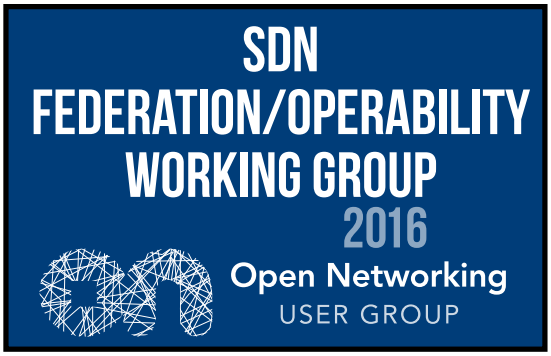
ONUG SDN Federation/Operability Orchestration

A white paper from the
ONUG SDN Federation/Operability
Working Group

May, 2016



Definition of Open Networking

Open networking is a suite of interoperable software and/or hardware that delivers choice and design options to IT business leaders, service and cloud providers. At its core, open networking is the separation or decoupling of specialized network hardware and software - all in an effort to give IT architects options in the way in which they choose to design, provision, and manage their networks. These technologies must be based on industry standards. The standards can be de-facto as adopted by a large consortium of the vendor community, open in the sense that they are community based, or defined as standards by the prevailing standards bodies. Open networking hopes to deliver on two promises:

1) Decoupling of network hardware and software which mitigates vendor lock-in and shifts network architecture structure options to users
2) Significant reduction of the total cost of ownership model, especially operational expense

SDN FEDERATION/OPERABILITY WORKING GROUP 2016
Open Networking USER GROUP

Executive Summary

The number of vendor-specific IT infrastructure controllers is growing in response to the short-term needs for network device management standardization. There is also an increasing requirement for harmonization between them. This working group looks at federating the controllers to manage north-south (and later east-west) communication across different controllers or network orchestration software.

Definition of the architecture: An Open Architecture Framework introduces an open approach in providing federated orchestration for a Software-Managed Infrastructure that is nimble, flexible and cost effective

Gap: There is a need for open application program interfaces (APIs) and programmable interfaces between controllers and orchestrators and between controllers and infrastructure layer elements.

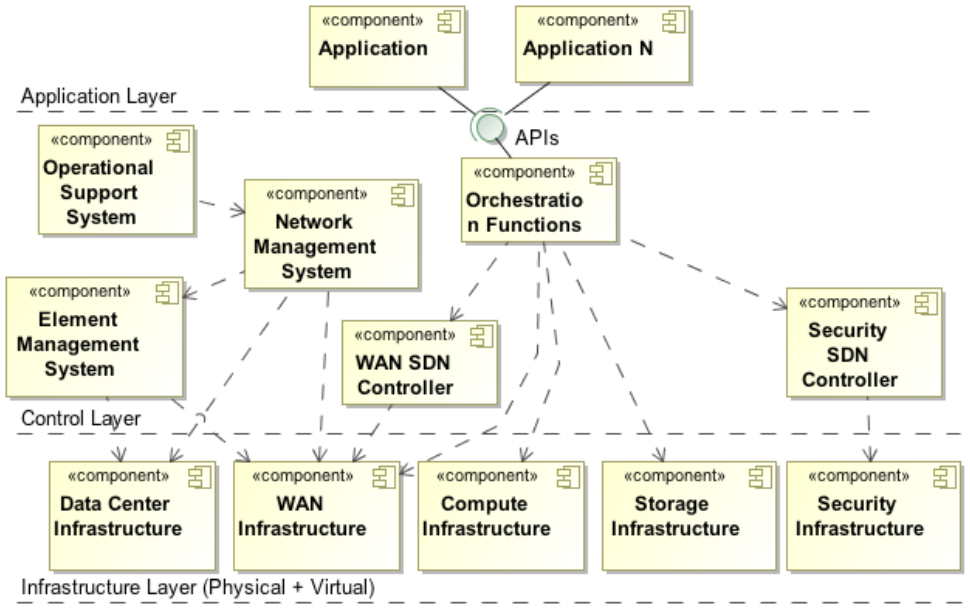


Figure 1. Present Day Landscape

Scope

This white paper is focused on north-south, rather than east-west communication.

