Independent market research and competitive analysis of next-generation business and technology solutions for service providers and vendors

HEAVY READING
WHITE PAPER

NFV Service Design & Operations:
Saving the Sorcerer's Apprentice

A Heavy Reading white paper produced for Accanto

AUTHOR: JAMES CRAWSHAW, SENIOR ANALYST, HEAVY READING

INTRODUCTION

In Goethe's *Sorcerer's Apprentice*, popularized by Disney's *Fantasia*, the wizard's assistant tires of his manual chores and uses magic to automate them – with disastrous consequences. With network functions virtualization (NFV), operations teams are also looking for automation, and the acronym soup of NFV can often resemble wizardry (or gobbledygook). To avoid drowning in a sea of alarms and bug-ridden services (and acronyms), communications service providers (CSPs) need new tools that take the best from the DevOps world of enterprise IT and add functionality that caters to the more complex environment of network services.

CSP and enterprise networks are becoming increasingly complex with the introduction of software-defined networking (SDN) and NFV. Ironically, this could make networks more costly to operate than traditional physical appliances, at least in the initial phase of deployment. NFV introduces additional layers of operational complexity that, in the words of AT&T's Irene Shannon and Jennifer Yates, in Building the Network of the Future, "put more onus on the operator to integrate technologies that were traditionally integrated by a vendor." Essentially, the disaggregation of network management systems, software, operating systems and underlying hardware (processors) leaves CSPs with the arduous task of gluing it all together themselves.

CSP operations teams have historically relied on operations support systems (OSS) tools to manage networks and services. The move to NFV will require that they learn new skills, such as the OpenStack cloud computing platform or the scripting language Python. Additionally, new DevOps tools and methodologies are needed to automate testing and deployment of changes across multiple environments. Feedback from operations on production networks can then be rolled into subsequent product sprints for iterative improvements.

One of the reasons that automation is such a hot topic in the telecom industry right now is that with the introduction of NFV, network management has suddenly become a lot more complicated. Unless CSPs plan to hire more operations staff (which we doubt), they are going to need new tools to automate the management of these new systems.

NFV SERVICE LIFECYCLE OPERATIONAL CHALLENGES

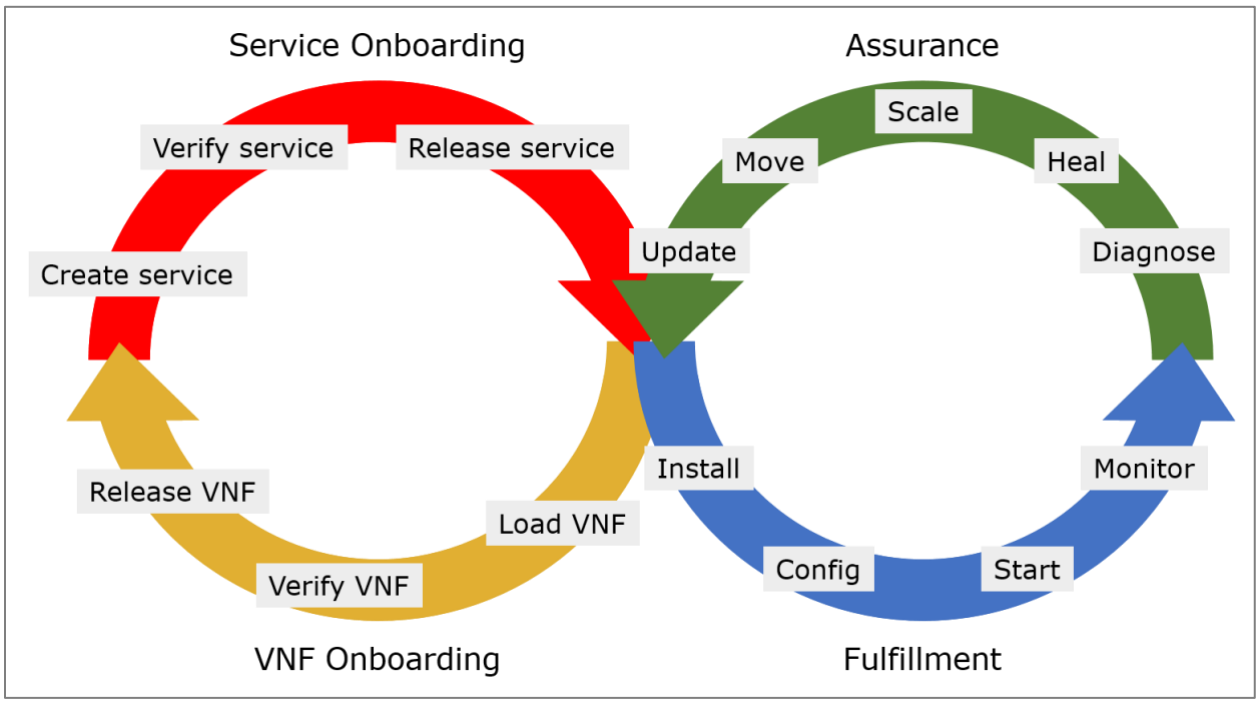
Physical network appliances have highly manual processes to manage their end-to-end lifecycle. The testing, installation, configuration and problem management of traditional network appliances involve many manual activities that often require a human touch, and even the infamous truck roll. The software paradigm of SDN and NFV promises fully automated lifecycle processes to take network services into production and maintain (scale, heal, patch) them on an ongoing basis. Virtualized network functions (VNFs) should support greater programmability of lifecycle tasks than their physical counterparts, enabling near-full automation of the creation and healing of services.

