Industry Report

2017 Open Source in Networking Report

The Trusted News and Resource Site for SDx, SDN, NFV, Cloud and Virtualization Infrastructure

Table of Contents

contents

contents

96%

OF IT PROFESSIONALS CONSIDER OPEN NETWORKING TO BE A RELEVANT BUYING CRITERION

– 2017 GARTNER DATA CENTER NETWORKING MAGIC QUADRANT

OPEN SOURCE REVOLVES AROUND LINUX.
CUMULUS IS BUILT ON LINUX.
THE PERFECT FIT.

LEARN MORE AT
CUMULUSNETWORKS.COM/LINUX

CUMULUS

DELL EMC

OS10
from Dell EMC Networking

Modern software for modern operations.

Open networking from Dell EMC

Make the switch.

Save The Date

ons

OPEN NETWORKING SUMMIT

Integrate // Automate // Accelerate
March 26-29, 2018 // Los Angeles, CA

JOIN THE NETWORKING INDUSTRY ECOSYSTEM

of executives, developers and IT architects across:

- // Enterprises, carriers and cloud service providers
- // Vendors, start-ups and investors
- // Open source and open standards projects in SDN/NFV

To discuss innovations in networking and orchestration to shape the future of open networking.

Key Themes from ONS 2017

Read the whitepaper here:

bit.ly/2hgAPvF

Join us in 2018 to continue the journey!

OPENNETSUMMIT.ORG

Produced by:
THE LINUX FOUNDATION

Executive Summary

With the advent of SDN and NFV, open source in networking has risen to prominence. From SDN controllers, to virtual switches and routers, to network operating systems on white box switches, even to the orchestration and monitoring systems, open source now plays a major role. Open source has even spread to hardware platforms with the rise of the Open Compute Project (OCP). These rapid changes are pushing traditional proprietary networking vendors to find ways to continue to provide value to their customers while figuring out how to embrace open source themselves.

Open source in networking isn't new, with open-source routing stacks like Quagga having been around for many years, network intrusion systems like Snort (released in 1998) gaining prominence more than 10 years ago and open-source BSD UNIX variants used as network operating systems by major networking vendors. However, SDN and NFV have pushed open source to the forefront and we are now faced with a burgeoning open-source ecosystem and new market dynamics that cloud service providers, communication service providers, enterprises, system integrators and technology vendors are all working hard at figuring out.

For all networking stakeholders, understanding best practices in open source, including how to work with prominent foundations like Linux Foundation, OpenStack Foundation, and Apache Foundation are now critical to business success. Likewise, understanding how to incorporate, contribute to, and benefit from open source projects has become a critical core competence for all organizations.

Finally, to benefit from open-source projects, organizations will need to get a handle on the vast library of projects being worked on, across all aspects of the networking ecosystem, including hardware, embedded software, operating systems and networking applications. While not exhaustive, this report attempts to capture the more prominent and popular open-source projects that impact networking and should serve as an initial map into the world of open source in networking.

Introduction to Open Source in Networking

Networking has been transformed with the advent of SDN and NFV. What had traditionally been a closed and proprietary environment dominated by a few vendors has opened up to innovation and a much more rapid pace of development than in the past decades. Developments in cloud architectures, virtualization and the open-source movement within application stacks and operating systems have spilled into networking.

As a result, open source has become an important part of the new networking landscape, with open source driving SDN controllers and NFV infrastructure, as well as the white box platforms making up the physical infrastructure. The SDxCentral Open Source in Networking Report aims to provide you with a primer in open-source networking, covering a little of the history of how we got here, the economics and drivers behind open-source networking, best practices in open-source networking, trends and evolution of the market and a quick map of the top open source projects impacting networking today.

We put together this report with input from members of the SDxCentral community, original research from the SDxCentral research team, and survey results from our audience polls and hope it provides you with a useful map of the open-source networking landscape today and a window into what's to come.

Why Open Source for Networking?

Open-source software has played a significant role in networking for over two decades, but has emerged as particularly important to the unprecedented industry transformation underway catalyzed by SDN and NFV. Open networking requires open source, and the industry has responded with a rapidly growing set of open-source networking and orchestration projects.

What is open source? Like so many networking terms (e.g., services, applications, virtualization, and even network, among many others), and tens of millions of projects, open source has taken on a number of meanings by various organizations. Our definition presents open source in the context of networking, which is characterized by the following attributes:

- A distribution model for software (and increasingly, hardware as well)
- Inclusive, with low barriers for a broad range of contributors to participate
- Deliverables are readily accessible, and consumers are free to distribute the code
- Sustainability, propelled by strong community support

For many, open source is synonymous with 'free', including no cost, and unrestricted access to source code. While most open-source networking projects do not entail a license fee, adopters of open source incur significant cost in integrating, testing, adapting, and maintaining open source code for real-world solutions. Adopters are subject to the terms of the many different open source license agreements governing the use of the code. Contributors are also subject to license agreements as well, typically imposed by the project.

Other misconceptions are summarized in Table 1 below

Myth	Reality
"Open source is Free"	*Fact:* Significant cost is required to adopt, test and maintain open source (like any software!)
"Open source spells the end for vendors"	*Fact:* Open source presents a number of opportunities for vendors to capitalize upon their core competencies
"Open source is created by individuals who are passionate about the code"	*Fact:* Many open-source networking projects are backed by corporate entities that fund the developers contributing to the project
"Open source is more vulnerable than proprietary software"	*Fact:* Open source is usually more secure than proprietary software because of the extensive vetting by the developers and public visibility of every line of code
"Open source is not mission critical"	*Fact:* Open source is integral to virtually all large-scale software systems and popular projects are more tested and scrutinized than proprietary software.
"Open source replaces proprietary software."	*Fact:* While open source replaces some proprietary software, it enables software developers to focus on proprietary, high-value software. Virtually all proprietary platforms today adopt significant open source components.

Table I Open Source Myths (and realities)

Open Source Taxonomy

As the telecommunications and cable industry increasingly relies upon open source for networking, a number of different projects have been established to address a wide range of requirements. In Figure 1, we present an open-source networking project taxonomy based on the desired outcomes, including some well-known examples.

Platform projects are largely scale open source frameworks that provide extensive functionality and integrate with a broad range of external *components* (see below). Examples include the Open Network Automation Platform (ONAP) orchestration and automation platform, **OpenStack** cloud operating environment, and **OpenDaylight** SDN controller framework. Platforms are typically large projects with dozens of members, and are intended for large-scale deployment.

Component projects provide more targeted functionality and are typically integrated into platforms to implement a broad solution. Examples include:

- OpenvSwitch (**OvS**) virtual switch
- Fast Data IO (**FD.io**), which enhances the data plane switching performance
- **OpenSwitch** embedded white box operating system
- Open Network Install Environment (**ONIE**), boot loader for embedded network operating systems
- Open Networking Operating System (**ONOS**), an embedded SDN controller adopted by the Central Office Re-architected as a Data Center (**CORD**) initiative.

Other open source component projects include VNFs such as OpenVPN, BIND (open source DNS), **freeRADIUS** (AAA server), SSHD (secure shell), BRO and Suricata (open-source IDS systems) and many others.

Open Source Reference Platforms projects typically provide a testbed platform for the software development lifecycle. Most implement aspects of the SDN and/or NFV reference architectures, such as Open Platform for NFV (**OPNFV**), which is aligned with the **ETSI NFV Architectural Framework**, **MEF** Open Lifecycle Services Orchestration (**OpenLSO**) platform (implementing **LSO**), and the **Open CORD project**, which aims to build a reference implementation of **CORD**.

Open source reference platforms are not intended to be directly deployable (but may be), but rather are to assist in development (testing, integration, validation), promotion (Proof of Concepts (PoCs), Demonstrations, and technology explorations.

Reference Code projects enable investigation of new concepts, validations, API development, and new capabilities through limited-scope open source development. As part of the overall SDN movement, the ONF established a forum for reference code projects, opensourcesdn.org, which seeded and kickstarted many **projects** which facilitated standards development, open source contributions, validation, and demonstrations.

Reference code is not intended to be directly deployable, but rather be channeled into open source component and platform projects.

As well, this taxonomy is not intended to be exhaustive, but rather illustrative of open-source networking projects. As momentum continues to grow, many other types of projects are likely to emerge.

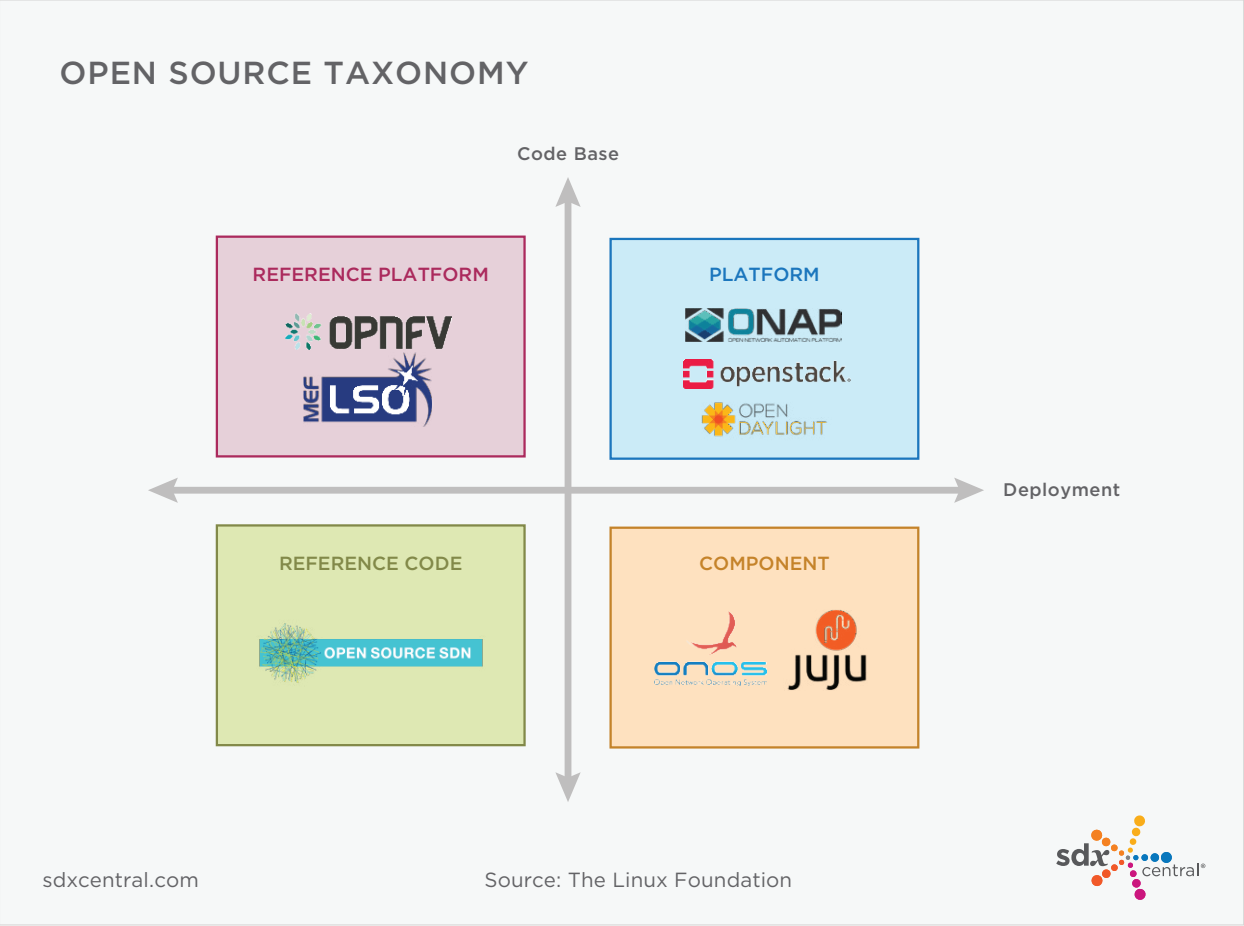


Figure 1

Open-Source Networking Project Lifecycle

Open-source networking projects share a great deal in common with more general, large-scale open-source projects. However, to serve the interests of the world's largest network operators spanning telecommunications, cable, enterprise, and cloud, there are distinct differences in some of the more comprehensive projects:

- Adopters of open-source networking projects find these projects inherently fit into a broader, more integrated ecosystem than being standalone
- Open-source networking projects, especially as they move up the stack, motivate more direct participation by network operators in addition to vendors
- Large-scale network operators expect that open source platforms and components are aligned reference architectures and comply with relevant standards
- Open-source networking projects must address how network operators can migrate from their vast installed bases to realize the new technology vision

These properties influence the open source project lifecycle, governance, organization, and operations. Figure 2 depicts a high-level lifecycle model.

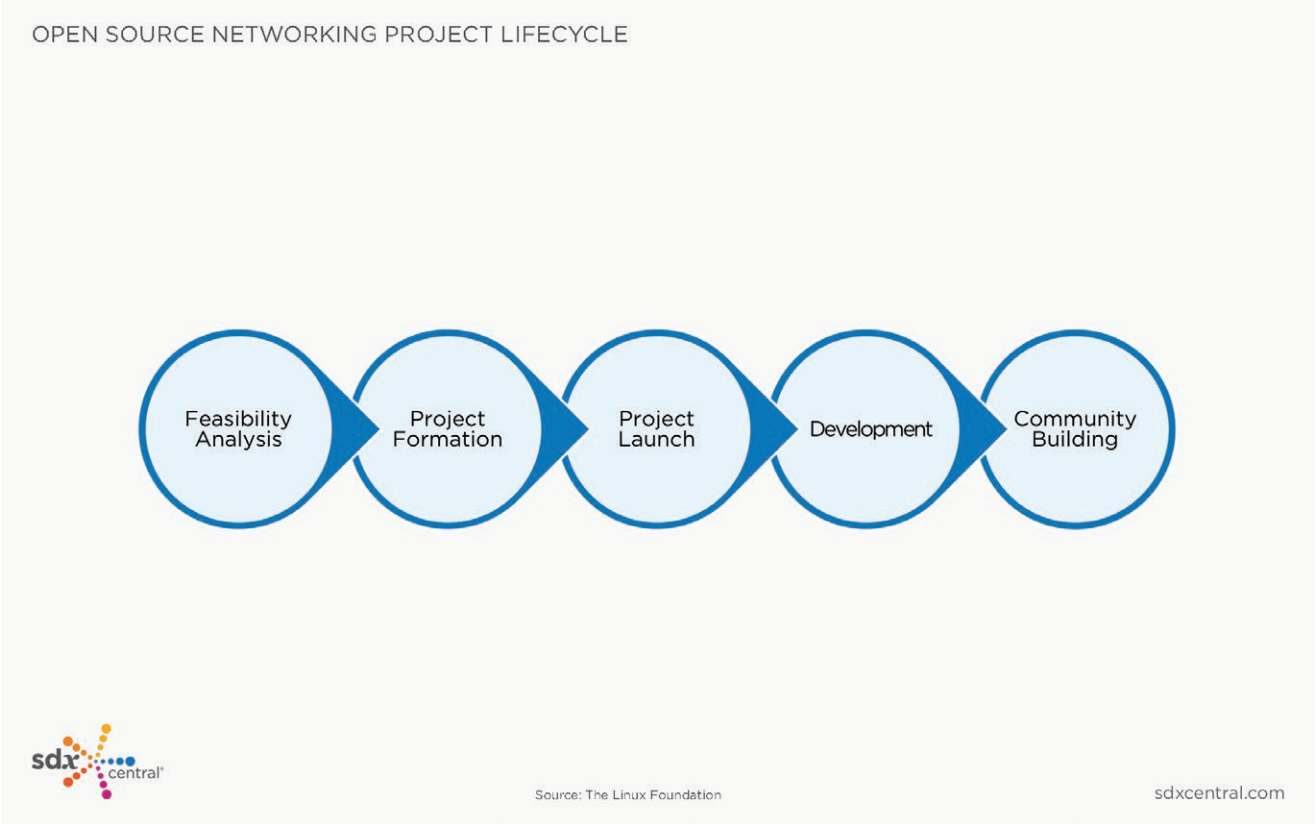


Figure 2

The project lifecycle has evolved over recent years, and is particularly appropriate for the higher level projects such as the Linux Foundation open orchestration project ONAP or OPNFV. This model has been adapted as necessary for the specific needs of individual projects, and consists of the following phases:

- **Feasibility Analysis** - Establish project feasibility, including a common problem statement and high-level scope definition, initial interest expressed by organizations interested in contributing, and statement on how this projects fits with the broader networking ecosystem.
- **Project Formation** - Bring the interested parties together, refine the project scope, forge the governance model, and discuss how the project will operate. Determine what seed code interested organizations are considering to contribute.
- **Project Launch** - Formally create the project, typically within an open source software foundation, including the project organization, governance model, membership, budget, development infrastructure and tools, and plans.