Problem and Challenges

The identified problems and challenges include:

1) Communication

Communication between equipment is too dependent on vendor-specific command-line interface (CLI) and should be standardized as follows:

Customer/Operator/Partners ↔ Business applications	WEB interface
Business applications ↔ Orchestrator	APIs
Orchestrator ↔ Controllers	APIs
Controllers ↔ Equipment	Programmable interfaces (HTTP, SNMP, NETCONF, CAPWAP, OpenFlow…)

Open Networking User Group (ONUG)

ONUG is one of the largest industry user groups in the networking and storage sectors. Its board is made up exclusively of IT business leaders, with representation from Fidelity Investments, FedEx, Bank of America, UBS, Cigna, Pfizer, JPMorgan Chase, Citigroup, Credit Suisse, Gap, Inc., and Symantec. The ONUG mission is to guide and accelerate the adoption of open networking solutions that meet user requirements as defined through use cases, proof of concepts, hackathons, and deployment examples to ensure open networking promises are kept.

The ONUG community is led by IT business leaders and aims to drive industry dialogue to set the technology direction and agenda with vendors. To that end, ONUG hosts two major conferences per year where use cases are defined and members vote to establish a prioritized list of early adopter, open networking projects that communicate propensity to buy and budget development. The vendor community stages proof of concepts based upon ONUG Use Cases, while standards and open source organizations prioritize their initiatives and investments based upon them. ONUG also hosts user summits and smaller, regional user-focused Fireside Chat Meet-Ups through the year.

ONUG defines six architectural areas that will open the networking industry and deliver choice and design options. To enable an open networking ecosystem, a common multivendor approach is necessary for the following six architecture components:

1) Device discovery, provisioning, and asset registration for physical and virtual devices
2) Automated "no hands on keyboards" configuration and change management tools that align DevOps and NetOps
3) A common controller and control protocol for both physical and virtual devices
4) A baseline policy manager that communicates to the common controller for enforcement
5) A mechanism for sharing (communicating or consuming) network state and a unified network state database that collects, at a minimum, MAC and IP address forwarding tables automatically
6) Integrated monitoring of overlays and underlays

Note: In this federated communications schema, southbound requests come from the top and northbound notifications come from the bottom.

2. Legacy equipment

An exception to the proposed scheme could be made for legacy equipment. There is a lack of controllers that are able to communicate with legacy equipment. We acknowledge that for legacy equipment, it would be best to use CLI or another legacy methodology until legacy equipment is phased out.

Open Architecture Framework

The ONUG SDN Federation/Operability Working Group has specified that SDN Controller interoperability be performed by orchestrators within the Control Layer. As Figure 2 illustrates, the architecture is viewed as a three-layer model.

- Applications reside at the top most level, which also represents the layer exposed to customers and/or users.
- The Control Layer represents the scope of this white paper and represents the control logic necessary to orchestration services required by applications in the Infrastructure Layer.
- The Infrastructure Layer represents the physical and/or virtual resources required to instantiate services such that a customer or end-user may use the application.

