IP Video, Cloud and NFV

NOKIA
The initial focus of network functions virtualization (NFV) was primarily to leverage NFV to create efficiency, both in operations (OPEX) and capital expenditure (CAPEX), by implementing a common, shared infrastructure platform for network services, with consistent operational tooling. As service providers and vendors have matured their NFV deployment and strategies, service providers have become increasingly motivated to leverage NFV for more complex use cases and scenarios. In order to realize increased service agility and ultimately benefit their organizations’ profitability and competitiveness, service providers are extending their NFV architectures to include the automation of common tasks such as auto-deployment, auto-scaling and self-healing.
As service providers’ NFV deployments have evolved and vendors have developed solutions to deploy within these new environments, both parties recognized the potential OpenStack has to provide the platform to provide Infrastructure-as-a-Service (IaaS) for NFV. Stimulated by NFV strategies, service providers have deployed OpenStack at scale, and vendors have provided Virtualized Network Functions (VNFs) to deploy within these private cloud platforms. Additionally, both parties have worked closely with the OpenStack community to enhance and extend OpenStack for NFV use cases.

This paper explores cloud and NFV, the importance of OpenStack and how Nokia IP Video products have been extended to include new components and new functionality, enabling the harmonious deployment of Nokia IP Video applications within public and private clouds and extensibility to support NFV use cases such as auto-deployment, auto-scaling and self-healing.
NFV introduction

What is Network Functions Virtualization (NFV)?

NFV is a new way to describe, create, and operate network services by replacing dedicated network appliances with shared infrastructure, software and automation. NFV moves service creation dependencies away from the need to deploy physical hardware components, which often lack flexibility and are proprietary, expensive and may result in underutilization of resources and assets.

In an NFV environment, specific network functions are provided by software components running on top of a common physical infrastructure; these components are referred to as Virtual Network Functions (VNF). VNFs may be deployed in this infrastructure as virtual machines (VMs), containers or bare metal, depending on the required workload and capabilities of the target deployment environment. Examples of VNFs are firewalls, content delivery network (CDN) caches and mobile components such as packet data network gateway (PGW).

Within an NFV deployment VNFs runs on general purpose COTS servers, within virtual machines, containers or bare metal and the overall environment is deployed and managed using standardized and open APIs, which are typically implemented using open source software.

The aim of NFV is to enable efficiencies to service providers by enabling automation. Other benefits of NFV include:

- Automation: Programmatic provisioning, capacity augmentation and recovery
- Speed: Faster time to market for new services and capabilities
- Efficiency: Unified physical infrastructure, COTS hardware (OPEX and CAPEX)
“IHS (NYSE: IHS) today released excerpts from its IHS Infonetics NFV Hardware, Software, and Services report, which forecasts the global network functions virtualization (NFV) hardware, software and services market to reach $11.6 billion in 2019, up from $2.3 billion in 2015.”

Campbell, CALIFORNIA (July 20, 2015)

“NFV represents operators’ shift from a hardware focus to software focus, and our forecasts show this. We believe NFV software will comprise over 80 percent of the $11.6 billion total NFV revenue in 2019.”

“The software is always a much larger investment than the server, storage and switch hardware, representing about $4 of every $5 spent on NFV.”

Michael Howard, Senior Research Director for carrier networks at IHS
NFV market highlights

Revenue from outsourced services for NFV projects is projected to grow at a 71 percent compound annual growth rate (CAGR) from 2014 to 2019.

Service providers are still early in the long-term, 10- to 15-year transformation to virtualized networks.

Revenue from software-only video content delivery network (CDN) functions for managing and distributing data is forecast by IHS to grow 30-fold from 2015 to 2019.

Additional information about NFV

NFV can be explored in greater detail by reading the following background documents:

- ETSI NFV specifications: http://www.etsi.org/technologies-clusters/technologies/nfv

Today OpenStack has largely been accepted by operators as a foundational component within NFV deployments. OpenStack provides the Infrastructure-as-a-Service platform required by NFV and is used for the deployment, orchestration and management of VNFs and their associated underlying infrastructure. OpenStack has become the go-to technology when building the functionality required of the Virtual Infrastructure Manager (VIM), as defined by the ETSI NFV model.

