A pattern for network functions virtualization

Eduardo B. Fernandez, Brahim Hamid

A pattern for Network Functions Virtualization

EDUARDO B. FERNANDEZ, Florida Atlantic University

BRAHIM HAMID, University of Toulouse

Cloud computing has brought a large variety of services available to potential consumers. A recent type of services are the provision of network functions using virtualization. Network Functions Virtualization (NFV) is a network architecture where network node functions such as load balancers, firewalls, IDS, and accelerators are built in software and offered as services. This approach results in reduced complexity in network design, better scalability and agility, as well as faster deployment. We present here a pattern for the NFV architecture.

Keywords: virtualization, architecture patterns, network functions, security patterns, cloud computing, telecommunications

INTRODUCTION

A cloud-based computing system involves a variety of users and devices connected to it and provides multiple kinds of services. Typically, clouds provide three levels of service: Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS). SaaS can be directly used by consumers to access its available applications but also to let these consumers become in turn service providers (SPs). Telecommunication (telco) companies have discovered that they can provide services to their customers by building their networks as services rented from some cloud provider or from their own private clouds [Bas14]. The provision of network functions using virtualization, *Network Functions Virtualization (NFV)*, is a network architecture where node functions such as load balancers, firewalls, IDS, and accelerators are built in software and offered as services. Each *Virtualized Network Function (VNF)* may use one or more virtual machines or containers running different software. This approach results in reduced complexity in network design, reduced cost, better scalability and faster deployment. While the telco company could build these services using its own hardware, using cloud services results in no upfront expenses. Note that this is virtualization using the services of the cloud at any level; it is not about virtualizing processors and networks as it is done normally in the cloud itself in its PaaS level. Of course, NFV can be performed using a dedicated cloud using its PaaS functions.

We present here a pattern for the NFV architecture. Our audience includes system architects and system designers as well as telco service providers. The NFV pattern provides network functions related to a cloud reference architecture and is an important part of cloud ecosystems. An ecosystem is the expansion of a software product line architecture to include systems outside the product which interact with the product [Bos09]. Figure 1 shows a partial cloud ecosystem. The Cloud Security Reference Architecture (SRA) is the main pattern (hub) that defines the ecosystem [Fer15a]. The SRA can be derived from a Cloud RA by adding security patterns to control its identified threats. Cloud Web Application Firewalls and Security Group Firewalls provide filtering functions that can be provided as services through VNFs or on their own. The Cloud Compliant Reference Architecture applies patterns to the Cloud RA to comply with regulations.

Authors' addresses: Eduardo B. Fernandez (corresponding author), Dept. of Computer and Electrical Eng. and Computer Scienc Florida Atlantic University, 777 Glades Rd., Boca Raton, FL33431, USA; email: ed@cse.fau.edu; Brahim Hamid, University of Toulouse, France, email: brahim.hamid@irit.fr

