OPEN DATA CENTER ALLIANCE[SM] MASTER USAGE MODEL: SOFTWARE-DEFINED NETWORKING REV. 1.0

TABLE OF CONTENTS

CONTRIBUTORS

The following individuals from the Software-Defined Networking/Infrastructure Work Group contributed to the contents of this document:

Vishy Ganti, Verizon

Vince Lubsey, Virtustream

Mrigank Shekhar, Intel

Chris Swan, CohesiveFT

LEGAL NOTICE

This "**Open Data Center Alliance^{SM} Master Usage Model: Software-Defined Networking**" document is proprietary to the Open Data Center Alliance (the "**Alliance**") and/or its successors and assigns.

NOTICE TO USERS WHO ARE NOT OPEN DATA CENTER ALLIANCE PARTICIPANTS: Non-Alliance Participants are only granted the right to review, and make reference to or cite this document. Any such references or citations to this document must give the Alliance full attribution and must acknowledge the Alliance's copyright in this document. The proper copyright notice is as follows: "**© 2013 Open Data Center Alliance, Inc. ALL RIGHTS RESERVED.**" Such users are not permitted to revise, alter, modify, make any derivatives of, or otherwise amend this document in any way without the prior express written permission of the Alliance.

NOTICE TO USERS WHO ARE OPEN DATA CENTER ALLIANCE PARTICIPANTS: Use of this document by Alliance Participants is subject to the Alliance's bylaws and its other policies and procedures.

NOTICE TO USERS GENERALLY: Users of this document should not reference any initial or recommended methodology, metric, requirements, criteria, or other content that may be contained in this document or in any other document distributed by the Alliance ("**Initial Models**") in any way that implies the user and/or its products or services are in compliance with, or have undergone any testing or certification to demonstrate compliance with, any of these Initial Models.

The contents of this document are intended for informational purposes only. Any proposals, recommendations or other content contained in this document, including, without limitation, the scope or content of any methodology, metric, requirements, or other criteria disclosed in this document (collectively, "**Criteria**"), does not constitute an endorsement or recommendation by Alliance of such Criteria and does not mean that the Alliance will in the future develop any certification or compliance or testing programs to verify any future implementation or compliance with any of the Criteria.

LEGAL DISCLAIMER: THIS DOCUMENT AND THE INFORMATION CONTAINED HEREIN IS PROVIDED ON AN "**AS IS**" BASIS. TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, THE ALLIANCE (ALONG WITH THE CONTRIBUTORS TO THIS DOCUMENT) HEREBY DISCLAIM ALL REPRESENTATIONS, WARRANTIES AND/OR COVENANTS, EITHER EXPRESS OR IMPLIED, STATUTORY OR AT COMMON LAW, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE, VALIDITY, AND/OR NONINFRINGEMENT. THE INFORMATION CONTAINED IN THIS DOCUMENT IS FOR INFORMATIONAL PURPOSES ONLY AND THE ALLIANCE MAKES NO REPRESENTATIONS, WARRANTIES AND/OR COVENANTS AS TO THE RESULTS THAT MAY BE OBTAINED FROM THE USE OF, OR RELIANCE ON, ANY INFORMATION SET FORTH IN THIS DOCUMENT, OR AS TO THE ACCURACY OR RELIABILITY OF SUCH INFORMATION. EXCEPT AS OTHERWISE EXPRESSLY SET FORTH HEREIN, NOTHING CONTAINED IN THIS DOCUMENT SHALL BE DEEMED AS GRANTING YOU ANY KIND OF LICENSE IN THE DOCUMENT, OR ANY OF ITS CONTENTS, EITHER EXPRESSLY OR IMPLIEDLY, OR TO ANY INTELLECTUAL PROPERTY OWNED OR CONTROLLED BY THE ALLIANCE, INCLUDING, WITHOUT LIMITATION, ANY TRADEMARKS OF THE ALLIANCE.

TRADEMARKS: OPEN CENTER DATA ALLIANCE^{SM}, ODCA^{SM}, and the OPEN DATA CENTER ALLIANCE logo® are trade names, trademarks, and/or service marks (collectively "**Marks**") owned by Open Data Center Alliance, Inc. and all rights are reserved therein. Unauthorized use is strictly prohibited. This document does not grant any user of this document any rights to use any of the ODCA's Marks. All other service marks, trademarks and trade names reference herein are those of their respective owners.

OPEN DATA CENTER ALLIANCE℠ MASTER USAGE MODEL: SOFTWARE-DEFINED NETWORKING REV. 1.0

EXECUTIVE SUMMARY

Software-defined networking (SDN) is an emerging set of technologies within network management that promises to solve many of the limitations imposed by current networking technology on rapidly evolving cloud-computing technologies.

Traditional networking architecture, which is tightly coupled with network interface identity, has been based on the principles of autonomous systems. As a proven networking model, it is simple (with plug-and-play connectivity at lower levels), resilient, and scalable. However, this architecture makes certain vital needs more difficult to achieve and administer in a straightforward, consistent way, including security, network segmentation into logical groups, access control, and quality of service (QoS) across network infrastructure. Also, since data packets can take independent routes, applications such as audio and video do not perform well without modifications to control flow and performing such complex procedures as packet sequencing. The key problem is that the network topology and identity of devices are simultaneously defined by the devices' network addresses. Networking software has evolved over several decades in a decentralized, monolithic manner, with the code embedded into network devices. Not all network functionality, however, need be distributed, and certain services can be more easily specified and managed if control at the autonomous network level is centralized. Migrating the level and locus of control across a network lets network operators specify network services without having to create specifications for each physical network interface and component that can be present in a network infrastructure. The emerging principles of SDN address this need.

Indications point toward rapid SDN adoption in cloud data centers. SDN technology also offers promising opportunities for high-throughput, high-volume applications, such as big data deployments in the financial and scientific sectors. To improve evaluations and decision making, IT departments and cloud subscribers will require standard features and defined metrics. The Open Data Center Alliance (ODCA) recognizes the need for the adoption of SDN in infrastructure as a service (IaaS) and management solutions that incorporate standard mechanisms to enable better management of network services. This usage model specifies actions and processes to advance development of practical solutions that seek to lower management complexity and costs, especially in heterogeneous, multi-vendor environments.

This document serves a variety of audiences. Business decision makers seeking specific solutions and enterprise IT groups involved in planning, operations, and procurement will find this document useful. Solution providers and technology vendors should benefit from its content to better understand customer needs and tailor service and product offerings. Standards organizations should find information that helps define open standards relevant to end users.

FRAMING THE CHALLENGE

SDN offers the following benefits:

- Separation of the physical network from the logical network
- Simplified and more centrally controlled network management, automation, and (potentially) programmability
- Greater virtual machine (VM) mobility across the data center
- Enhanced security capabilities
- Better traffic segmentation
- Dynamic network partitioning to handle multi-tenant networks

These capabilities of SDN promise better service, agility, and new capabilities.