OpenStack enables heterogeneous management of datacenters, with common security and identity services, API and user interfaces. Leveraging OpenStack also enables operators to flexibly architect NFV platforms themselves, by using the open and pluggable architecture of OpenStack and without being tied to deploying components they see as unnecessary within their context.

---

2 https://www.openstack.org/software/
3 http://www.etsi.org/technologies-clusters/technologies/nfv
“Network Functions Virtualization (NFV) is now synonymous with OpenStack. When people say NFV, there is an implication that they are talking about OpenStack.”

Within the ETSI NFV model OpenStack is leveraged to provide the functionality of the VIM. The VIM is part of the Management and Orchestration (MANO) layer and is responsible for the management of the NFV Infrastructure (NFVI), which is responsible for managing and allocating resources, such as compute and storage resources. OpenStack provides abstraction between the NFVI and higher-layer systems (such as Orchestrators and OSS) by exposing datacenter resources over a standard, consistent set of API. The following OpenStack interfaces are examples of interfaces which are key to NFV:

- **Neutron networking**
  Provides access to virtual networking

- **Cinder block storage**
  Provides access to storage resources and the ability to attach storage devices (block storage) to virtual machines

- **Nova compute**
  Provides access to compute resources and the ability to deploy and configure virtual machines

A more in-depth overview of OpenStack components and interfaces can be found here: https://www.openstack.org/software/

The above interfaces (in combination with other OpenStack interfaces and projects) provide the core building-blocks for operators to build private clouds and underpin their ability to apply OpenStack to NFV. However, operators have broad and complex requirements, across various functional and non-functional areas. The breadth of operator requirements for NFV has led to a substantial drive to enhance OpenStack, specifically for NFV.

4 Quote Reference: https://www.openstack.org/telecoms-and-nfv/
5 https://www.etsi.org/deliver/etsi_gs/NFV-MAN/001_099/001/01.01.01_60/gs_nfv-man001v010101p.pdf
As operators have applied OpenStack to their NFV architectures, it has led to a drive to develop OpenStack further and further to meet the needs of NFV:

**Flexible deployment models**
Before the introduction of NFV and private cloud platforms, operator network services were almost exclusively based upon the deployment of hardware-based network appliances, for functionality and capacity. By introducing cloud and NFV platforms, the dependency between hardware deployment and software-based functionality is decoupled. In order to support this deployment model, VNF vendors are required to provide software-based solutions, without tightly-coupled hardware dependencies, which support a greater range of deployment scenarios ranging from virtual machines, containers or (should the workload require it) bare metal.

**Life-cycle management**
Management of the life-cycle of VNFs refers to the ability to manage operational workflows such as the deployment and upgrade of VNFs. Moving toward NFV-style architectures comes hand-in-hand with the expectation that common operational workflows may be managed and also automated by leveraging common tooling and infrastructure. OpenStack provides common tooling and APIs to manage the life-cycle of VNFs, e.g., Glance provides a common image repository for VNF images and supports the back-up and cloning of image snapshots, Horizon provides a common UI to carry out operational tasks, and OpenStack also supports templated workflows over Heat.

**Scaling and resilience**
Beyond providing common workflows and interfaces for deployment and provisioning, operators also wish to enhance their NFV platforms by automating the life-cycle of applications. In particular, the support for automated scaling and automated self-healing are drivers for are key drivers for NFV deployment. In order to support deeper automation, where scaling or healing is automatically triggered based on feedback or metrics from the VNF, a deeper level of integration between the VNF(s) and the NFV MANO is required. For example, to automatically scale based on metrics such as load, the NFV MANO needs a method of obtaining a view of load. Furthermore it needs a method of understanding what that load means within the context of that specific VNF. Within the ETSI NFV model the VNF Manager (VNFM) provides the integration point to host these kind of interfaces between the VNF(s) and the NFV MANO. Within OpenStack, projects are addressing the complex requirements of the VNFM, for example the Tacker project. Tacker is an open-source VNFM which provides interfaces between the MANO and the VNFs through a pluggable-driver architecture, where each VNF provider may provide their own drivers to support monitoring of VNFs and configuration of VNFs. Tacker provides a north-bound TOSCA interface, for describing and modeling VNF applications, and Tacker also provides a south-bound HEAT interface, to drive OpenStack resource provisioning, enabling functions such as automated scaling.

