WHITE PAPER

Communications Service Provider
Network Functions Virtualization

Streamlining VNF on-boarding process

Learnings from over 200 VNFs on-boarded by HPE, Intel, TechMahindra and VMware

Authors

Vinay Saxena, Badri Natarajan, **HPE**

Petar Torre, Nikita Agarwal, Edel Curley, Adrian Hoban, **Intel**

Gururaj Phadnis, Sudha Narayanan, **Tech Mahindra**

Vanessa Little, Jambi Ganbar, **VMware**

Table of Contents

Executive Summary

Implementing Network Functions Virtualization (NFV) - running network functions in software, independent of any specific hardware - requires on-boarding many individual third party virtual network functions (VNFs).

HPE, Intel, Tech Mahindra and VMware have, between them, on-boarded over 200 VNFs for Communication Service Providers (CSPs). This paper consolidates lessons from their individual work.

It includes a clarification of terminology, a phased on-boarding process and details of the on-boarding methodology.

This methodology, focusing on VNFs implemented as virtual machines (VMs), will be current for most of 2017. We expect to iterate and improve this process by introducing a Common Data Model to describe VNFs in a uniform way.

1. Introduction and terminology

The transition towards Network Function Virtualization (NFV) is in its early stages. The process by which Communication Service Providers (CSPs) and their suppliers shift more and more telco workloads towards the cloud (private or even public) is an evolution rather than a big-bang type of migration. CSPs must mitigate risks, maintain their existing customer base and assess avenues where virtualization will lower cost of deployment and operations. Traditional Telco solution vendors (such as Telecom Equipment Manufacturers, TEMs) on the other hand, have years of investment in their existing code base; immense expertise in developing for their own fixed-function hardware and the benefit of a closed system in which they control the complete stack.

CSPs are finding that on-boarding each of the dozens of VNFs they need to support their revenue-generating services is taking weeks or months, and incurring more than $100k in associated costs. This is not meeting the promise of NFV – which is to deliver new and improved services faster and cheaper.

One CSP executive observed that each VNF is like a snowflake. They are unique, and licensed, deployed and managed in completely unique ways. Aside from the non-standard nature and challenges in deploying VNFs, there is a myriad of potential implementation and automation technologies to choose from.

The foundation of delivering services faster, cheaper and better is to leverage economies of scale, economies of scope, automation and faster and more complete learning curves.

This paper is intended to be used as practical guide for CSPs. It includes a methodology for on-boarding VNFs, the majority of which currently come packaged as Virtual Machines. It consolidates the experiences of four companies that have each on-boarded many VNFs. Hope is that this will assist CSPs in moving to NFV "automation" and NFV “at scale”.

By "VNF" in this paper is meant an entity defined by the vendor of that VNF, not taking literally the ETSI definition which requires at least one VNF component and a data model for VNF descriptors (VNFD). The difference is when the VNF catalog has multiple different definitions of similar entities. For example when the virtualized Enhanced Packet Core (vEPC) has various gateways that are all individual VNFs while simultaneously all gateways are one VNF.

1a – Phases in implementing VNFs

We have identified a sequence of phases in typical implementation:

- Software development by Independent Software Vendor (ISV) or Telecommunications Equipment Manufacturer (TEM), sometimes called Network Equipment Provider: The VNF software architecture is chosen and implemented by the vendor, which is typically an ISV or TEM. They develop software, optimize, describe and package to this architecture.
- Choosing Data Model to describe functions: Data Model for VNF Descriptors (VNFD) and Network Services Descriptor (NSD) are created to describe VNF attributes like deployment requirements and service chains.
- CSPs or System Integrator contracted by CSP:
  - VNF evaluation: Before any real on-boarding work is done with a VNF, CSP or their System Integrator (SI) need to evaluate the VNF against specific criteria.
  - VNF on-boarding: This is the phase where VNFs get loaded on to the virtualized platform and normally integrated with the NFVO. This is the point at which the VNF software architecture is exposed in more detail. Sometimes the software architecture is inadequate, and this is incorrectly attributed and reported as a VNF on-boarding problem.
  - VNF performance characterization: After functional on-boarding, this phase tests the performance characteristics of VNFs by applying load.
  - VNF operationalization: Full integration in the CSP environment, lifecycle management, service assurance and other aspects for fully operationalizing the VNF into the CSP's services.

