KEY FINDINGS

NFV founder members warn that NFV business models may be hampered by conflicting MANO implementations

The dual role of the NFV Orchestrator is a major source of market confusion, especially when vendors include NFV-specific extensions to OpenStack in their NFV-O products

"Build it yourself" is the only way to avoid NFV MANO vendor lock-in at this stage of the market

Network equipment providers have moved faster than IT players to implement NFV MANO, but the race is getting closer

Vendors that can link a strong B/OSS/ network service management story to their NFV MANO offer will attract market interest as NFV moves into an operational phase

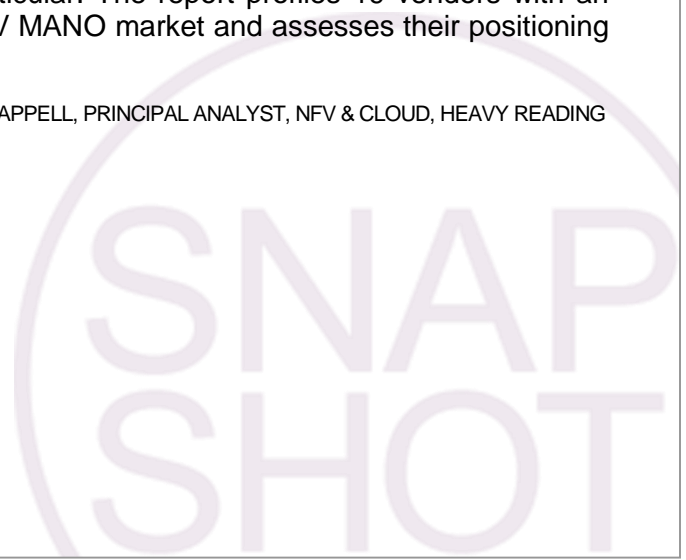VOL. 13, NO. 2, FEBRUARY 2015

NFV MANO: What's Wrong & How to Fix It

The NFV Management and Orchestration (MANO) architecture is responsible for managing cloud infrastructure, virtualized network functions (VNFs) and services composed of VNFs, but what are functionally described as separate orchestration layers in ETSI NFV documents are typically implemented in idiosyncratic ways by NFV MANO vendors at this point in the market.

As a result, the emerging "zoo" of NFV MANO products is causing operator confusion and represents a large barrier to NFV adoption. This situation is compounded by the immaturity of OpenStack, the leading candidate for the virtualized infrastructure manager (VIM) layer of the NFV MANO and a key reference point for functionality in the upper layers.

This report examines the issues influencing vendor interpretations of NFV MANO and the requirements for greater clarity around MANO functionality in general, and improved capabilities within OpenStack in particular. The report profiles 19 vendors with an interest in the NFV MANO market and assesses their positioning and products.

AUTHOR: CAROLINE CHAPPELL, PRINCIPAL ANALYST, NFV & CLOUD, HEAVY READING

Introduction & Key Findings

In 2014, Axel Clauberg of Deutsche Telekom – a key network functions virtualization (NFV) "mover and shaker" – coined the term "zoo of orchestrators" to describe different vendors' interpretations of what is still a weakly defined component of the NFV reference architecture: the NFV Management and Orchestration (MANO) stack.

Heavy Reading research confirms that NFV MANO is the largest source of operator confusion and the biggest barrier to NFV adoption today. Yet it is also critical to the NFV venture. The NFV MANO architecture is responsible for orchestrating and managing the cloud infrastructure on which virtualized network functions (VNFs) execute; the VNFs themselves, as they run in a cloud execution environment; and – because we are talking about the network, where network functions don't exist in isolation from one another – the services that are composed from multiple, chained VNFs, as they execute across the cloud.

At its simplest, NFV MANO consists of a cloud management system (CMS) and a service orchestration engine that knows how to deploy services composed of VNFs and how to assure those services throughout their lifetime in a dynamic cloud environment.