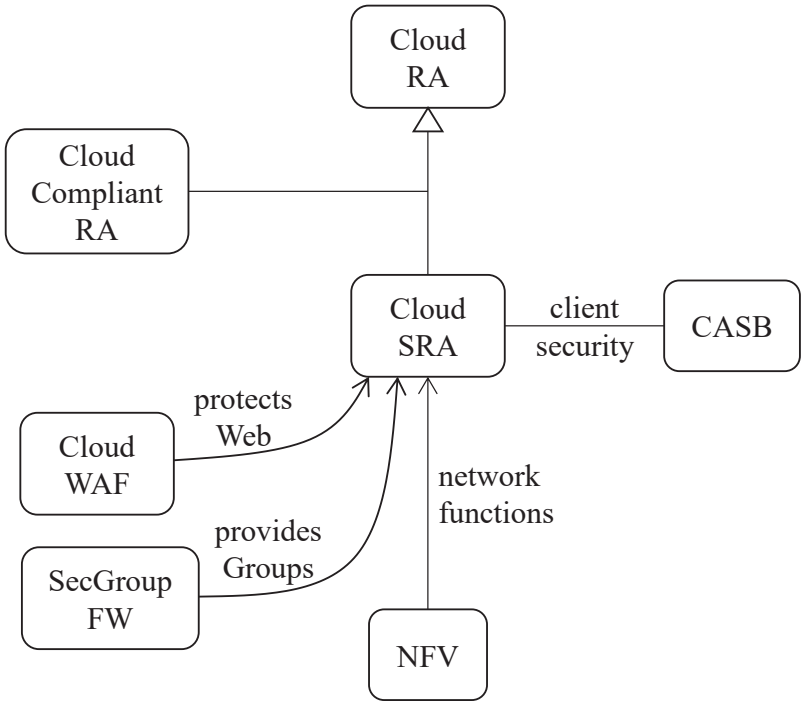

Figure 1. The relationship of NFV with other cloud patterns

NETWORK FUNCTIONS VIRTUALIZATION

Intent

Network Functions Virtualization (NFV) is an architecture for the construction of network services using software building blocks. The building blocks, *Virtual Network Functions (VNFs)*, are typically created from cloud services using virtual machines or containers.

Context

Telco companies provide network services to their customers. To do that, they set up physical networks. A cloud service provider may provide them with virtual networks but they still need to use hardware functions such as routers, switches, and firewalls, to interconnect their functions and their data, as shown in Figure 2. We call *Service Providers (SPs)* the clouds that provide services, and *Telco Service Providers (TSPs)* or just telcos, the companies that provide telecommunication services to their customers.

Problem

There is a variety of network devices that are required in order to set up communication networks. As indicated above, until recently, these devices were physical devices, but this approach implies a high up-front cost and is not scalable. When the needs of the users of the networks change we need to buy more or different devices. The new set up also takes time and we cannot be very responsive to the needs of our customers. If the devices have weak security protection it is difficult to harden them. How can we provide more flexible, responsive, and secure services to our customers? The solution to this problem is affected by the following forces:

- Heterogeneity—the variety of proprietary hardware appliances keeps increasing; this produces large costs and unnecessary complexity.

- Lifecycle—the lifecycle of hardware devices is becoming shorter because of rapid technological advances; this makes investments on them to depreciate very fast.

- Security—complex systems are easier to attack because of the higher probability that they will have vulnerabilities. It is also harder to apply common policies to a very diverse set of functions. Upgrading the security of a device may be costly or very complex. Different applications may need different degrees of security.

- Reliability/availability—the integration of a variety of devices increases the probabilities of failure because of the extra complexity.

- Scalability—increasing the number of units brings integration problems because of the variety of built-in interfaces and protocols. It is also expensive and time-consuming to procure hardware devices.

- Extensibility—adding new functions requires new specialized hardware that may not be available.

- Development—hardware development is a barrier for new products because they are hard to develop and test, requiring costly facilities.

- Location—the network functions should be dynamically moved to, or instantiated in any location in the system but this is hard to do with physical devices.

- Power usage—the use of many physical network devices requires large amounts of power; it is convenient to reduce these needs.

- Fast time to market—business requirements demand the production of functions in a short time; otherwise, the business loses competitiveness.

- Upgrades—when there are improvements, we need to upgrade all the functions of some type. These updates may take time when performed on hardware devices.

Solution

Service providers (SPs) can use SaaS services to provide network functions where we simulate the specific functions of a hardware device using software running in one or more virtual machines or containers in the cloud (Figure 2). Figure 3 shows that we can build network services using the PaaS and IaaS services and deploy them as SaaS services. These network services can then be used for the needs of the applications being accessed by the Consumers. The Network Services are provided by a telco company, which in turn rents virtual hardware from the cloud provider (or owns the cloud).