1b – The case for a streamlined VNF on-boarding methodology

There are two primary VNF on-boarding challenges which work to decrease time-to-deployment and the associated costs:

- Complexity of environment: Minimize the effort for VNFs to be tested, optimized and certified for every CSP's individual specific environment. Also needs to be considered specifics related to the integrator, the hardware vendor and the software environment.
- Manual process: The current VNF on-boarding, instantiation and maintenance processes are highly manual. This is very time consuming. Moreover, each process needs to be specifically tailored for CSP, Virtual Infrastructure Manager (VIM), NFV Infrastructure (NFVI), and VNF types.

The result of the factors above, and the necessarily labor-intensive on-boarding process, is a slowing of VNF innovation and a reduction of the pace at which more and better VNFs will become available.

What is needed is a more streamlined, automated and adaptable way to on-board VNFs. CSPs are increasingly deeply engaged in digital transformation – enabled by a cloud-like service delivery model. Those services are built on a mixture of existing physical and new virtual network functions. So the current complex, slow and expensive VNF on-boarding situation will significantly impact the broader business goals and direction of the CSP unless it is improved.

2. Before on-boarding

2.1. VNF software architecture: Making it cloud ready, resilient and decomposed

The initial definition of the NFV is to separate the network function software image from the appliance hardware, then virtualize that software image and run it on a virtualized environment shared between various network functions.

This can be implemented for simpler and for more complex network functions. Nevertheless, for more complex VNFs that were initially implemented on multiple blades in a chassis this approach fully replicates the physical layout of those blades and chassis along with any undesired dependencies. To that extent it lost some of the benefits of NFV.

The ETSI NFV Software Architecture team has documented various VNF design patterns and best practices[1]. These include:

- VNF internal structure (separating VNF Component functionalities in different VMs)
- VNF Instantiation (1:1 for failover or 1:many for scale out)
- VNF Component state (how to externalize the state from the load balancer and application logic layers into the last layer holding all state information, such as user or session data)
- VNF Load Balancing models (internal in VNF, using another load balancing VNF, calling client understanding VNF cluster, or part of data center networking infrastructure).

The design patterns are applied, for example, on vEPC, the load balancing and application logic become stateless. Separating the control and data planes allows for independent scaling of those VMs or for separating them in different data centers.

This way VNFs can be re-architected to be:

- Cloud Ready (scale out and in on virtualized infrastructure)
- Resilient (will keep delivering critical KPI even if the underlying virtualized infrastructure provides varying level of compute resources, or if the network has delays)
- Decomposed (separating functionalities into smaller, lighter VMs).

2.2. VNF evaluation

After the ISV/TEM completes the VNF software work, and before the CSP/SI starts on-boarding it, the CSP needs to go through an evaluation phase to select the VNF. After applying various criteria (documented in evaluation "checklists") they will select some VNFs to use. Evaluation documents will typically include commercial, procedural and technical questions to determine VNF characteristics and requirements on target environments.

3. On-boarding methodology

This section describes the on-boarding methodology from the overall concept to instantiation, scale, and updates/upgrades based on OpenStack/KVM and VMware vCloud NFV environments in CSP data centers.

A similar methodology can be used to on board a VNF to virtualized platforms for on-premise virtualized Enterprise Customers Premises Equipment (vECPE) or Edge Computing (for example Multi-access Edge Computing, MEC).

On-boarding has three stages:

- Create VNF Descriptors (VNFD) as templates - in order to for example instantiate/terminate, scale, update/upgrade
- Create package (templates and images - for dataplane functions or other demanding applications using Enhanced Platform Awareness concept to detect and configure virtualized infrastructure).
- Validate and load into the service catalog

Additionally, CSPs should plan for VNF lifecycle management at the time of on-boarding.