**Performance**
The performance requirements of VNF applications can range from high-volume packet-processing to high-volume of delivery of data from storage, or a mixture of both. When combined with the need for high-density deployments, these performance requirements typically dictate that applications/resources can access the hardware resources directly, to accelerate processing and achieve an acceptable density of performance. Technologies such as SR-IOV7 and DPDK8 are exposed by Neutron, and Nova allows the operator to control application to CPU allocation through the support of NUMA9 topology awareness. By supporting the control of mapping of appropriate hardware resources to application resources and allowing access to hardware acceleration technologies, OpenStack provides allows operators to provide high-density and high-performance VNFs.

---

Nokia IP Video applications are architected to meet the requirements of operator NFV deployments. Whether based on OpenStack, public cloud platforms or hybrid deployment models, Nokia IP Video applications are agnostic to the under-laying infrastructure, enabling flexible and future-proof deployment options. Nokia IP Video applications deal with the dynamism and elasticity of cloud environments by leveraging open-source solutions for configuration, deployment and scale-out storage, enabling the automation of common workflows.
Nokia IP Video applications are provided as pre-packaged VNFs, running on open-source OS (CentOS) and are architected to run harmoniously within cloud platforms such as OpenStack.

<table>
<thead>
<tr>
<th>Delivery VNF</th>
<th>Ingest VNF</th>
<th>Distributed Intelligence VNFs</th>
</tr>
</thead>
<tbody>
<tr>
<td>HTTP Delivery Application (DA)</td>
<td>Origin Application (OA)</td>
<td>Video Recording Manager (VRM)</td>
</tr>
<tr>
<td>RTSP Delivery Application (DA)</td>
<td>JITX (Just-in-time-transcoding)</td>
<td>Video Personalization Broker</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Reporting</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Network Abstraction Engine (NAE)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Request Router (RR)</td>
</tr>
</tbody>
</table>

**Video Cloud: Applications**

**IP Video Applications Unified Management system**
- Unified Management Console
- Unified API
- Database

**Distributed Intelligence Virtualised Network Functions (NFV)**
- Video Recording Manager
- Video Personalization Manager
- Reporting Engine
- Network Abstraction Engine
- Request Router

**Distributed ingest**
- Object store
- JITX
- Origin

**Distributed delivery**
- Delivery cache

**NFVI**
- Virtual compute
- Virtual storage
- Virtual networking

**Virtualization infrastructure manager**
- Amazon Web Services
- OpenStack
- Microsoft Azure
Use cases

Cloud and NFV enablers operators to take a more flexible and automated approach to managing network services. By leveraging cloud platforms and NFV MANO integration, operators are able to automate the management of common operational workflows, in doing so realizing greater agility of service enablement and efficiency of operations. Nokia IP Video applications include interfaces which provide NFV MANO layer integration, enabling a range of use cases.

Self-healing

Within the cloud, individual components are not intended to provide the “5 nines” level of redundancy which are typically associated with operator networks. Cloud application redundancy is provided by architecting the system-as-a-whole to dynamically react to failures and scaling requirements.

Nokia IP Video applications are architected to deal with failure scenarios at two levels: 1) internal application-layer load-balancing dynamically re-routes video traffic away from failed or heavily-utilized system components, automatically preventing system failures from effecting user experiences, and 2) at an infrastructure level by exposing interfaces to VNFM, enabling external MANO systems to detect application failures and respond accordingly. By leveraging the VNFM to detect application failures, operators are free to define flexible policies, to define how application failures should be responded to, e.g., policy may be based on pre-defined application topology and rules defined within TOSCA and/or HEAT templates. By exposing metrics and alarms over Monitoring API and supporting automated configuration, Nokia IP Video applications can deal with automated self-healing and interim service protection using application-layer load-balancing.

Network based self-healing

1. Cache health is monitored by the Unified Management System via Monitoring API.
2. In the case of rapid end user request increase where there are limited caching and delivery resources available in a customer solution, cache can become overwhelmed and struggle to serve the high number of requests.
3. The cache begins to reach its capacity threshold.
4. The UMS identifies the demand on the busy cache and uses routing behavior to allow other caches deployed in the network that would usually not be chosen to serve additional request.
5. This then allows the traffic on the video solution to be more widely distributed sacrificing latency to maintaining the delivery capacity of the solution.
**NFV based self-healing**

1. The VNFM monitors Nokia IP Video applications via the Monitoring API, exposed by both the UMS and individual VNFs.

2. The UMS detects a failure of a CDN cache and notifies the VNFM of the failure over the Monitoring API(s) and the resultant impact to capacity via the Load API.