SDN is a nascent field and new use cases, operational models, common practices, tools, and services are constantly emerging and will rapidly evolve. Initially, vendors and early adopter cloud providers will differentiate themselves by offering new capabilities and services, but there will be little interoperability and standardization. Currently, standards have only been established for the “Southbound APIs,” specifying the interface between the control plane and network devices. The “Northbound APIs,” which form the interface between the management and services planes on one side and the control plane on the other, are based entirely on software. It will take longer before standardization for these APIs can be established.

Consistent management is essential for any network environment. Ensuring interoperability of SDN solutions will let vendors easily develop interoperable management solutions that lower management complexity and cost, especially in a heterogeneous, multi-vendor environment. By supporting certain SDN management standards, for example, such as the Open Networking Foundation’s OpenFlow™ protocol, SDN deployments could be managed in a centralized, vendor-independent manner. To maximize interoperability, there should be a common command set that is supported by all SDN implementations, including the following:

- Central control of VM network parameters
- Definitions of QoS and security levels as a business policy
- Network control directives

With a standards-based SDN, the entire virtualized environment—including the network—could then be managed from a single management console.

Additional support for differentiating features should be monitored and reviewed on a regular basis. If features become adopted as standard practices they should be incorporated into the common command set.

This usage models extends the compute infrastructure as a service (CIaaS) master usage model, which serves as the framework for CIaaS services to be evaluated, acquired, and disposed of by enterprises in a way that reflects the ODCA member firms’ vision of a robust and vibrant market by the end of 2014.

TAXONOMY

Table 1 lists the standard terms and definitions used in this document.

Table 1. Terms and definitions.

Actor	Description
Layer 2 Multi-Path (L2MP)	Improves network utilization by eliminating the use of the Spanning Tree Protocol, which closes some ports to avoid network loops. L2MP implementations optimize traffic routing and enable East-West traffic. TRILL (RFC 5556), from IETF, and SPB (802.1aQ), from IEEE, are protocols that facilitate for L2MP networking.
Multi-Chassis Link Aggregation Group (MC-LAG)	Aggregates ports on separate network devices. MC-LAG increases bandwidth and provides redundancy at the node level, instead of just at the link level as does simple Link Aggregation. MC-LAG is not a standard and implementations are vendor proprietary.
Network Functions Virtualization (NFV)	A complement to SDN, NFV allows components of the network infrastructure to perform different functions at different times, based on the software being run. In contrast to dedicated hardware performing a specific function, NFV leverages standard IT virtualization technology to consolidate many network equipment types onto industry standard high-volume servers, switches and storage. These can be located in fixed and mobile network infrastructures. NFV does not require SDN (nor does SDN require NFV), but the two technologies can work together within a network production environment to lower cost, improve efficiency, and enhance agility.
Network Virtualization	Refers to technologies that create tunnels or overlays of a network infrastructure that make its logical topology differ from its physical topology. Network virtualization can be used to join geographically disjoint data centers and make them appear as a single logical entity. Conversely, network virtualization can be used to limit access to certain nodes, paths, or services within a physical network, with managing multi-tenant infrastructure as a service in a single data center as a primary usage model. Network virtualization can be implemented with or without SDN.
Network Virtualization using Generic Routing Encapsulation (NVGRE)	NVGRE—Network Virtualization using Generic Routing Encapsulation [RFC 2784, RFC 2890]—was designed to contend with the scalability problems associated with hyper-scale virtualization. This proposed IETF standard, co-authored by Microsoft, Emulex, and other organizations, addresses many of the challenges present by the growth of cloud computing.
Software-Defined Networking (SDN)	SDN is an emerging architecture that is dynamic, manageable, cost-effective, and adaptable, making it ideal for the high-bandwidth, dynamic nature of today's applications. This architecture decouples the network control and forwarding functions, enabling the network control to become directly programmable and the underlying infrastructure to be abstracted for applications and network services. The protocol or messaging scheme that forms the communication between the controller and switches is defined separately.
Stateless Transport Tunneling (STT)	Developed by Nicira, STT is a network virtualization specification and an encapsulation protocol for creating L2Overlay networks. STT utilizes the TCP Segmentation Offload Capabilities (TSO) available on most modern network interface cards.
Virtual Extensible LAN (VXLAN)	VmWare, in partnership with networking and silicon vendors, created the network virtualization technology VXLAN to overcome many of the limitations of VLAN technologies. VXLAN enables the creation of large cloud computing environments and provides a framework for overlaying virtualized Layer 2 networks over Layer 3 networks.

SOFTWARE-DEFINED NETWORKING

Challenges of Modern Networks

Traditional network architectures rely heavily on the network interface identity. In the OSI Model, network identity is represented at the physical layer by a hardware-based identifier, such as the MAC addresses in Ethernet, ATM, or FDDI network technologies. The Internet serves as a network of networks, and the interoperability of the constituent networks is enabled by the TCP/IP protocol suite. TCP/IP couples the physical address with a software-based addressing scheme that helps create a unified logical network out of the disparate physical networks. By promoting the autonomy of the constituent networks over centralized control as a way of achieving resiliency, the design of TCP/IP led to the rapid growth of the Internet. Because of the intelligence embedded in the individual network devices that function by progressively mapping the neighborhood topography, reliable, efficient communications between widely separated end points can be achieved.

However, this architecture has shortcomings in a number of critical areas that are related to identity, including the following:

- Security
- Logical grouping
- Access control
- QoS
- Flow control

As these features became widely adopted, follow-on IETF standards—including VLANs and VPNs—augmented those requirements that were specific to identity. These standards, however, increased the complexity of network element specifications and the resulting network interface configurations by network operators. For example, the implementation of more complex and resilient security architectures often requires inline processing of data that requires deep packet inspection, and voice and video applications do not operate well unless the order in which packets are sent and received is controlled.

Characteristically, this architecture was created when networks were predominantly static. The IP address included an inherently fixed network component and a host component. No problem exists if the hosts are rarely moved, but, as network usage models evolved, difficulties arose. The two distinguishing features of TCP/IP networks—network identity bound to location, and distributed intelligence—eventually led to serious shortcomings.

Networking software is typically embedded within the networking element (such as the router, switch, or firewall), so moving the location of hosts requires reconfiguring the network by making changes at the network element level. Networks have increasingly become more dynamic because of the proliferation of mobile hosts (such as laptops that roam among Wi-Fi hotspots and cell phones that roam among cell towers) and VMs that can be moved across racks or pods in a data center. Reconfiguring networks to accommodate these computing devices however presents enormous problems, primarily because network elements were never designed to be centrally managed (based on the principle of autonomy). Traditionally, each device has to be individually configured. Standards, such as VLANS, solved some of these problems, but networking software lagged behind the software industry in many categories including usability, maintainability, scalability, manageability, virtualization, portability, standards, and interoperability.