On-boarding can be completed either in the NFV platform vendors' ecosystem programs for VNF partners, or in a CSP's lab. Several very broad ecosystem programs exist to do just that, for example HPE OpenNFV Solution program[2], Tech Mahindra VNF-Xchange program[3] or VMware Ready for NFV program[4].

These programs test functional interoperability between a VNF and the NFVI/VIM of choice. Such certification programs run the VNF through a battery of tests to also asses its virtualization readiness, posing such questions as: does it benefit from dynamic allocation of resources; does it take advantage of high performance data-plane optimization capabilities of the platform; can it use some of the platform specific operations for typical virtualized environment activities such as providing high availability for the VNF, VNF backup capabilities, flexible configuration etc.

After the on-boarding phase, to get VNFs into service they also need to be fully integrated into the CSP's specific environment.

On-boarding the VNF in the CSP network involves the following stages:

- VNF vendor supplies the specific VNF requirements (like today for example for OpenStack from Heat templates and images)
- Infrastructure planning
- Certifying VNF in the staging environment. From the perspective of on-boarding a VNF, this is about behavioral characteristics and the impact on other network elements rather than the functional aspects of the VNF
- Deploying into production, monitoring, and scaling up and scaling down the instances

The process of on-boarding a VNF starts from receiving the Heat template and images from the VNF vendors. The first step is to validate compatibility and perform basic sanity testing. This is followed by setting up the required infrastructure (CPUs, memory, network setup) and then spawning the VM.

Next, the VNF goes through system testing in the staging environment. The VNF image is validated against its expected behavior in the integrated network architecture. IT validates images for malware, virus protection and other security functions, and approves them for production deployment. The approval includes checking the metadata, integration into the orchestration and management platform, and testing for VM bring-up and bring-down at runtime.

When testing is finished, the CSP submits and validates a Network Service Descriptor (NSD), including any related VNFFGD and VLD. Upon successful completion the Network Service Descriptor is stored in the NS catalog, and can be used for Network Service lifecycle management.

After a VNF is on-boarded, VNF Management functions are responsible for the VNF's lifecycle management including operations such as:

- Instantiating the VNF (creating a VNF using the VNF on-boarding artifacts)
- Scaling the VNF (increasing or reducing the capacity of the VNF)
- Updating and/or upgrading the VNF (supporting VNF software changes of various degrees of complexity)
- Terminating the VNF (releasing the VNF-associated NFVI resources and returning them to the NFVI resource pool)

The deployment and operational behavior requirements of each VNF are captured in a VNFD, and stored during the VNF on-boarding process in a catalog, for future use. The VNFD describes the attributes and requirements necessary to realize such a VNF and captures, in an abstracted manner, the requirements to manage its lifecycle.

There are many open source tools available from OpenStack, such as Glance, Nova, Neutron and Cinder, or Nagios and fully featured dashboards like Horizon for managing the entire lifecycle of a VNF.

4. After on-boarding: Integration into the CSP environment

Once a VNF is on-boarded it needs to be integrated into the existing CSP NFV environment to become part of the NFV services.

Typically, this means integration with an Operations Support System (OSS). This can constitute building a new NFV OSS environment designed for handling virtualized elements in a fully automated way for lifecycle management or FCAPS (Fault, Configuration, Accounting, Performance, and Security). It could also mean integrating with a complex existing OSS environment, grown over many years, and based on legacy technology.

4.1 The challenge of integrating NFV

NFV adoption has a significant impact on a CSP's OSS system estate.

- It creates new challenges in scalability, elasticity, agility, real-time demand and automation within the infrastructure layer
- NFV adoption makes workflow execution even more complex. Services are increasingly dynamic, personalized, contextualized (with reference to specific NFVI) and adaptable.
- While NFV installations are still ramping up, CSPs are expected to continue to invest in the physical network and its upstream IT ecosystem (NMS, OSS)

This creates a demand for an overarching layer to manage the coexistence of the Physical Network Functions (PNFs) and Virtual Network Functions (VNFs) as CSPs are looking at a mix of both traditional and virtual infrastructure as they fully scale-up to Software Defined Networking (SDN)/NFV.