This is the equivalent of the apprentice putting on the sorcerer's hat and commanding the broom to carry buckets of water to fill the cauldron. However, just as casting spells is not as simple as it seems to the apprentice, NFV today presents significantly greater operational complexity than traditional networking, and this is holding CSPs back from achieving their automation goals. Consider some of the challenges across the NFV service lifecycle:

- **Service Onboarding:** Before designing a service, one must first onboard and test a portfolio of VNFs from which that service will be composed. Given that many VNFs are simple rewrites of software that was designed to run on proprietary hardware and repurposed to run on a specific VM (i.e., not truly cloud-native), CSPs are struggling to get many VNFs to scale cost-effectively (or even work at all) in their environments.
- **Service Lifecycle Programming:** The service designer must assemble various VNFs to work together as a service. The service chain will depend on various pieces of independently managed VNFs (e.g., firewall, gateway, DNS server) cooperating in an interconnected web. The operational workflows (tasks needed to instantiate, configure, heal, upgrade, etc. the service) must be programmed to enable automation. The complexity of the underlying resources, infrastructure and VNFs, across domains and vendors, can cause service designers to end up designing complex, sub-optimal workflows.
- **Service Verification:** Service bundles of multi-vendor VNFs must be tested for inter-operability and performance. Complex pre-production test environments are required; operational service testing can be a highly manual and time-consuming activity.
- **Service Deployment:** When put into production, services and VNFs must be loaded and configured. Service deployments are highly error-prone. Ready-for-service testing is, again, highly manual.
- **Service Maintenance:** Once a service is deployed, the operations team must then be able to roll out regular (e.g., weekly) patches, manage customer upgrades and resolve performance issues. Services and VNFs must be updated, scaled, healed and migrated. Service, VNF, virtualized infrastructure management (VIM) fault resolution and upgrade procedures across VIMs are all highly manual activities today.

Many NFV orchestration tools have been developed to manage the initial phase of a service lifecycle – namely the installation, configuration, starting/stopping, reconfiguring and monitoring of VNFs. Collectively these steps form the fulfillment phase of the lifecycle (see **Figure 1**).

Figure 1: NFV DevOps Lifecycle



Source: Heavy Reading

As CSPs begin to operationalize NFV, they are finding that some orchestration platforms are failing to manage the entire service lifecycle – specifically, the steps of problem diagnosis, healing, scaling, moving (VNFs from one VM to another) and updating (patching). Collectively, these steps form the assurance phase of the lifecycle. Very few orchestration platforms encompass the entire DevOps lifecycle – i.e., the development steps of VNF loading, verification and release; and service creation, verification and release (VNF and service onboarding, respectively).

While NFV seems straightforward on paper (especially white ones), the devil, as they say, is in the details. While onboarding a single VNF has become straightforward for the early adopters, creating services that are a mesh of multiple VNFs is still complex, considering the heterogenous mix of platform environments upon which the individual VNFs may be running. If one VNF in a service chain misbehaves, there can be a domino effect on other VNFs and potentially other services.