But life is not so simple: first, because a CMS is no better defined than the MANO stack, so different candidate CMSs have different sets of capabilities, and second, because it turns out that, for various reasons outlined in this report, a CMS needs significant extension to support NFV. An NFV MANO vendor's implementation turns out to be contingent on a number of factors at this early stage in the market, including the kind of VNFs it needs to orchestrate and manage and, perhaps more critically, its CMS assumptions, starting point and partnerships.

Heavy Reading research confirms that OpenStack is the clear favorite of all possible candidates for the CMS – or to use NFV MANO terminology, the Virtualized Infrastructure Manager (VIM). OpenStack is immature today and needs significant proprietary extensions to support NFV. This means that each NFV MANO vendor is wrapping OpenStack with its own, non-standardized implementation of these extensions, often displacing them into higher levels of the MANO stack to avoid accusations of "forking" open source code. Moves are afoot through the Open Platform for NFV (OPNFV) Project and the OpenStack sub-team for NFV to fast-track the development of open source versions of NFV extensions, but at present, vendor support for OpenStack is no guarantee of interoperability between NFV MANO implementations and other components of the NFV reference architecture: VNFs and NFV infrastructure (NFVI) cloud platforms.

In its second, "implementation" phase, the European Telecommunications Standards Institute (ETSI) NFV Industry Specification Group (ISG) intends to focus on creating clearer, prescriptive definitions of the functionality that operators want to see at each level of the MANO stack. Since operators may wish to plug in different vendors at different MANO levels, they don't want to pay twice for the same functionality, or have to sort out conflicting capabilities if they want to offer advanced NFV services, such as NFVI as a service (NFVIaaS) or VNF(s) as a service (VNFaaS).

Industry dynamics such as vendor relationships, the project-based nature of service deployments, siloed organizational structures and sourcing policies will inevitably mean that operators will end up deploying multiple NFV MANO products, at least initially. In the longer term, interoperability between these products will be highly desirable if operators are to achieve the transformational benefits NFV promises. This will depend on a better definition of, and separation of concerns between, MANO layers of orchestration.

NFV MANO: What's Wrong & How to Fix It looks at the need for clarification around NFV MANO and new proposals and initiatives that could influence its implementation. It discusses the way in which OpenStack is influencing interpretations of NFV MANO and analyzes the approaches that different vendors are taking to fulfilling one or more layers of the NFV MANO stack.

The key findings of this report include the following:

NFV MANO is needed to manage workloads executing on cloud infrastructure, so its starting point is the IT CMS. Big differences between IT workloads and VNFs, as well as between the IT cloud platform and the NFVI, mean that IT CMSs don't support the MANO functionality needed for NFV. This functionality includes support for VNF placement on bare metal resources, distributed resource allocation and setting up WAN connectivity. Over time, NFV MANO vendors and users anticipate that their extensions of the IT CMS will be required by other types of enterprise beyond telcos, especially as the cloud edge becomes more distributed to support the Internet of Things.

Although the ETSI NFV ISG describes three functional layers of orchestration in the MANO stack, these are not being implemented separately in practice. IT CMSs don't distinguish between different layers of orchestration, so most MANO implementations don't either. The ETSI NFV MANO draft white paper acknowledges that implementations may not preserve the functional separation that operators originally envisaged between components.

NFV founder members, including BT and Telefónica, point out that interoperability between different operators' MANO layers and implementations will become important as they develop NFV services, such as NFVIaaS and VNFaaS. The development of rich NFV business cases depends on agreeing each layer's responsibilities and the application programming interfaces (APIs) and service-level agreements (SLAs) through which each exposes its functionality to others. Otherwise, BT points out, nonsensical scenarios will arise, such as an NFVIaaS provider ceding resource allocation control in its own infrastructure to its customer's NFV Orchestrator (NFV-O).

A key problem with the functional MANO architecture as it currently stands is that it splits responsibility for resource (NFVI) management between two functional layers – the VIM and the NFV-O – in a way that is causing confusion. This is partly due to weak definition of the NFVI itself: Is an NFVI a local domain, such as a data center, or is it a distributed aggregation of local domains, where each domain may be based on different technologies? Global resource management is parked in the NFV-O alongside service orchestration, but BT and others argue that it should be separated out into a fourth layer: the inter-domain resource orchestration layer.

