



OPEN NETWORKING
FOUNDATION

TR-521 SDN architecture

Issue 1.1
2016

Abstract

This document specifies the architecture of software defined networking (SDN). Issue 1.1 extends SDN architecture issue 1 in light of further work in the industry and in the ONF architecture working group. It also clarifies a number of topics in light of experience with issue 1.

Original document :

Text: Microsoft Word

Figures: PowerPoint

URL https://www.opennetworking.org/images/stories/downloads/sdn-resources/technical-reports/TR-521_SDN_Architecture_issue_1.1.pdf

This excerpt:

Acrobat tools

Contents

1	Introduction	5
2	Executive summary.....	6
3	Scope	8
4	Definitions.....	10
4.1	Abstraction.....	10
4.2	Association	10
4.3	Client	10
4.4	Client context.....	10
4.5	Domain	10
4.6	Management-control continuum (MCC).....	10
4.7	Orchestration	11
4.8	Policy	11
4.9	Recursion	11
4.10	Resource	11
4.11	Server	11
4.12	Server context.....	11
4.13	Service.....	11
4.14	Service context	11
4.15	Virtualization	11
5	Concepts.....	12
5.1	Principles of SDN.....	12
5.2	Roles	14
5.3	Service and resource oriented models	16
5.4	Primitives	18
5.5	Controllers and planes.....	21
6	SDN controller	21
6.1	SDN controller as feedback node	22
6.2	Orchestration	23
6.3	Virtualization	25
6.4	Resource sharing.....	26
6.5	Delegation	27
6.6	Client context.....	27
6.7	Multiple client management-control sessions	30
6.8	Service context	33
6.9	Server context.....	34
7	Applications.....	
	Tables included to show scope, size, structure of document	

8 Putting it together: the integrated architecture	36
8.1 Interfaces.....	38
8.2 Notifications	40
8.3 Peer controllers.....	41
9 Specific perspectives on the architecture	42
9.1 Security.....	42
9.2 Migration and coexistence	43
9.3 Relationship of SDN and NFV	43
Appendix A. Discussion of definitions	44
A.1. Abstraction.....	44
A.2. Association	44
A.3. Client	45
A.4. Client context.....	45
A.5. Domain	45
A.6. Management-control continuum (MCC).....	45
A.7. Orchestration	46
A.8. Policy	46
A.9. Recursion	47
A.10. Resource	48
A.11. Server.....	48
A.12. Server context.....	49
A.13. Service.....	49
A.14. Service context	49
A.15. Virtualization	50
Appendix B. Operational considerations	51
B.1. Reliability and availability.....	51
B.2. Identifiers	52
B.3. Realization considerations.....	52
B.4. Initialization	53
B.5. Complexity	54
B.6. Persistence	55
Appendix C. Evolution from issue 1 to issue 1.1.....	56
Appendix D. Back matter.....	58
D.1. Acronyms.....	58
D.2. References	58
D.3. Contributors	59

List of Figures

Figure 1 – Basic model	6
Figure 2 – Core of the SDN architecture	7
Figure 3 – Administrator role.....	14
Figure 4 – User and provider roles.....	15
Figure 5 – Client as service orchestrator.....	17
Figure 6 – Control as feedback.....	23
Figure 7 – Client context	28
Figure 8 – Multiple users of a client context.....	30
Figure 9 – Separated SDN controller Green	31
Figure 10 – Server context.....	34
Figure 11 – SDN controller illustrating contexts	37
Figure 12 – Supplementary function example.....	39
Figure 13 – Peers as symmetric requestors and providers.....	41
Figure 14 – Issue 1 architecture.....	56

2 Executive summary

An architecture is a necessarily incomplete collection of perspectives over a set of underlying ideas. A consistent architecture reveals no contradictions when viewed from any of these perspectives. A useful architecture facilitates productive development of concepts into realities. An open architecture minimizes the difficulty of extension in previously unforeseen directions. The architecture described in this document aspires to be consistent, useful, and open.