Overseeing this mesh of VNF dominoes requires the operations team to be experts in a vast range of NFV and cloud technologies to which they have previously had little exposure. Throwing more bodies at the problem to deal with this complexity means that costs can quickly get out of control – just as the apprentice's enchanted broom gets out of control, continuing to fill the overflowing cauldron with water and flooding the room. Furthermore, the complexity will force service designers to be more cautious, potentially stifling the innovation that NFV is supposed to unleash.

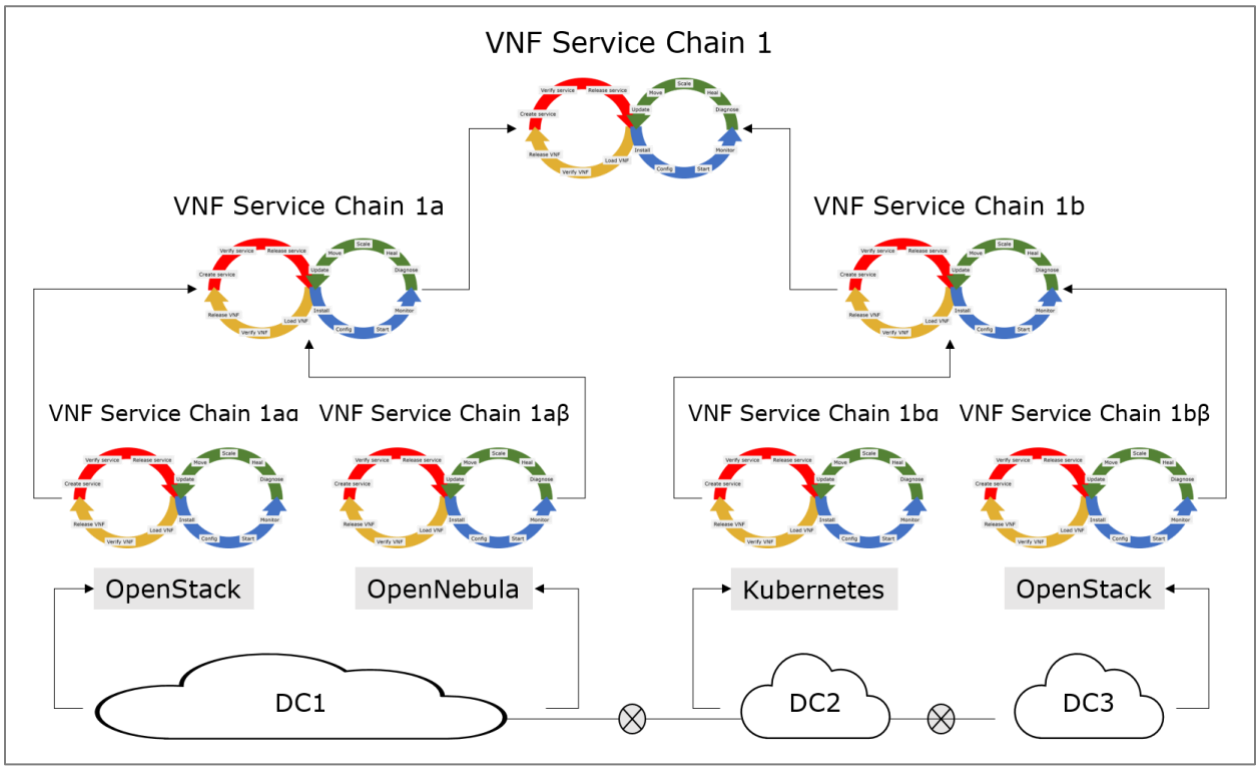
COMPLEXITY INCREASES WITH SERVICE CHAIN LENGTH

To move from proof-of-concept to operationalizing NFV at scale presents a huge challenge for CSPs. Assuming that services have been successfully deployed and their health is being monitored (along with that of the underlying VNFs), CSPs need to:

- Test VNF bug fixes
- Test new service bug fixes
- Upgrade services while in production
- Automate day 2 service operation use cases
- Create service diagnostic tools
- Train analytics to identify known issues
- Diagnose issue resolution
- Auto-heal service issues
- Auto-upgrade service versions

The above activities span both engineering and operations teams. Even with the simplest network service, complexity is very high. Consider **Figure 2**: Service chain 1 is dependent on two subordinate service chains, which in turn are dependent on two subordinates each. The VNFs that comprise each service chain might be running in different data centers and on different management systems (OpenStack, OpenNebula, Kubernetes, etc.). Stretching the analogy somewhat, this hierarchy of service chains is similar to the sorcerer's apprentice chopping the broom with an axe to try to fix the "VNF bug," but then finding that the split pieces come alive and turn into more brooms, carrying yet more buckets of water.

