White Paper:
"Operationalizing IT Services"
Software Defined Networking Federation/Operability Orchestration

Brian Hedstrom
Lead BSS/OSS Architect
bhedstrom@datavision-inc.com

Mark Abolafia
VP, SDN Practice Lead
mabolafia@datavision-inc.com

White Paper: "Operationalizing IT Services" - Software Defined Networking Federation/Operability Orchestration

In enterprise network and datacenter environments, the management of services is often accomplished through device management which is still highly vendor specific with multiple element management systems communicating with, and controlling their respective families of devices. With the advent of "open" approaches driving the replacement of these devices with commodity hardware, the cost of that replacement relegates that approach to a long term migration. In the short to medium term, a method of defining IT service lifecycles and standardizing network management across disparate vendor devices is needed.

Figure 1 illustrates a layered model typically found within the Communication Service Provider realm, with the mapping to an enterprise environment done for IT Services instead of Communication Services. At the highest level is the end user or IT Customer and the lowest level includes physical or virtual devices at the IT Resource Layer.

From the top-down approach, an IT Customer may be an external customer or an internal customer, such as a business unit within the enterprise. The "IT Product" Layer represents product offerings from the enterprise which are procured by External end-customers. "IT Products" are typically part of a portfolio as many large enterprises offer multiple products across the IT space. External IT customers may then browse publicly published portfolios for which IT product offerings they are interested in obtaining. The IT Service layer represents services designed, implemented and managed by the enterprise in order to fulfill customer business needs. Just as products make up a product portfolio, services make up a "service catalog".

For example, a single product offering might have three services associated with it. Internal enterprise customers may access the service catalog directly for their business application needs. External facing products are therefore realized through internal facing service definitions. The IT Resource Layer represents physical and/or virtual assets (resources) which realize a service or set of services, for example compute, networking, storage and security assets residing in the large enterprise data center and network. Fundamentally, these resources represent devices which require administration, provisioning, monitoring, and maintaining in order to ultimately fulfill a customer's business application needs.

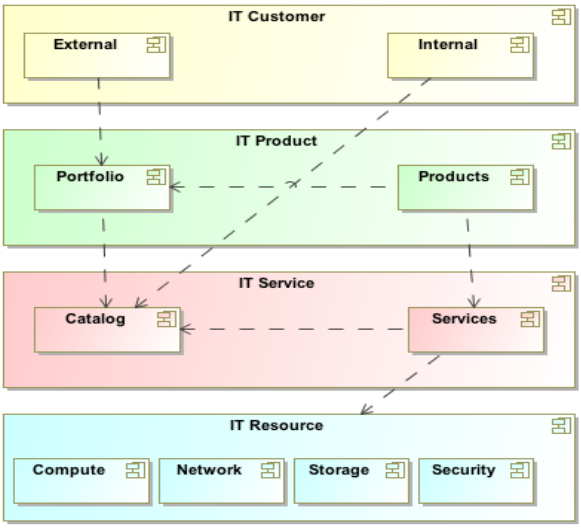

Figure 1 – IT Operationalization Layered View – Level 1

Figure 2 takes the operationalization of IT Services one-step further by defining a lifecycle for an IT Service based on the ITIL set of service management standards. The lifecycle begins with a Service Strategy that focuses on the business strategy, business use cases, market analysis and portfolio management.

Portfolio management is a key aspect of Service Strategy, as it defines what the product offerings are and what needs they fill. Service Design, the next stage of the lifecycle, focuses on the design of the IT service, processes required for the service as well as other service management activities. The service definitions should be technology agnostic at this stage in order to abstract the service from the IT Resource Layer (since technologies and protocols advance and change).

Catalog management is a key aspect of Service Design as it defines the services and how those services are associated with the business through service-oriented business processes, information and interfaces. Service Transition focuses on delivering services to customers. This is where orchestration of services lies, including fulfilling service requests from internal customers or via product instances from end-customers.

Asset Management, Configuration Management and Change management are key functional areas as new services require resources with specific feature configurations and incremental changes to instantiate services or modify existing services. With a service deployed live, Service Operations focuses on delivering services according to Service Level Agreements. This is where active monitoring of the services is performed, around fault and performance perspectives. Event and Incident Management relies on real-time monitoring of fault and error conditions whereas Problem Management looks at root cause analysis of fault or degradation conditions and may include analytics to view performance trends. Finally, Continual Service Improvement aims to improve the processes across the entire IT Service Lifecycle.