- **Development** - Converge on the functional architecture, contribute and integrate the seed code, and define initial use cases to focus on the initial release(s). Over time, create a roadmap, release plan, and develop, integrate, and test new capabilities, including documentation, tools, etc. to facilitate adoption.
- **Community Building** - Educate, promote, evangelize, and sustain the community through a variety of means. Among them: Hackfests/Hackathons, to ramp up developers, End User Advisory group to validate the approach, user groups to broaden awareness and interest, event participation, social media, market education/marketing, etc. The goal is to cultivate a sustainable community for the long term.

Open Networking Evolution

Open-source Networking Before SDN

While there has been significant coverage of SDN and the open source movement behind SDN, the reality is that open source existed within and contributed to networking in significant ways before SDN. In the early days of the Internet, many of the key Internet infrastructure applications such as Paul Vixie's BIND (DNS) and Eric Allman's Sendmail were open-source. Likewise, the BSD UNIX operating system and its derivatives including FreeBSD, OpenBSD, NetBSD powered a good number of network appliances and routers in the early days of the Internet—often as private modified versions that were shipped as proprietary products. And of course, Linux and its variants were also used in embedded networking devices and today continue to feature prominently on many open-source whitebox platforms.

On the networking security side, Martin Roesch's SNORT rose to prominence in the IDS (Intrusion Detection System) realm, and was eventually acquired by Cisco. And today, BRO and Suricata join its ranks, along with many other open-source projects in the networking security realm.

And in routing, projects like XORP (eXtensible Open Routing Platform), BIRD, Quagga evolved, with Quagga gaining prominence within many large cloud service providers, and Quagga now sees its future in the Free-Range Routing (FRR) project under the auspices of the Linux Foundation.

A Brief History of SDN and NFV

Circa 2007-2008, Software Defined Networking (SDN) ushered a new era of open networking with **fundamental concepts originating** in academia. At that time, the focus was on **OpenFlow**, an emerging standard control interface and switch abstraction to enable SDN Control.

SDN (and OpenFlow) was propelled to the forefront by the **formation of the Open Networking Foundation (ONF)** in 2011. The ONF forged the familiar **high-level SDN model** that characterized Software Defined Networking by the following attributes:

- Logically centralized control
- Network programmability and intelligence
- Abstraction
- Openness

SDN was defined more rigorously by the **ONF SDN architecture framework**, which motivated operators, solution providers, software innovators, and integrators alike to use SDN to transform their networks.

Network Functions Virtualization (**NFV**) was introduced in 2012 by a series of progressive operators who from the outset recognized the need for close collaboration to realize the vision for virtualizing the carrier network. The NFV Industry Specification Group was formed in the European Telecommunications Standards Institute (**ETSI**) just after the **seminal white paper** was released in October, 2012. Today there are close to 40 operators participating in the ETSI NFV ISG, which engaged in the third phase of its work program.

While emerging as distinct initiatives, many (particularly those in the telecommunications and cable industries) recognized that SDN and NFV are quite complementary. Whereas SDN virtualized the network infrastructure, and introduced intelligent connectivity services, NFV focused on virtualizing the higher layers, leveraging the cloud computing model. Bring the two initiatives together allowed network functions to be deployed where needed (and not just in a cloud data center), under SDN control.

Early innovators in leading industry groups such as the ETSI NFV ISG, ONF, **CableLabs**, along with more traditional standards development organizations including **MEF**, **3GPP**, **TMForum**, **BBF**, among others, recognized that open

source would play a vital role in accelerating NFV adoption, and embarked upon a series of initiatives within their respective bodies.

Among them, MEF created LifeCycle Services Orchestration (**LSO**), a reference architecture that addresses how telecommunications carriers can enhance their back-end operational systems, and automate new service delivery. Other efforts including TMForum's Zero-touch Orchestration, Operations and Management (**ZOOM**) project, **BBF 20/20** vision, and the 3GPP's 5G specifications all acknowledge open source as a vital aspect of technology and business transformation.

Closely related to the whole SDN and NFV movement was also the concept of disaggregation (white box movement). Just as mainframes and minicomputers shifted to PCs and x86 servers, proprietary networking gear was split apart and separated into horizontal pieces: hardware, OS, network application. Open-source enabled a new open ecosystem across this stack.

Why Open-Source Networking Became Prominent

Open source has evolved as a viable delivery model for decades, led by the **Linux revolution**. While many network operators have adopted open source for myriad applications, open-source networking is relatively new.

The advent of the Cloud set the stage for networking to come together with computing and storage, domains where open source have been widely adopted. Hyperscale cloud operators disrupted the entire technology industry by rethinking the operational model, capitalizing upon economies of scale and exploited open source to increase agility while driving down costs.

As early innovators, the cloud operators contributed select developments into the open community, offering a head start for network operators whose networks have traditionally built upon proprietary technologies.

Another major influence fueling open-source networking was the emergence of SDN, disaggregation and white box trends, followed by NFV. This move resulted in not only open source, but also open standards, open APIs, and a vision towards an open ecosystem.

The first large-scale open-source networking project was driven by SDN. OpenDaylight (ODL) was formed in 2013, and today has dozens of deployments and hundreds of contributors. ODL also opened the door to many other open-source networking projects, with more being introduced every month. A summary of key open-source networking projects will be presented in this report.

NFV represents an even more direct approach to leverage the power of the cloud. Though initially a telecommunications initiative in ETSI, the concept of virtualizing network functions was first embraced by the cloud community, and has been gaining acceptance in Enterprise as well.

From the outset, the ETSI NFV leaders became strong proponents for open source, which led to the formation of the Linux Foundation OPNFV project, which they envisioned would validate the NFV Infrastructure, thus accelerating NFV adoption.

The Role of the Open Source Foundations

One important factor that has contributed significantly to the growth of open-source networking is the rise of Open Source Foundations (OSF). While there are many, three of the most prominent in networking include:

- **Linux Foundation**
- **OpenStack Foundation**
- **Apache Software Foundation**

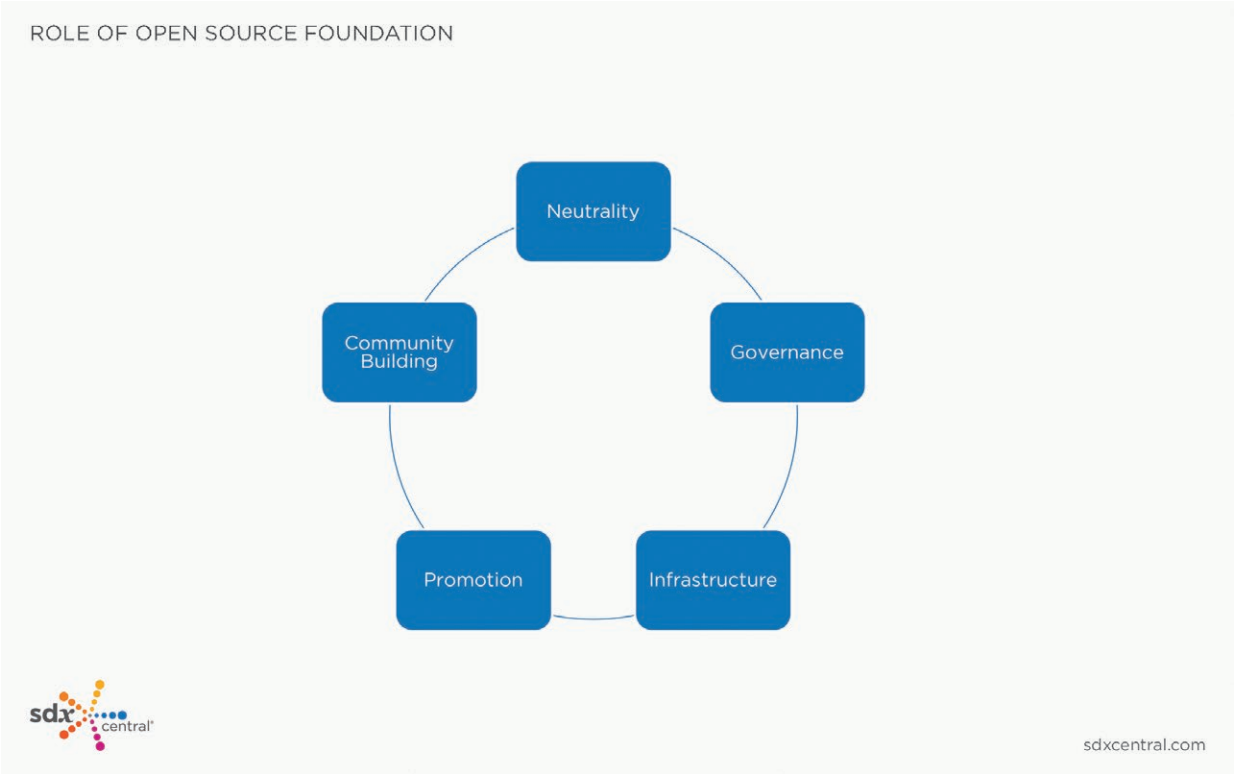


Figure 3

Figure 3 summarizes the key assets offered by OSFs to enable open-source networking projects.

- **Neutrality** - First and foremost, OSFs must provide an autonomous forum that is unbiased, and insulated from the commercial interests of its members and the community at large. The overarching goal is to maintain an inclusive and open forum that encourages diversity of opinion and participation, yet continues to progress.
- **Governance** - OSFs provide the legal and organizational expertise to allow projects to address their goals, while complying with applicable laws and standards. OSFs such as the Linux and Apache Foundation have organized **dozens of projects**, and asserted best practices based on extensive experience.
- **Infrastructure** - OSFs provide the development, marketing, and related infrastructure to support projects throughout their entire project lifecycle. Most have a combination of internal infrastructure and cloud services to provide the scalability and flexibility needed to support diverse projects.
- **Promotion** - With (tens of) millions of open source projects, it is critical to stand out in the crowd. OSFs provide varying degrees of marketing services to help position, promote, and market projects, news, new members, etc.
- **Community Building** - Perhaps the most important aspect of any large-scale open source project is the ability to effectively establish a sustainable community. This requires specialized expertise, presence, tools, and leadership to attract not only project members, but a vibrant development community, and many other participants who together can sustain a project over the long haul.

While OSFs are not a pre-requisite for a successful open-source project, they can be a critical factor in creating momentum and longevity for larger projects. Many projects incubated in other organizations (research organizations, universities and even private companies) often "grow-up" and migrate to OSFs.

A Look Back - Open Virtual Switch (OvS)

One of the first and most successful open-source networking projects is OpenvSwitch (**OvS**). Originally established in 2009 by the Nicira team (subsequently acquired by VMware), OpenvSwitch provided an open source virtual switching alternative to proprietary switches as server virtualization gained momentum. Prior to OvS, virtualized networking was relatively simple, providing Layer 2 connectivity (and standard **IEEE 802.1Q VLAN** support) and limited communications functionality. OvS experienced tremendous growth with the advent of cloud, along with a corresponding expansion of the feature set for multi-layer, virtualized networking.

Today, OvS is the default open-source (Apache 2.0 license) virtual switch for a number of open source and proprietary virtualization environments, including Xen and KVM, and is supported by most cloud environments including OpenStack. In the summer of 2016, **OpenvSwitch was moved to the Linux Foundation's** open networking and orchestration portfolio.

Open Source SDN and Rise of the Controllers

Open-source SDN controllers have been evolving since the beginnings of SDN.

(Refer to the link **here** for a brief history of SDN Controllers.)

SDN Controllers provide logically centralized control over both programmable devices, as well as traditional managed devices. Not surprisingly, introduction of the SDN Controller catalyzed a hotbed of open source projects:

- **OpenFlow**, which was contributed to the ONF in 2011, provides an open programmable interface and logical switch abstraction for programmability. While originally developed in academia, administration for OpenFlow was transferred to the Open Networking Foundation when it was founded in 2011, and served as the catalyst for open source SDN controller development.
- **NOX**, generally acknowledged as the first SDN/OpenFlow Controller, developed by Nicira Networks (acquired by VMware in 2013)
- **Beacon**, an open source, Java-based SDN/OpenFlow controller developed by Stanford University.
- **FloodLight**, another prominent SDN/OpenFlow Controller, developed by Big Switch Networks. FloodLight was a fork of the Beacon code base, and emerged as the basis of many commercial SDN Controllers
- **OpenDaylight**, the first Linux Foundation open-source networking project, which is widely supported by a strong community of developer, and deployed in dozens of production installations.
- **ONOS**, an OpenFlow controller developed by ON.LAB (**recently merged into the Open Networking Foundation**), which was targeted towards telecommunications applications. ONOS is now an integral element of the CORD architecture.
- **Ryu**, the first open SDN controller contributed by a telecommunications operator, NTT (Japan). Ryu has been widely investigated and utilized in the research community, especially in Asia.

As SDN emerged as a viable architecture for the data center, telecommunications/cable, and enterprise, vendors responded with several proprietary alternatives, including:

- **NEC** - introduced the **Programmable Flow Controller**, the first commercial SDN controller in 2011
- **Cisco** - introduced a number of controllers, including the Cisco Open Network Environment Controller, which is integral to the Cisco **ONE** architecture, the Extensible Network Controller (**XNC**), based on OpenDaylight, and Application Policy Infrastructure Controller (**APIC**) Controller (APIC), the SDN controller in Cisco's Application-Centric Infrastructure (**ACI**) architecture.
- **HPE** - HPE introduced the Virtualization Application Network (**VAN**) controller, which adopted the Beacon

open source framework. HP replaced the VAN controller with an SDN controller based on OpenDaylight.

- **Juniper** - Juniper's SDN controller was based on technology acquired from Contrail. Juniper now offers an open source version (**OpenContrail**) and commercial **Contrail** Controller.

As SDN continues to gain traction, a growing number of commercially available SDN Controllers have **adopted OpenDaylight**, which is emerging as the market leading open source controller framework.

Network Functions Virtualization and Open Source

Since the introduction of NFV in 2012, the world's leading telecommunications and cable operators sought to exploit the pervasive benefits of virtualization and the cloud. Hyperscale cloud operators, including "The Big 4": Amazon, Google, Microsoft, and Facebook. The corresponding cloud giants in China, BAT (Baidu, Alibaba, and Tencent), have been leveraging open source for years, and not surprisingly aggressively adopted open-source networking projects to accelerate their development activities.