Given the uneven capabilities of different VIM candidates and the implementation approaches of NFV MANO vendors, build-it-yourself is the only strategy for NFV MANO implementation guaranteed to protect operators from any level of vendor lock-in at this stage of the market. Only the very largest and most cloud-experienced operators are choosing to do this, however. Other operators need to accept that buying an OpenStack-based MANO solution doesn't necessarily free them from vendor lock-in either at VIM or higher levels of the NFV MANO architecture today. OpenStack is hard to deliver without vendor support and many carrier-grade extensions will remain proprietary in the near future. There are high hopes that the OPNFV Project will create an open source NFV reference implementation of OpenStack with open source versions of these extensions.

IT vendors and network equipment providers have shown different levels of commitment to NFV, but this situation is changing fast. IT vendors, despite their cloud presence and solutions, have been much later to line up their NFV ducks than the equipment vendors, although there have been leaders and laggards even within the equipment vendor community. This is no longer the case, with all the big equipment vendors now pushing their agendas for telco cloud, and IT vendors recently starting to articulate coherent NFV strategies that will give equipment vendors a run for their money.

Defining NFV MANO

Executing VNFs in the cloud requires new management functionality that has not previously been needed in the physical network. Conventional B/OSS, used to manage network functions running on dedicated hardware, do not provide this functionality and will need to be augmented to handle VNFs and the cloud infrastructure on which they depend. This process will have a transformational effect on conventional B/OSS, albeit one that is beyond the scope of this report. This report focuses on the new MANO capabilities mandated by NFV and industry approaches to its implementation.

Many early NFV proofs of concept (PoCs) focused on migrating functions onto hypervisor-enabled commodity servers to test their performance and reliability in what is still a novel environment from a network perspective. In 2014, the driving forces behind ETSI NFV, founding members such as AT&T, BT, Deutsche Telekom and Telefónica, made it clear that the cloud, not virtualized servers, is their platform of choice for NFV.

It is now widely accepted that the NFVI in the ETSI NFV reference architecture (see **Figure 1**) is a network-appropriate equivalent of an IT cloud platform. The cloud is not merely a collection of virtualized servers. It is a software-based abstraction of the infrastructure resources used to execute applications: compute, storage and latterly, through software-defined networking (SDN), connectivity. This infrastructure abstraction is insulated from the physical implementation of those resources (i.e., choice of server, switch and storage hardware) through a virtualization layer.

Figure 1: NFV Reference Architecture
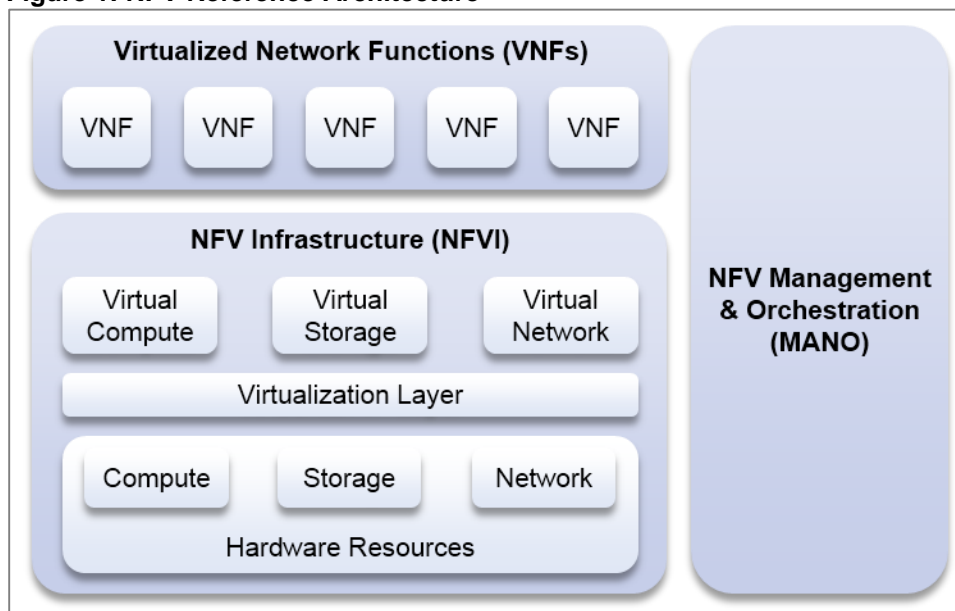
Source: ETSI NFV ISG

Due to its abstract (virtualized) nature, the cloud:

- **Makes virtual resources (compute, storage, connectivity) appear infinite** to workloads running on them. Since resources are apparently unlimited, workloads (VNFs) can scale on demand to consume more or less of them, efficiently using the right amount of resource at any given moment.

