OPNFV*: NFV system integration and development as an open community effort

Telecom operators are in the midst of modernizing their networks. They are leveraging the European Telecommunications Standards Institute (ETSI) Network Functions Virtualization (NFV) architectural framework to bring automation, agility, and flexibility to network infrastructure and services using proven IT virtualization and cloud technologies. NFV transformation requires newer, different capabilities than those available today in IT-based solutions. The Open Platform for Network Functions Virtualization (OPNFV) open-source project under the Linux Foundation is a carrier-grade, integrated open-source platform created to accelerate the introduction of new NFV products and services. This report describes OPNFV, its progress, and the joint efforts being made by Cisco, Ericsson, and other ICT industry players to bring the best-in-breed technologies and solutions to open source.

*Open Platform for NFV and OPNFV are trademarks of the Open Platform for NFV Project, Inc. This is an independent publication and is not affiliated with, authorized by, sponsored by, or otherwise approved by Open Platform for NFV Project, Inc.

Introduction

Operator networks have traditionally been created using purpose-built hardware, and they have often been complex and time-consuming to operate. To improve this, the ETSI NFV Industry Specification Group (ISG) [1] created an architectural framework to evolve networks to bring automation, flexibility, agility, and operational simplicity to infrastructure and network services by utilizing virtualization, software-defined networking (SDN), and cloud technologies with commercial off-the-shelf hardware and by fostering open ecosystems around NFV concepts.

The NFV framework has several components like compute, network, storage, virtualization layer, Virtualized Infrastructure Manager (VIM), and Virtual Network Functions (VNFs), as shown in Figure 1. These components can be fulfilled either by proprietary or open-source-based solutions. The use of open-source components provides the ability to leverage the faster pace of convergence and innovation in open-source communities, and it also helps to avoid vendor lock-in.

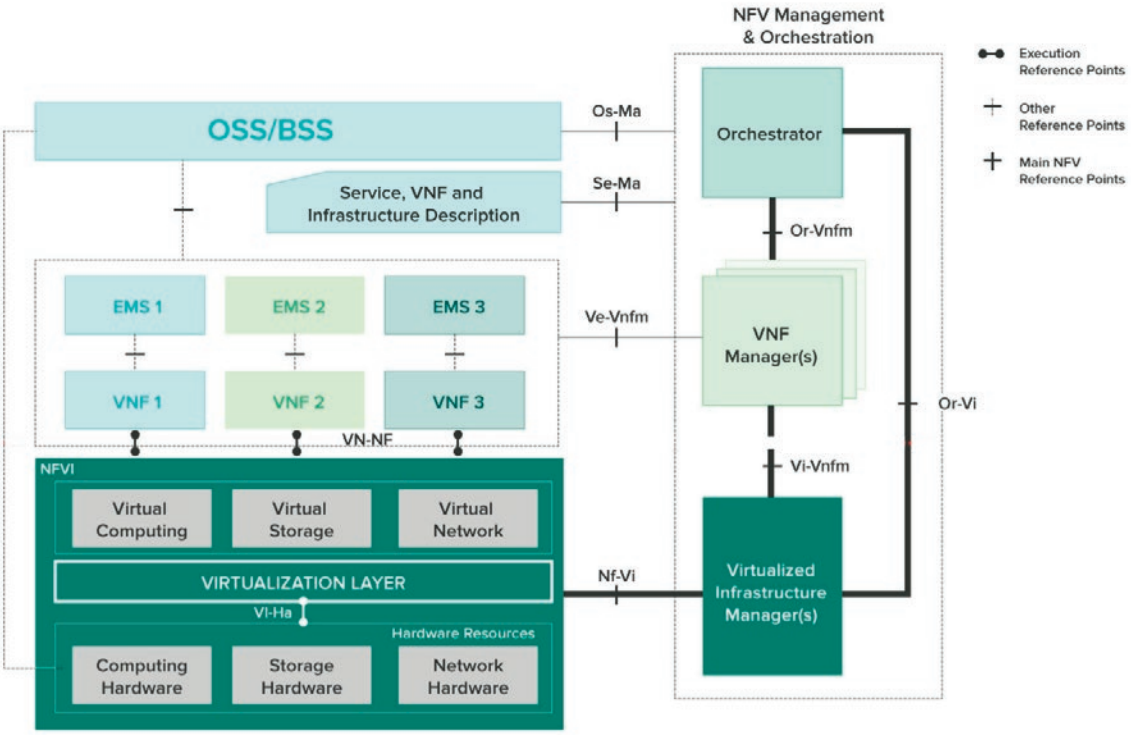

Figure 1: OPNFV's depiction of the ETSI architectural framework [2]