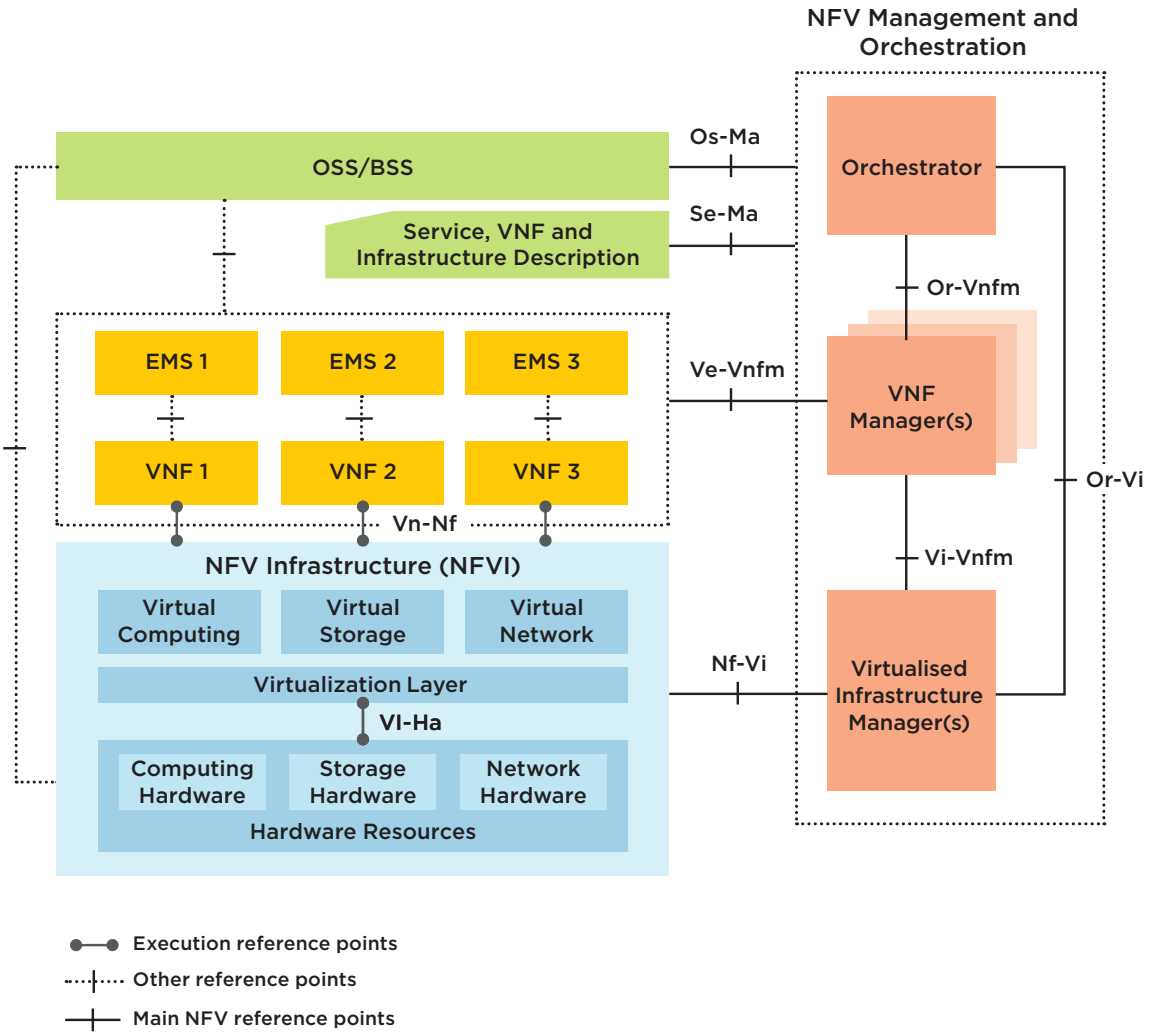


Figure 4: ETSI NFV NFV Reference Architecture
Source: **ETSI GS NFV 002 - V1.1.1**

As the telecommunications/cable industry observed the rapid progress in the cloud, they sought to transform their internal infrastructures to exploit the many benefits of virtualization, including agility to improve service velocity, while automating operations, and optimizing infrastructure. Along the way, they embraced open source, which was reflected in the **NFV Architectural Framework** (see Figure 4).

Major elements of the NFV architectural framework include:

- **NFV Infrastructure** (NFV-I), which encompasses the virtualization environment for Virtualized Network Functions (VNFs). NFV-I consists of the virtualized resources, as well as the physical resources for compute, storage, and networking.
- **Management and Orchestration** (MANO), which is responsible for the orchestration and management for the virtualized network functions and NFV-I, as defined in the **ETSI NFV MANO specification**.

For a more detailed description of NFV, visit the **What is NFV** page, or the **ETSI NFV site**.

NFV Infrastructure and Open Source

From the outset, the ETSI NFV leadership (especially the operators) recognized the need to cultivate a vibrant open NFV ecosystem to facilitate and accelerate NFV adoption. This entailed multi-vendor deployments that include both open source and proprietary building blocks that could be readily integrated together.

In 2014, the OPNFV project was established in the Linux Foundation. OPNFV was an open-source reference platform designed to validate NFV concepts, efficiently onboard new components and platforms, and establish an extensible methodology for testing and integration.

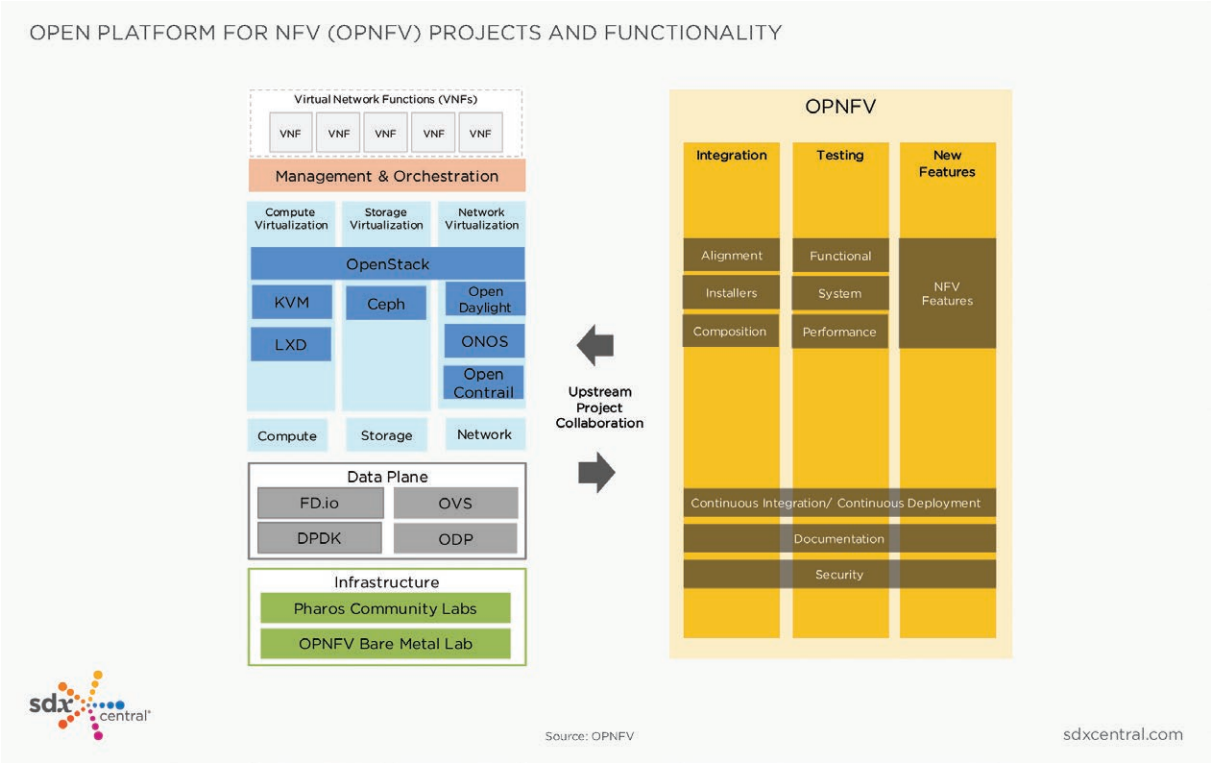


Figure 5
Source: **https://www.opnfv.org/software**

The project also served as a testbed for integrating common open source building blocks for the Cloud environment, as illustrated in Figure 5. Initially, OPNFV focused on the NFV-I layer, reasoning that the significant MANO complexity could not adequately be addressed until a solid NFV-I foundation was established. Many so-called open source upstream projects have been integrated using OPNFV, as depicted in Figure 4, including hypervisors, virtual disks, SDN controllers, data plane acceleration, etc.

Orchestration, Automation and Open Source

Within the telecommunications world, Service Orchestration provides model-driven, end-to-end connectivity across diverse network domains, technology, and multi-vendor hardware and software. Orchestration resides near the top of the stack, and typically will integrate with operator and end-user portals, carrier back-end systems (both Operational Support Systems (OSS) and Business Support Systems (BSS), as well as operator-specific applications.

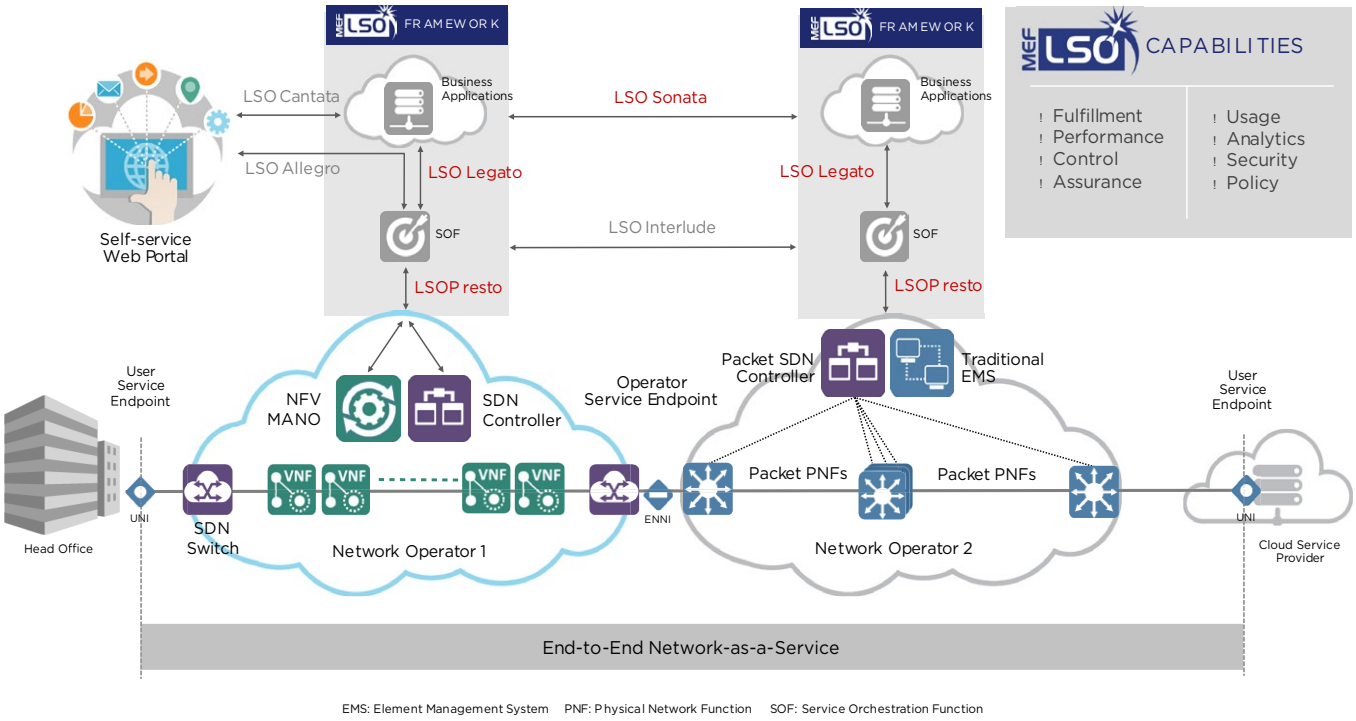


Figure 6
Source: **MEF LSO Capbilities Presentation**

The MEF LifeCycle Services Orchestration (LSO) is an orchestration reference architecture based on SDN and NFV, and depicted in Figure 6. Orchestration inherently integrates with a range of components and platforms, and serves to abstract Intent-based service requests from the underlying implementation.

One important aspect of service orchestration is the ETSI NFV ISG defined Management and Orchestration (MANO), which is an integral element of the NFV Architectural Framework introduced in Figure 3, and **explained on SDxCentral**.

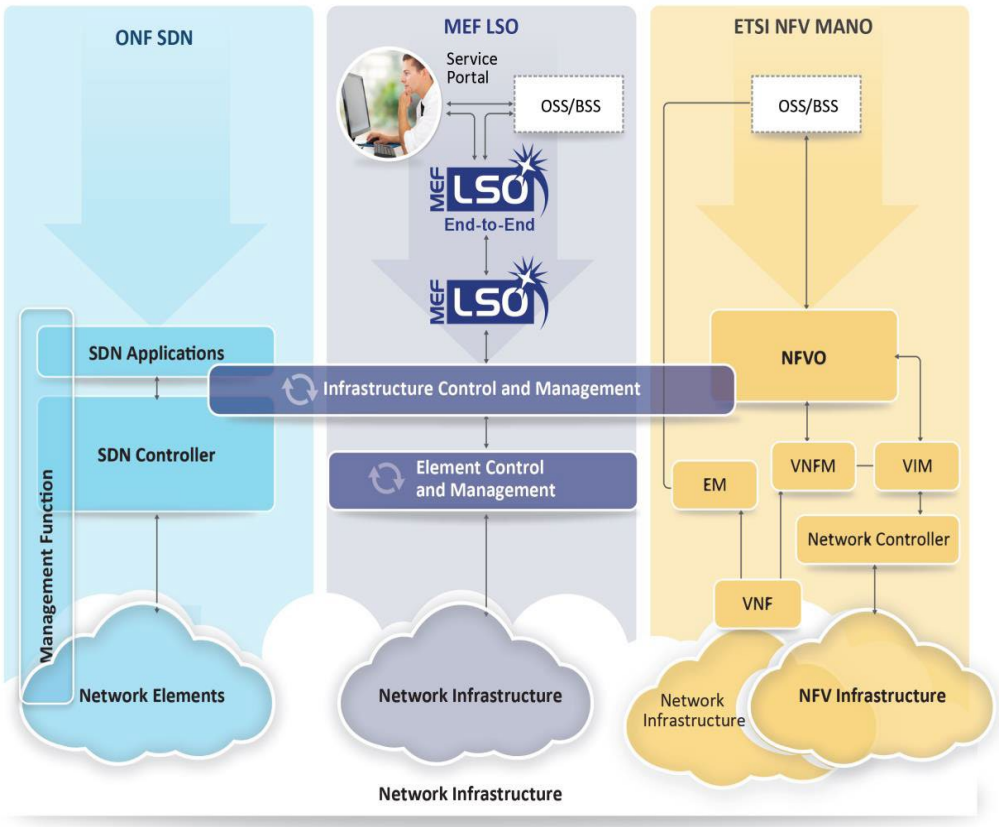


Figure 7
Source: **MEF Lifecycle Orchestration Vision Whitepaper 2015**

NFV MANO is a subset of the end-to-end service orchestration requirements captured in the MEF Lifecycle Services Orchestration (LSO) reference architecture. Figure 7 illustrates the role of ETSI NFV MANO in the broader LSO reference architecture. MANO encompasses management of the Virtualized Network Functions running in the NFV-Infrastructure, including the Virtualized Infrastructure Manager (**VIM**), VNF Lifecycle Manager (**VNFM**), and Resource Orchestrator. All of these functions have been implemented in the open-source community (examples below):

- VIM: OpenStack, **CloudStack**, **Kubernetes** (future), and others
- VNFM: **JuJu**, **OpenStack Tacker**, and others

Beginning in 2016, the open source community embraced orchestration with the introduction of a number of key initiatives. At Mobile World Congress2016 (Feb 22-26, 2016; Barcelona, Spain):

- The **Linux Foundation announced the Open Orchestrator** (**OPEN-O**) project
- **ETSI announced Open Source MANO** (**OSM**) project
- **AT&T expressed their intent** to contribute their Enhanced Control, Management and Policy (**ECOMP**) code into the open source community
- The **OPNFV project agreed to expand its charter** from NFV-I to address orchestration

OPEN-O was focused on providing an end-to-end service orchestration platform that bridged the gap between SDN and NFV. OSM was primarily focused on validating the ETSI NFV MANO specifications, and attracted many ETSI members. A comparison of the scope of the projects is provided **here**.