These factors require that CSPs align rather than transform their existing OSS with their MANO design strategy.

The way to circumvent these challenges for CSPs is by providing an overarching framework that:

- Mediates and controls information flows between applications in the physical and virtualized worlds
- Moves legacy application transactions to near real-time states
- Paves the way for service based deployment strategies rather than using a network deployment approach

With the very dynamic and real-time nature of new age network services, monitoring and management solutions themselves need to run in a virtualized environment. This facilitates automatic load balancing under peak conditions.

4.2 A possible approach

One approach to this issue is to build an open-source platform-based solution that adheres to decomposed light VMs. This platform enables the rapid creation, deployment and management of microservices. It accelerates the refactoring and rationalization of legacy applications at optimized cost. The key word is "refactoring". Our suggested approach does not call for a 'big bang' transformation of the existing OSS landscape.

This platform also provides an API gateway which helps publish and expose functional capabilities from disparate legacy back-end systems and new decomposed applications allowing unified and federated operations across the IT ecosystem.

Key benefits include:

- Legacy application refactoring: accelerated micro services development, deployment and management
- Federation and orchestration: accelerated API development, deployment and management
- Accelerated development: industry best practices in the form of integrated project archetypes and templates. Going towards Agile and later an integrated DevOps approach.
- Vendor independence: a vendor agnostic open source framework for automated testing and deployment - Continuous Integration (CI)/Continuous Development (CD)

5. Other related topics

5.1. How ONAP and Open Source MANO help with the Common Data Model and reference implementation

ONAP and Open Source MANO are open source communities developing implementations of the NFV and service orchestration layers.

Generally such communities follow the ETSI NFV Information Model and as basis to define data models are considering standards like OASIS TOSCA (Organization for the Advancement of Structured Information Standards, Topology and Orchestration Specification for Cloud Applications) and can be supported by YANG.

Industry interoperability and convergence could be facilitated by TOSCA/YANG agreement on VNFs so that VNF vendors develop their VNF package once, and each orchestrator knows what to do with it.

A Common Data Model is needed to describe those VNFs in a uniform way, and then any NFV Orchestration (NFVO) layer can import it and use it. Such an NFVO can be ETSI NFV MANO compatible or not, like orchestrators for public cloud services.

Resource orchestration refers to a set of operations for the allocation of compute, network and storage resources for the deployment of VNFs and their interconnection.

Service Orchestration (SO) is the set of operations for the automatic configuration of PNFs and VNFs, networks and traffic forwarding between PNFs and VNFs in a coordinated way. Configuration is driven by stimulus coming from the OSS, Element Manager of the VNF and infrastructure or VIM.

5.2. Options for Performance Characterization

As a last step of VNF on-boarding, or as separate step, many CSPs will want to performance characterize VNFs to quantify the platform resources they consume in a production environment. A similar methodology and tools can be used in preproduction environments by VNF vendors to provide input for CSP/SI VNF evaluation.

This VNF characterization is challenging because of VNF resource controls (rate control, scale-out orchestration overheads), many virtualization options (VM sizing, in a number of virtual CPUs or memory; network virtualization with GRE or VXLAN; usage of DPDK or SR-IOV) and data plane dependency on hardware configurations (Enhanced Platform Awareness type of workload placement, aware of the physical architecture of the volume servers).

Currently the CSP industry is lacking a comprehensive established industry standards that would facilitate a straightforward comparison between VNF vendors (as is the case for mature established workloads in the IT world). Such a comparison is complicated by the proliferation of network segments, use cases, VNFs and load modeling customer behavior (for example call models).

To support performance characterization, vendor comparison and capacity planning, within OPNFV Intel is leading the definition and initial implementation of a Network Services Benchmark[5] which will include an open source testing framework for various loads and several prototype VNFs.