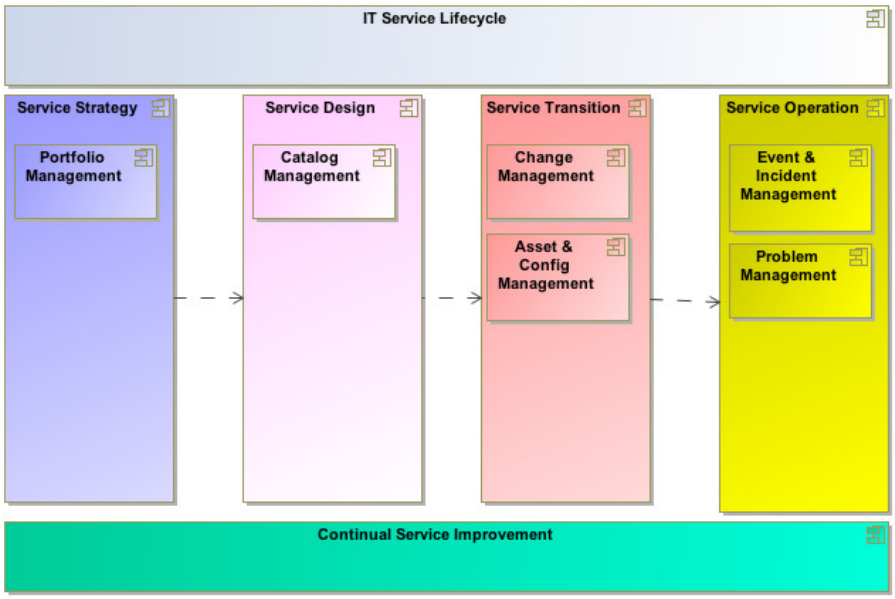

Figure 2 – IT Operationalization Layered View – Level 2

In examining Software Defined Network as a Service, among the challenges of network device management, the most challenging is that changes need to be "orchestrated"; meaning, there is a certain order in which changes have to be applied to ensure that the network remains operational. With more and more functionality contained in the typical end-device, this has become a complex undertaking. Vendors have been responding by introducing "controllers" that can manage many of their devices, and orchestrate changes between them. However, also controllers are highly vendor specific, with proprietary communication both north- and sound-bound. Figure 3, below, illustrates this present-day "mash-up" of controllers and EMSs.

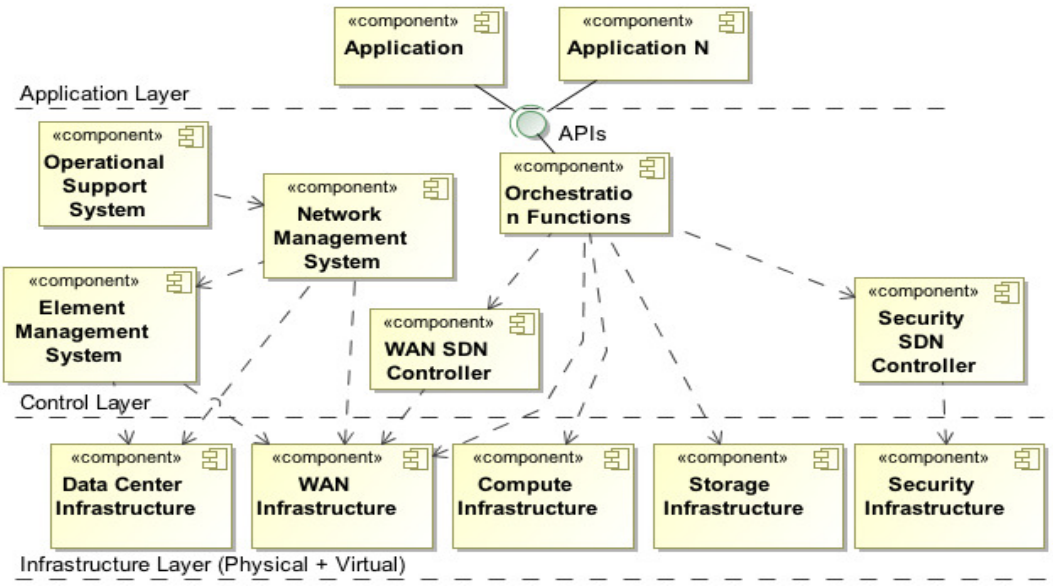

Figure 3 – Present Day Landscape

Problem and Challenges encountered in orchestrating today's networking environments include:

1. Communication between the Orchestrator and the controllers is not standardized or non-existent. The Orchestrator is often a human being and the configuration is done through Command Line Interface (CLI). This creates an error-prone and non-scalable bottleneck to rapidly deploying new services or service upgrades.
2. The mode of communication between the controllers and the infrastructure layer elements is typically CLI, with little automation.
3. In the case of legacy components, there is a lack of controllers. Legacy Operational Support System (OSS) applications as well as a hierarchy of Network Management System (NMS) and Element Management System (EMS) applications, are required to manage these legacy devices.

The solution to these issues is the adoption and use of an open API for controllers that is standardized among vendors, allowing them to consistently issue network management instructions that are deployed over that vendor's infrastructure.

Eventually, one could architect a “controller of controllers”, where a master controller orchestrates, via REST APIs, the other controllers to consistently manage a network. For example, the master controller would configure controllers from vendor X and vendor Y which themselves communicate to infrastructure layer elements of vendor Z. Eventually, we expect that hardware devices will adapt open standard APIs themselves such as NETCONF, so the mechanics of network orchestration will be standardized, intermediate controllers will be eliminated or reduced, and the master controller takes on much higher-level architectural control functions to ensure overall performance and reliability of complex networks.

This white paper identifies the potential Use Cases and puts forth an architectural framework that speaks to an Application Layer, Control Layer and Infrastructure Layer around which to realize a Federated SDN control system.

Requirements

1. Controller functions must provide northbound interfaces to orchestration functions.
2. Orchestration functions must provide east-west interfaces between different orchestrators.
3. Controller functions must support southbound heterogeneous (open & proprietary) interfaces.
4. The framework must support multiple administrative domains.
5. A controller may receive configuration demands or request from multiple orchestrators.
6. The orchestration function provides northbound interfaces to the service catalogs.
7. All APIs must be opened & published such as REST and NETCONF.
8. The controller must provide a discovery function.
9. Standardized APIs must be defined for most standard functions.
10. Must scale to "n" number orchestrators and controllers.

Use Cases

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

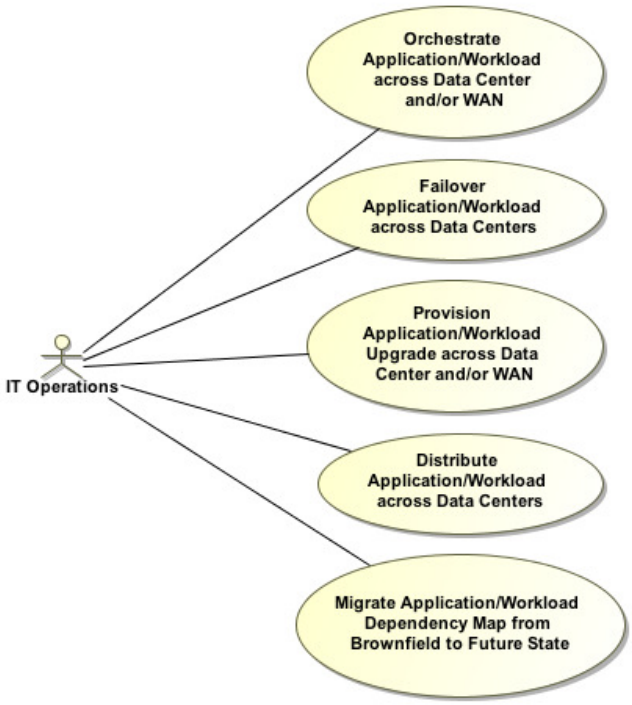

Figure 4 – SDN Federation and Operability Orchestration Use Cases

Use Case 1: Orchestrate Application/Workload across Data Center and/or WAN

Currently, data center orchestration and controllers are disjoint from wide area networking initiatives. This use case seeks to connect these software-defined infrastructure islands to establish network connectivity and network services for applications/workload that spans both local and wide area networking. In this use case, a workload is configured to span both data center and wide area network infrastructure. Specific SD-WAN and SDN virtualized networking/overlay connectivity require coordination and tunnel establishment. In addition, to support a workload, network services resident in branch offices and data center require configuration.