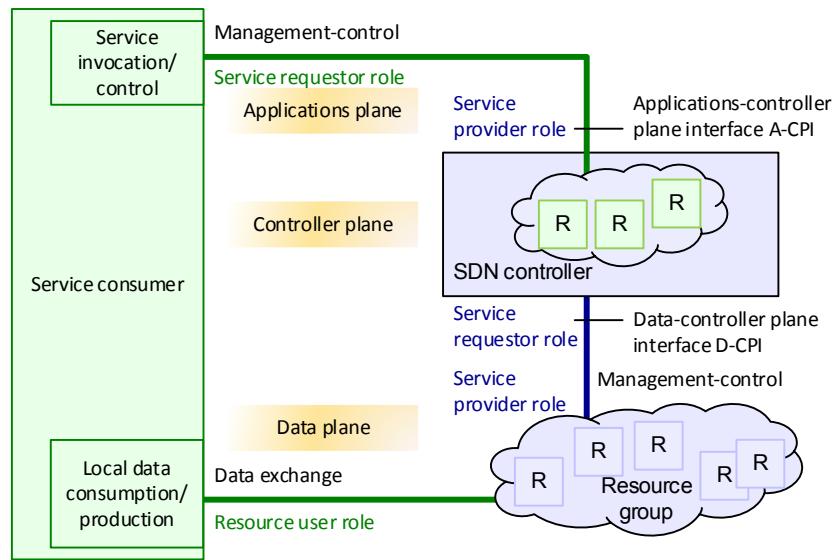


Figure 1 – Basic model

Figure 1 illustrates the basic model of SDN, that of a service consumer (client, user, customer) Green, who exchanges both data and management-control operations with some SDN server or provider, Blue. Although user data is ultimately forwarded or processed by some set of resources (R) that are owned by Blue, Green controls its service via a session contained in a management-control association. It does so by invoking actions on a set of virtual resources (R) that it perceives to be its own. Among other responsibilities, the SDN controller virtualizes and orchestrates the Green resource and service view onto its own underlying Blue resources and services. The concepts of resources and services are intentionally unbounded.

The SDN architecture extends the basic model and clarifies its implications. Key extensions include sharing resources a) among multiple clients, b) dynamically, c) in an optimum way. Other essentials of a complete architecture include management in the classical sense, both of network resources and of services.

This architecture usually portrays client and server in separate colors. The reason for this is information hiding, security and policy enforcement. The contractual relationship between client and server is relaxed in any particular deployment situation.

Illustrates use of color consistent with related PowerPoint explanation: Blue (B) through Green (G), and in other figures, to Red (R)

Character formats in colored boxes help clarify references to figure.

Tools: Word, PowerPoint

SDN controller relationships

The central entity in an SDN is the SDN controller. Figure 2 illustrates some of its key functions and interfaces.

SDN is modelled as a set of client-server relationships between SDN controllers and other entities that may themselves be SDN controllers. In its role as a server, an SDN controller may offer services to any number of clients, while an SDN controller acting as client may invoke services from any number of servers. As long as they exhibit appropriate interface behavior, the internal details of entities that are not SDN controllers are beyond the scope of the architecture.