This initiative aims to deliver a common set of benchmarks, open source tools, test suites and reference virtual network functions. The NFV Network Services Benchmark (NSB) Test harness is a test framework extending the OPNFV Yardstick solution with a Network Service testing capability. The NFV NSB enhances the Yardstick test framework with the following capabilities:

- NFV Infrastructure benchmarking of Network Service level including bare-metal and fully virtualized (SR-IOV and OVS based) infrastructures
- Definition of Network Service topology including control-plane and data-plane interface
- Benchmarking VNFs from different vendors
- Generalized VNF models written in Python supporting any type of NFV infrastructure defining instantiation/ termination and configuration
- Generalized real traffic profiles
- Generalized models for traffic generators supporting different vendors

Every NFV solution vendor and network operator requires optimizations for the specific environment where network functions will be implemented. The performance can be different where the same virtual network function is used in a pure software environment without any hardware accelerations and when this VNF uses specific Enhanced Platform Awareness (EPA) features and other accelerators.

The NFV NSB Test harness helps in characterizing VNFs by measuring the network, VNF and NFVI KPIs in bare metal, standalone virtualized, and managed virtualized environments.

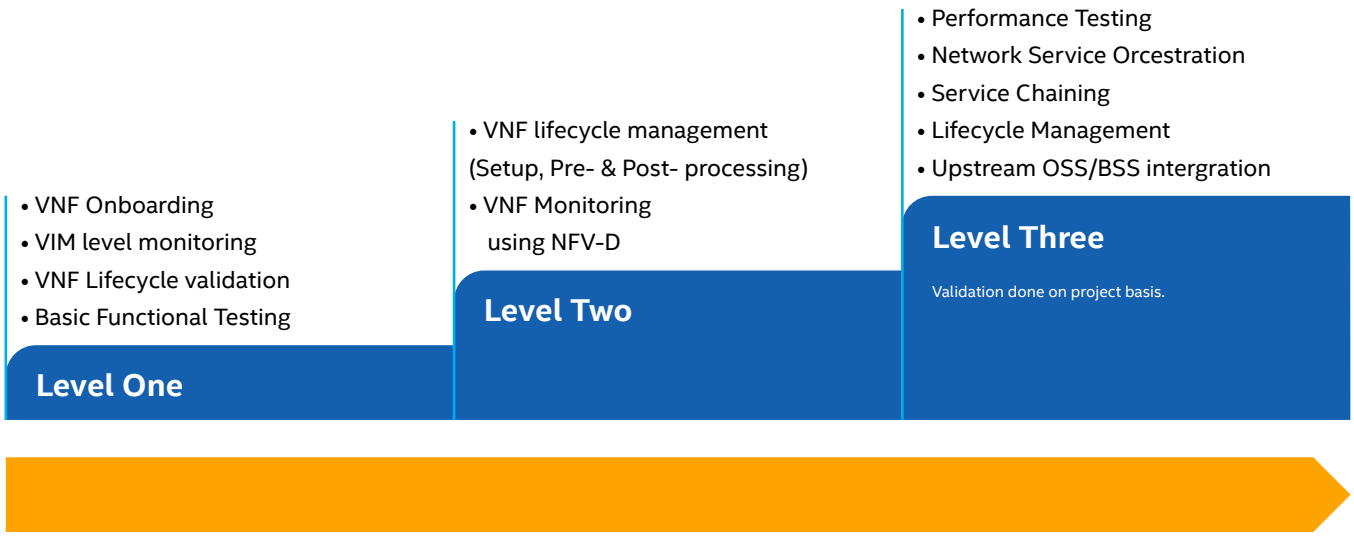

Figure 1: Example of levels in VNF on-boarding ecosystem program

[1] Information Modeling (IM) is modeling managed objects at a conceptual level, independent of any specific implementation or protocol details. Data Models (DM) are programmatic definitions at lower level with many details including protocol-specific constructs.

5.3. Towards Cloud Native VNFs

Most current NFV use cases involve virtualizing applications from existing carrier network appliances. As we have discussed, at present most are running on VMs. However, we are encountering substantial customer interest in replicating in the CSP environment principles from the cloud and IT worlds to make VNFs more "cloud native".