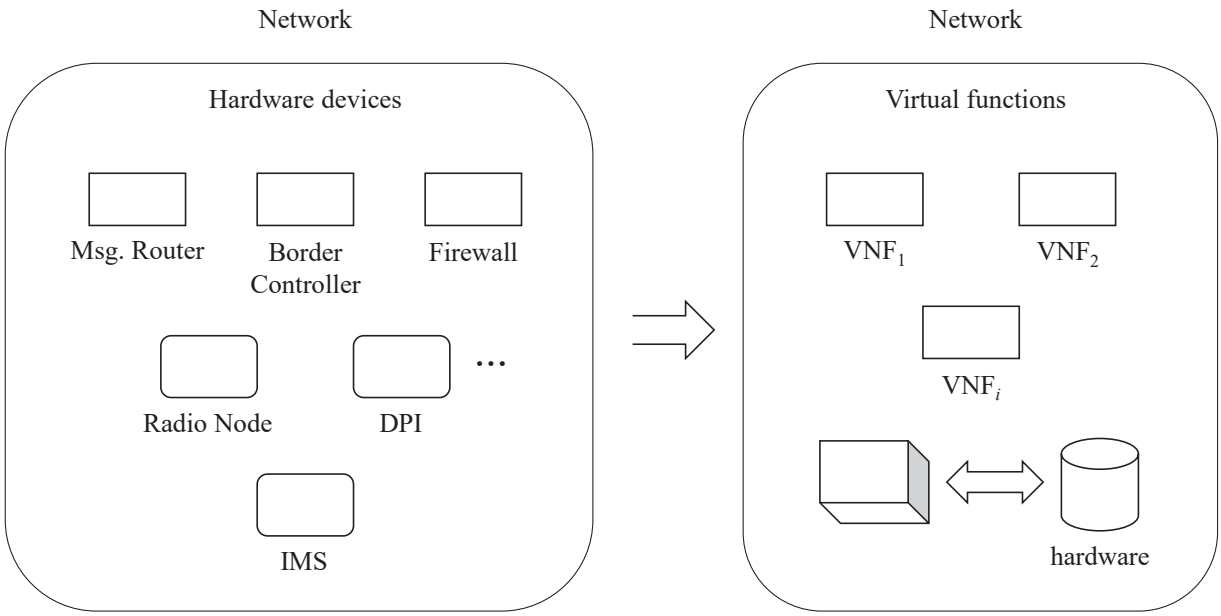

Figure 2. Change in network devices implementation

Structure

Figure 4 shows the class diagram of the NFV pattern. The cloud **Virtualization Layer** provides **Virtual Network Services (VNF)** implemented using one or more **Virtual Machines (VM)**. While the SaaS level is used to provide the services to telco companies (Consumer1…N), the provider of these services would build them using the PaaS of the cloud, utilizing one or more virtual machines to implement each service.

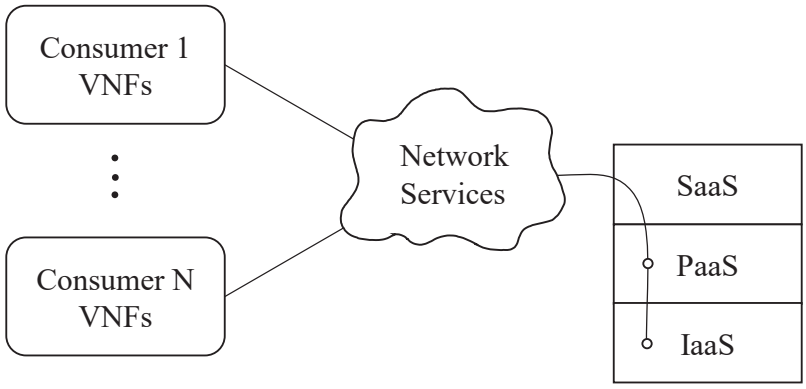

Figure 3. Idea of the NFV pattern

Dynamics

Figure 5 shows use case “Access a virtual function”. The Customer requests the use of a network function, which is supplied by the cloud provider. To access the service the request goes through a Virtual Router (VRouter), which uses one or more Virtual Machines (VM) for its implementation. Another use case could be “Set up a virtual network”, which would insert several VNFs in a virtual network provided by the cloud.