In early 2017 the **Linux Foundation announced** that the OPEN-O project was to be merged with AT&T's opens source ECOMP code base to form the Open Network Automation Platform (**ONAP**) project. ONAP was established to avoid open source fragmentation, and to achieve critical mass with among the largest operators and solution providers. ONAP has grown to include a dozen major operators spanning the globe, and in excess of 50 total members.

On the Enterprise and Cloud Service Provider side of the market, while OSS systems don't necessarily feature, the equivalent in terms of automation and orchestration systems still exist in a different form. Network element management, automation are still central to running a large scale network. The cloud service providers like AWS, Facebook, LinkedIn, Microsoft Azure, Google Cloud Platform, etc have their own home-grown orchestration systems while enterprises turn to automation and DevOps frameworks like Puppet, Chef, SaltStack and Ansible (RedHat) all of which have open-source versions of their products.

Open Source Hardware and Disaggregation

While distinct and separate from SDN and NFV, the market has often tied the concept of the white box to SDN and NFV. A white box is essentially a generic platform (x86 or ARM architecture coupled with merchant ASICs for packet switching and sometimes lower-cost optical components) with separate network OS and network stack (L2, L3 etc). With the move to SDN and NFV, the disaggregation of previous proprietary network appliances was made possible and new vendors moved in to take advantage of the disaggregation of the platform. These white box platforms often feature open-source network OS and open-source routing and switching stacks, with the ability sometimes to even mix and match hardware and software stacks.

And the open source movement in networking did not stop with software stacks and orchestration. With the Open Compute Project (OCP), the industry has come together to help define standardized hardware infrastructure for networking products. Some examples here include the innovative Wedge and 6-Pack switches driven by FaceBook within the OCP, suited for large-scale data center deployments. The OCP is also the home of other open-source hardware projects for server and rack designs. Associated with the hardware switches are a couple of important open-source projects, including ONIE (Open Network Install Environment), a network device pre-boot environment that allows the device to choose from and load an appropriate network OS, and ONL (Open Network Linux) a Linux distribution for "bare metal" switches. Also of importance is the Switch Abstraction Interface (SAI) specification, which provides a C API to program ASICS on a network switch, which has been adopted by vendors including Broadcom, Cavium and Marvell.

The Evolving Business Perspective on Open-source networking

Open-source networking and orchestration, not surprisingly, emerged as both SDN and NFV have progressed to open up the network. However, the inherent complexity of networking has made it more challenging to adopt open source to the same extent as it has been in the server and storage domains.

The rapid growth of open-source networking projects that now span the entire stack, places new emphasis on the business case. Vendors will not contribute to open source development without being rewarded for their investments. Major network operators are also actively investigating how they can benefit from SDN, NFV, and the opening up of the network.

As the large-scale open-source networking and orchestration projects continue to mature, the industry is expected to unleash unprecedented innovation to address these challenges, even though not all firms will benefit equally. The commercial landscape is expected to adapt to the new business realities in ways that are difficult to predict.

The Role of Open Source in Networking

The Changing Role of Standards

With the realization that **software is eating the (networking) world**, open source will play a major role in virtually all large-scale applications (and many small ones). Consequently, the technology adoption lifecycle is being revamped to reflect the role of open source in networking.

Traditionally, standards drove new technology adoption for telecommunications and networking, with venerable bodies including the ITU, 3GPP, IETF, IEEE, among others publishing the standards that defined a range of technologies with the intent of achieving multi-vendor interoperability. Such efforts required years from specifications to implementation, and were most suitable for hardware, where the cost of rework remains high, and updates are infrequent.

The familiar waterfall model breaks down for software-centric solutions, especially as the update cycle is converging on continuous, and systems are designed to be tailored into distinct environments. A more iterative lifecycle is needed that blends specification with implementation and accelerates the overall process. While a radical change is necessary, the end goal remains the same: multi-vendor interoperability.

Figure 8 illustrates the traditional technology adoption cycle (in the outer ring in blue) along with the proposed new technology lifecycle (inner ring), which turns the model upside down.

Challenges with Open Source

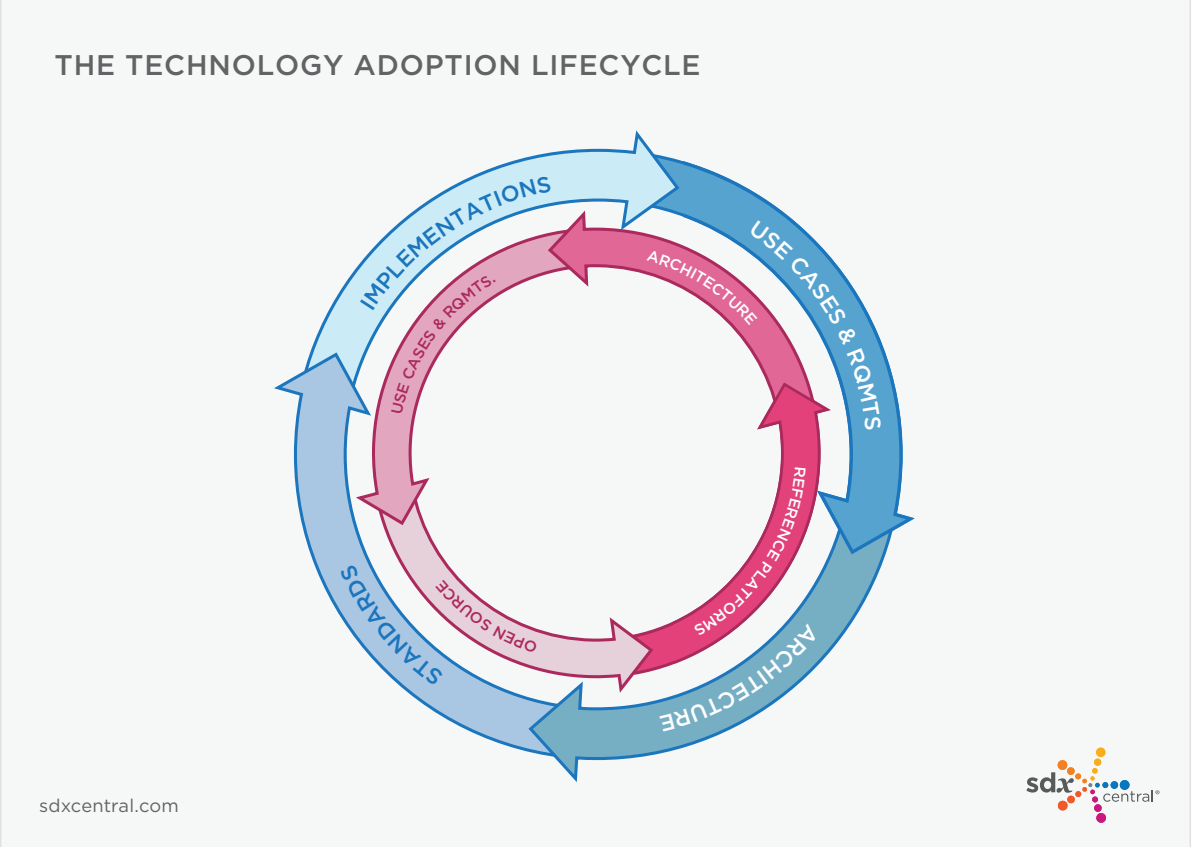


Figure 8

In this model, standards guide the scope and integration for open source components and platforms, while open source serves to validate the standards as a necessary step towards implementing them. One could not imagine standardizing the full complement of functionality for large-scale platforms such as ONAP, OpenStack, OpenDaylight. However, standardizing select APIs to facilitate integration is increasingly important, along with the certification programs to validate the integrations.

Open Source Evolution

In the early days of open source—think Linux or BSD or Apache httpd—there was not a formal methodology to guide the projects. Engineers and developers who were self-motivated contributed, and the magic of collaborative innovation transformed the individual outputs into open source building blocks that reshaped the entire software industry.

Over the past twenty five years the approach, tools, and mindset in the open source community has evolved. We are in the midst of another change as the telecommunications and networking industry leaps into open source.

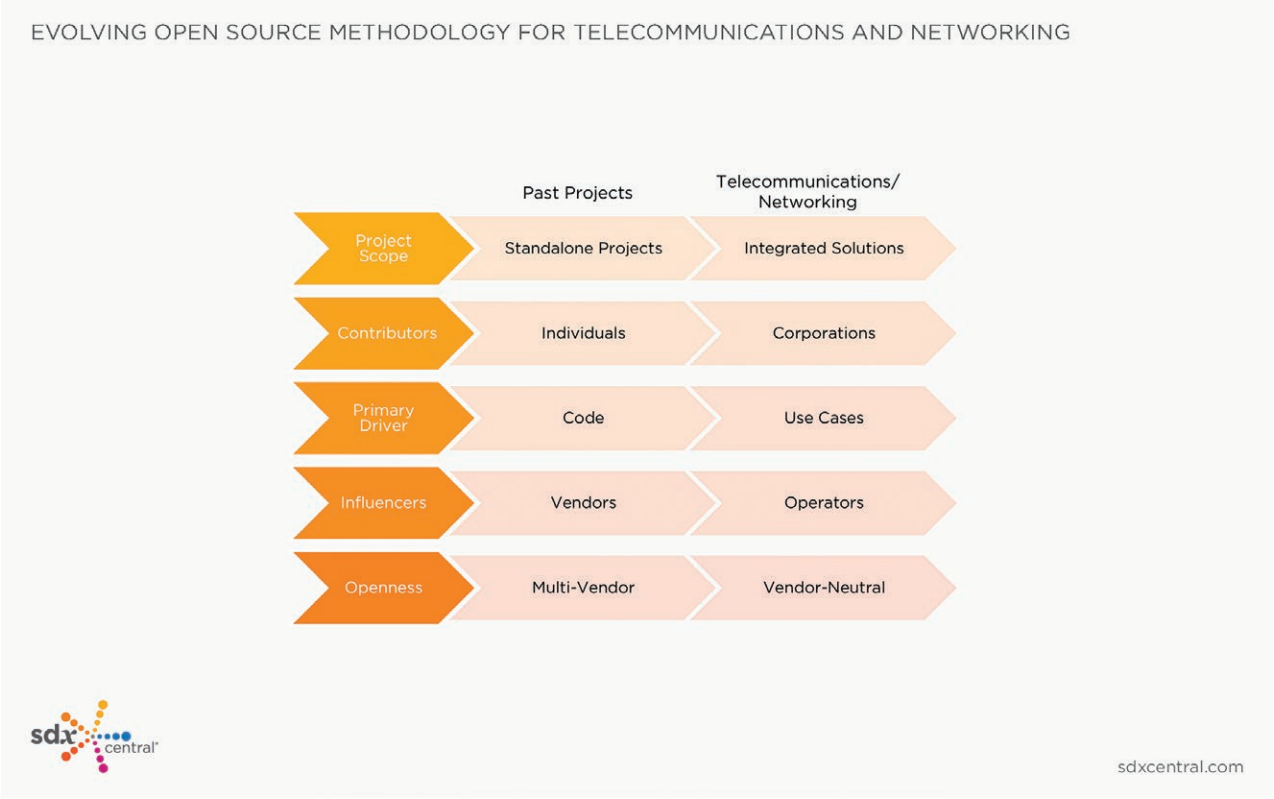


Figure 9

Figure 9 summarizes how open source has evolved to enable large-scale platforms for telecommunications and networking:

- **Projects must be integrated** - traditionally, open source projects have been organized and operated autonomously from other projects, albeit with varying degrees of liaisons; as networking is by its nature implies interconnection, open-source networking projects must facilitate integration with the many other projects that collectively provide a solution.

- **Contributors are sponsored** - The days of developers participating on their own have given way to corporate sponsorship. Vendor organizations are beginning to establish open source technology offices (with various names) to address the myriad of issues associated with open source development/adoption.
- **Code is NOT the only factor** - A mantra for open source was summed up by a notable quote "Code is the coin of the realm", attributed to David Meyers, the first Technical Steering Committee chair from the OpenDaylight project. While no doubt code remains critical, network operators who are considering adoption of open source are guiding the development with key use cases, and seek to validate the utility of the code produced.
- **The rise of the operator** - Open source development, like standards, have been driven by vendors on behalf of their network operator customers. Especially now that open source is moving up the stack, operators are becoming directly involved. Whereas OpenDaylight was formed without a single operator member, ONAP was formed with ten members.
- **Neutrality is the ticket** - While there are precedents for many aspects of SDN/NFV, the truly big change is the opening up of the network. Openness does not solely equate to multi-vendor support, but rather independently run projects that are not dominated by a single party.

Open Source: an Enterprise Perspective

Many enterprises have embraced Linux on servers within the data centers, but slow to use open source through the rest of their infrastructure. Even on the virtualization front, VMware still holds sway despite options like OpenStack and KVM. On the networking front, enterprises are less ready for open-source networking, with only the largest enterprises showing an eagerness to run proofs-of-concept within the data center. Having said that, the open source in networking movement has spurred all networking vendors to open up their platforms via open API RESTful interfaces (or equivalent), allowing end-users to automate and script their platforms. And enterprises now have the option of purchasing vendor solutions based on open-source software and hardware designs from vendors like Dell EMC with OS10, Big Switch, Cumulus, Pica8, and HPE, with their Altoline of OCP-compliant switches.

Benefits	Risks
• Many cost-effective solutions utilization open-source • Open APIs that can facilitate customization of networking gear for the business • Reduce vendor lock-in • Enable vendor collaboration	• Limited core competency in software development • No single 'throat to choke' (if purchase disaggregated solutions) • Less control than expected due to complexity of components

Table 1- Open Source Implications to Enterprise

Open Source: The Cloud Service Providers' View

Cloud Service Providers like Amazon's AWS, Google's GCP, Microsoft's Azure are driving the open source in networking movement. Many of them, along with large web properties like FaceBook and LinkedIn, have incubated and seeded many of the major projects. For them, open source is a competitive advantage and an integral part of their infrastructure. They also view open source projects as a way to maintain thought leadership in the ecosystem, as well as a recruiting tool to grab talented engineers and retain them. Nevertheless, the

question of whether creating and maintaining networking software is necessary to run their business will come up and many question whether these are side projects which contribute to distracting them from running their cloud business. For now though, as these companies continue to innovate and share their innovations as open source, the rest of the ecosystem stands to benefit.

Benefits	Challenges
• Opportunity to directly influence the outcome • Exploit shared technology investment and collaborative innovation • Reduce vendor lock-in • Recruit and retain talented engineers • Demonstrate thought leadership • Help drive supplier ecosystem (e.g. OCP)	• Increased technology investments required • No single 'throat to choke' • Dependence on a pool of specialized talent to maintain codebase • Distraction from running core business

Table 2- Open Source Implications to Cloud Service Providers

Open Source: the Telecom Operator's Perspective

Open-source networking is relatively new to the telecommunications industry, fueled by the rapid ascent of SDN and NFV. With the new technology adoption lifecycle (refer to Figure 8), operators have become increasingly involved in the process. The extent of the participation varies by organization, with some operators aggressively building an in-house network software development expertise, while some prefer to rely on their vendors. However, most operators are exploring how they can adopt and deploy solutions based on open source, and building up software expertise to facilitate the transformation under way.

Table 3 below summarizes the operator's perspective on open source.

Benefits	Challenges
• Opportunity to directly influence the outcome • Exploit shared technology investment and collaborative innovation • Reduce vendor lock-in • Enable vendor collaboration	• Limited core competency in software development • Increased technology investments required • No single 'throat to choke' • Less control than expected over open source roadmaps

Table 3- Open Source Implications to Operators

Open Source: The Vendor's Perspective

Network operators have traditionally relied upon their vendors for products and solutions with varying degrees of integration. Open source does not necessarily alter that relationship, even as many operators begin to develop an in-house software development capability (beyond their IT departments).

market summary

Open source offers vendors a low-cost building blocks that provide non-differentiating functionality, particularly where there is a vested interest in a common framework, such as virtual switches, network operating systems, SDN controllers, orchestration platforms, etc. Open-source deliverables and artifacts, however, are targeted towards the development (vs. end-user) community, requiring significant productization prior being suitable for deployment. Since software development expertise is concentrated in the vendors (vs. network operators), vendors have been predominant in the open source community, contributing the vast majority of development resources and project funding. As more network operators build in-house software development expertise, their participation in open source projects will increase, and a balance will be struck that will ultimately enhance the value of the open source deliverables.