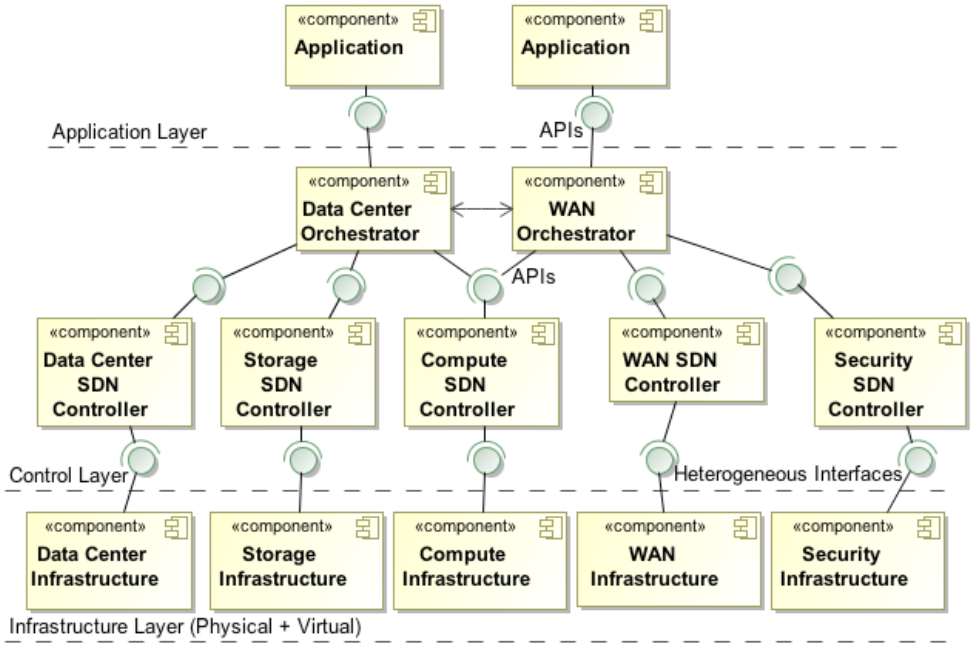


Figure 2. SDN Federation and Operability Orchestration Architectural Framework

Application Layer

The Application Layer includes the applications and workloads required to meet business needs. Users, whether internal to a business organization or as an end customer to the organization, access applications and execute workloads to perform various business tasks. Applications use the services provided by the Control and Infrastructure Layers. More specifically, applications are not aware of the underlying physical and/or logical resources, rather, they are aware of the services they need such as networking, storage,

security and/or compute. In turn, applications, generally in response to user interaction, request such services from the underlying layers. For example, deploying a new application, based on a user's need, requires application connectivity to a network; storage and backup access to disk; and authorization and authentication through application and network security policies. It also requires a computing platform to run the application, whether distributed across multiple virtual machines (VMs) or residing on a single VM. There is a significant dependency on the underlying resources in supporting the deployment of the new application. Once the application is running, questions exist on various change management activities, including dynamically increasing the available storage pool and modifying network security (based on tightened security policies)

Control Layer

The Control Layer includes the various orchestrators and controllers. Orchestrators provide end-to-end lifecycle management capabilities to the upper Application Layer. The orchestrators receive requests from the Application Layers (such as change requests or new deployment requests), and fulfill these requests through direct communication with underlying controllers. These controllers, in turn, monitor portions of the Infrastructure Layer. Orchestrators are therefore service aware, whether the service request includes network, compute, storage, security, or a combination of all functions. Orchestrators may be specific to a service, however inter-orchestration capabilities are required to hand off requests in a distributed or domain specific model. For example, one orchestrator may handle data center service requests while another may handle WAN networking requests. Management domain boundaries may also exist, whether within a single business enterprise, or across enterprise boundaries.

The controllers provide resource abstraction to the orchestrators. An Orchestrator does not need to understand the underlying infrastructure, however the controllers provide this mapping from a service view to a resource view. If an orchestrator receives a request to deploy a workload on a VM, the controllers actually make that happen in the Infrastructure Layer. The orchestrator has the knowledge of which controllers to make the request of, while the controllers have the knowledge of which resources to make the requests to. For example, an application request may be received by an orchestrator to deploy the application on a new VM, with certain compute, storage and networking attributes. The orchestrator parses the request into how it gets realized via the controllers it communicates with, across the entire visible environment. One controller may be called to provision a new VM in Data Center X, while another may be called to provision available storage capacity in Data Center Y. Yet another might be called to establish a secure network across the two Data Centers to ensure the new VM has storage resources. Controllers handle infrastructure demands, in response to requested services.