- **Supports multiple types of workloads**, multiplexing them in highly efficient ways across physical infrastructure. Leading operators want to take advantage of the cost efficiencies of the cloud to lower opex. They adamantly want cloud platforms that are vendor-agnostic when it comes to running VNFs, so that they can repurpose cloud resources to execute any VNF. Telefónica, for example, says it will not work with VNF vendors that only allow their products to execute on a vendor's proprietary cloud platform.

- **Provides redundancy at an infrastructure level**, instead of the level of individual physical components. The cloud's abstraction properties prevent workloads running on it from being affected by the failure of individual physical server blades, storage devices or connections.

- **Enables virtual resources and VNFs to be manipulated dynamically**, and in an automated fashion, through software APIs. This programmatic orchestration of cloud components ensures that each workload at all times has an appropriately configured and sized execution environment available to it.

In the ETSI NFV reference architecture:

- **NFV infrastructure (NFVI)** is the term for the cloud platform that provides a virtual execution environment for **virtualized network functions** (VNFs) on top of physical server, storage and networking resources.

- **NFV Management and Orchestration (MANO)** is the term for the set of management capabilities needed to manage VNFs and one or more VNFs stitched together as a "network service," plus their virtual execution environment(s) in the NFVI. This set of capabilities supports scalability, resource allocation (workload multiplexing), high availability and programmatic orchestration of workloads executing in the cloud.

NFV MANO & the IT Cloud Management System

NFV MANO both is and isn't a telco equivalent of the cloud operating system (OS) or CMS that enterprises use to orchestrate cloud infrastructure in support of IT applications. There are similarities between the MANO stack and a CMS, but also big differences. In the IT world:

- CMSs don't make the functional distinction the ETSI NFV architecture does between NFV MANO and the virtualization layer in the NFVI. Typically, cloud management and orchestration and the virtualization layer are tightly coupled within an IT CMS. The combined intelligence of the CMS and virtualization layer drives virtual machine (VM) scheduling (placement), migration, scaling and recovery mechanisms that protect VMs from failure.

- The distinct layers of the NFV MANO architecture are not always evident in an IT CMS. The IT CMS equivalent of the VNF Manager used to deploy and manage IT applications, sometimes called a platform as a service (PaaS), is somewhat distinguishable from the IT CMS equivalent of the VIM, although the two functions are usually closely coupled in IT CMS implementations. The CMS may understand IT application needs sufficiently well that it can prevent them suffering performance degradation and business policy breaches.

- There is no equivalent in the IT CMS for many of the additional management functions needed for NFV. This is because many network functions, as they are currently virtualized, require the NFVI to support them in ways that IT workloads, so far, have not demanded, and which therefore have not been reflected in IT CMS.

- IT applications are by nature discrete, "cookie cutter" executables that can be packed onto standard servers and managed in a uniform way. VNFs are often assembled into services, where a service is an entity that needs to be managed and executed in addition to the individual VNFs it contains. In an NFV network service context, the scaling of one workload element (VNF) may have knock-on implications for other VNFs participating in the network service, a situation that doesn't typically arise in the IT cloud.

It is worth noting that a "service" from an IT perspective is an application or application component, such as a database server, firewall or router. In a networking context, the term "service" has very different connotations: It implies a customer-specific configuration of a network function or assembly of network functions. An NFV network service is an example of a resource-facing service (RFS) in TM Forum parlance, although it perhaps would have been more helpful to have called it a virtual RFS (vRFS). The current MANO terminology reflects the fact that OSS interests were not involved when it was first defined.