The benefits are to be able to provide automatic configuration, simpler repeatable processes less prone to user errors, faster provisioning.

Use Case 2: Failover Application/Workload across Data Centers

In this Use Case, an application is running in a data center A and it 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 to activate the application in data center B.

Benefits of this approach include automatic failover. For the cases of high cost and complex high-availability (HA) architecture, this could help to contain the cost and simplify the architecture with having almost the same benefits of HA.

Use Case 3: Provision Application/Workload Upgrade across Data Center and/or WAN

In this Use Case, a new application is provisioned or an update 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 resource consumption needs of new services.

Use Case 4: Distribute Application/Workload across Data Centers

In this Use Case, an application spans between 2 data centers. We would envision that the orchestrator will configure the different equipment to provision compute, storage and network (router, load balance, security) resources, and bandwidth between data centers to properly distribute a workload.

Benefits are the application is deployed quickly with minimal human intervention. The application could be redeployed exactly the same way between 2 other data centers. Additionally, during periods of high usage (Black Friday retail events, Open Enrollment in Health care businesses, additional automatic provisioning of resources on high trading volume days/hours, etc.) these resources could be orchestrated to ramp in as the demand on the applications ramp.

Use Case 5: Migrate Application/Workload Dependence Map from Brownfield to Future State

In this Use Case, the multiple controllers are orchestrated to configure their respective devices/components to establish a workload's dependency map. The dependency map may be wide area communications or WAN devices, firewalls, IPS, switches, routers etc., that are all hardware devices and/or software components that an application depends upon to deliver its service at a high level of performance and/or availability.

Benefits are to automate workload dependency map configuration to decrease the time of IT Service delivery.

SDN Federation

The Open Networking User Group (ONUG) SDN Federation and Operability Orchestration working group has defined SDN Controller interoperability to be performed by Orchestrators within the Control Layer. As Figure 5 illustrates, the architecture is viewed as a three layer model where Applications reside at the top most level, which also represents the layer exposed to customers and/or users. The middle layer, the Control Layer, represents the control logic necessary to orchestrate services required by applications in the Infrastructure Layer. The Infrastructure Layer, or IT Resource Layer, represents the physical and/or virtual resources required to instantiate services such that a customer or end-user may use the application.

