



OPEN NETWORKING  
FOUNDATION

# SDN Architecture Overview

Version 1.0  
December 12, 2013



## Disclaimer

THIS SPECIFICATION IS PROVIDED "AS IS" WITH NO WARRANTIES WHATSOEVER, INCLUDING ANY WARRANTY OF MERCHANTABILITY, NONINFRINGEMENT, FITNESS FOR ANY PARTICULAR PURPOSE, OR ANY WARRANTY OTHERWISE ARISING OUT OF ANY PROPOSAL, SPECIFICATION OR SAMPLE.

Without limitation, ONF disclaims all liability, including liability for infringement of any proprietary rights, relating to use of information in this specification and to the implementation of this specification, and ONF disclaims all liability for cost of procurement of substitute goods or services, lost profits, loss of use, loss of data or any incidental, consequential, direct, indirect, or special damages, whether under contract, tort, warranty or otherwise, arising in any way out of use or reliance upon this specification or any information herein.

No license, express or implied, by estoppel or otherwise, to any Open Networking Foundation or Open Networking Foundation member intellectual property rights is granted herein.

Except that a license is hereby granted by ONF to copy and reproduce this specification for internal use only.

Contact the Open Networking Foundation at <https://www.opennetworking.org> for information on specification licensing through membership agreements.

Any marks and brands contained herein are the property of their respective owners.

## Note

This document does not claim global authority over terms used in this document and their definition. All the terms are to be considered in relation to the ONFs SDN architecture, void of any claim of applicability outside the ONFs SDN architecture context. However, for the sake of readability and brevity this document does not always prepend the string "ONF" in front of each term.

This version of the overview architecture document is intentionally on a high-level and kept simple. Future more detailed versions may and will extend on this architecture. Example extensions include but are not limited to controller federation, slicing/virtualization, hybrid networks and hybrid switch considerations, layer 4-7 considerations, OA&M, security, charging, performance, northbound interface abstraction layers and functional groups, ...

## Credits

Contributors to this document: Stuart Bailey, Deepak Bansal, Linda Dunbar, Dave Hood, Zoltán Lajos Kis, Ben Mack-Crane, Jeff Maguire, Dan Malek, David Meyer, Manuel Paul, Sibylle Schaller, Fabian Schneider, Rob Sherwood, Johann Tonsing, Tina Tsou, Eve Varma