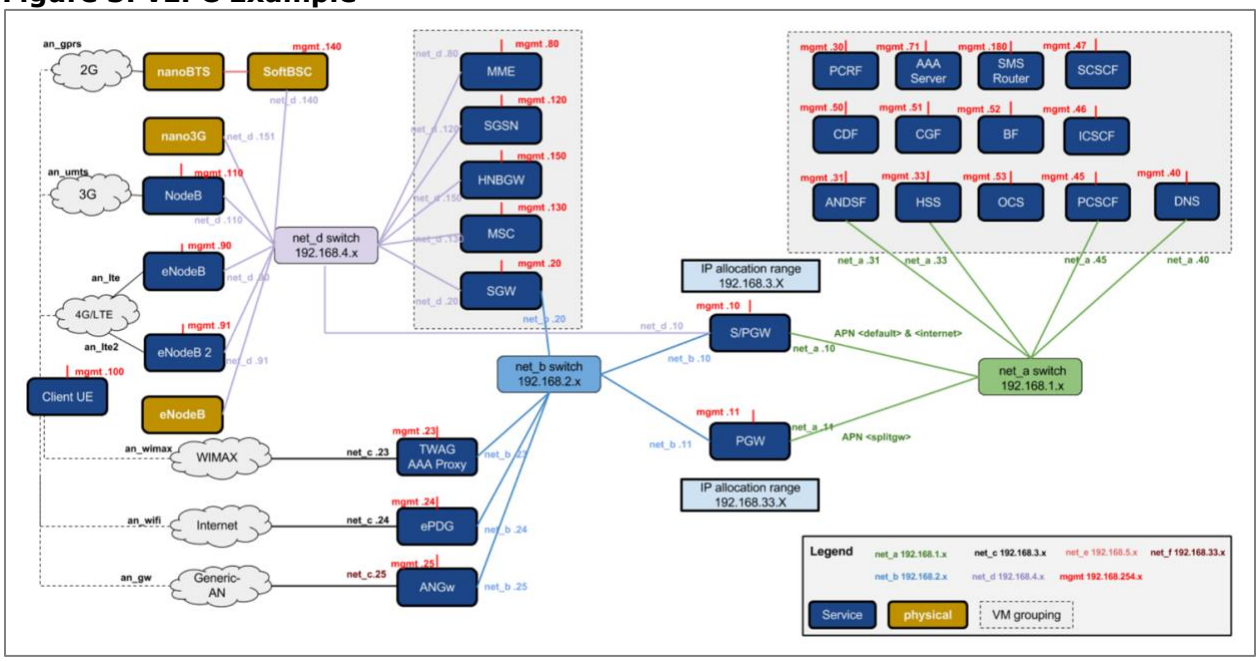
Figure 2: Complexity Increases Exponentially as Service Chains Lengthen



Source: Heavy Reading

Now consider the real-world example of the virtual Evolved Packet Core (vEPC). **Figure 3** shows a basic deployment model for vEPC, with some simplified configuration requirements.