Table 4 below summarizes the vendor's perspective on open source.

Benefits	Challenges
• Free up resources that can be redirected to deliver differentiated value • Validate the market for new technologies • Open up new market opportunities that would otherwise be inaccessible • Recruit and retain talented engineers • Forge closer relationships with customers	• Threaten investment in existing products • Must collaborate with the competition • Reduce product stickiness, making it easier for customers to swap out products • Less control over features and product roadmap

Table 4- Open Source Implications to Networking Vendors

Best Practices for Open Source

Now that open-source networking and orchestration is gaining momentum, the collective knowledge base is rapidly growing. Building on the success of each project, best practices for open-source networking are beginning to emerge. Figure 10 illustrates the Big Picture perspective on open source success- It's all in the 'C's':

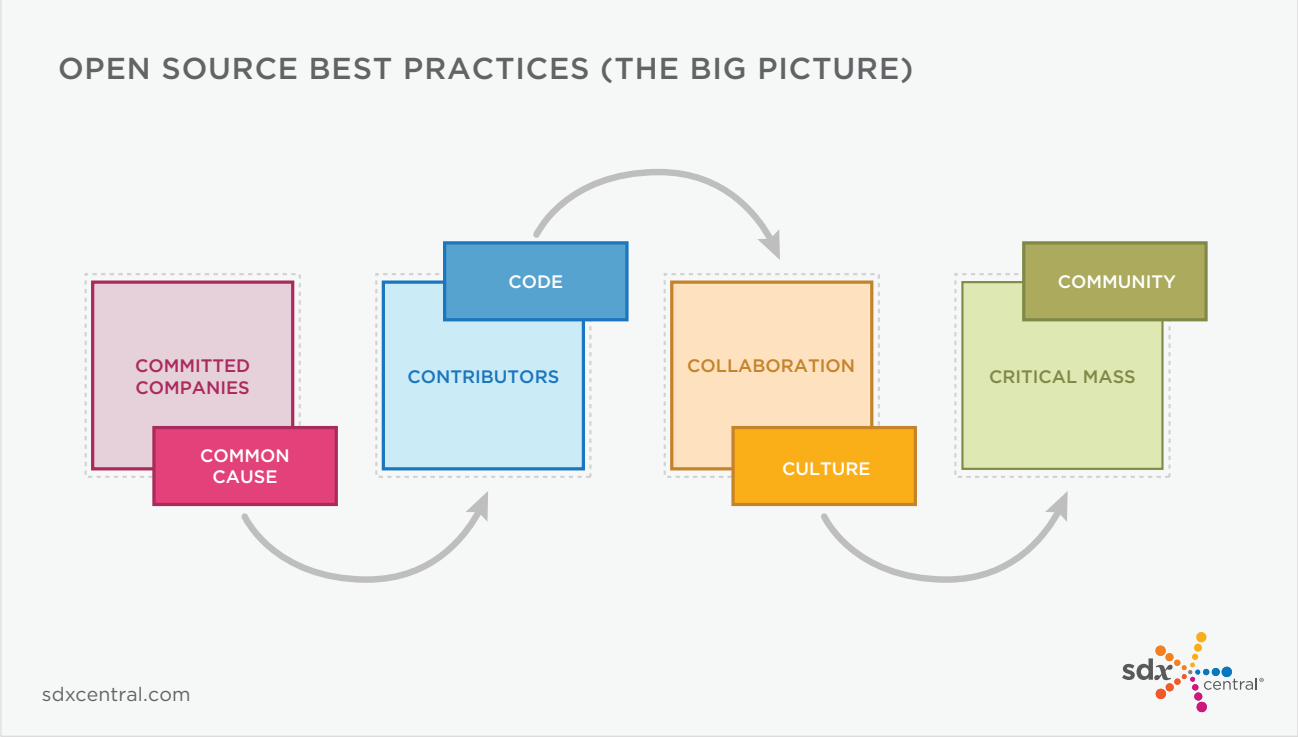


Figure 10

- **Common Cause**—Convergence around a common problem statement with a committed set of companies
- **Collective Contributions**—Code is critical, especially from a diverse set of contributors willing to share seed code, enhance, and support it
- **Collaborative Culture**—Creating and maintaining a high-degree of comraderie and cooperation, especially among competitors, remains critical to success
- **Community**—Cultivating a sustainable community is essential towards attaining critical mass and the ultimate goal of a viable ecosystem

Attaining these lofty goals requires a value system focused on the project vs. individual organizations, true openness avoiding domination by a single organization, diversity across up and down the value chain, global participation, and perhaps most importantly trust.

Best Practices for Open-source networking Development

- **Involve network operators (applies to both telcos and enterprises) upfront**—the sooner the operators can engage, the better, as they can contribute use cases, share priorities, refine requirements, etc. Strong operator participation will also attract significant vendor participation.

- **Narrow the project scope**—Exercise discipline to avoid features, functionality, and project proposals that deviate from the project mission. Establishment of an architecture committee, with a charter to consider how each proposal fits with the code base is one prudent step successful projects have taken.
- **Encourage contributors and not observers**—While recruiting large numbers of members can benefit the project, emphasis should be placed on proactively involving organizations who intend to contribute code and development resources.
- **Strive for (open source) diversity**—An effective means to validate the need for new functionality, and avoid being dominated by a single organization, require all new project proposals include support from 3-5 organizations (minimum) prior to granting approval. In the open source community, this is referred to as open source diversity, not to be confused with cultural, geographic, gender, experience- based diversity, which is obviously also highly desirable.
- **Open up the development community to all**—To maximize participation, implement governance that allows all individuals to participate without restriction, including the requirement to execute a membership or participation agreement. Contributor License Agreements (**CLAs**), however, are required to protect contributors and adopters alike.

Best Practices for Open-source networking Adoption

Open-source networking adoption typically involves both vendors and operators working closely together. Project deliverables are intended for developers and not end users; significant additional development and documentation to transform the codebase and artifacts into deployable products.

- **Establish reasonable expectations**—Large-scale open-source networking projects are inherently complex, broad in scope. While there is often a great deal of pressure to deliver, networking projects take multiple releases and years to be sufficient ready for widespread deployment. The old adage, "under-promise and over-deliver" is a prudcent strategy to avoid hype and its detrimental fallout.
- **Pay attention to license agreements**—There are many different open source license agreements with widely varying implications. Consult an experienced open source attorney prior to adopting code, especially for commercial distribution.
- **Upstream first**—To capitalize on the benefits of open source, it is imperative that any changes to the core functionality are contributed back to the project. If not, it is as if an organization adopts a proprietary branch, which will rapidly become obsolete. Instituting a comprehensive Upstream First methodology and culture will ensure an network operator derives the most from the open source project.
- **Participate in End-user Advisory Groups**—Many large-scale open-source networking projects have established a forum for network operators to share their experiences in the evaluation, trial, and deployment of open-source networking projects. By participating in the End-user advisory groups, network operators can benefit from their peers, as well as directly influence the priorities and scope for the project.

Open Source in Networking Foundations

The following represent a short summary of the more prominent foundations that are involved in open-source networking. There are certainly many more non-profit organizations that have been founded to help power the overall open-source ecosystem that we have not included, and we are open to feedback from our readers on other organizations we should include in future editions of this report.

The Linux Foundation
www.linuxfoundation.org

The Linux Foundation is the leading open source foundation for the networking industry, hosting many of the major projects. With over 25 years of experience, featuring support for the Linux platform, the Linux Foundation has applied its extensive experience in open source development, best practices, community building, and industry presence to progress and promote open-source networking. The foundation is home to many of the most prominent open-source networking projects, including OpenDaylight, ONOS and CORD, DPDK, FD.io, ONAP, to name but just a few.

European Telecommunications Standards Institute (ETSI)
www.etsi.org

ETSI is the standardization body hosting the **Network Functions Virtualization** initiative. In 2016, ETSI launched its first open source project - **Open Source MANO** (as described below). ETSI plans to expand its open source orchestration and networking portfolio based on a number of new initiatives including Multi-access Edge Computing (**MEC**), Next Generation Protocols (**NGP**), Experiential Networked Intelligence (**ENI**).

OpenStack Foundation
www.openstack.org

Launched in 2010, with contributions from Rackspace, NASA and others, OpenStack is an open platform for creating cloud infrastructure. OpenStack has rapidly become one of the largest open-source projects, with thousands of followers. It has been adopted by most of the open orchestration projects, including ONAP, OPEN-O, OSM, OpenBaton, etc.

Apache Software Foundation
www.apache.org

The Apache Software Foundation (ASF) was founded in 1999 as a way to support Apache software projects, including the eponymous Apache HTTP server, which powered a large part of the Internet back then. Today, the foundation hosts many key projects that help power cloud infrastructure, including Hadoop, Kafka, Cassandra, CouchDB, often used as components in major clouds. And while the ASF hosts less pure networking open-source projects, many ASF projects are used to power projects like ONAP and OpenStack etc.

Open Networking Foundation
www.opennetworking.org

The Open Networking Foundation (ONF) is the industry group that has been widely acknowledged as the champion for open SDN. In 2016, the ONF merged with ON.LAB, and shifted its focus to CORD. The ONF is developing "Software-Driven Standards' and has kickstarted a number of open source initiatives as a result, including **XOS**, **ONOS**, **MININET**, among them.

Open Compute Project (OCP)
www.opencompute.org

The Open Compute Project (OCP), originally launched by Facebook was the first organization to adapt open source techniques to hardware platforms. Since the launch in 2009, OCP has published a range of specifications encompassing networking devices (Top of Rack and Optical switches), along with network software.

Telecom Infra Project
telecominfraproject.com

Founded February 2016, the Telecom Infra Project, Inc. (TIP) is an engineering-focused initiative driven by operators, suppliers, developers, integrators, and startups to disaggregate the traditional network deployment approach. It was founded by FaceBook though core contributors today include many of the world's largest telecommunications companies. TIP is working on solutions to open up the carrier network, including OpenRAN (opening up the Radio Access Network), Edge Computing (to innovate with infrastructure at the access edge) and OpenCellular, to develop open rugged platforms to deploy mobile in difficult-to-reach places on the globe.

Harmonization and the Role of Foundations

As end users deepen their direct involvement on open source projects, expectations will increasingly shift from features and functionality to deployment. Many network operators assume that open-source networking projects will inherently snap together like Lego bricks, since networking implies interconnectivity.

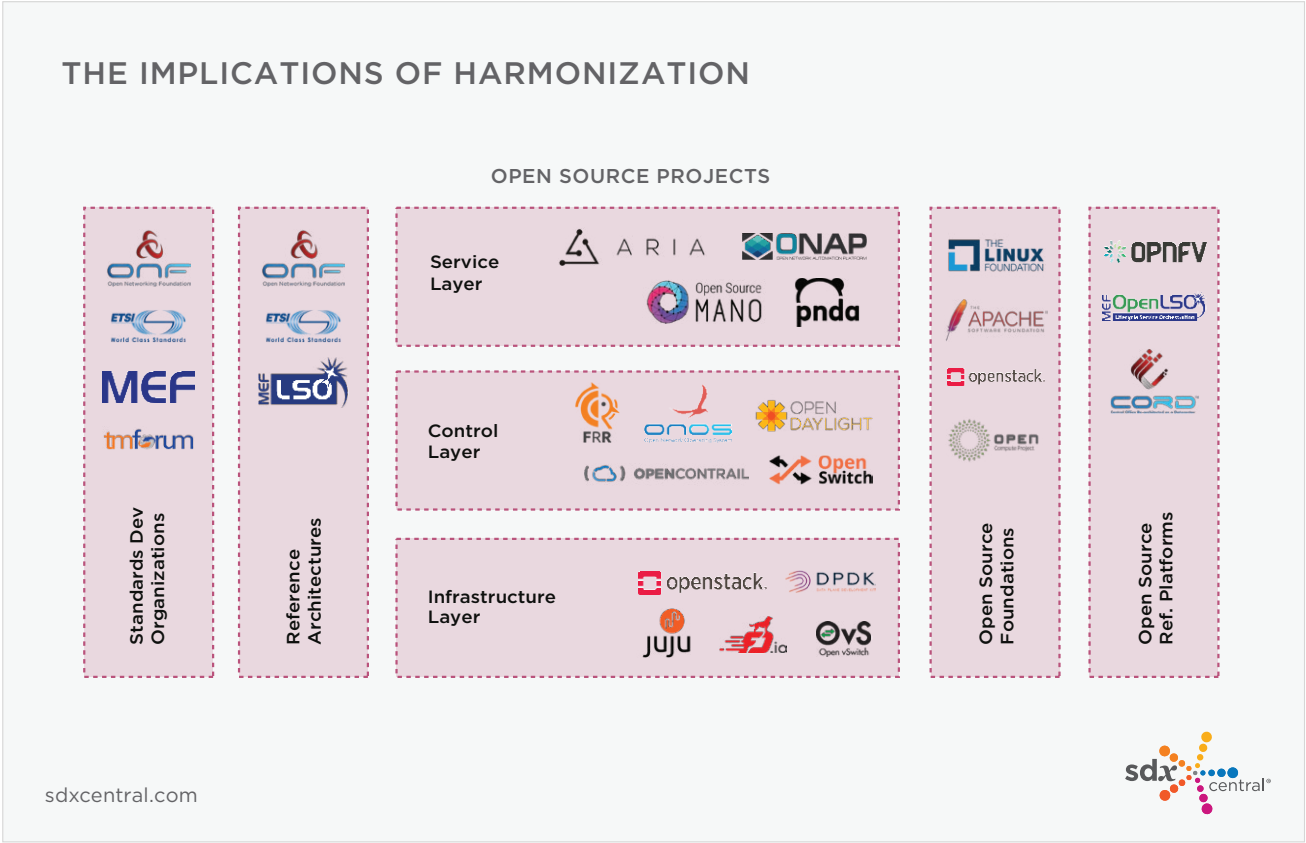


Figure 11

market summary

Today, most open-source networking projects have been developed autonomously from one another. This is not to imply that they do not integrate with one another, because they do, but rather that the development environment, user experience, testing, tooling, etc. are distinct. In addition, release plans, roadmaps, are loosely coordinated at best.

The proliferation and rapid growth of open-source networking and orchestration projects, prompted by major members, motivated the Linux Foundation to address the challenge of more effective coordination, in an ambitious initiative referred to as 'Harmonization'. First introduced at the Open Networking Summit (ONS2017), and described in a white paper, Harmonization is vital towards achieving wide-spread adoption.

In addition to forging an open source solution framework that coalesces individual open source projects into an integrated stack, Harmonization also represents a call to action to the industry to re-examine the role of standards and open source, and their relationship, to forge a new model for collaboration. Figure 11 offers an initial perspective on Harmonization, which is expected to evolve over time.

Open Source in Networking Landscape

The **SDxCentral directory** provides a collection of the major open-source networking projects available today. And the second half of this report provides a rich list of the more popular open source projects. Nevertheless, in this section, we will provide you with a quick run through, by major category, of projects you should pay attention to and that you might consider utilizing in your own products or deploy in your infrastructure. If you think we've missed a project that deserves more prominence, we would love to hear from you at **research@sdxcentral.com**.