The IT CMS Needs Extending to Become an NFV MANO Solution

IT workloads take advantage of the centralization, consolidation and standardization of commodity IT hardware in large scale data centers. Cloud providers rely on the very high utilization, often based on oversubscription, of this hardware to wring out capex benefits and the automation of management tasks to achieve opex savings.

In the early days of NFV, it was assumed that VNFs would run on IT cloud platforms with perhaps a few tweaks to the hypervisor to ensure "carrier-grade" performance. It is now becoming clearer that because many VNF workloads are different from IT workloads, there are specific NFV-related requirements for the "telco cloud." As Telefónica points out:

> *IT cloud computing is consistent. There is a pre-defined set of "flavors" (the OpenStack term for virtual compute server templates), a predefined number of zones that are always the same shape and pre-created network segments. In an NFV environment, you are continuously creating and reshaping the network, creating random numbers of zones of different shapes. As you scale, you create new zones and may need to apply new rules to them. [IT] Cloud is dynamic in one direction, NFV is dynamic in multiple dimensions.*

For example, many VNF workloads:

- Are highly variable, in contrast to IT workloads, and many have very specific, fine-grained requirements for their execution environment.
- Need consistent, predictable performance with extremely low levels of latency.
- Exchange enormous amounts of traffic with the external network in contrast with IT workloads, which generate high volumes of intra-data center (East/West) traffic.
- Are interdependent by definition, as component functions of an interconnected network and network services.

These differences have implications for cloud management. A CMS managing and orchestrating an NFVI needs to:

- **Avoid resource oversubscription when required**. Many VNFs cannot be shoehorned onto servers to boost utilization if there is a risk that critical processing power or memory may not be available the instant they need it. A CMS for NFV must have a fine-grained, disaggregated view of the components that make up virtual compute resources in order accurately to assign VNFs to virtual resources. For example, it should have NUMA (Non-Uniform Memory Architecture) awareness to provide visibility of the way in which server resources are partitioned and CPUs, I/O devices and memory are attached to sockets. It also needs detailed memory management capabilities, so that VNFs that carry out high-speed I/O operations can be assigned continuous memory access on servers that support huge pages. This is in contrast with IT cloud computing where huge pages are deliberately disabled to minimize memory fragmentation when memory is allocated to multiple VMs.

- **Support the distribution of VNFs** across multiple locations to minimize latency and security risks to traffic. Eventually, operators envisage being able to distribute VNFs across tens or even hundreds of locations, including hyperscale data centers, central offices, base stations and customer edge devices, so that they can move VNFs dynamically in and out of locations depending on parameters such as traffic volumes, bandwidth availability, user experience and delivery cost. Centralization in a few conventional data centers is not appropriate for many VNFs.

- **Manage the software-defined transport layer of the network** to support high traffic throughput. In an NFV context, where VNFs are distributed across multiple network locations/data centers, the CMS needs to orchestrate connectivity across both the data center and WAN domains. This involves the programmatic stitching together of data center and MPLS networking (although DT's Terastream proposes a radically simpler way to support SDN in the WAN) so that end-to-end connectivity can be spun up and down on demand.

NFV MANO capabilities such as support for VNF placement on bare metal resources, distributed resource allocation and establishing WAN connectivity are not typically implemented in IT CMS today. Over time, NFV MANO vendors and users anticipate that their extensions of the IT CMS will be required by other types of enterprise beyond telcos, especially as the cloud edge becomes more distributed to support the Internet of Things.

Supplier Strategies for NFV MANO

CMS vendors sell a VIM capability, typically integrated with an infrastructure virtualization layer (e.g., hypervisor) and, in some cases, an OS (Linux), to create a CMS. Vendors in this category have different strategies for coupling with the virtualization layer and/or OS, e.g., partnering versus owning all the pieces in the CMS stack.

This group of vendors hold the key to NFVI virtualization and management. When it comes to NFV, there are really only two VIM/NFVI virtualization layer games in town: VMware and OpenStack/KVM at present. VMware and the clutch of OpenStack vendors are not only pursuing operators, they are also forging partnerships with the network equipment providers, in a bid to become a strategic component of the equipment vendors' NFV reference architecture implementations.