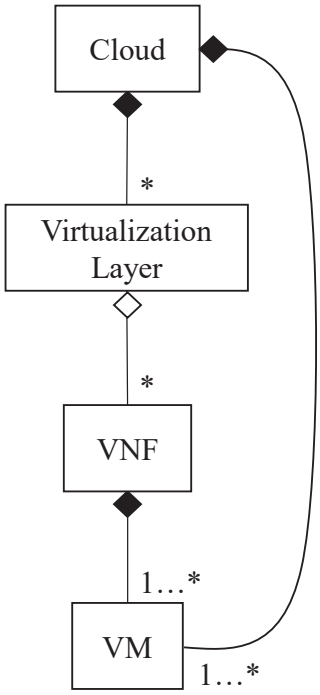

Figure 4. Class diagram of the NFV pattern

.

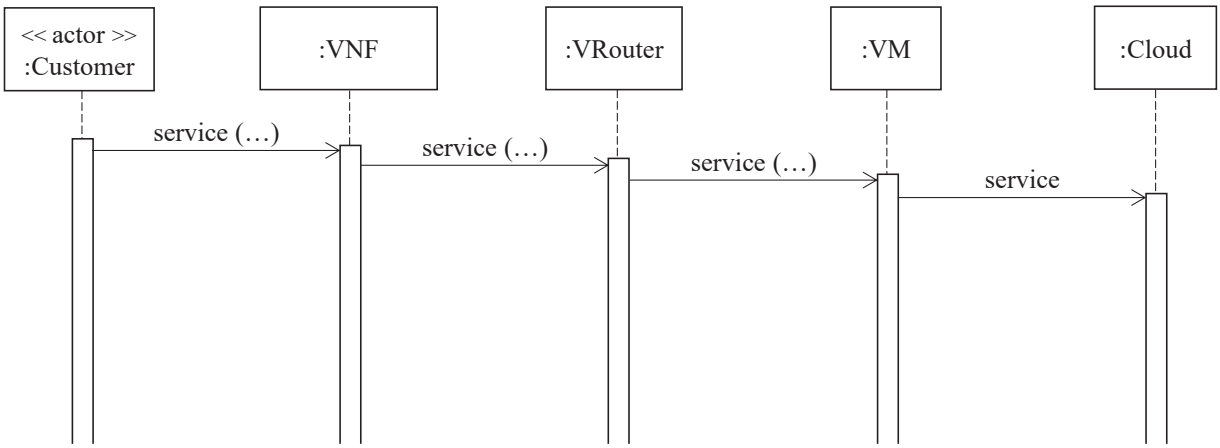

Figure 5. Use case “Access a virtual function”

Implementation

An SP may implement some VNFs, which must be composed (chained) to build more complex services. The network must be able to instantiate VNF instances, monitor them, repair them, and bill for the provided services (orchestration process) [Wik15]. The network service must also provide availability and

security. The Telco Provider may locate the VNFs in different locations according to the needs of their customers.

Each VNF may be implemented as a stand-alone function but it might be better to use Software Defined Networking (SDN). SDN decouples the data and control plane from each other and it is a complementary technology that simplifies the orchestration of services.

Figure 6 shows the ETSI architectural framework for NFV [ETS13-2]. The NFVI (NFV Infrastructure) provides the virtual resources to support the execution of the VNFs. [Bat13] shows the implementation of an NFV router using the Open Flow protocol (Open Flow is the most common protocol for implementing SDN).

Strictly, a cloud is not needed to provide VNFs; however, given the objective of cost-effective solutions available anywhere, it is almost the only practical way. Given the specialized type of services, private clouds will probably be the most common way to provide VNFs [ETS13]. A VNF can be built using virtual machines or containers [Fer15b]. Current implementations are using OpenStack APIs [Kav15].

For security, typical implementations use authentication, authorization (RBAC or Rule-Based Access Control), and encryption [Fer13].

Known uses

- The Cisco ESP uses NFV, SDN, open APIs, and advanced orchestration capabilities to create a flexible and modular platform [Cis].
- Alcatel-Lucent has a variety of NFV products [Alc].
- Ericsson has implemented VoLTE (Voice over LTE) services using NFV [Jel14]. They have also extended Neutron, the OpenStack API with appropriate functions [Kav15].