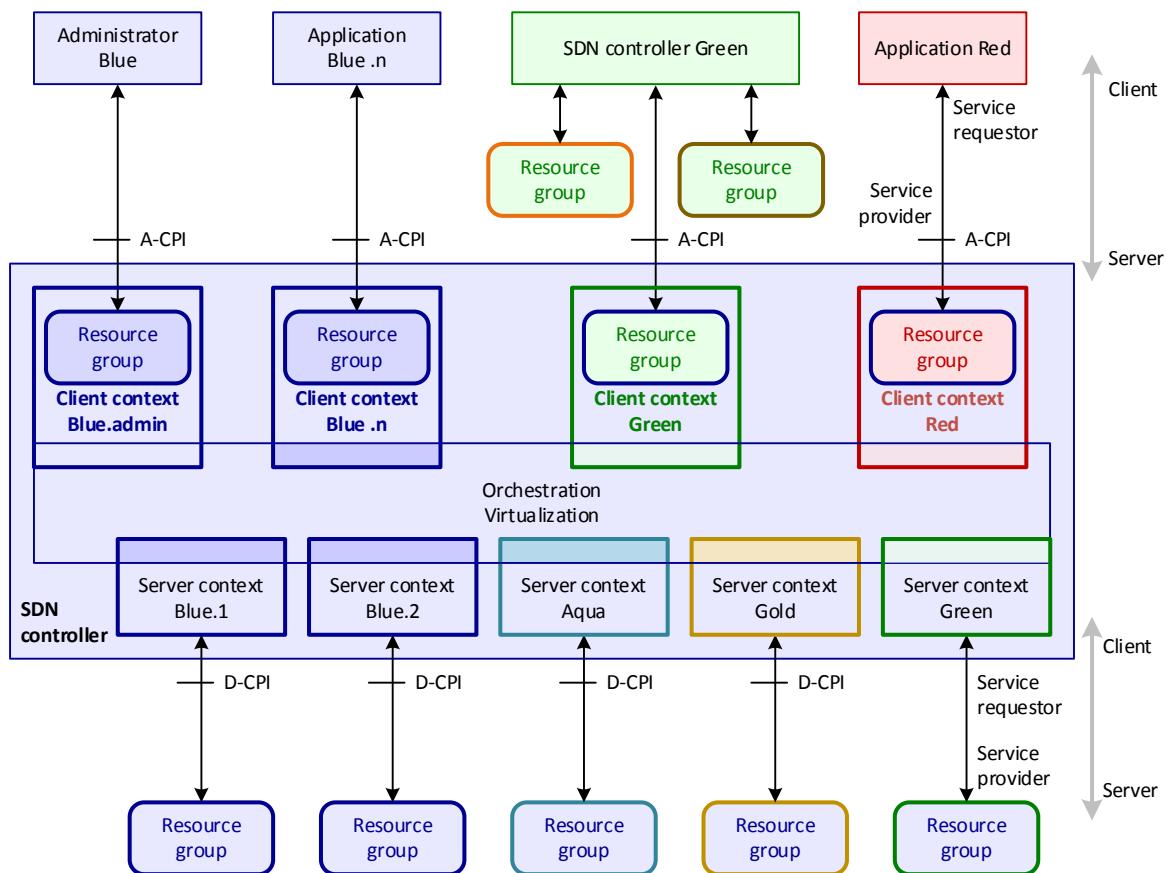


Figure 2 – Core of the SDN architecture

The architecture recognizes dual perspectives on the nature of client-server interfaces. The services perspective is particularly appropriate from a top-down or customer-provider viewpoint. The resources perspective is particularly appropriate from the bottom-up viewpoint of a resource owner, especially an internal administrator. The perspectives are complementary, but emphasize different things. The construction of views and mappings on a common underlying information model helps tie these perspectives together.

The SDN controller satisfies client requests from various sources. As the network environment changes, it is responsible for continuously updating network configuration.

Primary colors distinguish major stakeholders.

Shading, boundaries, shapes distinguish subcategories and conceptual overlaps.

Tools: Word, PowerPoint

4 Definitions

A good definition is concise and correct, avoids direct or indirect self-reference, and aligns well with at least one of the understandings in common usage. If multiple understandings exist in common usage, a definition must take a position, rather than sanctioning confusion.

Further, a good definition should be narrow enough and crisp enough to exclude invalid candidates. Fuzzy outer bounds are a major flaw in many definitions in the industry.

The implications of a definition are often important, but not obvious. A definition may therefore require informative material that is not essential to the definition itself, but that assists in its understanding. Informal descriptions of the category and qualifiers may be helpful. Examples are often useful, both of candidates that do, and also of candidates that do not, satisfy the criteria. For brevity in the main text, these discussions appear in Appendix A.

4.1 Abstraction

Definition: The representation of an entity or group of entities according to some set of criteria, while ignoring aspects that do not match the criteria.

4.2 Association

Definition: The information needed by two contracting entities as a precondition to establishing management-control sessions between each other.

4.3 Client

Definition: An entity that receives services from a server.

4.4 Client context

Definition: The conceptual component of a server that represents all information about a given client and is responsible for participation in active server-client management-control operations.

4.5 Domain

Definition: A grouping of entities according to some set of criteria.

4.6 Management-control continuum (MCC)

Definition: The principle that the function of the controller is to manage the network, and the function of the controller is to control the network, the same.