3. The VNFM can optionally act upon the failure notifications, i.e., policy (expressed in templates such as TOSCA) can define that a new cache is deployed should a failure be detected.

4. A new cache is deployed and automatically configured by the UMS.
Auto-scaling
Prior to NFV scaling, video services for peaks in demand, such as popular content and unexpected high volumes of traffic required that the operator deploy hardware-capacity which scaled to the highest predictable peak. Today operators are taking a more efficient approach. NFV and cloud technologies allow applications to dynamically react to capacity demands and where necessary deploy new capacity only when and where it is required.
1. The unified Management System is constantly engaged with edge caches that are deployed within the customer network. These caches provide data on their load and ability to continue serving requests for content from end user clients.

2. In the event of an unexpected spike in video demand, for example a breaking news story or a live sporting event, the deployed caches will soon reach capacity and require additional resources to deliver the higher load and maintain the overall customer experience.

3. The Nokia UMS exposes load-levels and metrics over API(s) to the VNFM, enabling the VNFM to trigger automated deployment of new video caches, either based upon thresholds or by predicting peaks based on trend.

4. Caches are then able to be deployed within the solution to maintain the delivery performance of the solution and grow to meet the additional requests from end users. As requests reduce, additional capacity thresholds trigger caches to be "spun down" as they are no longer required.

Nokia IP Video applications support auto-scaling scenarios within operator NFV platforms, where the scaling decisions may be made by operators’ own MANO-layer systems, such as the VNFM. The Nokia Unified Management System and associated VNFs expose a set of Monitoring API, which provide external systems with a view of load across all VNFs and the video workloads they support, i.e., video delivery load, video ingest load, transcoding load, etc. Additionally, the Monitoring API provides not only a global view of load but also a view of load within specific geographic locations. Exposing load-metrics to MANO systems enables the automation of capacity-related workflows. For example, based on a scenario where there is a high load of video traffic within a specific location, the MANO will be informed of the high-load and may then automatically deploy additional CDN capacity, by launching additional virtual machines (using OpenStack).

Once capacity is deployed, the Nokia Unified Management System automatically registers new application resources (e.g. CDN DA) post-booting. Once registered, configuration is automatically applied to new resources by Automatic Application Configuration Management.
Hybrid cloud

Today OpenStack is the go-to VIM for NFV deployments. However, as NFV evolves operators will look to other cloud platforms to provide this function, either in addition to or in place of OpenStack. Technology such as Tacker\(^\text{10}\) is evolving to consider extending beyond OpenStack, and existing technologies such as Terraform\(^\text{11}\) already provides operators with the ability to abstract between OpenStack and other cloud platforms, such as AWS\(^\text{12}\), Azure\(^\text{13}\) and Rackspace\(^\text{14}\), enabling operators to express infrastructure-as-code consistently across a range of cloud platforms.

By leveraging multiple cloud platforms, operators are able to enhance their IP Video services by:

- Extend reach beyond their network and cloud footprint
- Efficiently react to high demands by overloading capacity to 3rd party clouds
- Efficiently recover from disasters by leveraging 3rd party clouds

Nokia IP Video applications are designed to be agnostic of the underlying cloud platform. By leveraging open-source software and standardized tooling, the applications may be deployed on all major cloud platforms, inclusive of OpenStack, AWS, Azure and Rackspace.

---

\(^\text{10}\) [https://wiki.openstack.org/wiki/Tacker](https://wiki.openstack.org/wiki/Tacker)
\(^\text{11}\) [https://www.terraform.io/](https://www.terraform.io/)
\(^\text{12}\) [https://aws.amazon.com/](https://aws.amazon.com/)
\(^\text{13}\) [https://azure.microsoft.com/en-gb/](https://azure.microsoft.com/en-gb/)
\(^\text{14}\) [http://www.rackspace.co.uk/openstack](http://www.rackspace.co.uk/openstack)
Nokia IP Video: Building blocks for cloud and NFV

Nokia IP Video applications are architected with the needs of evolving cloud and NFV deployment in mind. Based upon a platform of loosely-coupled building blocks and open-source technology components, Nokia IP Video applications can be deployed within a range of environments, from bare-metal hardware appliances to cloud-based virtual machines. Regardless of the deployment environment, the same architecture can be applied, achieving a consistent platform that can be applied to a broad range of video-workloads and automated use cases.