The increasing use of virtualization has created a new set of problems for network management. VMs are not visible to network monitoring tools, making it difficult for the network administrator to adapt to traffic requirements of physical servers running multiple VMs, virtual network interface cards (NICs), and virtual switches.

Finally, changing traffic patterns into and within data centers requires the rethinking of the design of network switching core and edge architectures. Data center densities are quickly rising as computing workloads move to the cloud and virtualization becomes widely adopted. This, in turn, is leading to significant changes in data flow patterns. The volume of data flowing into data centers and physical servers is rapidly spiraling upward at the same time as East-West traffic (data flowing between servers inside the data center) is experiencing double-digit growth.

Requirements of Modern Networks

Modern networks are characterized by rapid physical and virtual server and bandwidth growth, increasing geographic concentration, elasticity, and mobility. Among the most critical requirements are the following:

- **Adaptability.** Networks must adjust and respond dynamically, based on application needs, business policy, and network conditions.
- **Automation.** Policy changes must be automatically propagated so that manual work and errors can be reduced.
- **Maintainability.** Introduction of new features and capabilities (software upgrades, patches), must be seamless with minimal disruption of operations.
- **Model management.** Network management software must allow management of the network at a model level, rather than implementing conceptual changes by reconfiguring individual network elements.
- **Mobility.** Control functionality must accommodate mobility, for
  - Next-generation IP-based mobile networks
  - VM mobility within the data center
  - VM mobility across disparate data centers
- **Integrated security.** Network applications must integrate seamless security as a core service instead of as an add-on solution.
- **On-demand scaling.** Implementations must have the ability to scale up or scale down the network and its services to support on-demand requests.

The Concepts of Software-Defined Networking

SDN addresses many of the issues that IT administrators and managers of modern networks face. By incorporating the advances of traditional application software into a dynamic network model, an SDN provides centralized programmability of networks, separating the control and data forwarding functions of network switching. A centralized controller with knowledge of all the networking components can instruct switches how to manage traffic flows by means of software messaging. The behavior of the network is then defined by software in contrast to each individual component being independently configured by means of a command-line interface or a graphical user interface.

Certain fundamental principles have emerged in this new field, including:

- **Separation of the network layers.** As shown in Figure 1, networking software lends itself to layers (or “planes”) of abstraction—in this case, an application plane that includes management and services, a control plane, and a forwarding plane. By separating out the current monolithic software stack into these layers, new models of managing the network can be realized, offering flexibility, greater control, and innovative new service models. Specifically, layer separation enables the programmability of the network through APIs and application platforms that can take advantage of distributed processing and facilitate the modular implementation of network applications for the management, services, and control planes.
  - **Application platforms.** New software platforms that streamline control, service, and management layer functions can lead to new management frameworks and business solutions. This is the area of greatest promise in SDN.
  - **Network stack implementation.** All four planes of the network stack can be efficiently implemented as software on commodity servers that can benefit from cloud methodologies including horizontal scalability, flexibility, and usage-based pricing. The forwarding plane can also be implemented in software (and will be for many usage models), but critical applications that require high bandwidth and low latency will require implementation in custom ASICs.
- **Centralized management.** Network design and operations can be simplified by centralizing many of the functions of the management, services, and control planes. Historically, networking has been decentralized and that design principle has several advantages. There is, however, no reason for the entire network to be decentralized. Control of autonomous networks (such as branch, campus, data center, or metro networks), can safely be logically centralized without affecting the ability to communicate among networks or reducing the autonomy and resiliency of the Internet as a whole. In fact, centralized control of constituent networks can solve many of the current problems of network management. In addition, with centralized cloud orchestration systems, it is essential to have a central management that can connect compute resources with the network.
- **Standardized protocols.** Adoption of standardized protocols helps achieve multi-vendor interoperability, and enables choice and low cost.

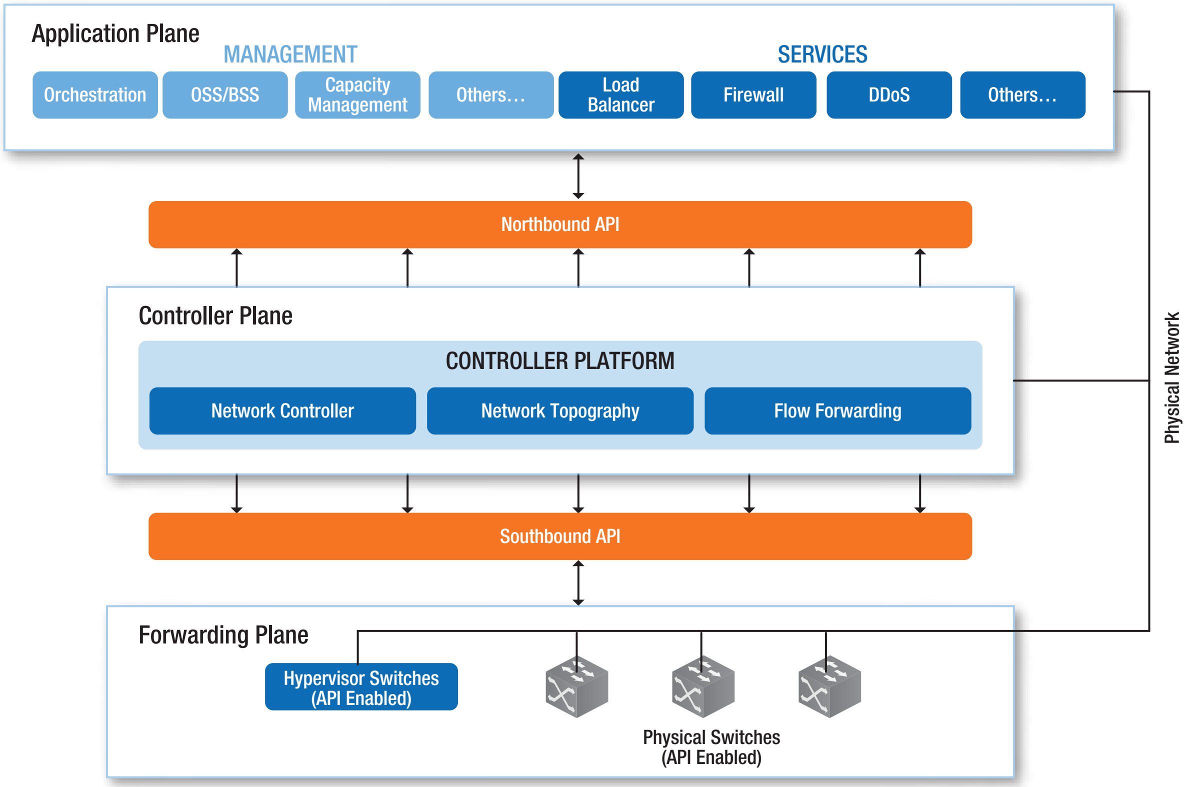

Figure 1. The conceptual architecture of software-defined networking.