Several open-source solutions are addressing IT needs, which results in a set of components that are not well suited for telecom products and applications, as they do not meet the strict requirements in areas like Service Level Agreement (SLA) enforcement or fault management. Introducing new, NFV-specific features into these solutions is a challenge, due to the difference in mindset between these two parts of the industry.

Each operator is unique in its business needs and can make choices for NFV components to meet their use case requirements. The breadth of choices available for the NFV components means that interoperability becomes a key issue among the components. In fact, market research [3] shows that interoperability is a major concern for the operators in the NFV transformation journey.

OPNFV

Over the past 10 years or so, open-source development has become increasingly relevant to all parts of the industry. As a result of this trend, more and more open-source components have been added to both enterprise IT and telecom products. In parallel with this tendency, the emerging presence of NFV and SDN has been moving the industry toward network and application virtualization.

As the latest trends in virtualization pushed several parts of the IT world on top of a cloud-based platform, following this pattern also became a logical next step for telecom applications. However, it soon became clear that fitting all these pieces together required a collaborative effort.

This is how OPNFV [4] started to formalize, and it was launched in September 2014 under the umbrella of the Linux Foundation as a collaborative development project. Both Ericsson and Cisco are platinum-level founding members, along with several other telecom vendors, service providers, and IT companies. The primary purpose of OPNFV is to create an integrated, carrier-grade, open platform to support the evolution of NFV.

OPNFV has an ambitious mission in trying to build an NFV platform from pure open-source components that is capable of running VNFs on top, while at the same time, it is working to improve the parts to make the environment increasingly NFV-ready. To prove the viability of this concept, the first release of OPNFV, Arno, focused on the infrastructure layer (as shown in Figure 2). The OPNFV Arno release contains the first version of the integrated platform using well-known upstream projects like OpenStack, OpenDaylight (ODL), Ceph, and Kernal-based Virtual Machine (KVM) as building blocks.
The community also produced a basic set of functional test cases to verify and validate the deployed environment.

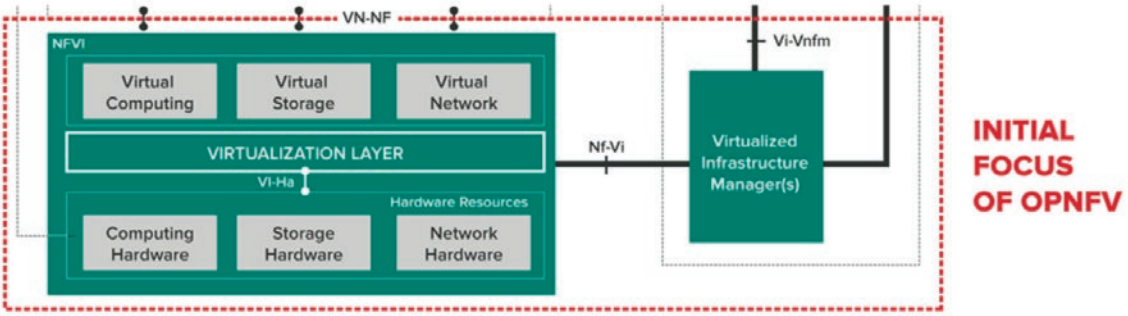

Figure 2: Initial focus of OPNFV (applicable for the Arno and Brahmaputra releases) [2]

By choosing a six-month release cycle, the OPNFV contributors are working continuously to enhance the platform by adding improvements and new functionality to the upstream projects on which it relies. This indicates the special nature of OPNFV, which is why it is often referred to as a 'mid-stream' project. The term depicts the different ways OPNFV interacts with the different open-source projects and communities.