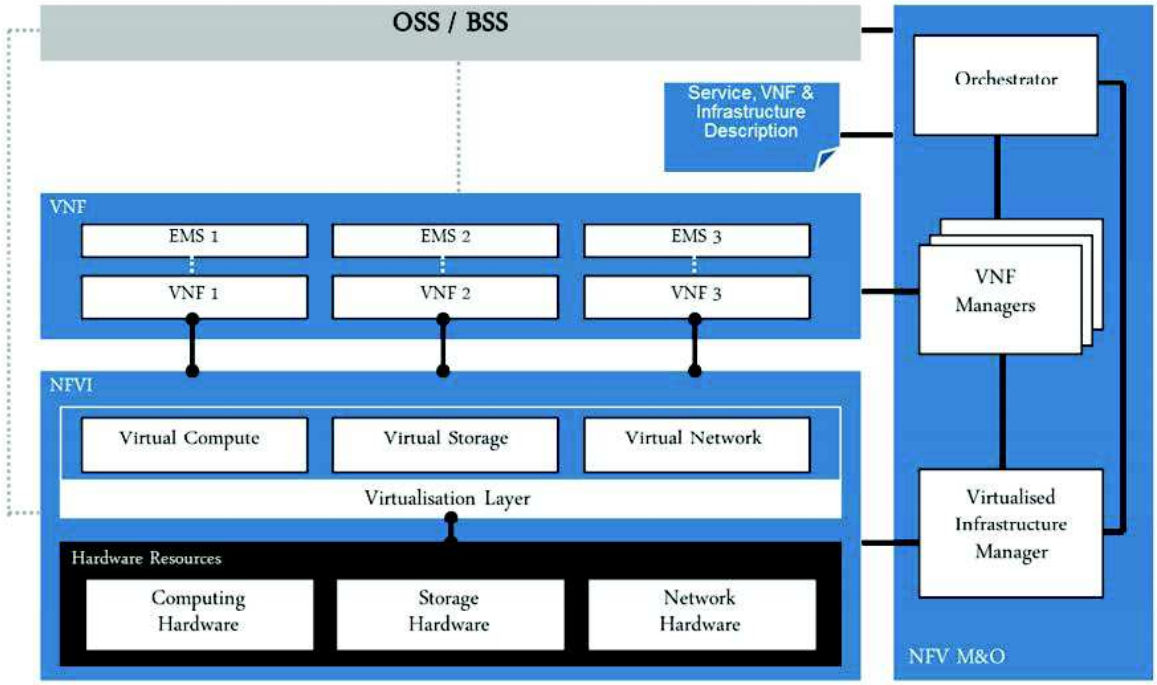

Figure 6. NFV implementation framework (from [ETS13-2]

Consequences

The NFV pattern presents the following advantages:

- Heterogeneity—by implementing standardized functions a good part of this heterogeneity disappears, all the functions are handled uniformly.
- Lifecycle—software-implemented functions are not affected by hardware technological changes
- Security—a unified system can be protected in a systematic way using institution policies that are applied to all devices. It is possible to have different degrees of security, tailored to different types of users or applications. The reduced complexity also contributes to security.
- Reliability/availability—error propagation is easier to contain, a hardware failure can be solved by restating a VM in another hardware.
- Scalability (elasticity)—increasing the number of units just requires making more software copies and can be dynamic.
- Extensibility—adding new functions requires new software that might not even have any hardware counterpart. New functions are pluggable units.
- Development—new vendors can develop products conveniently, without need for complex hardware development and testing facilities.
- Location—it is possible to move network functions or instantiate them, anywhere in the network.
- Power usage--reduced power usage is obtained by migrating workloads and powering down unused hardware [ETS13].
- Fast time to market—new functions can be set in a short time.
- Upgrades—can be applied automatically to all the corresponding functions.

It also has some liabilities:

- There is some performance overhead in providing a VNF as opposed to a function obtained from a hardware device.
- Without standardized functions, interoperability may not be possible; i.e. the telcos have to form federations to use NFVs.
- There will be new types of security threats. Software in the Internet and in the cloud is subject to a variety of threats, and the specific implementation of the VNFs can bring new threats.