Open-source software is at the heart of all Nokia IP Video components supporting cloud and NFV deployment.

Reference platform
All Nokia IP Video applications are certified on top of a recommended reference platform. This platform is designed and tested to provide the highest level of performance, redundancy and scalability - on top of a tested and supported infrastructure environment. The reference platform is a flexible environment where, depending on the targeted deployment model, varying levels of the environment can be provided by Nokia, i.e., for dedicated hardware deployments, high-performance appliances are available and for cloud and NFV deployments software-only is available.

- Bare-metal deployment reference platform:
  - HP Gen 9 Hardware:
    - HP Proliant Gen9 DL360
    - HP Proliant Gen9 DL380
    - HP Apollo Gen9 4200
    - HP Apollo Gen9 4510
- Virtualization support: KVM, VMWare ESX 5.1 update 1 or later
- OpenStack

For cloud and NFV deployments, applications are available as pre-built cloud images (QCOW2), compatible with OpenStack15.

Nokia IP Video applications support a range of flexible deployment options, ranging from virtualized deployment within NFV environments, private/public cloud deployment and dedicated hardware appliances. Regardless of the deployment model, the applications share a consistent codebase, common tooling and the same control-plane system (Unified Management System). This allows operators to choose the deployment model, with the flexibility to apply the right solution to the workload required - while allowing the mixing of cloud deployment with dedicated hardware deployment, should it make sense for the desired workload.

15 https://www.openstack.org/
Automated cloud bootstrapping

Nokia IP Video applications are provided as pre-built cloud images and include cloud bootstrapping tooling compatible with all major cloud environments, including OpenStack and public clouds. Based on open-source software projects, such as cloud-init\(^{16}\), and integrated into pre-built cloud images, Automated Cloud Bootstrapping can be flexibly applied to a range of post-deployment use cases to enable automated image customization and configuration. In addition to image customization and configuration, Automated Cloud Bootstrapping may be leveraged to inject application layer configuration e.g., internal application configuration policy (CDN, cDVR, etc.) can be controlled/influenced at image deployment, enabling automated configuration control and DevOps-style provisioning.

---

\(^{16}\) https://cloudinit.readthedocs.io/en/latest/
Automated application configuration management
When deploying applications within the cloud, having to apply post-installation configuration or tuning dilutes the benefits of cloud deployment, by adding time-consuming manual steps to bring capacity and functionality into service. The Nokia IP Video Unified Management System (UMS) hosts a centralized Automated Application Configuration Management service, which removes the need for manual configuration steps post-deployment. Once new instances of capacity are deployed (e.g., CDN Delivery Application), they automatically discover and register with a centralized service (API), which is then responsible for delivering the video service-layer configuration to the new instance. All configuration is pre-defined centrally, within a policy-based format. Configuration policy is applied based on a flexible rules-engine, allowing instances to be configured based upon a range of automatically detected properties e.g. source IP address, hostname, software version, location and opaque user-specified data injected at instance creation time. Application Configuration Management is exposed over REST APIs, which enable automated registration, deregistration and a centralized common user interface for defining configuration policy and rules.

Application state control API
Automation of operational workflows, like creating and deleting instances, requires that the system within the NFV MANO layer can understand and control the state of the applications running as VNFs. Application State Control API is a REST based interface which facilitates automated use cases by providing the following functions:

- View Application State: Provide a view of all instances (such as CDN cache) operational state.
- Control Application State: Bring instances (such as CDN DA) in and out of service.
- Gracefully Delete Instances: Notify instances (such as CDN DA) that they are going to be deleted, stimulating them to offload state such as logging data and currently active video sessions and to notify when complete.

Application State Control API supports the NFV MANO when performing automated tasks such as capacity scaling, deployment, upgrade, self-healing and monitoring.

https://puppet.com/product
**Application monitoring API(s)**

All Nokia IP Video application VNFs and the Unified Management System VNF include monitoring interfaces which expose data and metrics which describe the utilization of resources and the load and state of the video services which they support. The Application Monitoring API(s) are exposed over both REST and SNMP based interfaces. The data accessible over these interfaces is an important input to supporting automated workflows, orchestrated by NFV MANO systems, i.e., exposing load on specific components and/or geographic regions is an input into decision-making for scale-out or scale-in operations, whether automated or not.