Infrastructure Layer

The Infrastructure Layer includes the physical and/or virtual network, compute and storae resources, including routers, switches, bare-metal servers, hypervisors, virtual machines, storage pools and firewalls, to name a few The Infrastructure Layer resources are described in the Common Management Tools Across Network, Storage and Compute Product/RFI Requirements ONUG white paper, under the General Reference Architecture section. Each resource in the Infrastructure Layer includes an application environment, operating environment and physical or virtual chassis. The controllers communicate with resources at this layer using an array of heterogeneous interfaces, including open standard and vendor proprietary interfaces.

Architectural Detailed View

Figure 3 illustrates a deeper view into the Control and Infrastructure Layers, where interfaces and functions are highlighted.

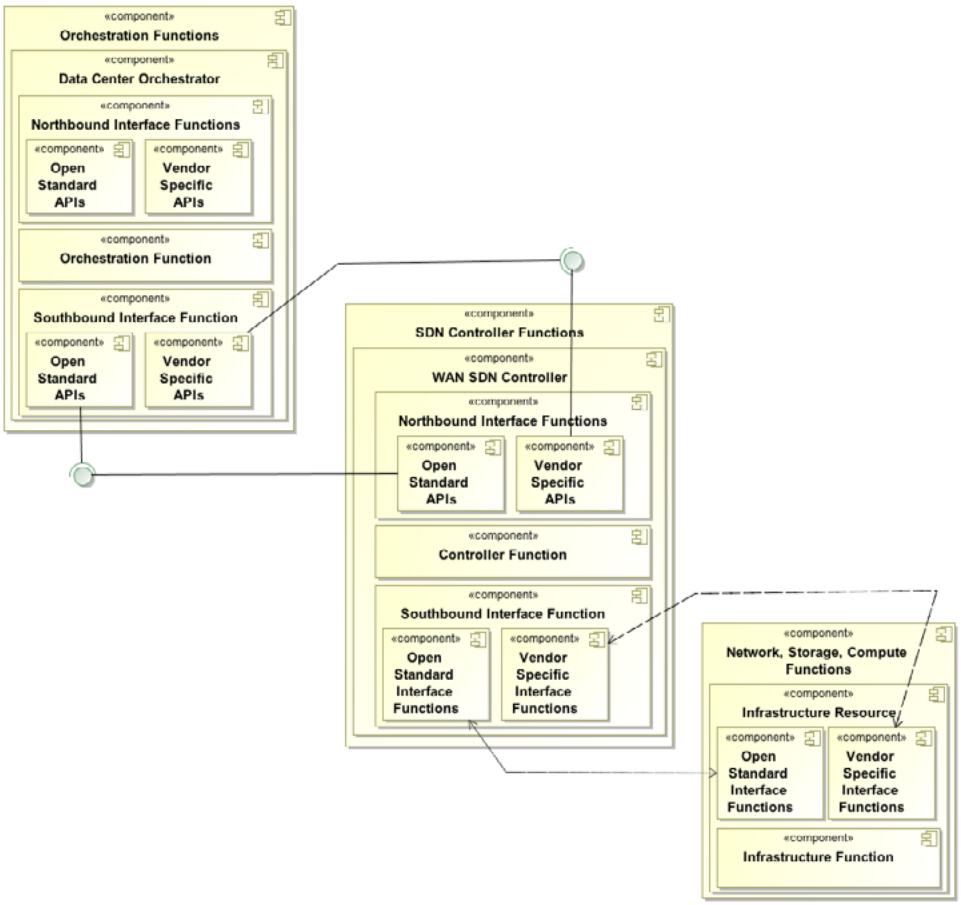


Figure 3. SDN Federation and Operability Orchestration Architectural Framework Detailed

From a top-down perspective, orchestration functions may include any number of orchestrator types, such as a Data Center Orchestrator, WAN Orchestrator, etc. Each orchestrator instance has a northbound interface function to serve application requests in the Application Layer. This interface includes open standard APIs with vendor API extensions. The southbound interface function serves as the interface to the specified controllers. This interface is structured like the northbound interface. Sitting between the northbound and southbound interface functions is the orchestration function itself, performing all the business logic of the orchestration capabilities.