Related patterns

- Virtual Machine Operating System architecture [Fer13]. Includes a Virtual Machine Monitor that creates virtual machines. The virtualization of processors, storage, and networks is clearly the basis for NFV.
- A pattern for SaaS is given in [Has12].
- Three potential patterns for Network I/O virtualization are given in [Luo10]. We are in the process of writing them.
- VoIP patterns [Pel07] are services that could be implemented in a low cost way by operators using VNFs [Jel14].
- Authenticator [Fer13]—Authenticates consumers to the SP and vice versa.
- Authorizer [Fer13]—Applies SPs' access policies to their services, including RBAC.
- Security Cloud Reference Architecture [Fer14]—Defines the context for the NFVs by describing the structure of the cloud system and the places where the NFV would interact with it.
- SDN patterns (not yet written)–As indicated, SDN decouples the data and control plane from each other and it is a complementary technology.
- A. Chesla proposes the concept of disaggregated security functions, an additional concept for which there are no patterns yet [Che14]. In this approach, elementary security functions are virtualized and composed to build more complex functions.

ACKNOWLEDGEMENTS

We thank our shepherd, Klaus Marquardt, for his valuable comments that contributed to improve our paper. The participants in the EuroPLoP workshop also provided useful suggestions.

REFERENCES

[Alc] Alcatel-Lucent, http://www.alcatel-lucent.com/solutions/nfv

[Bas14] H. Basilier, M. Darula, and J. Wilke, "Virtualization network services—the telecom cloud", Ericsson Review, March 28, 2014, 2-9.

[Bat13] J Batalle, J Ferrer Riera, E Escalona, JA Garcia-Espin, "On the implementation of NFV over an OpenFlow infrastructure: Routing Function Virtualization", Future Networks and Services (SDN4FNS), 2013 IEEE

[Bos09] J. Bosch, "From software product lines to software ecosystems", Procs. 13thInt. Software Product Line Conf. (SPLC'09), August 2009, 111-119.

[Che14] A. Chesla, "Exploring the concept of disaggregated security functions", Security Week, November 12, 2014.

[Cis] Cisco, Network Functions Virtualization, http://www.cisco.com/c/en/us/solutions/service-provider/network-functions-virtualization-nfv/index.html

[ETS13] ETSI GS NFV 001 v. 1.1.1, Network functions virtualization (NFV); use cases, October 2013.

[ETS13-2] ETSI Network functions virtualization (NFV), Network operator perspectives on industry progress, October 2013.

[Fer13] E.B.Fernandez, "Security patterns in practice: Building secure architectures using software patterns", Wiley Series on Software Design Patterns, 2013.

[Fer15a] E.B.Fernandez, Raul Monge, and Keiko Hashizume, "Building a security reference architecture for cloud systems", Requirements Engineering, 2015. DOI: 10.1007/s00766-014-0218-7

[Fer15b] Madiha H. Syed and E.B.Fernandez, "The Software Container pattern", submitted for publication.

[Has12] Keiko Hashizume, E.B.Fernandez, and Maria M. Larrondo-Petrie, "A pattern for Software-as-a-Service in Clouds", RISE'12,Workshop on Redefining and Integrating Security Engineering, part of the ASE Int. Conf. on Cyber Security,

Washington, DC, December 12-14, 2012.

[Jel14] B. Jellema and M. Vorwerk, “Communications as a cloud service: a new take on Telecoms”, Ericsson Rev., July 22, 2014, 2-9.

[Kav15] A. Kavanagh, “OpenStack as the API framework for NFV: the benefits, and the extensions needed”, Ericsson Review, 2015, No 3, 2-7.

[Luo10] Y. Luo, “Network I/O virtualization for cloud computing”, IT Pro, IEEE Sept./Oct. 2010, 36-41.

[Pel07] J. C. Pelaez, E.B.Fernandez, and C. Wieser, "Patterns for VoIP signaling protocol architectures", Procs. EuroPLoP 2007. http://hillside.net/europlop/home.html

[Sev12] Several authors, “Network Functions Virtualization”, SDN and Openflow World Congress, Darmstadt, Germany, October 2012.

[Wik15] Wikipedia, “Network Functions Virtualization”, http://en.wikipedia.org/wiki/Network_Functions_Virtualization