Depending on the needs of an NFV deployment, the composition of components that form an NFV system, as well as their configuration and version, may vary. A composition of components and their configuration is commonly referred to as a "deployment scenario," or simply a "scenario." The OPNFV project revolves around providing automated and ongoing deployment and testing of scenarios. This is commonly referred to as the "downstream" direction, which covers the integration work, where certain releases of components created by upstream projects are integrated and tested. The OPNFV community is focused on the evolution and enhancement of scenarios by further developing the components that make up the scenarios and/or developing new components, along with their automated deployment and test infrastructure.

The second major part of the community's activities is to collaborate with the targeted open-source upstream communities in order to add missing pieces to their code base to be able to cover an end-to-end implementation of new functionality at platform level or to meet the strict carrier-grade requirements. This upstream notion of the tasks comes from the principle that OPNFV is only intended to host and maintain integration, deployment, and test-related code; the OPNFV contributors are working toward the repositories of the underlying open-source projects to add new features and enhancements.

Knowing the significance of standards in the telecom industry, it is important to note that OPNFV also serves as a bridge between standardization bodies and the activities and actual development work that add speed and faster realization to the flow. In its pursuit to develop critical NFV capabilities in open-source components, OPNFV leverages architectural insights and takes guidance from standards organizations such as ETSI and IETF (Internet Engineering Task Force).

One of the community's key focus points is to align with guidelines from ETSI NFV ISG. By actively doing implementation work upstream, OPNFV sets out to create de-facto standards in the market, while ensuring quality and interoperability.

By creating this open-source community, all of its members can synchronize requirements and team up to jointly build solutions within the upstream communities. Such a concerted effort is more successful than contributions from individuals and independent organizations. The collaborative effort also provides the ability to execute both functional and performance tests on different scenarios in order to better understand how the components work together and to find the sources of errors and bottlenecks easier and faster. By having a growing number of OPNFV community labs (also known as "Pharos" labs) around the world, these benefits can be shared with other communities to help them with the work they are doing and to have even better collaboration between them and OPNFV.

By providing integration for NFV as an open-community effort, OPNFV reduces the overall effort required by the NFV industry for testing (testing of scenarios is done once, in the open, for the entire industry, as opposed to by each company individually in a redundant fashion), and at the same time it fosters the development of new features and functionality, and helps identify system-level integration issues and bugs very early in the development cycle.

Approaching the second and third releases of the platform in 2016, the community is ready to open up the initial scope (see Figure 1) and take up new challenges on the road to an integrated, cloud-based platform that fulfills the VNF requirements.

As an open-source project, OPNFV [5] is uniquely positioned to bring together the work of standards bodies, open-source communities, and commercial suppliers to deliver a de facto standard open-source NFV platform for the industry. Participation is open to anyone, including employees of member companies and even those who are just passionate about network transformation.

OPNFV progress

The number of OPNFV member organizations has steadily increased, and there are now 55 members [6] representing various segments of the industry. The number of committers and contributors to the projects has also increased, to more than 150, representing a growing interest in the work happening in the OPNFV community.

Since Arno, the first OPNFV release, the OPNFV community has made tremendous progress. As shown in Figure 3, there are about 40 incubated projects in the second OPNFV release, called Brahmaputra. The projects cover requirements and new functionalities targeted specifically for service providers in modernizing their networks and integration and testing projects that test common use case scenarios. These projects provide the rich functionality necessary for NFV deployments, and also help the easier integration, rapid deployment, and creation of high-quality products.