Infrastructure Layer

The Infrastructure Layer projects are primarily focused on accelerating network data plane processing on general-purpose hardware. Some of the popular ones include **Data Plane Development Kit (DPDK)** which is a set of development tools to accelerate packet processing. DPDK was originally contributed to the open source community in 2010s by Intel, and is now available on multiple platforms. In early 2017, the DPDK project joined the Linux Foundation open-source networking portfolio. Another similar project is **Fast Data I/O Project (FD.IO)**, another network acceleration initiative that is developing a "modular and extensible I/O services framework" agnostic to the hardware and virtualization architecture. FD.io is complementary to DPDK, and may be deployed in tandem. The core code for FD.IO was contributed by Cisco, and was derived from **Cisco's Vector Packet Processing**, which has been productized for over a decade.

No list for open source in networking would be complete without mentioning the venerable **Open Virtual Switch (OvS)**. Now part of the Linux Foundation, it was originally created by the Nicira/VMware team. Open vSwitch (OVS) is the predominant virtual switch for virtualization environments, originally derived from work in the early days of virtualization. It has been widely deployed, and integrated with many open source packages.

P4 is a slight lesser known open-source initiative that is a high-level language for programming protocol-independent packet processors. Created by some of the early proponents of SDN, P4 was designed to realize the vision of SDN- truly programmable networks with a focus on enabling protocol-independent silicon.

Network Virtualization

Network virtualization is the layer that provides an abstraction over and above physical transport, allowing for a virtual network to be run above an underlay network that doesn't have to be aware of the complexities of the virtual network running above it. Prominent projects in this area include **Project Calico**, a layer-3 networking solution that has gained a significant following within the container networking industry.

Calico is currently maintained by Tigera, a commercial entity that provides solutions built around Calico. Another project is **OVN** (Open Virtual Network), which is part of the Open vSwitch project and complements the existing capabilities of OVS to add native support for virtual network abstractions, such as virtual L2 and L3 overlays and security groups.

Control and Management Layer

At the control layer, we highlight some of the projects responsible for ushering in the era of SDN, the SDN controllers and the open-source routing stacks, many of which have been around in the pre-SDN days.

SDN Controllers

Starting with SDN controllers, **OpenDaylight** has emerged as the predominant large-scale open SDN Controller Framework. ODL was the first Linux Foundation open networking project, and has dozens of deployments across carriers, cloud operators, and enterprises. Along with ODL, another popular controller is **Open Network Operating System (ONOS)**. ONOS is an SDN Controller Framework, contributed by ON.LAB, that was targeted

to carrier-grade SDN networks. ONOS promoted OpenFlow switches, and has since expanded its southbound interfaces. ONOS is designed to be a more integrated network operating system, and has been adopted for the ONF Central Office Re-Architected as a Data Center (CORD) reference architecture.

Other controllers include **OpenContrail**, an SDN Controller Framework designed to enable interworking with the legacy routed infrastructure and also quite popular for creating virtual multi-tenant networks at large cloud providers. It is an open source version of **Juniper's commercial offering**.

Another controller of prominence, **Ryu**, is an early controller framework that was deployed in the first carrier SDN network by NTT in Japan. More recently, Ryu has been popular in the academic community, especially in Japan and Taiwan.

Control Plane

When it comes to L3 routing, we have a project with a rich history—Zebra/ Quagga / OpenSourceRouting. **Zebra** was the first open source TCP/IP routing stack, originating in the mid-1990s, and commercialized by IP Infusion (founded in 1999 by the Zebra key contributor, Kunihiro Ishigura). After the Zebra community waned, **Quagga** was launched in 2005 as a fork of Zebra, maintained by organizations such as **NetDef**. In 2011, the non-profit **open source routing** foundation adopted Quagga, and has been maintaining it since.

The story doesn't quite end there though—enter the Linux Foundation and the **Free Range Routing (FRR)** project. FRR provides a full-featured routing stack, backed by a diverse set of routing veterans. FRR is a fork of the Quagga code base, and joined the Linux Foundation in early 2017.

Network Operating Systems

On the topic of network operating systems, Linux certainly plays a strong role. We have **Open Networking Linux (ONL)** hosted by the Linux Foundation. Open Networking Linux is a Linux-based Network Operating Systems launched in 2014 for bare-metal (and white box) switches. Big Switch Networks contributed the code base to the OCP, where it has been significantly enhanced.

Another related project hosted by the OCP is **Open Network Install Environment (ONIE)**. Launched in 2013, the Open Network Install Environment is a boot loader for white box switches, in order to install both open and proprietary network operating systems. Cumulus Networks contributed the code based for ONIE to the OCP, where it has been maintained.

The **OpenSwitch** project is another Linux-based Network Operating System targeting enterprise-class switches. Originally seeded by HPE, OpenSwitch integrates select L2/L3 control plane support, multi-platform capabilities, and APIs to integrate with other open source projects (e.g., OpenStack). Today, OpenSwitch is used and supported by companies like SnapRoute and Dell EMC, which uses OpenSwitch in its Dell EMC OS10 open networking platform and actively supports the **OpenSwitch (OPX)** project.

Service and Orchestration Layer

In networking, we sometimes tend to focus on the L2, L3 and L4-7 components, forgetting that we need systems to provision, monitor, orchestrate these devices in order to have a working, maintainable network. Fortunately, the open source community didn't have the same blinders on and we have a rich set of projects in this space as well.

We start with the **Open Network Automation Platform (ONAP)**, a project hosted by the Linux Foundation, and one we've covered quite extensively earlier in the report. ONAP provides an open platform for policy-driven orchestration and automation of physical and virtual network functions that will enable service providers to rapidly automate new services and lifecycle management. ONAP was formed as the merger between the Linux

Foundation's OPEN-O project (which China Mobile and Huawei contributed to), and AT&T's open-source ECOMP contribution.

A similar framework, under the auspices of ETSI, is **Open Source MANO (OSM)**. Launched in 2016 with code seeded by Telefonica and Rift.io, the Open Source MANO project is focused on implementing the ETSI NFV Management and Orchestration (**MANO**) specifications. OSM is the first open source project supported by ETSI.

Another alternative to OSM is **OpenBaton**, another open implementation of the ETSI NFV MANO specifications. OpenBaton is intended for the research community, and has been largely funded by the European Union with primary contributions from **Fraunhofer Fokus**.

Two other projects of importance in the orchestration of services and important in managing Virtual Network Functions (VNFs) as part of an NFV framework are **Juju**, whose primary contributor is Canonical (Ubuntu), and **Tacker**, an OpenStack Foundation project.

Finally, to wrap this section up, we have the **Platform for Network Data Analytics (PNDA)**, created by Cisco and now hosted under the Linux Foundation. PNDA is Big Data platform that is optimized for networking applications. PNDA enables Big Data applications to be developed without the need to be an expert in Big Data. The goal is to use PNDA insights to facilitate better network visibility, leading to improved troubleshooting and optimization of network resources.

Open Source Reference Architectures

Moving on to reference architectures, these projects focus on providing a an open platform framework that incorporates other open-source projects in a way that solves real-world problems. Many of these will focus on integration and testing, as well as introduce glue components that may be necessary to tie other projects together.

A platform well-known by in telecommunications industry is **Open Platform for NFV (OPNFV)**. Launched in 2014 and hosted by the Linux Foundation, OPNFV is a development tool to enable investigation, validation, demonstration, and testing for NFV concepts, products, and solutions. OPNFV provides a Continuous Integration/Continuous Deployment (CI/CD) environment, integrations of existing open source cloud compute, storage, and networking components/platforms, and a series of global (Pharos) Labs to facilitate NFV developments.

Another platform is **Open Lifecycle Services Orchestration (OpenLSO)**, a framework and set of tools that enables orchestration investigations and demonstration based on the MEF **LifeCycle Services Orchestration** concepts. The open source reference architecture is hosted on the **MEFnet** testbed, which is currently restricted to MEF members.

Applications and Security (L4-7)

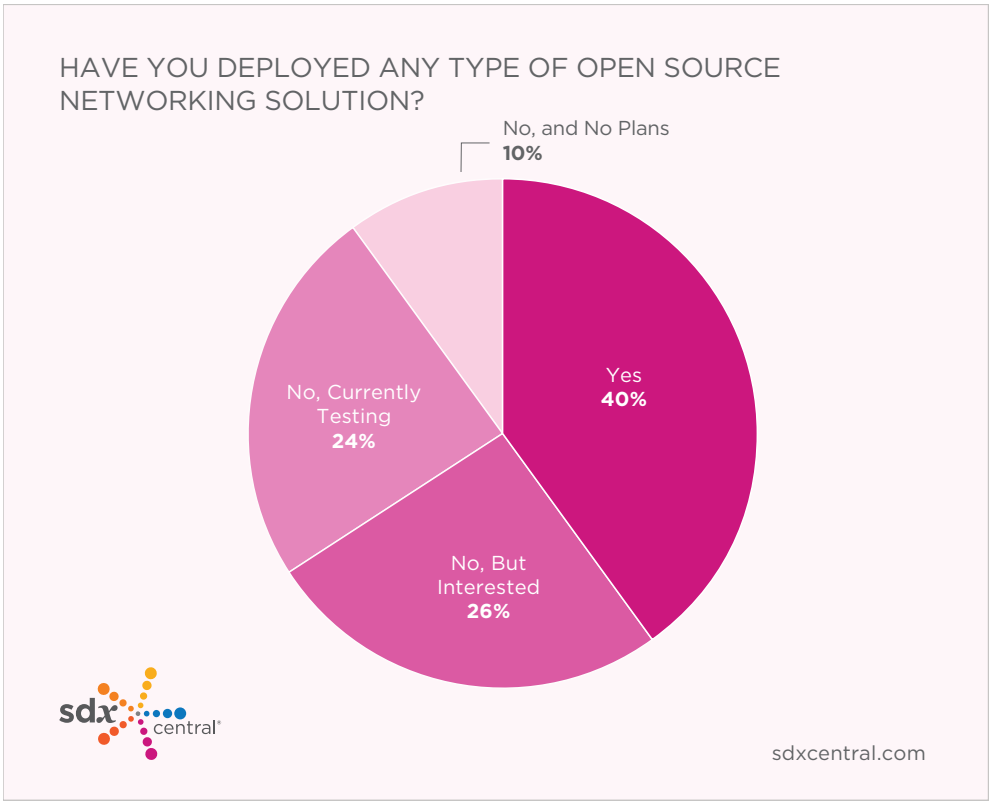
While many of us tend to focus on just SDN and NFV components in open-source networking, there are other projects that bear mention in the L4-7 space. Some of them include **NGINX** and **HAProxy**, which power a large part of the scaling infrastructure today across the Internet and provide security and load distribution. In addition, **linkerd** and **Cilium** are key projects focused on connecting multiple web services together, popular in container and micro-service-based applications. linkerd provides a service mesh architecture that deals with the nitty-gritty details of service discovery, fault handling and other key services for interconnecting application components while Cilium focuses on enhancing the security of service-to-service communication. Finally, on the security side of things, **BRO** and **Suricata** have snatched the open-source IDS mantle from Snort, and which are used both directly by enterprises and service providers worldwide and also embedded in other solutions.

Open Source in Networking Reader Survey Results

To help provide a reader perspective in this report, and to get a sense of where the SDxCentral audience believes the market is going, SDxCentral's Research Team asked community members what they think about open-source technologies and what they are doing or planning on doing with. For the purposes of the survey, open-source networking solutions were defined as "any software platform that relates to networking, including L2 to L4 stacks, as well as projects that manage or orchestrate networking equipment." There were 110 participants - 46% were "Technology Vendors," 22% "Telecommunications Service Providers," 15% "System Integrators," 10% "Other," and 6% "Enterprises."

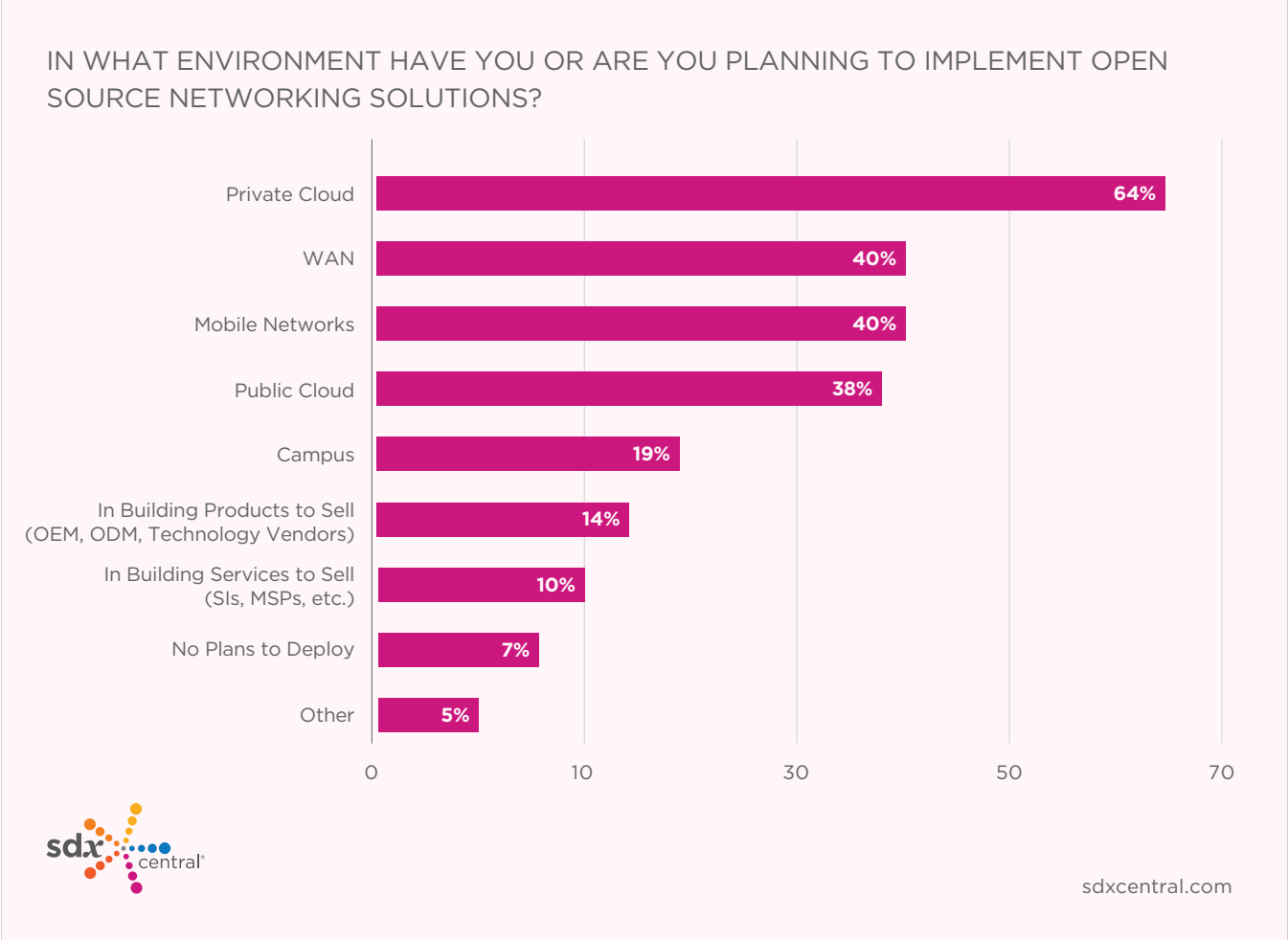
End User Open Source Implementations

When end users—cloud service providers, telecommunications service providers, enterprises, and others—were asked if they had already incorporated any open-source networking solutions, 40% said "yes." Another 24% indicated they were currently testing/qualifying open source solutions. Only 10% said they had "no plans to incorporate" open source solutions into their network, at this time.