The Cloud Native Computing Foundation[6] defined cloud native systems as:

- Container packaged: Running applications and processes in software containers as an isolated unit of application deployment, and as a mechanism to achieve high levels of resource isolation. This improves overall developer experience, fosters code and component reuse, and simplifies operations for cloud native applications.
- Dynamically managed: Actively scheduled and actively managed by a central orchestrating process. This radically improves machine efficiency and resource utilization while reducing the cost associated with maintenance and operations.
- Microservices oriented: Loosely coupled with dependencies explicitly described (e.g. through service endpoints). This significantly increases the overall agility and maintainability of applications. The foundation will shape the evolution of the technology to advance the state of the art for application management, and to make the technology ubiquitous and easily available through reliable interfaces.

This definition of cloud native systems is fine for regular web scale environments of public cloud services and accurately describes the evolution of IT environments. For CSPs, though, there are additional considerations in building cloud native VNFs.

One such consideration is to look at full service and VNF lifecycle management, ensuring that component failure (including potentially offloading hardware failure) does not lead to VNF or service failure. It is essential to ensure that there is no negative impact on service level agreements (SLAs) and that FCAPS (Fault, Configuration, Accounting, Performance, Security) is handled correctly.

All of the above requirements for VNFs can be implemented with small, light virtual machines from cloud and IT environments.

Find out more about cloud native architectures for VNFs in the webinar referenced[7] at the end of this document, where Affirmed Networks explains how they created them for several VNFs including vEPC.

5.4. What will change with NFV in Containers?

OS Containers allow higher density of functions per server with faster instantiation times. This is expected to help with many apps that can be decomposed into microservices.

Everything begins with developing new cloud native web-scale applications where they are complementary Cloud Service Providers' DevOps. Outside of those environments Containers have these three challenges:

1. Commercial motivation of VNF ISVs: Most VNF vendors today packaged their VNFs in VMs, looking for commercial success of those VMs before investing in re-architecting and re-packaging them. Also those VNF vendors look at platform ISVs to first make clear choices about which container management environments and engines will be on their commercial path to market.
2. VNF workload characteristics (data plane, transcoding…): Containers still need to be proven for long-running data plane heavy-network-consuming or transcoding heavy-CPU-consuming apps. Data Plane Development Kit (DPDK) Poll Mode Driver requires is own core, so one server cannot run hundreds of such containers. Transcoding given amount will need appropriate CPU cycles and decomposing it into multiple containers will not reduce that number of CPU cycles.
3. VNF packaging, management/lifecycle: The current VNF lifecycle is defined around VMs with clear abstraction from the hypervisor below. With containers this changes, creating dependencies between VNFs and the host OS that require different operational environments, which is easier to deal with in DevOps environments

6. References, links

1. ETSI NFV VNF Architecture document: www.etsi.org/deliver/etsi_gs/NFV-SWA/001_099/001/01.01.01_60/gs_NFV-SWA001v010101p.pdf
2. HPE NFV: www.hpe.com/us/en/solutions/telecom-digital-infrastructure-nfv, OpenNFV Partner Program: www.hpe.com/us/en/solutions/telecom-nfv-partners, OpenNFV Solution Portal: hpenfv.com, blog describing Solution Portal: community.hpe.com/t5/Telecom-IQ/OpenNFV-Solution-Portal-A-practical-approach-leveraging-the/ba-p/6827112, VITAL tool
3. Tech Mahindra VNF-Xchange: www.vnfxchange.org
4. VMware NFV: www.vmware.com/solutions/industry/telco, Solution Exchange for NFV: solutionexchange.vmware.com/network-function-virtualization
5. Network Services Benchmark: http://docs.opnfv.org/en/latest/submodules/yardstick/docs/testing/user/userguide/13-nsb-overview.html
6. Cloud Native Computing Foundation on Cloud Native systems: www.cncf.io/about/charter/
7. December 2016: Intel Affirmed Networks webinar on “The Journey towards Cloud Native Network Functions”: www.brighttalk.com/webcast/12229/236357