The Four Layers of Network Software

Network software has four layers.

- **Forwarding plane.** Moves the data packets on the physical network layer to their next destination.
- **Control plane.** Implements the intelligence behind data transportation. The control plane establishes and defines the neighborhood topology of the network, examines data packets, and decodes the encoded protocols, deciding where each packet should go and how it should be handled. The control plane continually accumulates information and adapts to the changes in the network as they happen, ensuring smooth traffic flows, enhancing the resiliency of IP networks. The control plane also interacts with other systems (for example, the compute orchestration system) to dynamically make the changes.
- **Services plane.** Handles special tasks that require much closer scrutiny and processing of the information contained in the packets than is required for the simpler switching/routing tasks that the control plane performs. Firewalls, video streaming, and other such applications are implemented at the services layer.
- **Management plane.** The layer at which the individual network devices are configured with instructions about how to interact with the network. Functions—such as turning ports on or off, the appropriate routing protocols to use, and manual specifications of routes—are performed at this layer.

Benefits of SDN

While SDN offers promising solutions for the requirements listed previously, the benefits for different stakeholders, including enterprise data centers, cloud service providers, long-haul telecom providers, and branch networks, vary. The broadest array of benefits clearly belongs to IaaS implementations, both for public and hybrid cloud models. The primary SDN benefits are the following:

- **Centralized network control and programmability.** One of the core features of SDN is separation of the control function from the forwarding function. This enables centralized control of the network through programmable and remote control of the network elements (switches, routers, firewalls, and so on) without requiring physical access to the devices. More importantly, the entire network can be managed as an abstraction and programmed to respond dynamically to application needs, changes in the network condition, and business policy enforcement requirements. IaaS deployments gain the most benefits from these features as SDN capabilities to create, migrate, and tear down VMs—without requiring manual network configurations—maximize the value of large-scale server virtualization. All of this makes it possible to separate the physical network from the logical/tenant network, using abstraction to define the elements.
- **Dynamic network segmentation.** VLANs provide an effective solution to logically group workstations at the enterprise or branch network level. However, the 12-bit VLAN ID cannot accommodate more than 4096 virtual networks and this presents a problem at the hyper-scale data center level, where thousands of physical servers and a 20x factor of VMs will soon be the norm. Reconfiguring VLANs is also an administratively daunting task: Multiple switches and routers have to be reconfigured whenever VMs are relocated. SDN's support for centralized network management and network element programmability allows highly flexible VM grouping (virtual VM LANs), and enables easy relocation of VMs (while keeping the static IPs associated with the VMs).
- **VMs visibility.** In virtualization platforms, the virtual hypervisor switch and all the VMs running in a physical server use only one or two NICs to communicate with the physical network. Managed by server management tools, these VMs are not visible to network management tools. This makes it difficult for network designers and administrators to understand traffic patterns being generated by VMs. It also complicates design planning and network administration. SDN-enabled hypervisor switches and VMs alleviate this visibility problem.
- **Capacity utilization.** With centralized control and programmability, SDN facilitates VM migration across servers in the same rack or across pods in the same data center, or with the help of data center bridging, even among distributed data centers. This provides an opportunity for capacity-planning applications to automatically optimize physical server utilization. Such applications belong in the management and services tiers of SDN.
- **Network capacity optimization.** The classic tri-level design of data center networks consisting of core, aggregation, and access layer switches (North-South design), is facing scalability limits and poses inefficiencies for server-to-server (East-West) traffic. Innovative new designs—including link aggregation, multi-chassis link aggregation, top-of-rack, and end-of-row designs, fabric extenders, and Layer 2 multi-path protocols—have created solutions for load balancing, resiliency, and performance of dense data centers. However, these solutions are all complex, expensive, and difficult to maintain. SDN makes it easier to create and maintain network fabrics that extend across multiple data centers. This capability could lead to new usability models for managing workload deployments.
- **Distributed application load balancing.** With current technology, load balancing switches can select a server to receive a service request, but this is limited to only servers directly connected to that switch. With SDN, a new load balancing architecture is being developed that can choose the server, as well as the network path. This architecture is based on the premise that load balancing has become a network primitive. This opens possibilities for content delivery networks, essentially creating geographically distributed load-balancing capabilities.
- **Distributed network security services.** Conventionally, network security services, such as firewalls, and other security services, such as denial of service or IDS/IPS services are handled by inline hardware devices that perform the necessary deep packet inspection and divert suspect traffic away from the critical servers. With SDN, the concept of distributed security enforcement points eliminates the need to deploy dedicated, special-purpose hardware for security functions. Instead, using SDN's forwarding and programmability features, commodity servers can perform security services—such as packet filtering, and intrusion detection and prevention—when required. The general-purpose servers can be redeployed for other services during the troughs in the security services demand cycle. These network security services can also be load balanced and distributed anywhere on the network.

Challenge of Security within an SDN environment

SDN offers numerous benefits, including much more effective use of the network and compute resources, but it also presents a new paradigm to network administrators that requires understanding to effectively address security concerns.

Within a traditional network environment, administrators understand where each physical host is located and can engineer the environment to protect the most vital assets. In SDN, even though the same security controls may still be available, the physical location is less important than the way controls are implemented within an abstraction layer. Enforcement of policy depends more upon what function a service performs, instead of where it is physically located.

To improve security, administrators need to maintain close surveillance of assets within the environment and track the ways in which assets are being protected. Security services such as Security Information and Event Monitoring (SIEM) and Event Correlation to determine anomalous behavior become much more important tools to understand what is happening in the network.

A full discussion of SDN security is beyond the scope of this document but will be covered in a future document release from ODCA.

USAGE MODEL REPRESENTATIONS

Operating an SDN environment is a significant undertaking that is expected to become simpler as standards establish a foundation for open solutions, minimizing complexity and lowering cost. This usage model discussion focuses on running simple network IaaS workloads—based on several VMs in a reference cloud model using an SDN controller. It's our plan that future usage models will investigate the characteristics of SDN instances that implement the management and service layer applications. In the scenarios described, the cloud provider could be either a cloud service provider company or the enterprise IT team providing services to their software developers as cloud subscribers.