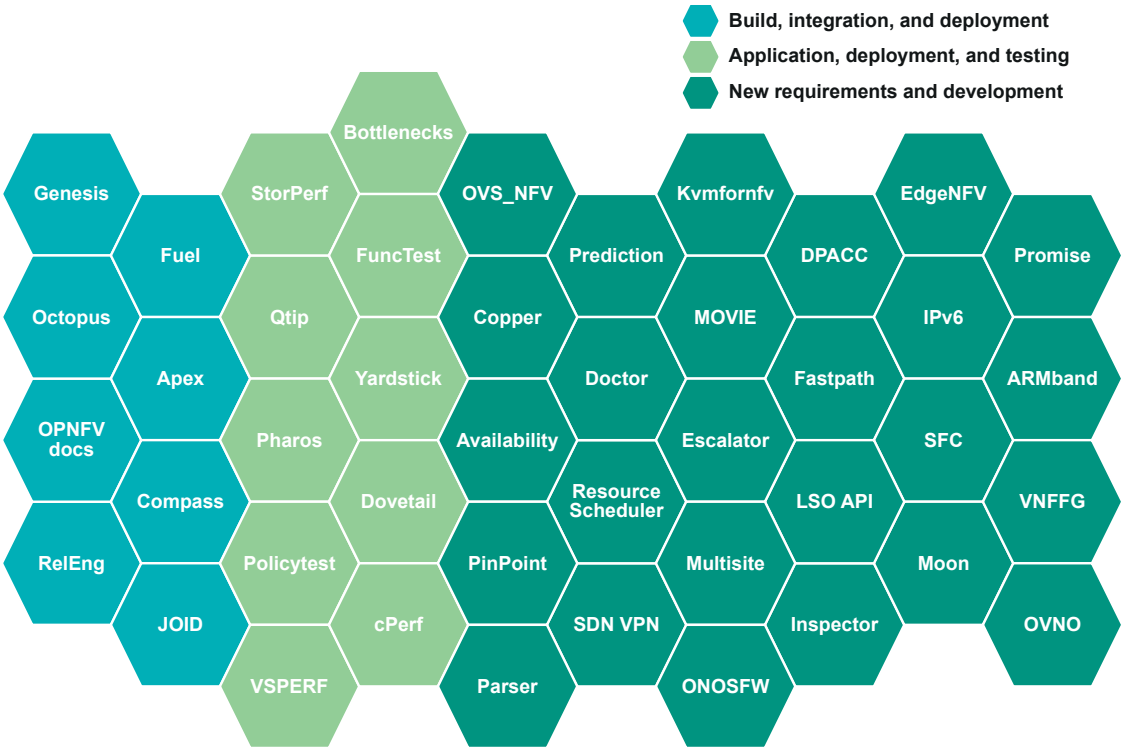

Figure 3: OPNFV projects [2]

OPNFV scenarios are deployed on both x86 and ARM architectures. The Brahmaputra release includes scenarios that leverage different SDN controllers, for example, ODL as well as OpenContrail and ONOS (Open Network Operating System). OPNFV scenario deployments use installers based on Juju, Compass, Fuel, and RDO-Manager. Brahmaputra increases the number of OPNFV community labs (Pharos) to more than 10. Initially, the scope of OPNFV was limited to the NFV Infrastructure (NFVI) and VIM of the ETSI NFV architectural framework. However, this scope restriction was recently lifted. OPNFV C release is expected to contain significantly more projects, including coverage of the management and orchestration components.

The following list of projects provides an insight into the breadth of functionality being developed, integrated, and tested within OPNFV.

Apex	RDO-Manager based OPNFV installer
Availability	Requirements for high availability in NFV
Bottlenecks	System limitation testing
Compass	Compass-based OPNFV installer
Copper	Virtualized infrastructure deployment strategies
Doctor	Requirements for fault management and maintenance
DPACC	Data Plane Acceleration requirements
Escalator	Requirements for NFVI/VIM upgrades
Fuel	Fuel-based OPNFV installer
FuncTest	OPNFV platform functionality testing
Genesis	Deployment tools requirements for common user experience for OPNFV
IPv6	Enabling IPv6 in OPNFV
JOID	Juju OPNFV installer
Kvmfornfv	NFVI enhancements in KVM hypervisor
Multisite	Requirements and Gap analysis for multisite virtualized infrastructure
Octopus	Continuous integration
ONOSFW	ONOS framework with ONOS as an SDN controller
OPNFV docs	OPNFV documentation
OVNO	OpenContrail Virtual Networking for OPNFV
OVS_NFV	Open vSwitch for NFV
Pharos	OPNFV community labs
Promise	Requirements for resource reservation and management
Qtip	Platform performance benchmark test suite with emphasis on quantitative testing
SDN VPN	SDN distributed routing and VPN
SFC	Service Function Chaining
StorPerf	NFVI storage performance benchmarking
Yardstick	Infrastructure compliance and SLA verification