When selling direct to operators, CMS vendors emphasize their hardware independence. On the other hand, an equipment provider's NFV reference implementation may certify the CMS on its chosen server platform, giving operator customers confidence in its performance.

One enterprise IT vendor (GigaSpaces) has been included in this report because its open source technology is the basis for a number of telco vendors' NFV MANO implementations.

Vendor Profile: GigaSpaces

GigaSpaces' latest version of its Cloudify 3.0 pure-play open source CMS brings orchestration, monitoring and real time analytics together to enable the automation and self-management of applications on OpenStack and other clouds. GigaSpaces points out that other cloud management tools, focused on initial deployment of applications in the cloud but not lifetime management activities such as monitoring, SLA management, upgrades, scaling, failover and recovery. Such tasks were often carried out through custom scripts and integration of various monitoring, log management tools on top of configuration management tools such as Chef or Puppet.

Cloudify 3.0 integrates a variety of DevOps tools such as Chef, Puppet, OpenStack Heat, Docker, CollectD, Celery and allows users to plug in their own DevOps tool of choice. Cloudify 3.0 uses real-time analytics to drive automated actions, such as scaling, failover and the deployment of new resources/application instances, based on policies.

GigaSpaces has completely rewritten Cloudify according to OpenStack design principles and guidelines, in the Python language, so that Cloudify "looks and feels" to OpenStackers like a native plugin. Cloudify 3.0 will work with any OpenStack distro – GigaSpaces has certified it with Red Hat and HP Helion and intends to add Mirantis and Ubuntu to the list.

Although the company has targeted Cloudify at OpenStack first, it intends the system to support other clouds, including VMware private clouds and public clouds such as Google Compute Engine, AWS and Azure.

Cloudify 3.0 implements the OASIS TOSCA specification. GigaSpaces joins IBM, Huawei and a number of other vendors in an effort to bring TOSCA into the OpenStack Heat project (currently in incubation stage). GigaSpaces claims that NFV is a motivation for bringing TOSCA and Heat together since the latter is unable to handle complex orchestrations needed for standing up and managing VNFs in the cloud. Cloudify 3.0's Intelligent Orchestration engine also integrates directly with Heat, so companies that wish to deploy infrastructure natively in OpenStack can do so, bringing Cloudify in later for ongoing application lifecycle management.

Cloudify 3.0 comes with network orchestration support that enables users to set up network infrastructure elements such as connectivity (VLANs), floating-IP, routers, security groups, availability zones, as part of the overall application deployment, using the Neutron API. Tying network orchestration together with application deployment enables tightened security and isolation of VMs as prescribed in an application blueprint.

A major new component in Cloudify 3.0 is its integrated analytics engine based on combination of Riemann, open source network monitoring, Elasticsearch's Logstash open source log management system and a built-in workflow engine. Cloudify aggregates data from agents and external sensors that it sets up to instrument the application environment as well as from existing logs and metrics (e.g., from OpenStack Ceilometer). Cloudify correlates this data, compares it against policies and uses it to drive pre-scripted workflows, for example, software upgrades or disaster recovery actions.

In an NFV context, where virtual appliances are "black boxes" that won't support the installation of an external agent, Cloudify uses a remote agent to monitor whatever parameters the appliance exposes through its API, feeding these into its analytics engine. Agents monitor common CPU and memory usage metrics, in addition to metrics that are important to specific applications, such as the number of concurrent sessions or transaction latency. In an NFV context, probe data on network performance could also be correlated and used to drive remediation and other processes in the future.

GigaSpaces positions Cloudify 3.0 as an alternative to Amazon Opsworks or Red Hat CloudForms: the latter, like Cloudify, is available as open source software. GigaSpaces agrees with the growing chorus of voices that suggests that OpenStack focus on a core set of functionality that does not compete with the broader ecosystem, including providers of application lifecycle management.

GigaSpaces' partners include IBM SoftLayer, VMware vCloud and HP Cloud. In an NFV context, Alcatel-Lucent is the best-known user of Cloudify, as a foundation for CloudBand Carrier PaaS.