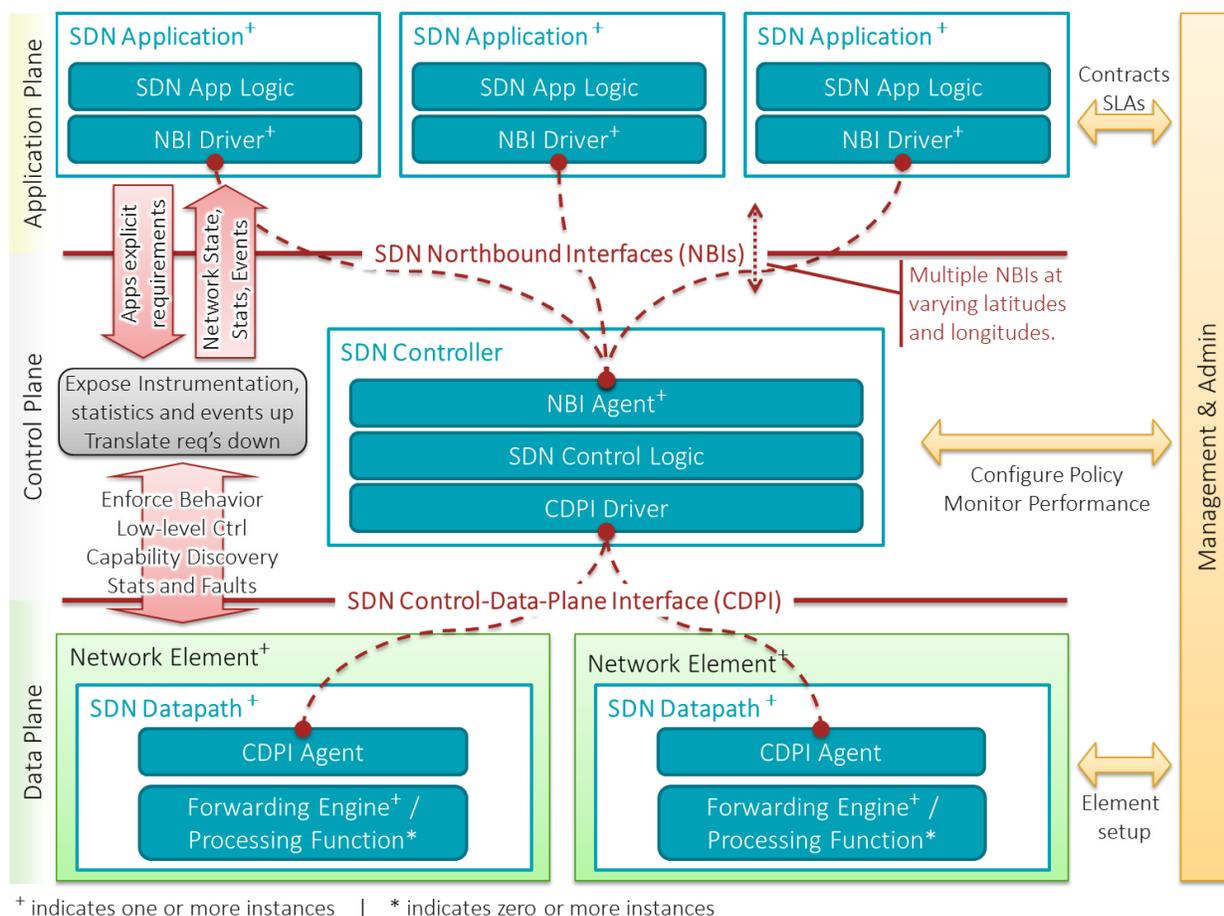


Figure 1: Overview of Software-Defined Networking Architecture

# 1 SDN Architecture Overview

This document presents the high-level view of the Software-Defined Network (SDN) architecture as seen by the ONF along with key architectural principles of SDN. Precise implementation details allowed within this SDN architecture are provided in more detailed ONF architecture documents. The aim of SDN is to provide open interfaces enabling development of software that can control the connectivity provided by a set of network resources and the flow of network traffic through them, along with possible inspection and modification of traffic that may be performed in the network. Figure 1 is a graphical representation of the architectural components and their interactions.

The original ONF white paper “Software-Defined Networking: The New Norm for Networks”<sup>1</sup> shows infrastructure, control and application layers. In this architecture we refer to them as data, control, and application planes. At bottom, the data plane is comprised of network elements, whose *SDN Datapaths* expose their capabilities through the *Control-Data-Plane Interface (CDPI) Agent*. On top, *SDN Applications* exist in the application plane, and communicate their requirements via *NorthBound Interface (NBI) Drivers*. In the middle, the *SDN Controller* translates these requirements and exerts low-level control over the SDN Datapaths, while providing relevant information up to the SDN Applications. The *Management & Admin* plane is responsible for setting up the network

<sup>1</sup> Link to white paper: <https://www.opennetworking.org/images/stories/downloads/sdn-resources/white-papers/wp-sdn-newnorm.pdf>

elements, assigning the SDN Datapaths their SDN Controller, and configuring policies defining the scope of control given to the SDN Controller or SDN Application. This SDN network architecture can coexist with a non-SDN network, especially for the purpose of a migration to a fully enabled SDN network.

## 2 Architectural components

The following list *defines* and explains the architectural components in Figure 1. ONF is continuously updating and evolving the terminology, which is tracked in the ONF Glossary project.

- **SDN Application (SDN App):** *SDN Applications are programs that explicitly, directly, and programmatically communicate their network requirements and desired network behavior to the SDN Controller via NBIs. In addition they may consume an abstracted view of the network for their internal decision making purposes.*

An SDN Application consists of one SDN Application Logic and one or more NBI Drivers. SDN Applications may themselves expose another layer of abstracted network control, thus offering one or more higher-level NBI(s) through respective NBI agent(s) (not shown in Figure 1).
- **SDN Controller:** *The SDN Controller is a logically centralized entity in charge of (i) translating the requirements from the SDN Application layer down to the SDN Datapaths and (ii) providing the SDN Applications with an abstract view of the network (which may include statistics and events).*