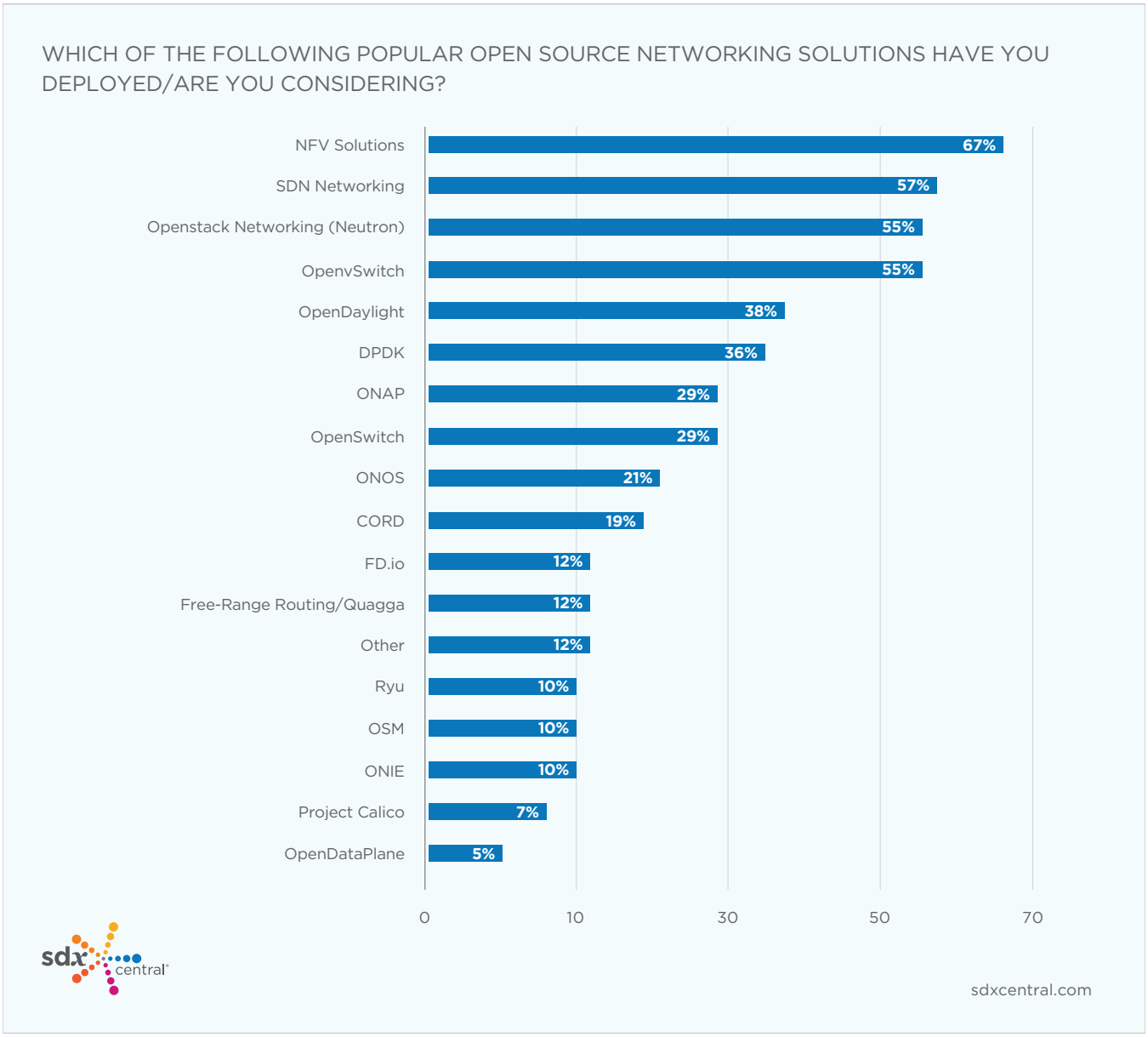
market summary

When probed on all the places they are deploying or plan to deploy open source solutions, the majority (64%) said in a private cloud. 40% said they were using or going to use open source in their mobile and WAN networks, while 38% said they are integrating or are looking to integrate open-source networking solutions in their public cloud environments.



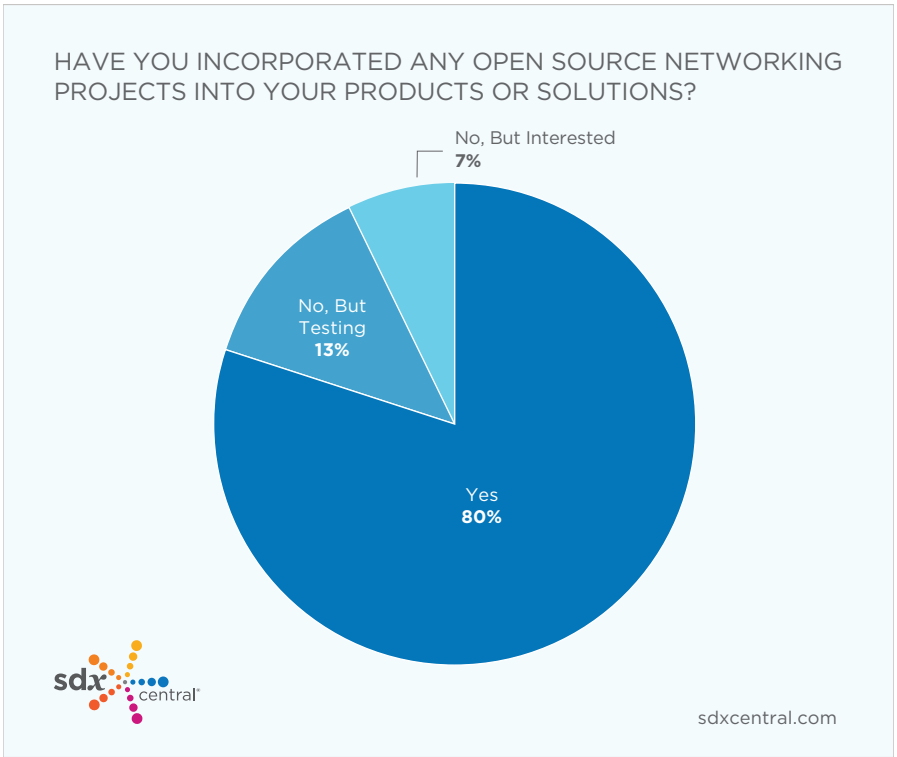
market summary

The most popular open-source networking solutions being deployed or considered are OpenStack solutions (55%), OpenvSwitch (55%), OpenDaylight (38%), DPDK (36%), ONAP (29%) and OpenSwitch (29%).



Vendor and System Integrator Implementations

The survey asked technology vendors and system integrators (SIs) if they had incorporated any open-source networking projects into their products or solutions. 80% said "yes," while 13% said they were currently testing/qualifying open source solutions. The 7% who haven't incorporated open source are researching solutions, with plans to implement in the next year.

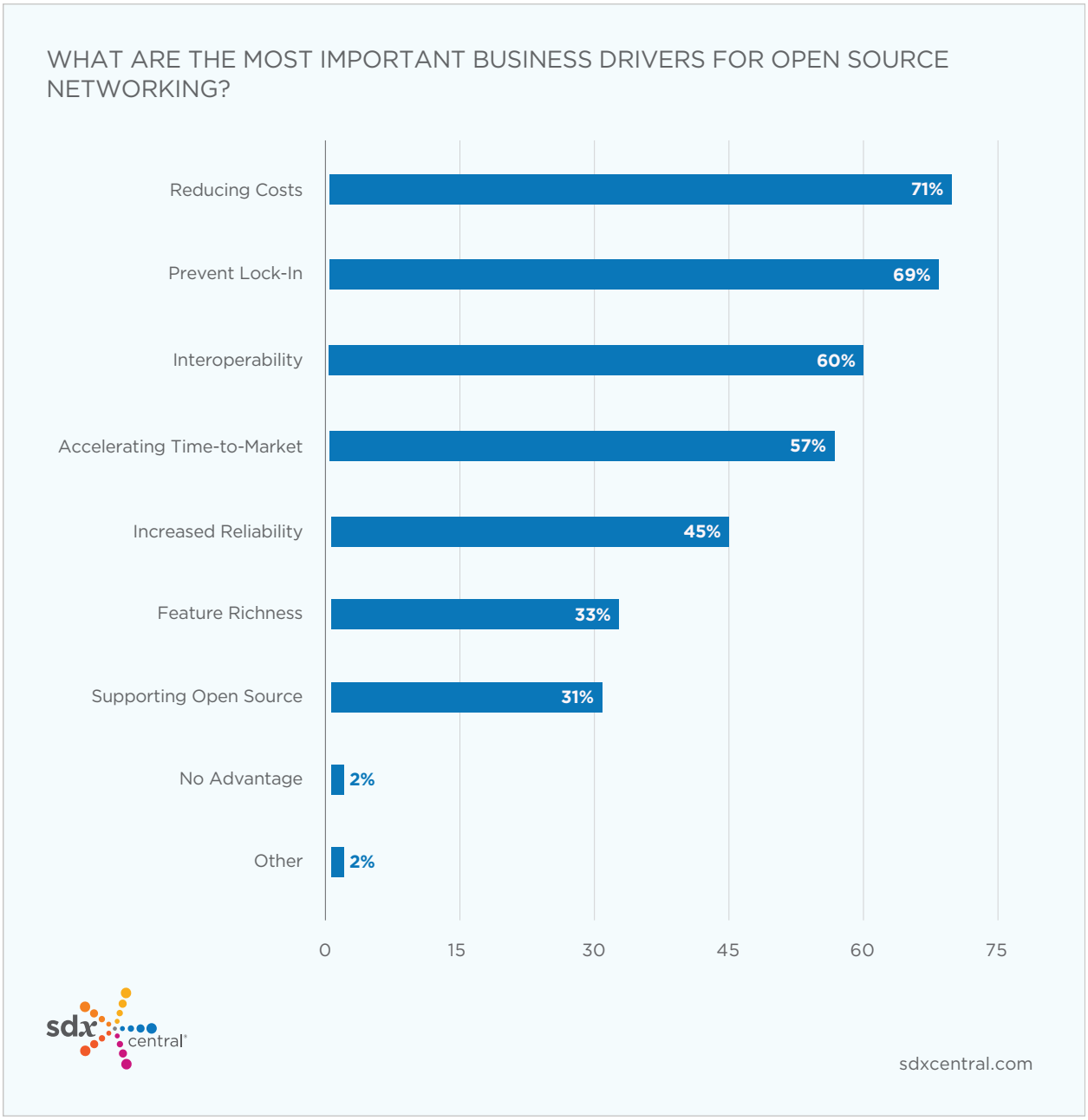


66% of the survey vendor/SI participants have deployed or are planning on deploying OpenStack solutions and networking projects (Neutron), 56% are deploying general SDN networking solutions (including controllers), while 54% are deploying network function virtualization solutions. The specific solutions include mentioned most were OpenDaylight, at 50%, OpenvSwitch, at 47%, and DPDK, at 46%.

Drivers for Open Source in Networking

The main reason end-user respondents said they are using or thinking about using open-source networking solutions is to reduce costs (71%). They felt "saving money by using a pre-built, validated codebase" was the primary business driver. Of those end-users who have already deployed open source solutions, 32% said they were seeing 25% – 50% savings over commercial solutions. A quarter of the respondents saw 25% savings, while another quarter said they really didn't see a difference in total costs between open source and commercial solutions.

Other important reasons to adopt open source solutions, according to respondents, who could check all the answers they felt applied, were to "prevent vendor/product lock-in," at 69%, ensure "interoperability with a common source-base," at 60%, "velocity – accelerating time to market" at 57%, "reliability," at 45%, and "feature richness," at 33%. Interestingly, 33% also noted there was a compelling community effect associated with using open-source networking – participating in the open-source ecosystem can have recruiting and marketing benefits.

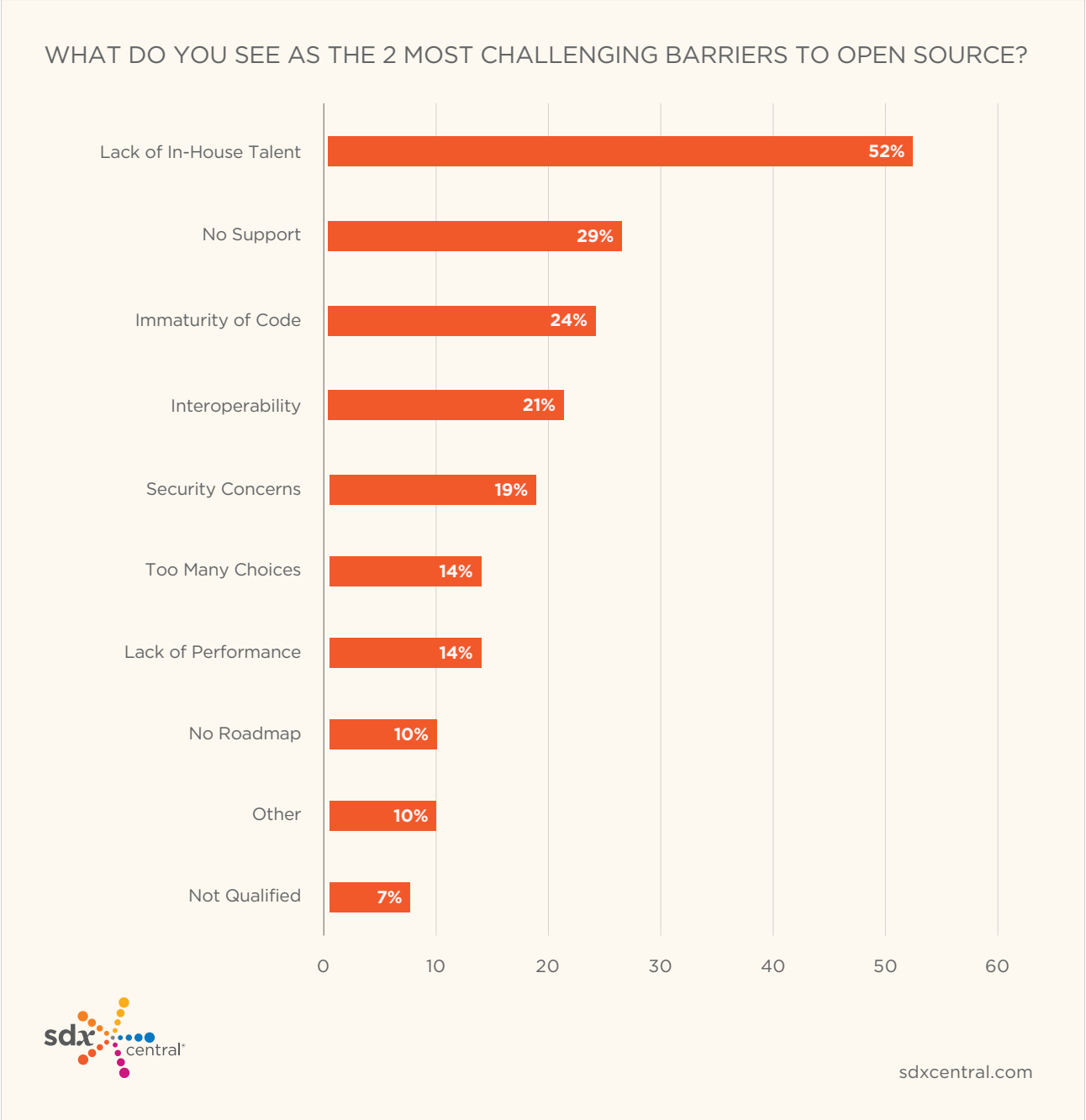


Technology vendors and system integrator respondents had a slightly different take on what they saw as the most important business drivers for using open source in their solutions. They indicated ensuring "interoperability with other systems" was the primary driver, at 65%, followed by "reducing deployment costs" at 63%, and "velocity - accelerating time-to-market," at 62%. Interestingly, there is a marketing value for vendors/SIs too – 43% felt that the "open-source ecosystem makes customers feel more comfortable and helps project 'open' image."

When probed on the cost savings vendors/SIs who have deployed open source solutions are seeing, 31% indicated it was between 25% and 50%. Another 31% noted it was less than 25%, and 23% found the costs were similar between open source solutions and the solutions they are developing in-house.