USAGE SCENARIOS

SDN enables diverse capabilities in the network layer and creates new capabilities for managing the network as a set of abstracted services. Some of the capabilities demonstrated by these abstractions include the following:

- **Decoupling VMs from the network.** SDN enables the build-out and management of hyper-scale data centers by decoupling VMs from their physical network identity. SDN ensures that VMs can communicate with each other when moved to different LANs in a data center or even across data centers on the WAN.
- **SDN-enabled security.** SDN-based networks can improve security in multiple ways, including the virtualization of security appliances that can be scaled dynamically and the integration of LAN and WLAN security policy and control into a centralized control platform.
- **Software-based load balancing.** At a fundamental level, SDN and load balancing are similar approaches to optimizing traffic flows. Current efforts seek to establish load balancing as a network primitive. With SDN-enabled load balancing, application servers can be distributed across multiple data centers to improve service levels to end users.
- **Video traffic optimizations.** Service providers can perform traffic engineering by using SDN controller in a network operations center to route and distribute high-volume video transmission traffic dynamically to the servers that are close to the point of consumption.
- **Virtual patch panels.** With a virtual patch panel, ports on multiple switches can be programmatically connected to each other using an SDN Controller to insert static flows into the flow tables of the switches. The inherent programmability of SDN ensures that the patch panel is dynamic and—unlike physical patch panels or switches—traffic flows can be modified quickly. A wide variety of use cases—including network monitoring and security applications—are well suited to the virtual patch panel approach. While this functionality is currently available from vendors today, SDN can enable the functionality to work across multiple vendors.

Beyond the use cases described, three distinct types of SDN-enabled overlay networks are becoming well established throughout the industry. The following sections examine these usage scenarios in greater detail. It's our plan that future releases of this document will include additional usage scenarios as they become recognized and adopted in the industry.

Usage Scenario 1 - SDN-Enabled Network Services

An SDN-enabled data center provides abstracted control of network capabilities, including the following:

- Capabilities for defining application profiles (for example, SLAs for latency, QoS, and other attributes) to compute clusters
- Dynamic network segmentation and logical grouping of workloads through virtual links
- Enabling of higher level services, such as firewalls, compliance checking, and scoped diversity to the established virtual links
- Deployment of the application on the network and ongoing management of operations

An SDN stack provides these capabilities through connectivity service APIs and a centralized controller that manages the physical network. Application architects typically access the connectivity APIs through a higher-level orchestration tool in the SDN stack and define application attributes and SLAs at an abstracted level instead of defining behavior by setting many individual parameters in the network elements. The SDN controller translates these business-level application profiles into network element specifications and manages the ongoing provisioning and dynamic configurations of resources. For example, as the scalability needs of a business application increase and new VMs must be spun up due to hardware cluster limits, these new VMs may need to be set up in different physical LANs. Existing VMs may also need to be moved to different LANs. The SDN controller automatically manages changes to the physical network elements based upon the initially defined profiles for the workloads.

Business drivers

Network abstraction, response, latency, service level

Goal

Cloud provider must provide SDN software stack, consisting of a centralized network controller, connectivity service APIs, and a user interface for the cloud subscriber to design application profiles.

- Reduce time to design virtualized networks, and operational parameter changes, including configuration time for VM changes

Assumptions

- The cloud provider implements the Open Data Center Alliance "Service Catalog" usage.[1]
- Cloud subscriber has signed up for the service and the provider has provisioned an administration account

Steps

1. Cloud provider provides clear documentation and orchestration tools which define the scope of available configuration and customization features of the underlying network.
2. Cloud subscriber designs the application profiles and virtual connectivity links.
3. Subscriber assigns services to the virtual communication links.
4. Subscriber deploys application clusters.

Failure condition 1

New service cannot be deployed or deployment does not behave as specified.

Failure handling 1

Cloud subscriber and cloud provider administrative accounts notified.

Success scenario 1

Cloud subscriber administrator is able to deploy and test new virtual network successfully.

[1] See www.opendatacenteralliance.org/library

Usage Scenario 2 - Enable Layer 2 Multi-Path (East-West) Traffic

Traffic patterns in modern data centers are changing rapidly so that East-West traffic predominates (consisting of traffic between servers within the data center). By some estimates, over 70 percent of traffic is now East-West. The traditional three-layer data center network design is ill-equipped to handle this kind of traffic. The use of Spanning Tree Protocol (STP), which closes many links to avoid loops, reduces the available bandwidth.

SDN provides an effective solution to this problem because the network path is determined by a central controller and does not require the use of STP. Thus, all inter-switch links can be fully utilized and traffic can be directed between multiple LANs without the need for first routing the traffic to higher-level switches.

As shown in Figure 2, Multiple Layer 2 paths can exist between points. The controller directs flows to use specific paths to avoid loops. This architecture helps alleviate choke points in the network.

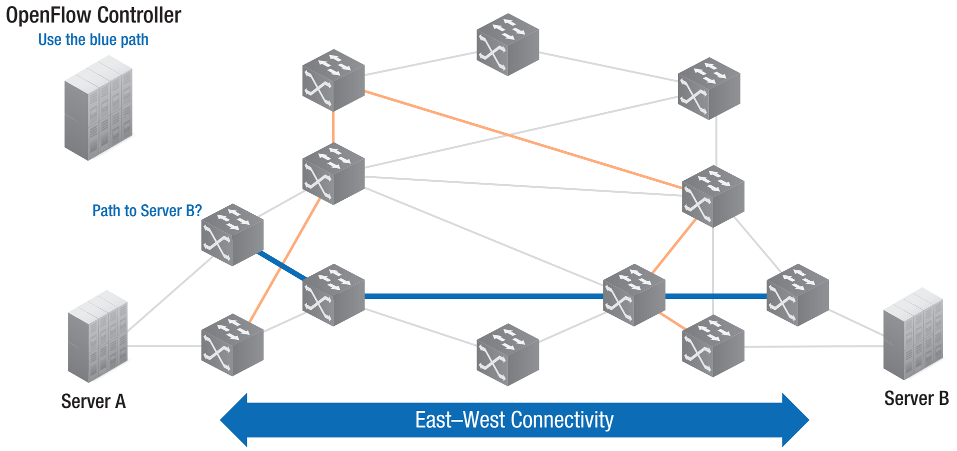

Figure 2. Layer 2 multi-path with software-defined networking.

Business drivers

Response, latency, service level

Goal

Cloud provider must enable Layer 2 multi-path traffic utilizing an SDN controller, reducing the latencies in East-West traffic.

Assumptions

- The cloud provider implements the Open Data Center Alliance "Service Catalog" usage.[2]
- The cloud provider is utilizing SDN-based software controller for managing network traffic and is not utilizing the TRILL (RFC 5556) or SPB (802.1aQ) standard protocols.
- The cloud provider is not utilizing any vendor specific Layer 2 multi-path technologies.
- Technical details of the current workload are known.
- VMs within the same data center are exchanging data.
- The cloud provider is asked programmatically, through a user interface, if it can fulfill the requirements defined by the stated technical and operational specifications.