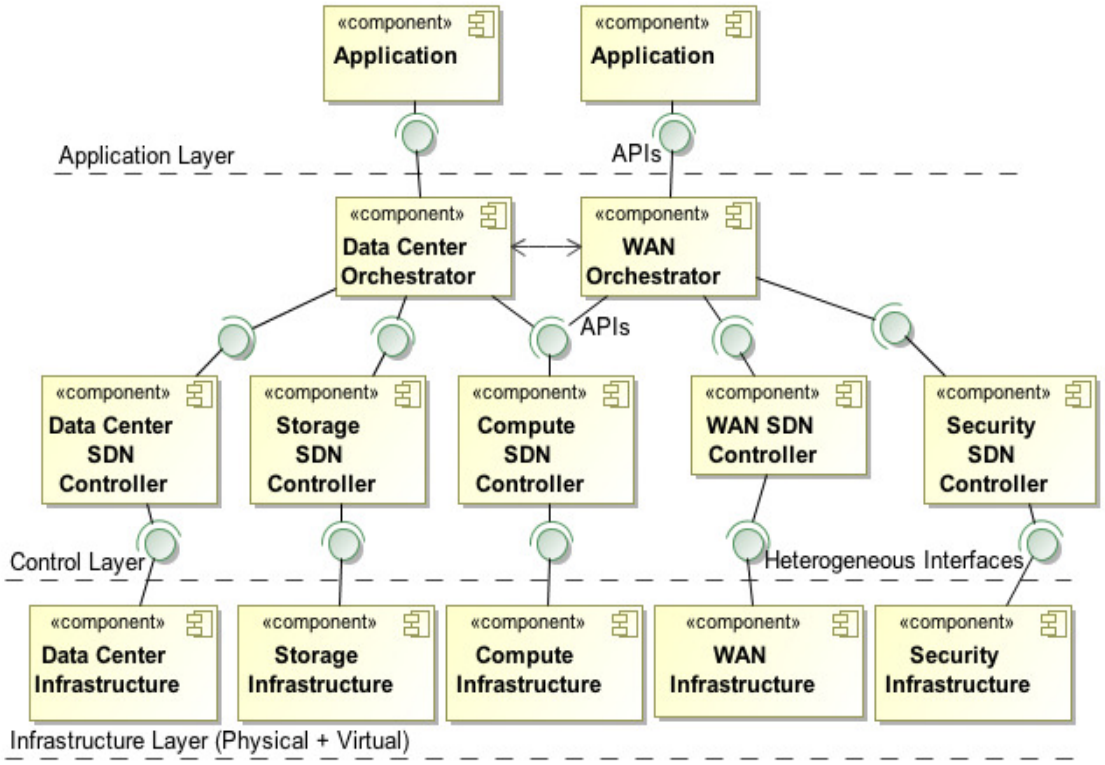

Figure 5 – SDN Federation and Operability Orchestration Architectural Framework

Application Layer

The Application Layer includes 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, they are aware of the services they need such as networking, storage, security and/or compute.

Applications, generally based on user interaction, request such services from the underlying layers. As an example, deploying a new application, based on a user's need, requires application connectivity to a network, storage and backup access to disk, authorization and authentication through application and network security policies, and a computing platform to run the application, whether distributed across multiple VMs or residing on a single VM. There is a significant dependency on the underlying resources in supporting the deployment of the new application. Secondarily, once the application is running, how are change management activities performed, such as dynamically increasing the available storage pool or modifying the network security due to tightened security policies?

Control Layer

The Control Layer includes the various orchestrators and controllers. Orchestrators provide end-to-end IT Service Lifecycle management capabilities to the upper Application Layer. The orchestrators receive requests from the Application Layer, such as a change request or new deployment request, while fulfilling the request through direct communication with underlying controllers, which control portions of the Infrastructure Layer. Orchestrators are therefore service aware, whether the service request includes network, compute, storage or security, or a combination of all. Orchestrators may be specific to a service, however inter-orchestration capabilities are required to hand off requests in a distributed or domain specific way. For example, one orchestrator may handle data center service request 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 IT 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 at an Orchestrator to deploy the application on a new VM, with certain compute, storage and networking attributes. The Orchestrator parses the request into how this gets realized via the controllers it communicates with across the entire environment it has visibility into. One Controller may be called to provision a new VM in Data Center X, another may be called to provision available storage capacity in Data Center Y, while another may be called to establish a secure network across the two Data Centers such that the new VM has storage resources. The Controllers handle infrastructure requests to instantiate the requested services.

Infrastructure Layer

The Infrastructure Layer, or IT Resource Layer, includes the physical and/or virtual network, compute and storage resources, including routers, switches, bare-metal servers, hypervisors, virtual machines, storage pools, firewalls, etc. 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 interfaces such as NETCONF and vendor proprietary interfaces such as CLI.

Architectural Detailed View

Figure 6 illustrates a deeper view into the Control and Infrastructure Layers, where interfaces and functions are highlighted.