Note – In keeping with SDN terminology, an SDN controller. The session is managed by the controller.

Definition discussion text appears in an appendix to avoid congestion at the front of the document.

Indented note as an inline paragraph aligns with ITU-T style; it avoids the discontinuity that can arise from separate footnotes. Recommended here because ONF had not defined its own style preferences.

Appendix A. Discussion of definitions

A definition should be concise and precise. However, informative material is often helpful in interpreting the definition, drawing out its implications, and relating it to other concepts. Clause 4 contains the definitions used in this document; this appendix contains their discussions.

A.1. Abstraction

Definition: The representation of an entity or group of entities according to some set of criteria, while ignoring aspects that do not match the criteria.

Discussion: Even when the underlying entity is completely physical, it is always viewed through a filter of some sort. The filter necessarily loses some aspects of the original and thus necessarily provides an abstract view. Some filters are natural, for example human visual or tactile perceptions, but those of interest in SDN are software representations chosen to expose aspects of interest. Any number of filters (abstractions) can exist for a given underlying entity. Abstractions can be aggregated and/or further filtered to create additional abstractions.

It is sometimes considered that, at the bottom of a recursive hierarchy, the lowest level SDN controller controls a bare metal switch, for example through OpenFlow. It must be understood that, even in this case, the hypothetical bare metal switch necessarily exposes abstractions, namely one or more OpenFlow logical switches, upon which OpenFlow commands act.

A similar case can be made about any other management-control interface, and about even further descent into the internal structure of managed entities.

The inclusion of groups of entities in the definition is recognition that entities may be merged or combined in an abstraction.

A.2. Association

Definition: The information needed by two contracting entities as a precondition to establishing management-control sessions between each other.

Discussion: A typical association would include mutually agreed security credentials and corresponding local policy regarding acceptable session encryption characteristics.

Such an association may be established by offline negotiation and installed manually by administrators of the entities to be associated. An association may also be bootstrapped from almost nothing. An example is the willingness of a web portal to accept an anonymous session from any other entity, during the course of which the necessary identity, security and policy information can be negotiated. Upon satisfactory completion, the resulting association would allow subsequent sessions to perform a far wider scope of operations, but only between entities included by the association agreement.

Any entity that is willing to support sessions would normally offer an “almost nothing” association, sufficient to recognize and respond to sessions would then be mapped into existing supported.

To ensure consistency, the formal definition text is hyperlinked to its earlier appearance in the document.

A typical association, as described above, may also be augmented by the definition of roles and logins, with different profiles for different logins (profiles being understood as open to any interpretation).

An association allows for management-control sessions between any pair of devices that satisfy the security criteria. The security criteria may be restricted to specific devices, e.g., by including MAC address or SIM card credentials.

The difference between an association and a management-control session is that an association represents the potential for communication, while a session represents an open communications channel between two entities subsumed by the association.

A.3. Client

Definition: An entity that receives services from a server.

Discussion: Refer to clauses 5.3.2 Resource-oriented model, 5.4.2 Resources and resource groups, A.11 Server

A.4. Client context

Definition: The conceptual component of a server that represents all information about a given client and is responsible for participation in active server-client management-control operations.

Discussion: Refer to clauses 6.6 Client context and 6.7 Multiple client management-control sessions

A.5. Domain

Definition: A grouping of entities according to some set of criteria.

Discussion: Common domains of interest to SDN include

- Administrative domains, the complete set of resources owned or controlled by a given business entity.
- Geographic domains, partitioned within an administration for scaling, or across administrations for other reasons, for example the North American [telephony] Numbering Plan.
- SDN control domain, the set of resources directly controlled by a given SDN controller.
- Technology or deployment domains, for example transport, access, wireless, cloud. NFV may be a separate domain. Non-SDN domains would typically be distinguished from SDN domains.

B.6. Management-control continuum (MCC)

Definition: The principle that the functions of management and of control are largely, if not entirely, the same.

Successive screenshots from animated
 PowerPoint presentation
 Illustrates progression of explanation,
 building from organizations designated
 Blue (B) through Green (G) to Red (R)
 Drawings: PowerPoint
 This page: PowerPoint, screenshot, Photo-
 shop, Illustrator