Success scenario 1

The cloud provider can answer "Yes" to the question of whether the SDN actions are supported.

Failure condition 1

- The cloud provider does not understand the question, is not able to answer it, or is unable to provide any details pertaining to the question.
- SDN controller actions for network routing are not supported.

Failure condition 2

The cloud provider is not able to fulfill some requirements but can define the differences.

[2] See www.opendatacenteralliance.org/library

Usage Scenario 3 - Enable Data Center L2 Network Bridging

SDN enables bridging of Layer 2 networks across multiple data centers, providing a single overlay network. With this capability, application architects can place compute clusters closer to their point of usage and support effective disaster recovery.

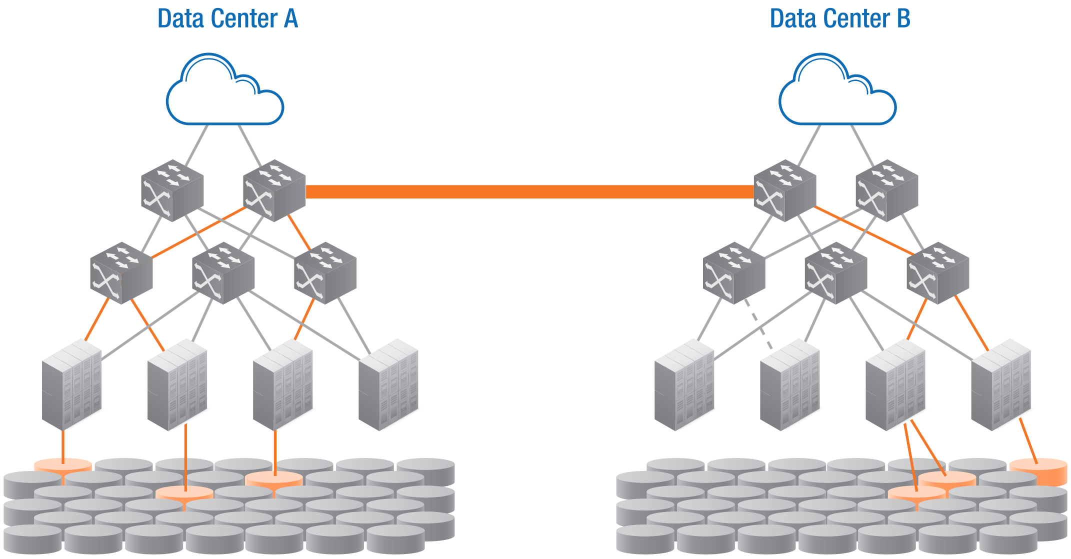

Figure 3. Layer 2 data center bridging - virtual machines from either data center appear on the same L2 network.

Business drivers

Flexibility, responsiveness, latency, service level

Goal

Cloud provider must enable Layer 2 bridging among multiple data centers so that VMs in different data centers are logically located on the same Layer 2 network.

Assumptions

- The cloud provider implements the Open Data Center Alliance "Service Catalog" usage.[3]
- The cloud provider operates multiple data centers.
- Customer workloads can be provisioned at any of the cloud provider data centers on-demand.
- Technical details of the current workload are known.
  - VMs located at multiple data centers need to exchange data and be on the same Layer 2 network.
- The cloud subscriber has access to all details about the SLA, operating-level agreement, and any specific controls.
  - Specific functions, such as remote copy or disaster recovery
  - Security root of trust
  - Consistency of input/output management (I/O controls)
  - Security and compliance (from compliance monitoring)
  - Carbon measurement
  - Geo hosting requirements
  - Licensing model and associated requirements
- The cloud provider is asked programmatically, through a user interface, if they can fulfill the requirements defined by the stated technical and operational specifications.

[3] See www.opendatacenteralliance.org/library

Success scenario 1

The cloud provider can answer "Yes" to the question of whether the SDN actions are supported.

Failure condition 1

- The cloud provider does not understand the question, is not able to answer it, or is unable to provide any details pertaining to the question.
- SDN actions are not supported

Failure condition 2

The cloud provider is not able to fulfill some requirements, but can define the differences.

LIFECYCLE MODEL

The SDN lifecycle is similar to the general IaaS lifecycle. With both, services must be discovered, designed, provisioned, implemented, used, modified, and terminated. The crucial difference lies in exposing the underlying network services as an abstraction. SDN achieves this through tools in the management layer. Another difference is that the involvement of the cloud provider in the design of the subscriber's application design is minimized because of the automation that the SDN stack can provide. As long as the suite of underlying network services are exposed to the subscriber, application architects can (either through the user interface or programmatically) specify application profiles and resources, and the SDN controller will automate the provisioning, monitoring, and maintenance of services and SLAs. This document addresses only the differences between general services orchestration and the differences for a SDN-enabled cloud provider. For an in-depth look at service orchestration for cloud services, refer to the ODCA Master Usage Model "Service Orchestration."[4]

Figure 4 shows the SDN lifecycle and associated functions.

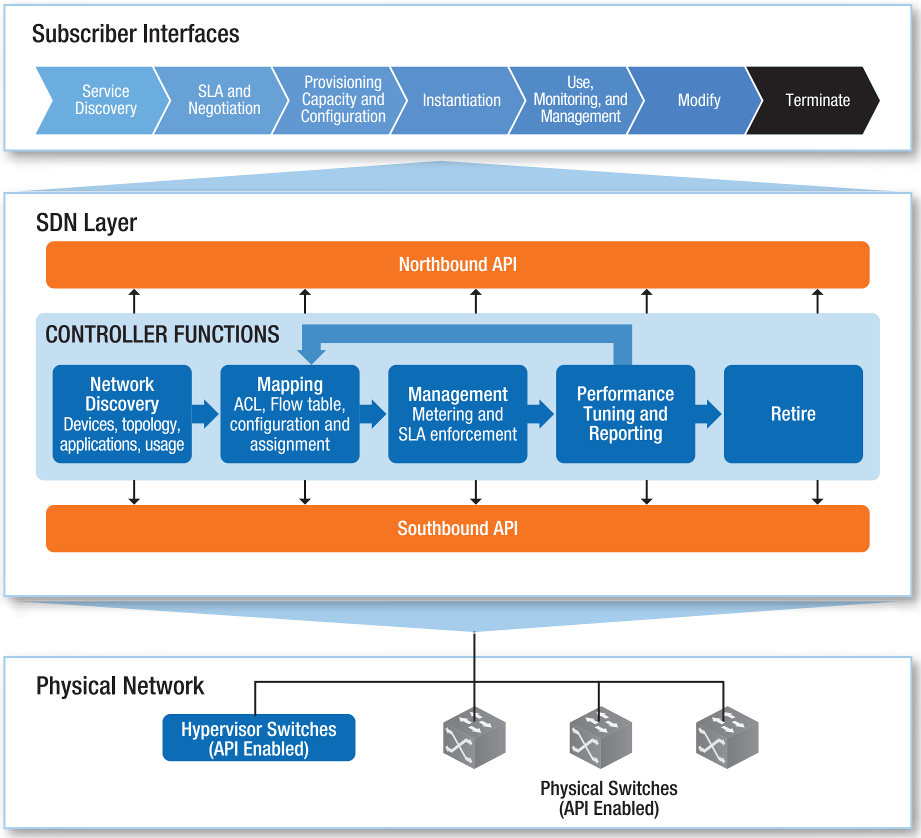
Subscriber Interfaces
Service Discovery
SLA and Negotiation
Provisioning Capacity and Configuration
Instantiation
Use, Monitoring, and Management
Modify
Terminate
SDN Layer
Northbound API
CONTROLLER FUNCTIONS
Network Discovery Devices, topology, applications, usage
Mapping ACL, Flow table, configuration and assignment
Management Metering and SLA enforcement
Performance Tuning and Reporting
Retire
Southbound API
Physical Network
Hypervisor Switches (API Enabled)
Physical Switches (API Enabled)