For a complete list of projects and detailed information, please visit the OPNFV approved project list page at: **https://wiki.opnfv.org/approved_projects.**

Ericsson and Cisco: joint efforts in OPNFV and open source

Provisioning complex network services manually is time-consuming and prone to error; it also often takes weeks to deploy the services in production environments. Ericsson and Cisco have joined forces to simplify and automate this operationally complex aspect of the network. Cisco and Ericsson are contributing heavily to the open-source projects, and people from both organizations are leading and working together in open-source initiatives in the OPNFV and upstream ODL project to make networks easier to operate. Service Function Chaining (SFC) [7] is one such example where the open collaboration between people from both companies is resulting in cutting-edge features in the open-source projects, ultimately helping the whole user community.

The SFC project makes provisioning of complex network services faster and programmable. This project also provides the ability to define an ordered list of network services – such as firewalls, Network Address Translation, and QoS – and stitch them together in the network to create a service chain, allowing the traffic between two end points to flow through the service functions in the order defined by the service chain. The service chains can be as simple as inserting a service function like a firewall between two end points, and as complex as inserting multiple services in a defined order between two end points. SFC can also allow a flow to traverse through different service chains based on the direction of the flow between the end points.

Service function chains can be tailored based on the needs of a tenant or subscriber. Additionally, the service functions can be scaled up and down based on the traffic load. They can also be dynamically inserted or deleted based on the tenant requirements. Ideally, multiple service chains would be created, allowing for complex use cases to be achieved quickly that previously would have taken weeks or even months to deploy. Once multiple service chains have been created, the tenant traffic has to be “mapped” to one of the pre-provisioned service chains. This mapping is called classification, and it is a crucial counterpart to SFC.

Cisco, Ericsson and other companies worked together with the ODL community in an open, collaborative fashion to cleanly abstract out the classification functionality from SFC. This effort made SFC classification agnostic, and it allowed classifier implementations to query SFC to get the necessary information to send packets to different pre-provisioned service chains. ODL Group Based Policy (GBP) is a classifier, and the two companies worked closely to integrate the SFC and GBP projects into the ODL project. Ericsson and Cisco later joined forces again to introduce the ODL GBP and SFC integration into the OPNFV SFC project.

Figure 4 shows a simple, preliminary SFC use case. In this use case, there are two simple service chains with a firewall acting as a service function. Based on subscriber information, the classifier maps the traffic from Client-1 to service chain S1 and Client-2 traffic to service chain S2. The Service Function Forwarder (SFF) OpenFlow switch steers the traffic to the appropriate service function. In the service chain S1, the firewall blocks the HTTP traffic, which means Client-1 cannot access the webpages served by the HTTP server, but it can use Secure Shell (SSH) to connect to the HTTP server. In the service chain S2, the firewall blocks the SSH traffic, which means Client-2 cannot use SSH to connect to the HTTP server, but it can access webpages served by the HTTP server. The advantage of SFC is that it can allow far more complex and more realistic use cases.