Figure 3: vEPC Example



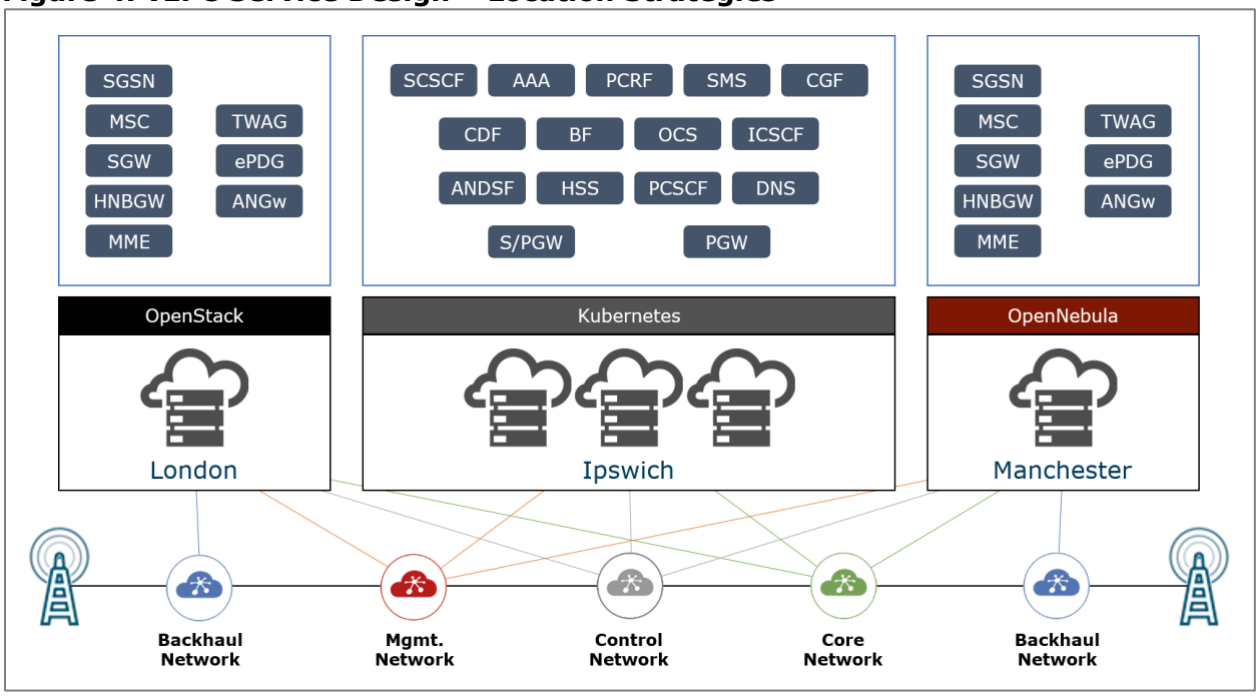
Source: Core Network Dynamics

There is a grouping of VNFs in the center of the diagram that includes MME, SGN, MSC and SGW. What the acronyms stand for is not important here; the point is that they are all complex VNFs in their own right, and they form part of an even more complex system (the EPC). There is another cluster of VNFs on the right (PCRF, AAA, etc.).

Each VNF cluster could reside in distinct data centers (centralized and edge). Each VNF must be deployed in a special configuration defined by the service designer. Each VNF block in the diagram may actually comprise multiple software instances running on different virtualization management technology (VM, containers). The system has many moving parts, with multiple elements that must be moved, healed, scaled and monitored across a heterogenous set of environments.

The service designer needs to take this software architecture and map it to their network topology, as shown in **Figure 4**.

Figure 4: vEPC Service Design – Location Strategies



Source: Accanto

Deciding which components of the vEPC should be deployed in centralized data centers and which should be closer to the customer is a complex trade-off of latency and cost. Each data center might have different cloud and virtualization technologies deployed (e.g., OpenStack, Kubernetes, OpenNebula). The data centers will be connected with an underlay network, the performance characteristics of which need to be factored into the overall service design.

Within each data center, the service designer will need to consider security or networking zones that each VNF will be attached to. If these zones do not already exist, then they must be created when the vEPC is instantiated. The service designer will need to consider application-layer relationships (between VNFs running in different data centers) that must be put in place to make the collective system work. This could include passing parameters (e.g., credentials or IP addresses) from one VNF to another.

AUTOMATED VNF & SERVICE LIFECYCLE MANAGEMENT

Now that the NFV operations team (and the sorcerer's apprentice) is neck deep in water, it is time to throw them a life preserver (or have the wizard take control). Both the operations team and the engineering team need a common NFV management tool that can simplify service design and operations.

This management tool should complement virtual infrastructure resource managers (e.g., OSM, ONAP) and other systems, such as SDN controllers, IT DevOps tools (Ansible, Docker, Kubernetes) and legacy EMS/OSS, which will form part of a hybrid physical-virtualized environment. In a distributed software environment, the entire lifecycle of VNFs and services should, ideally, be automated.

