reference points and open interfaces that enable the development of software that can control the connectivity provided by a set of network resources and the flow of network traffic though them, along with possible inspection and modification of traffic that may be performed in the network. Virtualization permits abstract views of network resources. These resources can be tailored to a particular client or application, and can be interrogated and manipulated by those clients or applications.

A logically centralized, scalable control plane manages a wide range of data plane resources of possibly diverse data plane technologies. The SDN Architecture allows modelling of forwarding and processing behavior, supporting a variety of media and connectivity types; where processing includes any compute, storage or network functions. Network functions and services may cover all OSI Layers (L0-7), and may be either physical or virtual.

Further, architectural considerations and specifications include co-existence with non-SDN environments and migration issues. If SDN is to be successful, it must be deployable within the context of largely pre-existing multi-player environments, comprising many organizations or businesses, with the consequent need for policy and security boundaries of information sharing and trust. Real-world constraints include the need to co-exist with existing business and operations support systems, and other administrative or control technology domains.

# 3 SDN Architecture Principles

The basic principles of the SDN Architecture are three-fold:

1. Decoupling of controller and data planes

   This principle calls for separable controller and data planes. However, it is understood that control must necessarily be exercised within data plane systems. The controller plane interface (CPI) between SDN controller and network element is defined in such a way that the SDN controller can delegate significant functionality to network elements (NEs), while remaining awareness of NE state. Clause 4.3.4 of [SDN ARCH] lists criteria for deciding what to delegate and what to retain in the SDN controller itself.

2. Logically centralized control

   In comparison to local control, a centralized controller has a broader perspective of the resources under its control, and can potentially make better decisions about how to deploy them. Scalability is improved both by decoupling and centralizing control, allowing for increasingly global but less detailed views of network resources. SDN controllers may be recursively stacked for scaling or trust boundary reasons, a topic described in [SDN ARCH] clause 5.

3. Exposure of abstract network resources and state to external applications

   Applications may exist at any level of abstraction or granularity, as described further in Section 4.

# 4 High-level SDN Architecture

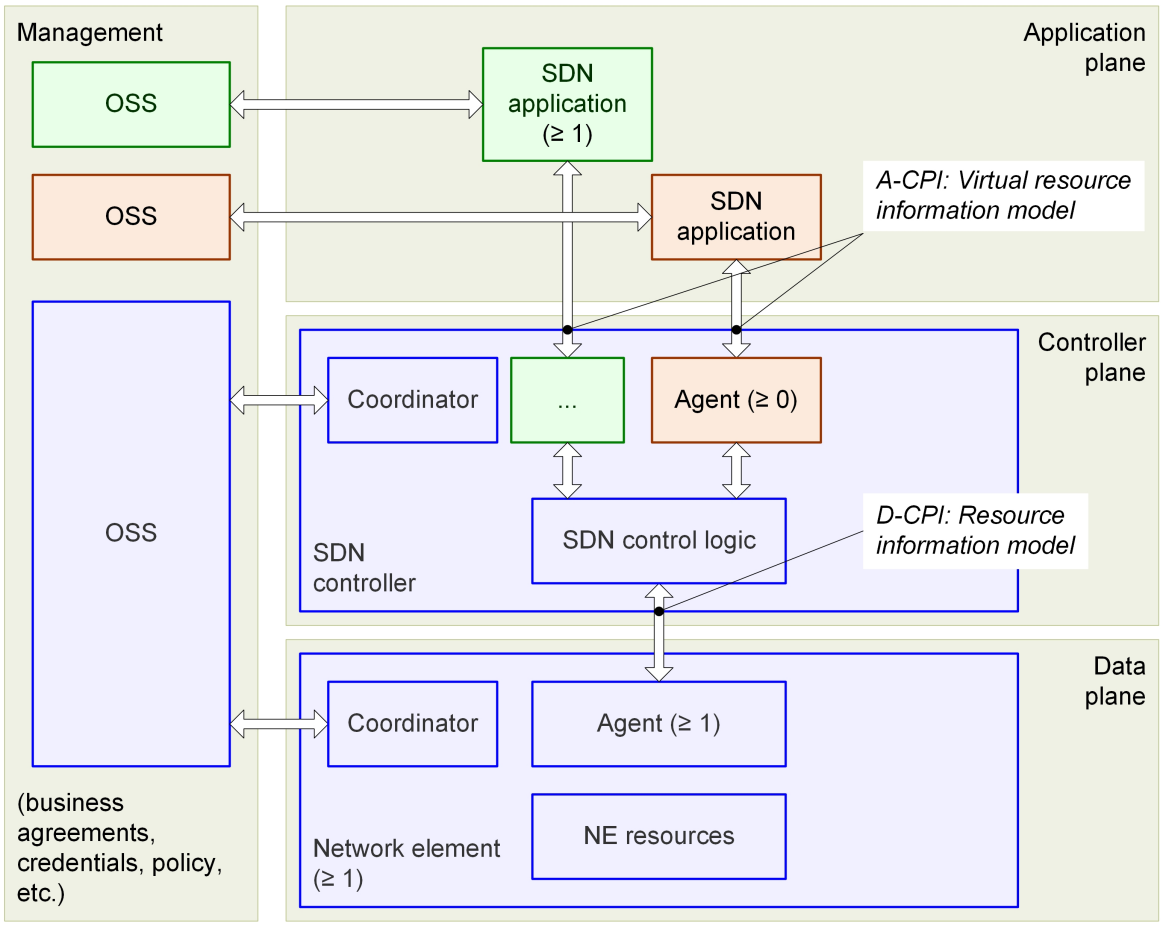



Figure 1: SDN Architecture Overview

The SDN Architecture comprises three layers:

- The **Data Plane** comprises network elements, which expose their capabilities toward the control layer (Controller Plane) via the data-controller plane interface (D-CPI).
- In the **Controller Plane**, the SDN controller translates the applications' requirements and exerts more granular control over the network elements, while providing relevant information up to the SDN applications. Services are offered to applications via the application-controller plane interface (A-CPI, often called NBI) by way of an information model instance that is derived from the underlying resources, management-installed policy, and local or externally available support functions. An SDN controller may orchestrate competing application demands for limited network resources.
- SDN applications reside in the **Application Plane**, and communicate their network requirements toward the Controller Plane via the A-CPI.

Although many traditional management functions may be bypassed by the direct application-controller plane interface (A-CPI), certain management functions are still essential. In the Data Plane, management is at least required for initially setting up the network elements and assigning resources to the respective SDN controller. In the Controller Plane, management needs to configure the SDN controller and the policies defining the scope of control given to each SDN application, and to monitor the performance of the system. In the application plane,

management typically configures the contracts and service level agreements (SLAs), which are enforced by the Controller Plane. In all planes, management configures the security associations that allow distributed functions to safely intercommunicate.

As any interface that exposes resources and state can be considered a controller interface, the distinction between application and control is a matter of perspective. The same functional interface may be viewed differently by the various stakeholders. To an SDN controller, everything further south is a data plane; everything further north is an application plane. Details of abstraction and functionality may differ, and interface protocols may differ, but the interfaces are all fundamentally alike. Just like controllers, applications may relate to other applications as peers, or as clients and servers.

The concept of hierarchically recursive application/controller layers and trust domains allows application programs to be created that may combine a number of component applications into a more comprehensive service.

The architecture shown in Figure 1 uses colors as a visual aid to emphasize trust domains. Blue is the default, and may be thought of as a network provider, while other colors, such as green and red, indicate customers, tenants, or distinct organizational or application entities within the overall Blue trust domain.

The architecture also recognizes the need for software interfaces among any number of separate business or organizational entities, and identifies functional partitions with trust and policy boundaries that facilitate the design of systems to satisfy these constraints.

The agents support the concept of sharing or virtualizing the underlying resources, for example, which network element ports are SDN-controlled (as opposed to hybrid or legacy ports), or the details of the virtual network that are exposed to the SDN applications, while isolating one customer's service from another's. In the SDN controller, different agents may expose control over the network at different levels of abstraction (latitudes) or function sets (longitudes).

The coordinators in both the network element and the SDN controller install customer/tenant-specific resources and policies received from management. Multiple agents may exist at the same time in any one network element and SDN controller, but there is only one logical management interface, and therefore only one coordinator per network element or SDN controller.

An SDN controller is expected to coordinate a number of interrelated resources, often distributed across a number of subordinate platforms, and sometimes to assure transactional integrity as part of the process. This is commonly called orchestration. An orchestrator is sometimes considered to be an SDN controller in its own right, but the reduced scope of a lower-level controller does not eliminate the need for the lower-level SDN controller to perform orchestration across its own domain of control.

# 5 Further Reading

The information in Table 5.1 points to additional detail on various architectural topics of interest.

Table 5.1: Further Reading

| Details on | Location information |
|---|---|
| Delegation of control | [SDN ARCH] clause 4.3.4 |
| Control hierarchies | [SDN ARCH] clause 5 |
| Virtualization | [SDN ARCH] clause 4.5 |
| Information Model | [SDN ARCH] clause 4.7 |
| Interworking with non-SDN environments | [SDN ARCH] clause 6.5 |
| Deployment considerations | [SDN ARCH] clause 6 |
| Management | [SDN ARCH] clause 4.6 |
| Security | [SDN ARCH] clause 6.1 |
| ONF high-level view of SDN from 2012 | [SDN WP] |

# 6 References

[SDN ARCH] SDN Architecture, Issue 1, June 2014, available at https://www.opennetworking.org/images/stories/downloads/sdn-resources/technical-reports/TR_SDN_ARCH_1.0_06062014.pdf

[SDN WP] ONF White Paper on SDN, Software-Defined Networking: The New Norm for Networks (2012), available at https://www.opennetworking.org/images/stories/downloads/sdn-resources/white-papers/wp-sdn-newnorm.pdf