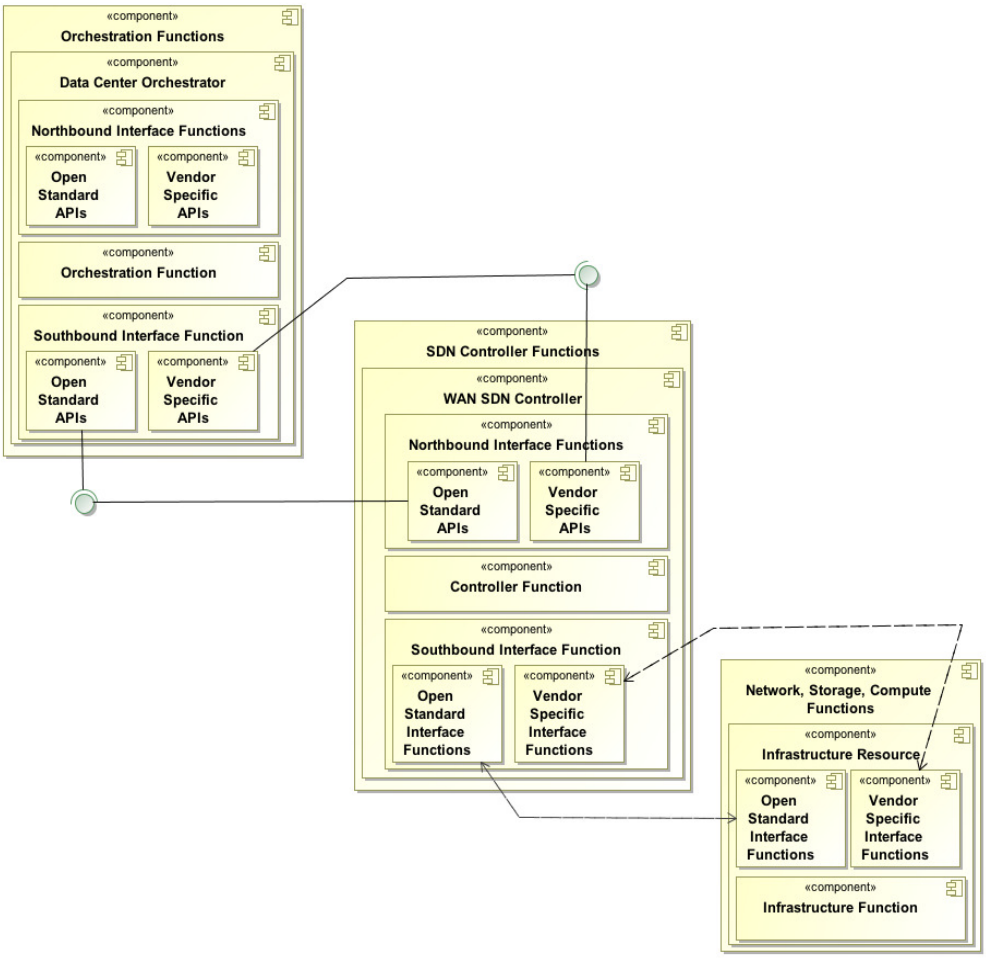

Figure 6 – SDN Federation and Operability Orchestration Architectural Framework Detailed

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 Controllers for which each Orchestrator has responsibility. This interface is structured like the northbound interface. Sitting between the northbound and southbound interface functions is the orchestration function itself, performing all business logic of the orchestration capabilities.

Controller functions may include any number of Controller types, such as WAN Controllers, compute Controllers, etc. 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, 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.**

A Solution:

In the course of work on behalf of service providers and large enterprises, we have developed a potential solution to address the aforementioned Use Cases that would consist of a Network Orchestration system, and a Data Center or Cloud Orchestration system. This is illustrated in Figure 7. These would be married on the Northbound side through a Portal that would be accessible by either or both IT personnel for provisioning network and data center resources as needed by the business, or, in a more sophisticated use, by actual departmental users seeking to provision incremental IT resources on demand.

On the southbound side, the Network Orchestration system would communicate directly to LAN and WAN resources to provision those resources according to the need at hand. The Data Center/Cloud Orchestration system has the ability to call up compute and storage resources on-demand. The combination of the two, with an ability to terminate VPN tunnels across a network, would be able to not only connect company to company sites, but also connect company data center private cloud infrastructure to AWS, Azure, etc. and be able to manage connectivity across all data centers and the resources utilized by all applications.

1. Network Orchestration Features:

- **View Service and Service Parameters**: View available services and associated service parameters relative to an end user/group. This may also include viewing network device configurations implementing the service.
- **Modify Service Parameter**: modify service parameters (e.g., Bandwidth, QoS, etc.) associated with an end user's service.

2. Data Center/Cloud Orchestration Features:

- Deploy any number of Virtual Machines.

- **Create Virtual Networks**: Create a virtual network utilizing the deployed VNFs (CPLANE Networks Overlay Gateway Router) with network connectivity to deployed VMs.
- **Connect Virtual Network to Transport Network**: Connect the virtual LAN network to the physical transport network (e.g., MPLS L3 VPN service)
- Connect private cloud to AWS or Azure, etc.