Controller functions may include any number of controller types, such as WAN Controllers, compute controllers, and others. As illustrated for the orchestrator functions, the northbound interface includes open standard APIs with vendor API extensions to serve the requests from the orchestrators. The southbound interface function serves as the interface into resources in the Infrastructure Layer and includes both Open Standard Interface functions (e.g., NETCONF, SNMP, OpenFlow, etc.) and vendor specific interface functions (e.g., CLI, proprietary messaging, etc.). Sitting between the northbound and southbound interface functions is the controller function, performing all business logic of the controller capabilities.

Physical and/or virtual network, storage, and compute resources realize infrastructure functions. Each resource includes the core network, storage and/or compute function(s) along with open standard or vendor specific interface functions as indicated for the controller southbound interface function. Therefore, each resource includes both control and data plane functionality, where most control plan logic resides in the controller.

Recommendations

The ONUG SDN Federation/Operability Working Group recommends using open standard API and programmable interfaces. Vendor specific API and programmable interfaces should be the exceptions.

- Application layer - control layer: APIs
- Within the control layer: APIs
- Control layer – infrastructure layer: Programmable Interfaces

Conclusion

The core of SDN federation is application centric, cross-vendor resource management, spanning compute, storage, and networking. APIs should be open and equally adopted between vendors, with vendor specific APIs for specialized functionality being separated out.

Use Cases

The following sections define several Use Cases for SDN Federation/Operability. The use cases are identified in Figure 4.

Figure 4. SDN Federation and Operability Orchestration Use Cases

Failover Application/Workload across Data Centers

Field	Description
Use Case Number	1
Use Case Name	Application resiliency between data centers
Description	In this use case, an application is running in Data Center A and is in a dormant state in Data Center B. If the application starts misbehaving or the application stops running, the orchestrator will reconfigure the environment to isolate the application in Data Center A and activate the application in Data Center B. Benefits include: • Automatic failover: In the cases of high-cost and complex high-availability (HA) architecture, this could help to contain the cost and simplify the architecture while preserving the similar benefits provided by HA. • Manual failover: Once the business makes the decision to invoke a BCP condition, it will take milliseconds to execute.
Scenarios	1. Automatic failover: Application failure is detected and the orchestrator is notified. 2. Manual failover: It is left to business processes how the manual failover is invoked (this could also be used for a maintenance window).
Actors	Data Center & Orchestrator
Pre-Conditions	1. There is a process that detects that the application is failing and notifies appropriate parties. A decision has been made either to automate the failover or to have a manual intervention. 2. The orchestrator has all information to reconfigure the environment.
Process Steps	**Automatic failover**: Prepare redundant application servers in advance. **Manual failover:** Prepare instruction and manual configuration of the application server.
Post-Conditions	No more application traffic outage.
Alternative Paths	The backup data center becomes the production data center.
Assumptions	Orchestrators are connected and communicating to controllers, while controllers have full authority to act upon their supporting devices on behalf of orchestrator directives.