An SDN Controller consists of one or more NBI Agents, the SDN Control Logic, and the CDPI driver. Definition as a logically centralized entity neither prescribes nor precludes implementation details such as the federation of multiple controllers, the hierarchical connection of controllers, communication interfaces between controllers, nor virtualization or slicing of network resources.
- **SDN Datapath:** *The SDN Datapath is a logical network device, which exposes visibility and uncontended control over its advertised forwarding and data processing capabilities. The logical representation may encompass all or a subset of the physical substrate resources.*

An SDN Datapath comprises a CDPI agent and a set of one or more traffic forwarding engines and zero or more traffic processing functions. These engines and functions may include simple forwarding between the datapath's external interfaces or internal traffic processing or termination functions. One or more SDN Datapaths may be contained in a single (physical) network element—an integrated physical combination of communications resources, managed as a unit. An SDN Datapath may also be defined across multiple physical network elements. This logical definition neither prescribes nor precludes implementation details such as the logical to physical mapping, management of shared physical resources, virtualization or slicing of the SDN Datapath, interoperability with non-SDN networking, nor the data processing functionality, which can include L4-7 functions.
- **SDN Control to Data-Plane Interface (CDPI):** *The SDN CDPI is the interface defined between an SDN Controller and an SDN Datapath, which provides at least (i) programmatic control of all forwarding operations, (ii) capabilities advertisement, (iii) statistics reporting, and (iv) event notification.*

One value of SDN lies in the expectation that the CDPI is implemented in in an open, vendor-neutral and interoperable way.
- **SDN Northbound Interfaces (NBI):** *SDN NBIs are interfaces between SDN Applications and SDN Controllers and typically provide abstract network views and enable direct expression of network behavior and requirements. This may occur at any level of abstraction (latitude) and across different sets of functionality (longitude).*

One value of SDN lies in the expectation that these interfaces are implemented in in an open, vendor-neutral and interoperable way.

- **Interface Drivers & Agents:** Each interface is implemented by a driver-agent pair, the agent representing the “southern”, bottom, or infrastructure facing side and the driver representing the “northern”, top, or application facing side.
- **Management & Admin:** The Management plane covers static tasks that are better handled outside the application, control and data planes. Examples include business relationship management between provider and client, assigning resources to clients, physical equipment setup, coordinating reachability and credentials among logical and physical entities, configuring bootstrapping. Each business entity has its own management entities. Communication among management entities is beyond the scope of this SDN architecture. One goal of SDN is to subsume many management tasks known from legacy network into the CDPI.

### 3 Key Principles of this SDN Architecture

With SDN, the **applications can be network aware**, as opposed to traditional networks where the network is application aware (or rather, application ambivalent):

- Traditional (i.e. non-SDN) applications only implicitly and indirectly describe their network requirements, typically involving several human processing steps, e.g., to negotiate if there are sufficient resources and policy controls to support the application.
- Traditional networks (e.g. the current Internet and its services like web browsing, media streaming) do not offer a (dynamic) way to express the full range of user requirements, for example throughput, delay, delay variation or availability. Packet headers can encode priority requests, but network providers typically do not trust user traffic markings. Therefore some networks try to infer the users requirements on their own (e.g. through traffic analysis), which may incur additional cost and sometimes leads to misclassification. SDN offers the ability for a user to fully specify its needs in the context of a trusted relationship that can be monetized.
- Traditional (i.e. non-SDN) networks do not expose information and network state to the applications using them. Using an SDN approach, SDN Applications can monitor network state and adapt accordingly.

The **control plane is (1) logically centralized** and **(2) decoupled from the data plane**. The SDN Controller summarizes the network state for applications and translates application requirements to low-level rules.

- This does not imply that the controller is physically centralized. For performance, scalability, and/or reliability reasons, the logically centralized SDN Controller can be distributed so that several physical controller instances cooperate to control the network and serve the applications.
- Control decisions are made on an up-to-date global view of the network state, rather than distributed in isolated behavior at each network hop. With SDN, the control plane acts as a single, logically centralized network operating system in terms of both scheduling and resolving resource conflicts, as well as abstracting away low-level device details, e.g., electrical vs. optical transmission.

The SDN Controller has **complete control of the SDN Datapaths**, subject to the limit of their capabilities, and thus does not have to compete/contend with other control plane elements, which simplifies scheduling and resource allocation. This allows networks to run with complex and precise policies with greater network resource utilization and quality of service guarantees. This occurs through a well-understood common information model (e.g. as the one defined by OpenFlow).