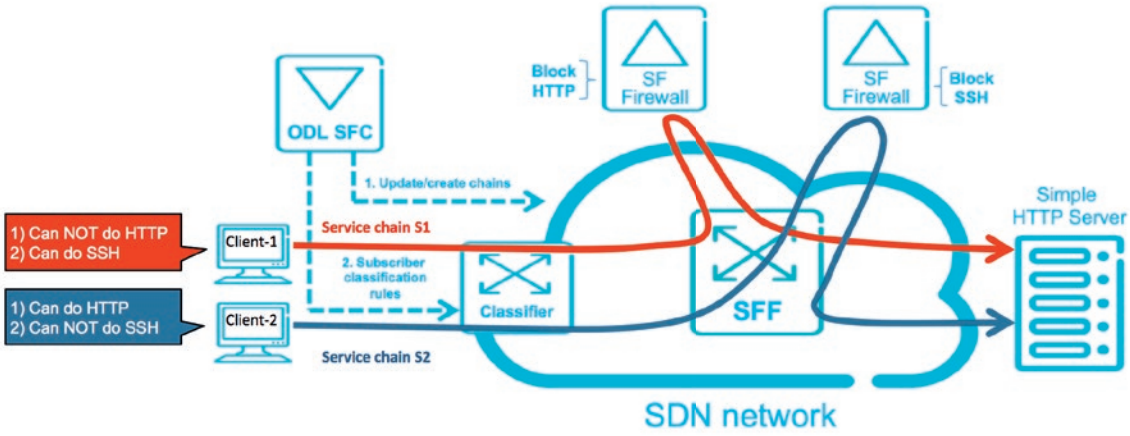

Figure 4: Initial OPNFV SFC use case [7]

OPNFV benefits

ETSI NFV ISG has recognized the role that OPNFV plays as a key forum for service providers and vendors to work together to address the implementation challenges in the NFV transformation journey. OPNFV brings the following benefits to the service provider community:

1. It identifies feature and functionality gaps in existing open-source projects like OpenStack, ODL, and KVM, and fills those gaps by implementing corresponding solutions in those projects.

2. It validates NFV concepts and use case scenarios that are critical to service providers.

3. It validates features and functionality across a broad range of hardware platforms through labs, and proactively identifies interoperability issues.

4. It reduces the overall efforts required by the NFV industry for integration testing.

5. It lowers integration costs through continuous integration and deployment.

6. It provides a forum to discuss, agree, and solve technological challenges with NFV.

7. It acts as a collective voice for the user community when interacting with open-source projects for upstream tasks.

8. It accelerates NFV progress through openness and collaborative efforts, and avoids vendor lock-in.

Conclusion

NFV is modernizing operator networks by bringing automation to infrastructure and network services. At the same time, it is introducing new interoperability issues because of the tremendous choices available to the operator when it comes to selecting components. OPNFV is helping to solve these challenges by leveraging open-source technologies to validate the key NFV concepts and use case scenarios through development, integration, and testing efforts. Cisco and Ericsson are working together with the community to implement the missing features and address the requirements to make NFV transformation a reality and a great success.

Glossary

ETSI	European Telecommunications Standards Institute
GBP	Group Based Policy
ISG	Industry Specification Group
KVM	Kernel-based Virtual Machine
NFV	Network Functions Virtualization
NFVI	NFV Infrastructure
ODL	OpenDaylight
ONOS	Open Network Operating System
OPNFV	Open Platform for NFV
OSS/BSS	operations support systems / business support systems
SDN	software-defined networking
SFC	Service Function Chaining
SFF	Service Function Forwarder
SLA	Service Level Agreement
SSH	Secure Shell
VIM	Virtualized Infrastructure Manager
VNF	Virtual Network Function

References

1. ETSI, Network Functions Virtualisation, available at:
   http://www.etsi.org/technologies-clusters/technologies/nfv

2. OPNFV, November 30, 2015, Introducing Open Platform for NFV, available at:
   http://go.linuxfoundation.org/l/6342/2015-12-01/2xwdcl/6342/136949/OPNFV_Overview_Deck_113015.pdf

3. Heavy Reading, 2015, Roz Roseboro, OPNFV Survey Results, available at:
   http://events.linuxfoundation.org/sites/events/files/slides/OPNFV%20survey%20results%20presentation-%20R%20Roseboro_0.pdf

4. OPNFV, available at:
   http://www.opnfv.org

5. OPNFV, About, available at:
   https://www.opnfv.org/about

6. OPNFV, Members, available at:
   https://www.opnfv.org/about/memberslist

7. OPNFV, Project: Service Function Chaining, available at:
   https://wiki.opnfv.org/service_function_chaining