**Software defined storage**

Cloud applications are increasingly moving toward consuming Software Defined Storage for their backend storage needs. Within OpenStack, Ceph has quickly evolved to become the de-facto backend storage technology. Ceph provides a horizontally scalable, fault-tolerant and self-healing storage system for the cloud. Ceph provides access to storage resources over a range of interfaces such as RDB (RADOS block device) for OpenStack Cinder and Glance, block-storage over a SWIFT compatible API and native POSIX interface over CephFS. Nokia IP Video applications leverage Ceph heavily for high-volume and high-throughput storage of large video files; for example, the Nokia Cloud DVR application consumes Ceph directly to provide extremely dense storage for cloud-based video recording workloads.

18 http://docs.ceph.com/docs/master/
Software defined networking
Network operators are increasingly leveraging new technologies and network architectures in the datacenter to provide dense carrier-scale datacenter networks with programmable and open interfaces to the network layer, allowing application automation to be handled hand-in-hand with the automation of network connectivity. Nokia IP Video applications are architected to reside on top of software defined networking platforms; for example, Nokia Cloud DVR is built on top of a network platform based upon Nuage Networks[^19], enabling the automated provisioning of Ceph storage, KVM based virtual machines and the supporting underlying network connectivity.

Application layer intelligence
By deploying applications within NFV platforms operators can leverage the power of automated NFV infrastructure to build automated workflows for a range of operational use cases, such as deployment, auto-scaling and self-healing. However, beyond the infrastructure layer, a cloud-based application should also be able to cope with the dynamic nature of today’s infrastructure-layer at an application level. Nokia IP Video applications have distributed intelligence to dynamically cope with changes at an infrastructure level. The Nokia Velocix CDN Request Router provides distributed application-layer load-balancing across the Nokia IP Video cloud, dynamically detecting infrastructure failures and load conditions and where necessary routing video workloads around problem areas, enabling cloud-based self-healing and self-optimization of video workloads.

As operator networks have evolved toward the deployment of network services within the cloud and have deployed NFV platforms, vendors and application providers have been driven to evolve their applications to fit better with these new environments. The applications and components used to construct network services within the cloud have needed to evolve to deal with automation, flexible deployment models, dynamic scaling and redundancy models and the dynamism which comes hand-in-hand with the use cases enabled by the cloud.

Vendors provide applications as VNFs today by providing software-only solutions, which are typically deployed as virtual machines. These VNFs often provide interfaces to MANO systems to support varying levels of automation. The VNFs provided today are largely self-contained instances, which are software-only versions of the hardware appliances which operators have traditionally deployed.

Today, applications provided as VNFs, are evolving through similar stages:

1. Decoupling: Network functions are separated from underlying hardware platforms. Decoupling from the hardware platform enables more efficient operations to be realized through the common hardware layer.

2. Virtualization: Network functions are deployed as virtual machines; hypervisors abstract infrastructure resources. Virtualization enables operational efficiencies, rapid scaling/deployment and co-location of application resources.

3. Cloud: Network functions are enhanced to behave less like monolithic components. Cloud innovation enables rapid deployment, elasticity and agility.

Today most VNF vendors are providing capability which aligns either with steps 2 and 3; for example this document describes how Nokia IP Video applications are already capable of the cloud-like behaviors necessary for rapid deployment, elasticity and agility. For most VNFs this means that network services are built upon VNFs deployed as virtual machines. However, as NFV evolves operators are already seeking to drive more and more efficiency and agility from their NFV platforms and this will drive the continued evolution of NFV platforms and VNF applications.
The current approach of VNF deployment, as virtual machines, serves its purpose of migrating the network services from a hardware-centric model to a software-centric model. However, by mapping traditional hardware appliances directly to software-based VNFs (as virtual machines) the VNF application itself often remains as a monolithic stack, usually comprised of VNF application components, a guest OS, a virtual machine and host OS with a hypervisor.

VNFs deployed as virtual machines are also often partitioned statically, meaning that infrastructure resources are statically reserved by VNFs. This method of deploying and scaling VNFs results in a twofold level of inefficiency: unnecessary deployment of components such as multiple guest OS and underutilized system resources as a result of static partitioning. Additionally, within monolithic VNF stacks, there is repetition of application resources, such as databases, storage gateways, message-bus, configuration-management and other such “infrastructure” services.