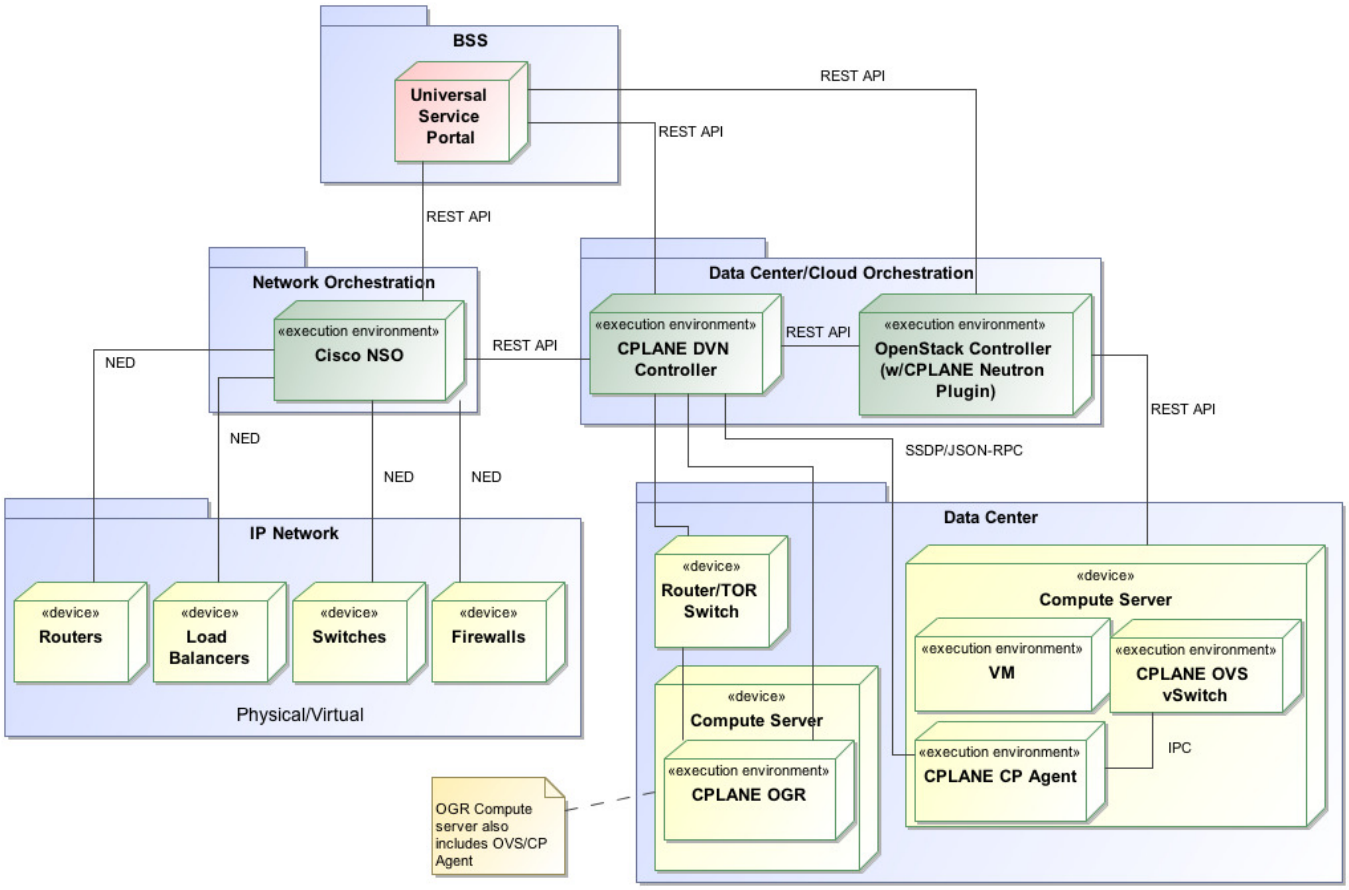

Figure 7: SDN Orchestration for the WAN and Datacenter

Conclusion

We have put forth a set of requirements and use cases to demonstrate the need and potential solution for SDN Federated Orchestration in the Data Center and the WAN based on the IT Service Lifecycle model to more effectively operationalize IT. Additional components of the architecture could be expanded to include switch fabrics as well. One could also potentially overlay this entire solution with an orchestrator of orchestrator such as OpenDaylight to include other functional silos of infrastructure such as optical transport and/or more sophisticated storage networks in legacy configurations.