The scope of this new tool (or magic spell) should include:

- Service design
  - Service design (ongoing to resolve any recurring performance issues)
  - VNF and service onboarding (simplified with standardized lifecycle models)
  - VNF and service testing – integration with test tools for automated testing (locally, and in a preproduction environment). Service bundles of multivendor VNFs should be tested for interoperability.
  - Release management of VNF software packages – third-party VNFs should be "wrapped" in a standardized way so they can be managed via APIs, avoiding the need for custom integration and potential errors in production.
- Operations
  - Deployment – automated programming of all the steps required to take a service from concept to reality. VNFs and services must be continually created, configured, updated, scaled, healed and migrated with no human intervention.
  - Automated VNF and service testing, monitoring and diagnostics – services are continuously monitored and, where possible, automated corrective action is taken, removing the need for manual resolution handling. If customer-service issues pass undetected by the monitoring or test systems, then the diagnostic tool should present coherent information to operations staff to reduce time to resolve and obviate the need to escalate issues to higher-tier support.
  - Optimization of the production environment to maximize performance
  - Automated healing, scaling, upgrading and migration of production VNFs and infrastructure

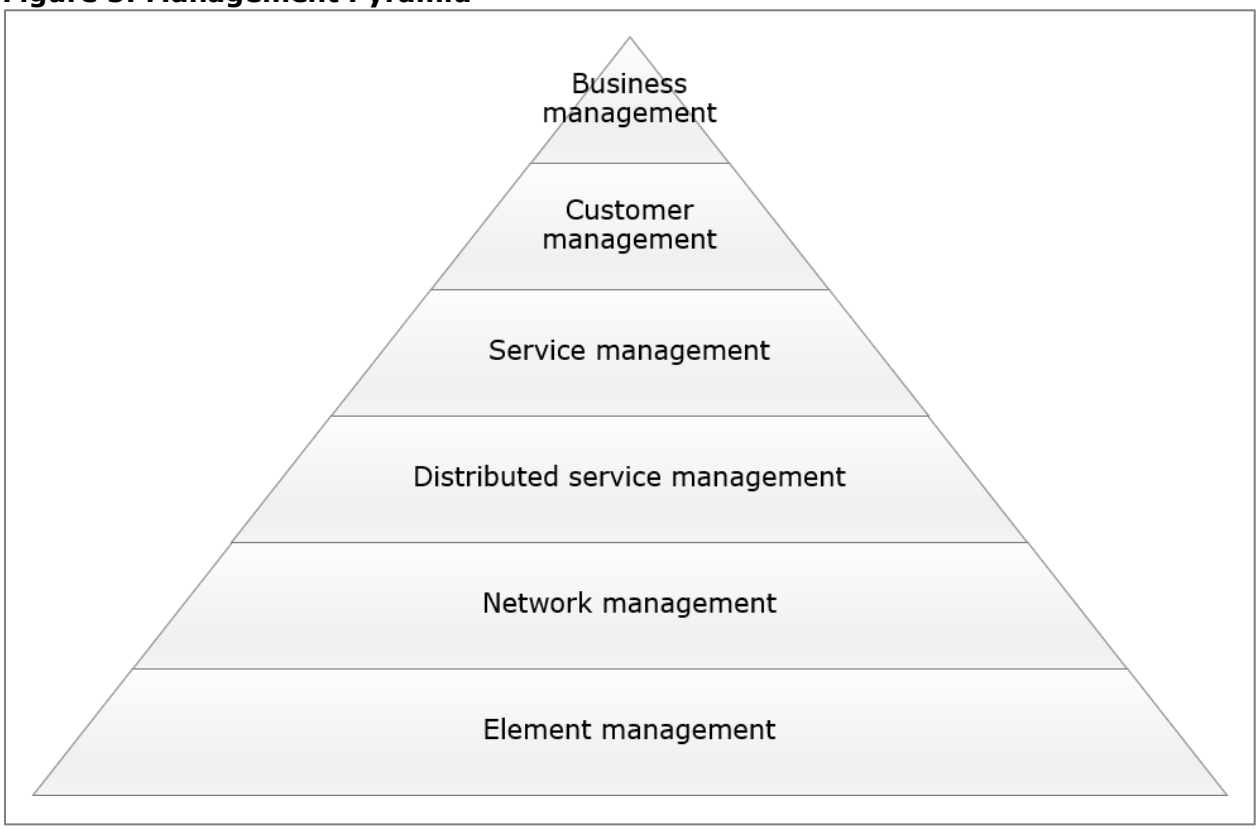
By removing the need for manual monitoring and resolution handling, operations staff can focus on more value-added activities, such as predictive maintenance. Damaged service components that are causing faults can be replaced without human interaction, or sent back to engineering for modification. Greater interaction between engineering and operations should benefit both teams, leading to better overall service delivery across design, test and deployment. And by hiding the complexity of the underlying multi-vendor infrastructure and VNFs, residing in multiple data centers, service designers can be freed from defining complex workflows for each component, and operations teams can focus on managing the system on a more holistic basis.