Network services can also scale down faster, making the reuse of infrastructure resources easier and faster. However, being able to deploy and scale faster is not useful if there is still the requirement for human intervention to trigger deployment. Beyond the containerization of application resources, new technologies are emerging to manage the automated allocation of containers (and the overall workload) to the most appropriate resources within a cluster – schedulers e.g. Docker Swarm, Kubernetes and Apache Mesos. By combining container-based micro-services with scheduling of resources, NFV platforms can take a more workload-centric approach to the deployment of network services, with fully automated resource allocation, scaling and healing – with greater efficiency and speed than with today’s hypervisor-based architectures.
Container-based micro-services will disaggregate the VNF software-stack, in the same manner as NFV has already disaggregated the hardware-based network appliances. Combined with container scheduling and orchestration, to manage deployment, scaling, concurrency and resilience, network operators will couple this new architecture with shared cloud infrastructure services based on open-source projects (configuration, database, message buses, etc.) and shared commodity hardware clusters, to define the next generation of NFV architecture, which will deliver a currently unavailable level of automation, programmability, modularity and agility – at greater speed and efficiency than today.

Nokia IP Video VNFS will continue to evolve, as network operator NFV architectures evolve and mature. Nokia IP Video products already leverage open-source software for common cloud infrastructure services and are already available as automated, elastic applications. The next generation micro-service architecture from Nokia IP Video will enable network operators to build highly-efficient and autonomous IP Video clouds, suitable for the next generation of NFV.
Glossary

- **Bare metal** - A bare metal environment is a computer system or network in which a virtual machine is installed directly on hardware rather than within the host operating system (OS). The term “bare metal” refers to a hard disk, the usual medium on which a computer’s OS is installed.\(^\text{20}\)

- **CAPEX** - A capital expenditure; money invested by a company to acquire or upgrade fixed, physical, non-consumable assets, such as buildings and equipment or a new business.\(^\text{21}\)

- **Cloud** - a term where computing resources are available as a service abstracted from the provider

- **COTS** – Commercial of the shelf

- **DA** – Delivery application (sometimes referred to as a cache). Terms that describe the delivery function of the Nokia IP video solution.

- **ETSI** - European Telecommunications Standards Institute

- **JITX** – Just in time transcoding

- **NFV** - Network functions virtualization

- **NFVI** - Network functions virtualization infrastructure

- **Nokia Velocix CDN** – An enterprise grade networking solution that allows service providers to quickly launch new revenue-generating services at a reduced cost that cannot be met by competing over-the-top (OTT) services. The Velocix CDN offers outstanding delivery performance levels, allowing service providers become attractive partners for content providers, or for content providers to deliver content directly to end users with high speed and quality with low latency.

- **OA** – Origin application, component within the Nokia IP video solution

- **OpenStack** - Open source software for creating private and public clouds

- **OPEX** - An operating expense, operating expenditure, operational expense, operational expenditure or OPEX is an ongoing cost for running a product, business, or system. Its counterpart, a capital expenditure (CAPEX), is the cost of developing or providing non-consumable parts for the product or system.\(^\text{22}\)

- **RTSP** – Delivery protocol used for video streaming. This protocol is supported by Nokia IP Video solutions.

- **Server / Pop** – location of deploying components within a network

- **Service providers** – a company or organization that aggregates and delivers video content to end users / subscribers

- **Unified Management System (UMS)** – a proprietary component of the a Nokia IP Video solution that enables the service provider, network operations center (NOC) team or content provider to configure, operate and manage Nokia's Video portfolio products including CDN, High-Scale Origin, Recording Manager and Personalization Broker

- **Video IP applications** – applications that comprise a Nokia IP video solution

- **VIM** – Virtual infrastructure manager

- **VNF** - Virtualized Network Function

---

\(^{20}\) [http://searchservervirtualization.techtarget.com/definition/bare-metal-environment](http://searchservervirtualization.techtarget.com/definition/bare-metal-environment)

\(^{21}\) [http://whatis.techtarget.com/definition/CAPEX-capital-expenditure](http://whatis.techtarget.com/definition/CAPEX-capital-expenditure)