Figure 4. Typical software-defined networking lifecycle as related to data center infrastructure as a service.

[4] See www.opendatacenteralliance.org/library

Service establishment and service changes require adjustment of the network elements and components. Logical network elements that make up the service must be identified and adjusted in accordance with predetermined policies associated with the application SLAs. SDN tools allow specification and manipulation of the logical aspects of the network and minimize the need to make manual changes to the individual physical network devices. Table 2 lists the stages of the SDN lifecycle.

Table 2. Stages of the software-defined networking lifecycle.

Transition States	Description
Discovery	The cloud subscriber obtains from the ClaaS provider a list of available abstracted network services including data centers with geographic details, and compute, storage, and network resources presented as business abstractions.
Design	Subscriber defines application profiles, creates virtual links connecting compute clusters, assigns clusters to physical data centers as needed, and assigns network services to virtual links.
Deployment	Cloud provider facilitates automated provisioning of the virtual network services and compute and storage clusters
Use	Subscriber uses provisioned services on an ongoing basis. Provider meters usage, bills the subscriber, and provides operational and billing support.
Modify	Subscriber specifies modifications to services as and when needed. Provider enables change management, provisioning and decommissioning of service parts.
Terminate	Subscriber specifies termination of individual components or entire virtual networks. Provider must enable any archival, storage, or migration services as contracted.

USAGE REQUIREMENTS

As a general principle, usage requirements are expected to be multi-vendor and open. Key requirements need to be met across vendors and SDN implementations. However, SDN is a rapidly evolving technology area and standards have not yet been formalized for many possible uses. As an example, while standards have been defined for controlling the network elements (Southbound APIs), most applications consisting of the management and service tiers are proprietary implementations (Northbound APIs). At this time the ODCA shall specify usage requirements that may be met by the OpenFlow Switch Specification (a Southbound API). In addition, the ODCA may also recommend requirements that it believes are important for functional usage scenarios. Many of these requirements have been specified as "Should" in the MoSCoW framework rather than as "Must" directives because common models are yet to emerge.

Usage requirements are determined by a combination of requirements associated with SDN vendors and cloud service providers.

MOSCOW REQUIREMENTS

Another view of the requirements that are important to VM Interoperability can be seen in the MoSCoW prioritization table (Table 3), which enables stakeholders to understand the importance of each requirement in relation to one another and enable evaluation of responses to the RFP questions.

As documented in Wikipedia,[5] the MoSCoW categories are as follows:

- **MUST.** Describes a requirement that must be satisfied in the final solution for the solution to be considered a success.
- **SHOULD.** Represents a high-priority item that should be included in the solution if it is possible. This is often a critical requirement, but one which can be satisfied in other ways if strictly necessary.
- **COULD.** Describes a requirement which is considered desirable but not necessary. This will be included if time and resources permit.
- **WON'T.** Represents a requirement that stakeholders have agreed will not be implemented in a given release, but may be considered for the future.

As a principle, all requirements are expected to be multi-vendor and open. Key requirements need to be met across vendors and hypervisors. These requirements are determined by a combination of requirements set to SDN product vendors and cloud service providers.

[5] https://en.wikipedia.org/wiki/MoSCoW_Method

Table 3. MoSCow Prioritization Requirements.

Requirements for Phase	Required Title	Description	MoSCoW
All Phases	Standardized	At a minimum service providers must implement the Open Networking Foundation's OpenFlow™ Switch Specification	Must
	Open	Available from the public domain	Should
	Secure	Security attributes (security zone restrictions, compliance requirements)	Must
	Portable	Specification of all service features (Application Profiles) should be compliant with formats that are portable across any service provider that implements recognized standards	Should
	Efficient	A standard recommended process must exist for specifying any of the software-defined networking (SDN) lifecycle actions	Should
	Extensible	The current services should demonstrate the potential and capability to be extended as technology functionality evolves (usually through use of open documented standards)	Should
	Internationalized	Localizable attributes (monetary symbols, language specification)	Could
	Identified	Feature attributes to determine the stakeholders related to all resources (compute, storage, virtual machines, services) at all times	Should
Discovery	Management	Support standard and consistent ways of discovering, configuring, and managing SDN services	Should
	Single Data Center	All SDN services available for a single data center that subscriber is setting up primary services in	Must
	Multi-Location	All data centers across which SDN services may be configured should be available through a common user interface to the subscriber	Could
	Programmatic Interfaces	Interfaces to support programmatic specifications of all available SDN services	Should
Design	Application Profiles	Ability to design application profiles	Must
	Cluster Assignment	Ability to configure compute clusters to one physical data center	Must
	Cluster Assignment	Ability to configure compute clusters and assign sub-groups to multiple data centers	Should
	Virtual Link Assignment	Ability to assign network links to compute clusters	Must
	Network Services Assignment	Ability to assign network resources to virtual links	Must
	Templates	Templates of commonly used application profile patterns	Could
Deployment	Licenses	Identify relevant products and licenses and license models	Must
	Automation	Hands-free deployment and automatic provisioning of the full physical and virtual environments	Should
Use, Monitoring, and Management	Lifecycle Management	Support standard and consistent ways of discovering, configuring, and managing the lifecycle of SDN services	Must
	Monitoring	Monitoring: Detection and tracking of all deployed environments. Service-level agreement, availability, performance monitoring, and usage statistics	Should
	Diagnostics	Consistent set of attributes and functions to provide correlation between Application Profiles and operational environments	Could
	Collection	Workload-based Logical Collections: Workload-specific collections of VMs, networking, and storage components that can be managed as a whole, with policy-based automation	Should
	Modify	Ability to modify Application Profiles and apply changes dynamically	Should
Terminate	Analyze	Analyze impact of retirement by checking stakeholders and integration/dependencies	Could
	Stop	Announce to stakeholders that the specified service is being stopped	Must
	Decommission	Delete a system or service and perform the service termination activities, such as data and memory deletion and scrubbing, and returning resources to resource pool	Must
	Remove from Profile DB	Remove specified Application Profile from Profile data base	Must

RFP REQUIREMENTS

The ODCA considers the following requirements important for inclusion in requests for proposal (RFPs) to cloud providers to help proposed solutions support standard SDN actions and consistency among management solutions.