INVESTING FOR COST REDUCTION AND AGILITY

As Hegering, Abeck and Neumair note in Integrated Management of Networked Systems:

> "The economic success of management is primarily manifested in the top end of the pyramid [see **Figure 5**] with the lower levels acting as the 'enablers.' Generally, level N cannot be effective if level N-1 is not operating effectively. On the other hand, IT managers expect a technical management system to provide everything. They are the ones wedged between the end users and business management for whom the system is only a cost center."

Figure 5: Management Pyramid



Source: Integrated Management of Networked Systems

As **Figure 5** shows, justifying investment in new network management tools is difficult, as operations is often seen as a cost center. The investment case therefore largely hinges on the ability to take out cost or prevent an anticipated increase in cost. As NFV is rolled out, healing, upgrading and maintaining VNFs is likely to account for most of the effort (and hence cost) associated with operating a service. As we have identified in earlier sections of this paper, there is a risk that the complexity of NFV leads to spiraling operational costs.

A priori, an automated NFV DevOps tool as described in the previous section – one that can automatically heal, upgrade and migrate VNFs and services – might reduce the manual operations costs of NFV from around 30 percent of revenue to less than 10 percent. The value of that software therefore depends on the amount of revenue likely to be generated by the services it supports.

A further argument for investment in better NFV management software is the ability to reduce infrastructure costs by dynamically scaling and moving VNFs to optimize the use of infrastructure resources while ensuring customer experience is continuously delivered. Exactly how much infrastructure costs can be reduced will depend on the current level of server capacity utilization and the impact of moving VNFs from one data center to another.

There is also a softer justification for investment: business agility. By reducing the time taken to onboard and test VNFs and associated services, the engineering team will be more able to support the marketing organization in introducing niche services to target new revenue opportunities. Quantifying the revenue potential of this increased agility is clearly speculative, but we note marketing departments today express a great deal of frustration with their OSS/BSS colleagues in their lead times to support new product initiatives.

CONCLUSIONS

Six years on from the initial NFV white paper, leading CSPs have moved beyond the proof-of-concept and trial phase to widespread deployment of VNFs. The challenge they now face is how to increase the level of automation throughout the lifecycle of a service chain, from initial installation, configuration and monitoring to ongoing software updates and self-healing of VNFs. This is about moving beyond the initial orchestration phase to the full lifecycle management of VNFs and their associated services.

The next step is to have a feedback loop from operations to the service designers, in a similar way to the DevOps methodology seen in IT. NFV service design, test and deployment must be coordinated across service engineering and operations to bring new services into production with a high degree of confidence, despite the highly dynamic nature of NFV.

Although we have used the *Sorcerer's Apprentice* analogy throughout this paper, we are not suggesting that the solution to operationalizing NFV requires supernatural powers. Instead, what service designers and engineers need are DevOps and software lifecycle tools to create pre-production environments where they can test service on-boarding. At the same time, operations staff need tools to manage the ongoing maintenance of services and their component VNFs. By sharing common tools, engineering and operations can better collaborate to deliver and maintain more reliable NFV-based services.

ABOUT ACCANTO

Headquartered in Finland and operating globally, Accanto (www.accantosystems.com) provides a software platform called StratOSS™, which helps customers to simplify and automate the production of NFV services. In providing tools that unify engineering and operations, we decrease the time it takes to design, test and simplify the NFV production process, improving diagnostics – leading to smarter operations.

To learn more about Accanto's NFV solutions, see: NFV Service Lifecycle Management – Leveraging DevOps Tool Chains and Intent-Driven Operations for Increased Automation.

To receive a copy of the Accanto NFV Business Value Assessment, or for general information, please contact: info@accantosystems.com.