Provision Application/Workload Upgrade across Data Center and/or WAN

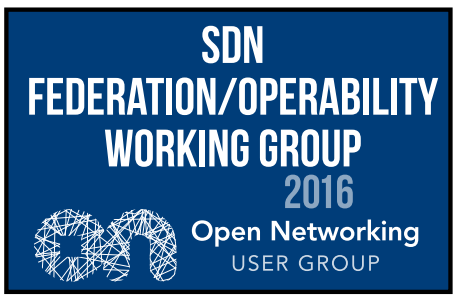
Field	Description
Use Case Number	2
Use Case Name	Provision Application/Workload Upgrade across Data Center and/or WAN
Description	In this use case, a new application is provisioned or a service pack is pushed to an application that will bring new services. The network needs to be reconfigured to prioritize the traffic of that application based on specific parameters. The orchestrator will direct the different controllers to reconfigure the network components in the field (routers, firewalls, switches, access points, WAN optimization, etc.). Benefits are that it will take seconds to reconfigure the network to accommodate the needs of new services.
Actors	Data Center & Orchestrator & WAN
Pre-Conditions	Network parameters of the applications are known.
Process Steps	All the monitoring tools are configured to push the new template.
Post-Conditions	The traffic of the application has the correct prioritization across throughout the enterprise.
Alternative Paths	Manual configuration or delay the rollout of the application until the problem is corrected.
Assumptions	Orchestrators are connected and communicating to controllers, while controllers have full authority to act upon their supporting devices on behalf of orchestrator directives.

Distribute Application/Workload across Data Centers

Field	Description
Use Case Number	3
Use Case Name	Distribute Application/Workload across Data Centers
Description	In this use case, an application spans between 2 data centers (public, private, or both). We envision that the orchestrator would configure the different equipment to provision virtual compute servers, databases / storage systems, network equipment and bandwidth between data centers. Benefits are the application is deployed quickly with minimal human intervention. The application could be redeployed exactly the same way between two other data centers.
Scenarios	1. The compute is in one data center & the storage is in another data center. 2. Application resiliency between the data centers. 3. Duplicate the application to 2 other data centers (for example for load balancing or geographic presence to user based).
Actors	Data Center & Orchestrator & WAN
Pre-Conditions	There is enough spare capacity in the data center and the overall infrastructure to accommodate the need of that new application. Network resiliency between the data centers.
Process Steps	Prepare necessary capacity (servers, network, etc.) for redundancy and provisioning. Confirm that the monitoring is in place. Generate auto-tests.
Post-Conditions	The application is fully operational between two data centers.
Alternative Paths	Manual configuration or delay the rollout of the application until the problem is corrected.
Assumptions	Orchestrators are connected and communicating to controllers, while controllers have full authority to act upon their supporting devices on behalf of orchestrator directives.

Migrate Application/Workload Dependence Map from Brownfield to Future State

Field	Description
Use Case Number	4
Use Case Name	Migrate Application/Workload Dependency Map from Brownfield to Future State
Description	In this use case, there is an application that is on legacy equipment (bare metal servers, switches, load balancers, IDS/IPS, etc.). We want to virtualize the brownfield infrastructure as-is and keep the same dependencies. There are multiple controllers that are orchestrated to configure their respective devices/components to establish a workload dependency map. The dependency map may be wide area communications or WAN devices, firewalls, IPS, switches, routers etc., that is all hardware devices and/or software components that an application depends upon to deliver its service at a high experience level. This use case seeks to automate workload dependency map configuration to decrease the time of IT delivery. Based upon ONUG community data, the time to install a physical server is approximately 200 days starting at the time the order is sent to procurement and 42 days to configure a workload dependency map after the physical service has been installed.
Actors	Data Center, WAN et al Orchestrators
Pre-Conditions	1. The workload and supporting services must be identified and provisioned. 2. The network controllers and resources participating in the service chain must be identified and deployed and activated.
Process Steps	Configure the orchestrator to configure appliances. Follow the orchestrator configuration guide and do the service chaining.
Post-Conditions	The orchestrator has necessary information for dependency map authorization, creation, and configuration.
Alternative Paths	Manual configuration or write your own program to enable service chaining.
Assumptions	Orchestrators are connected and communicating to controllers, while controllers have full authority to act upon their supporting devices on behalf of orchestrator directives. The full dependency map is known to orchestrators.



ONUG SDN Federation/Operability Working Group

Maxime Bugat, Chair

Snehal Patel — Gap Inc.

Nick Lippis — Open Networking User Group

Aziz Abdul — Cisco

Stefan Dietrich — glue Networks

Mike Haugh — glue Networks

Jenny Oshima — NEC