The RFP questions are distilled from the parameters and models identified in the usage model. They represent a conceptual evaluation base, rather than a specific technology analysis. These questions are intended to provide a departure basis to assist a potential cloud subscriber in becoming aware of all of the relevant dimensions which they should consider in their evaluation of a proposed cloud service. These conceptual areas may lead to the potential cloud subscriber extending them with specifics pertinent to their own organization's requirements, as relevant.

The conceptual RFP questions for the SDN usage model are as follows:

- **ODCA Principle Requirement.** Solution is open, works on multiple virtual and non-virtual infrastructure platforms, and is standards-based. Describe how the solution meets this principle and communicate any limitations it has in meeting these goals.
- **ODCA Software-Defined Networking Usage Model Rev. 1.0.** Solution should provide discovery capability of abstracted network features, both programmatically and through a user interface.
- **ODCA Software-Defined Networking Usage Model Rev. 1.0.** Solution should provide the ability to specify application profiles, compute and storage clusters, and virtual links both programmatically and through an online design workbench application. Upon receiving the command to provision the application, solution should automatically create VMs, assign storage, and configure network switches as specified in the application profiles.
- **ODCA Software-Defined Networking Usage Model Rev. 1.0.** Solution should be able to specify modifications to application profiles and changes to compute clusters including moving VMs to any physical server in the data center. Solution should automatically configure network switches with the new configuration.
- **ODCA Software-Defined Networking Usage Model Rev. 1.0.** Solution should be able to decommission the service and perform all teardown activities including the deletion and archival of VMs, wiping storage according to SLAs, and releasing all network resources.

Click here for an online assistant, Proposed Engine Assistant Tool (PEAT),[6] to help you detail your RFP requirements.

[6] www.opendatacenteralliance.org/ourwork/proposalengineassistant

SUMMARY OF REQUIRED INDUSTRY ACTIONS

To guide in the creation and deployment of solutions that are open, multi-vendor, and interoperable, the ODCA has identified specific areas where we suggest there should be open specifications, formal or de facto standards, or common IP-free implementations. Where the ODCA has a specific recommendation on the specification, standard or open implementation, it is called out in this usage model. In other cases, we plan to work with the industry to evaluate and recommend specifications in future releases of this document.

It should also be noted that the ODCA may look to extend this usage model (or develop a separate usage model) to discuss how to effectively overlay security services in an SDN environment.

SDN APIs, Emerging Protocols, and Industry Actions

As network software is abstracted into levels, protocols and APIs are required for communications between the layers and interoperability among multi-vendor systems. The controller application configures the individual network elements (switches and routers) through APIs termed Southbound APIs, essentially enabling programmatic control of the parameters that are provided by the device's operating system. In comparison, the Northbound APIs of the controller present the network in terms of business abstractions to the management applications (such as cloud orchestration systems), and service layer applications (such as firewalls, and denial-of-service attack mitigation systems). Northbound APIs manage higher level functions, including path computation, STP loop avoidance, routing control, deep packet inspection, and other functions.

The OpenFlow protocol has become the de facto standard for Southbound APIs, but currently the Northbound APIs are proprietary, with different controller platform vendors specifying their own interfaces. As common models for higher-level network abstractions are established, Northbound API standards will certainly emerge.

Developments within the open-source community have begun to impact how SDN takes shape across the industry. Previously, open standards ensured that multi-vendor interoperability could be achieved; vendors could differentiate products by extending protocols to enhance solution capabilities. As the need for common capabilities becomes vitally important, revised standards accommodate the new requirements. However, these standards take a long time to develop and usually lag market demand for innovation. Standards wars typically result in market fragmentation. In the past few years, however, vendors are increasingly turning away from open standards in favor of collaborative open-source implementations. This is illustrated by the success of OpenStack in the cloud industry and the recent announcement of OpenDaylight.

Other important industry groups are user-driven and adopter-driven organizations (such as the Open Networking User Group and the Open Networking Foundation (ONF)). These groups focus significantly on technology adoption and can influence the adoption of standards and common practices by vendors. These groups adopt a variety of methods such as sharing best practices, creating common requirements and holding joint negotiations with vendors, and through certifications and testing of conformance with standards. As a next step, the ODCA intends to invest in conformance tools including work with its partners such as ONF to accelerate vendor conformance. Such programs create a degree of confidence to end-user customers who want to future-proof their investments."

The following industry actions are required to refine this usage model:

- **Collaboration between the Open Data Center Alliance and ONF** on Southbound interfaces (with their corresponding forwarding plane implementations) and their utilization, particularly OpenFlow. Focus areas shall be to align to or document support for:
  - Dynamic network segmentation into logical groups
  - Multi-path traffic
  - Layer 2 Network bridging
- **Collaboration between the Open Data Center Alliance and other groups** as appropriate on other areas of SDN.
  - Abstracted control of network capabilities
  - Integration of network automation into overall provisioning and orchestration (in line with CIaaS MUM)
- **Collaborative work to accelerate the creation of an open and non-proprietary network API** that can be freely licensed by the member organizations.
- **Adoption and full production implementations of SDNs** in cloud environments as quickly as practical.
- **Support for initiatives, including OpenFlow,** across all cloud-provider platforms.

Appendix 1: Standards Groups and Bodies, and Open Source Projects

The following is a list of relevant and related standards groups and bodies:

- Open Networking Foundation (ONF): www.opennetworking.org
- OpenFlow: www.opennetworking.org/sdn-resources/onf-specifications/openflow
- Telecom Industry Association (TIA): www.tiaonline.org
- European Telecommunications Standards Institute (ETSI): www.etsi.org/
- Association for Telecom Industry Solutions (ATIS): www.atis.org/topsc/sdn.asp
- Internet Engineering Task Force (IETF): www.ietf.org

The following is a list of relevant and related open source projects:

- OpenStack Quantum: https://wiki.openstack.org/wiki/Quantum
- OpenDaylight: www.opendaylight.org