Barriers to Using Open Source

End-user participants were asked what they saw as the biggest challenges associated with adopting open-source networking solutions. 52% responded that "lack of in-house talent" was their biggest hurdle, while "lack of enterprise level support – no support" topped the list for 29% and "interoperability concerns with existing infrastructure" for 21%.



38% of vendors/SIs noted the "immaturity of the codebase or code-quality issues," was the biggest barrier for them. Another 31% noted that "lack of control over roadmap and features" could make it hard for them to incorporate open-source networking in their solutions. Other challenges included "interoperability concerns," at 24%, as well as "performance issues" and "lack of in-house talent," at 22%.

Future State of Open Source in Networking

56% of all survey participants plan to expand their use of open-source networking in the next year - 45% of end-users and 68% of vendors/SIs. The percentage of respondents who are using open source, but don't think they will expand their deployment is low, at only 14%. The percentage who are not using open source, but plan to is 26%.

When asked which use cases would be ideal for open networking solutions, based on their experience to date, SD-WAN and virtual CPE stood out. On the flip side, participants indicated they would still be hesitant to use open source for mission critical use cases.

Unsurprisingly, open source in networking is viewed favorably by our community, with a large percentage looking to expand their use of open-source projects in their network deployments over the course of the next year. This bodes well for all those open-source projects looking for new contributors and new end-users who can benefit from all the person-years of development already invested in those codebases.

What Lies Ahead for Open Source in Networking

Just ten years ago, it would have been difficult to imagine the entire networking industry being disrupted by open source. While as an industry we benefited from open source at the operating system level, including in embedded networking devices, and at the software component level, we took for granted that the core routers, switches and L4-7 appliances in our data centers, campuses and WAN would be proprietary boxes that came from networking vendors dueling for dominance in a closed ecosystem.

Today, we see quite a different landscape, with open source gaining sufficient prominence that we see open source projects powering networks at large telcos like AT&T, Telefonica, China Mobile, NTT, to name just a few. And open-source software and hardware powers a good part of the network infrastructure at the world's largest web properties and cloud service providers, such as LinkedIn, FaceBook, Google, Microsoft, and Amazon.

As well, large system integrators worldwide have embraced open source and have built new practices with large teams familiar with the most popular open-source projects in networking, ready to help enterprises and service providers benefit from the hundreds of millions of lines of code already developed and ready to be used.

And for technology vendors, one of the keys to future success will be transitioning to using open source to their advantage in their business. Whether it be incorporating more open source into their codebase or hardware designs, learning how to participate in open-source communities, or adding more service offerings that help their customers take advantage of open source, these vendors cannot ignore open source, nor can they combat it head-on—that's a losing proposition.

The wave of open source in networking is cresting and over the next year we expect to see the ecosystem for open-source projects explode, with many new ones being added, old ones falling out of popularity and getting culled. There will continue to be confusion surrounding the vast treasure trove of open-source projects in networking—there are many to keep track of. As with the natural evolution of any space, we expect consolidation and harmonization to occur, with the most well-funded and promising projects absorbing the less well-funded projects. For long-term success, we will need this to happen before the ecosystem gets too fragmented and we, as an industry, dilute our overall investment across too many projects. As such, we believe that open-source foundations will play a key role in ensuring that harmonization and culling happens, in a form of "guided darwinism".

In conclusion, we at SDxCentral see exciting times ahead for networking, and particularly in open source in networking. Let us be your guide and key information resource as open source transforms our industry, providing you with cutting-edge research and breaking news. And as always, we welcome your feedback at **research@sdxcentral.com**.

Open Source in Networking Projects and Related Vendor Products

The following section of this report profiles some of the more popular open-source projects in networking. We have also included a number of featured commercial products that extensively leverage open-source projects. **Extended profiles can be viewed online**. The information was gathered via a collaborative effort between the SDxCentral's Research Team, vendors' product experts and the hosting foundation. As an inaugural report, our list of projects will not be as extensive as some of our more established industry reports. We welcome feedback and we ask that you contact the research team to start a conversation about having your products included in future editions of this report.

While every attempt has been made to validate the capabilities listed in the profiles, SDxCentral advises end users to verify the veracity of each claim for themselves in their actual deployment environments. SDxCentral cannot be held liable for unexpected operations, damages or incorrect operation due to any inaccuracies listed here.

SDxCentral welcomes feedback and additional information from end users based on their real-world experiences with the products and technologies listed. The SDxCentral Research Team can be reached at **research@sdxcentral.com**.

Organizations covered include: Organizations covered included: Apache Software Foundation, Buoyant, Covalent, CZ.NIC, Dell EMC, Digital Ocean, ETSI, Facebook, Faucet, GoBGP, HAProxy, Intel, Juniper Networks, Linaro, Linux Foundation, MEF, NGINX, NTT, Open Compute Project (OCP), OpenConfig, Open Networking Foundation (ONF), OpenStack Foundation, P4.org, Tigera, VMware

Dell EMC Network OS10

(Click to View More Details Online)
http://www.dell.com/us/business/p/open-platform-software/pd

Dell EMC | Networking
PUBLIC | **PRIVATE**
176 South Street
Hopkington, MA, 01748, United States
kiran_khanna@dell.com
1 (866) 438 3622
http://www.dellemc.com

Description of Company: Dell EMC, a part of Dell Technologies, enables organizations to modernize, automate and transform their data center using industry-leading open networking infrastructure, servers, storage and data protection technologies. Our Networking division provides a trusted foundation for businesses by enabling a hybrid cloud for cloud-native apps and big data solutions. Dell EMC has customers in 180 countries with the industry's most comprehensive and innovative portfolio from edge to core to cloud.

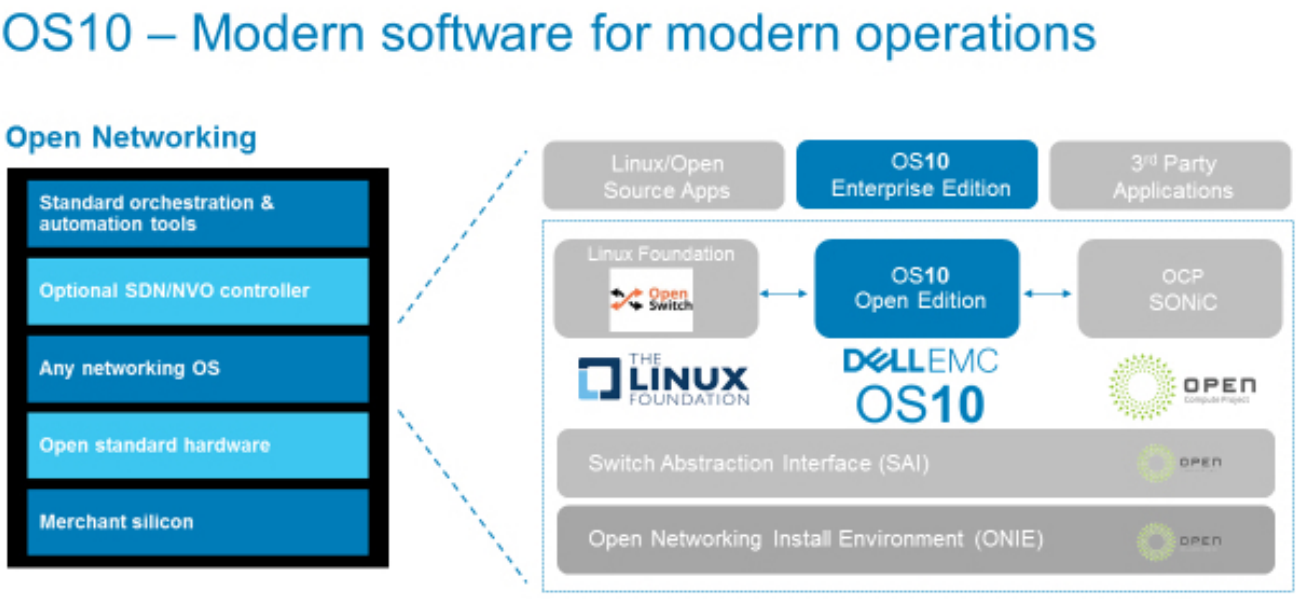
Dell EMC | Networking in SDxCentral Company Directory

Description of Product: Dell EMC Network OS10 combines the best of Linux, open computing and networking to advance open networking disaggregation. OS10 Open Edition embodies the evolution of networking software and was designed to introduce new levels of software flexibility and programmability in large-scale data center environments for Cloud and Communications Services Providers (CSPs). It easily integrates with DevOps tools and has been contributed to OpenSwitch through Linux Foundation.

Dell EMC Network OS10 in SDxCentral Product Directory

Relevant Categories
NFVI - Network Operating Systems
Value Proposition
OS10 Open Edition introduces new levels of software flexibility & programmability in large-scale data centers for Cloud and Communication SPs. It disaggregates network software and leverages unmodified, open-source Linux (Debian) with platform portability, rich application composition & assembly capabilities. It has been contributed to OpenSwitch thru Linux Foundation. It allows right-sized software footprints for different operating models, breaking down traditional IT/CT administrative silos.
Key Partners
Quagga. Commercial packages are coming soon.

Applicable Verticals
Cloud Service Providers, Financials, Government & Education, Healthcare, Retail, Telecom
Related Foundation
Linux Foundation, ONF/ON.LAB, Open Compute Project
Related Projects
OpenSwitch, SONiC
Use Cases
1. Enterprise data center 2. Webtech and Service providers 3. Telco Cloud providers



Edgent

(Click to View More Details Online)
http://edgent.apache.org/docs/downloads

Apache Software Foundation
PUBLIC | **PRIVATE**
https://www.apache.org/

Description of Project: Apache Edgent is an open source development tool that makes it easier for developers to create Internet of Things (IoT) applications to analyze data on the edge of their networks.

Relevant Categories
Network Analytics
Value Proposition
Working in conjunction with centralized analytic systems, Apache Edgent provides efficient and timely analytics across the whole IoT ecosystem: from the center to the edge. Edgent enables network operators to shift from sending a continuous flow of trivial data to the server to sending only essential and meaningful data as it occurs. This is especially important when the cost of communication is high, such as when using a cellular network to transmit data, or when bandwidth is limited.
Licensing
Apache

Launch Date
2016-12
Use Cases
1. Reduced Communication Costs - Edgent performs real-time analytics on the edge device, separating the interesting from the mundane, so you don't have to send every sensor reading over a network. 2. Local and Faster Time to Action - Edgent makes devices more intelligent, enabling them to take immediate action. For example, a connected vehicle running Edgent can adjust traction control based on the weight of the cargo/passengers. 3. Learning From Related Devices - Edgent enables connected devices to learn from related devices.

linkerd

(Click to View More Details Online)
http://github.com/linkerd/linkerd/releases

Buoyant, Inc
PUBLIC | **PRIVATE**
https://buoyant.io

Description of Project: Linkerd is an open source network proxy designed to be deployed as a service mesh, or a dedicated layer for managing, controlling, and monitoring service-to- service communication within an application. Linkerd gives service owners the freedom to choose whichever language is most appropriate for their service. By decoupling communication mechanics from application code, linkerd allows visibility and control over these mechanics without changing the application itself.

Relevant Categories
NFVI - Infrastructure Layer, VNF - Layer 4-7 Acceleration and Caching
Value Proposition
Linkerd adds visibility, control, and reliability to application environments with a wide array of powerful techniques that include circuit-breaking, latency-aware load balancing, eventually consistent-advisory service discovery, deadline propagation, and tracing and instrumentation. Linkerd automatically instruments top-line service metrics such as request volume, success rates, and latency distributions to provide a resilient service mesh for cloud native applications.
Licensing
Apache

Launch Date
2016-04
Use Cases
1. Load balancing - Because linkerd operates at the RPC layer, it can balance load based on observed RPC latencies and queue sizes, rather than heuristics such as LRU or TCP activity. 2. Circuit breaking - Circuit breaking is a mechanism used by linkerd to remove unhealthy service instances from load balancing. By employing circuit breaking, linkerd can minimize the amount of time spent trying to route requests that ultimately fail. 3. Service discovery - linkerd has the ability to treat service discovery as authoritative or advisory. Authoritative service discovery is configured by default, but can be toggled via linkerd's enableProbation load balancer configuration.

Cilium

(Click to View More Details Online)
http://github.com/cilium/cilium.git

Covalent
PUBLIC | **PRIVATE**
http://covalent.io/

Description of Project: Cilium is open source software for transparently securing the network connectivity between application services deployed using Linux container management platforms like Docker and Kubernetes. At the foundation of Cilium is a new Linux kernel technology called BPF, which enables the dynamic insertion of powerful security visibility and control logic within Linux itself. Because BPF runs inside the Linux kernel, Cilium security policies can be applied and updated.

Relevant Categories	Launch Date
NFVI - Infrastructure Layer, VNF - Layer 4-7 Security	2017-03
Value Proposition	**Use Cases**
Cilium acts as a middle layer, plugs into container runtimes and orchestrators such as Kubernetes, Docker or CNI, and can generate and atomically update eBPF programs on the fly without requiring a container to restart. Thus, unlike connection proxies, an update of the data-path does not cause connections to be dropped. These programs are specifically tailored and optimized for each container.	1. Security - Enforce network security policies at both the network and HTTP layer for Docker containers or Kubernetes pods. 2. Containers - Generate kernel-level BPF programs that work directly with containers. Rather than create overlay networks for containers
Licensing	
Apache	

BiRD

(Click to View More Details Online)
https://gitlab.labs.nic.cz/labs/bird/tree/master

CZ.NIC
PUBLIC | **PRIVATE**
http://www.nic.cz/

Description of Project: BIRD is an open source implementation for routing Internet Protocol packets on Unix-like operating systems. It was developed as a school project at the Faculty of Mathematics and Physics, Charles University, Prague, and is distributed under the GNU General Public License. BIRD supports Internet Protocol version 4 and version 6 by running separate daemons. It establishes multiple routing tables, and uses BGP, RIP, and OSPF routing protocols, as well as statically defined routes.

Relevant Categories	Launch Date
NFVI - Switching and Routing	2013-03
Value Proposition	**Use Cases**
The BIRD project aims to develop a dynamic IP routing daemon with full support of all modern routing protocols, easy to use configuration interface and powerful route filtering language, primarily targeted on (but not limited to) Linux and other UNIX-like systems and distributed under the GNU General Public License. BIRD can be compiled to support either IPv4, or IPv6. To route both protocols, you have to use two completely separated bird processes.	1. IPv4 and IPv6 routing - Both IPv4 and IPv6 (use --enable -ipv6 when configuring) 2. BIRD used as an OSPF router - Static protocol is used for leaf networks hidden behind a neighbor router that does not speak OSPF. 3. BIRD as a BGP route server - It does not use BGP community policing. All the peers are connected into main (master) routing table.
Licensing	
GPLv2	

NetBox

(Click to View More Details Online)
https://github.com/digitalocean/netbox

Digital Ocean
PUBLIC | **PRIVATE**
https://www.digitalocean.com/

Description of Project: NetBox is an open source web application designed to help manage and document computer networks. Initially conceived by the network engineering team at DigitalOcean, NetBox was developed specifically to address the needs of network and infrastructure engineers. It encompasses the following aspects of network management: IP address management, equipment racks, devices, connections, virtualization, and data circuits.

Relevant Categories	Launch Date
NFVI - Switching and Routing	2016-06
Value Proposition	**Use Cases**
NetBox intends to represent the *desired* state of a network versus its *operational* state. As such, automated import of live network state is strongly discouraged. All data created in NetBox should first be vetted by a human to ensure its integrity. NetBox can then be used to populate monitoring and provisioning systems with a high degree of confidence.	1. Single converged database - Whereas most open source IPAM tools available today provide little functionality around tracking physical connections, NetBox provides both IPAM and DCIM functions under one roof. 2. Network modeling - NetBox's data model seeks to replicate the real world as closely as possible. IP addresses are assigned not to devices but to specific interfaces within a device, and interfaces may have multiple IP addresses. 3. Support a robust IP hierarchy - NetBox employs PostgreSQL's native network data types to construct a robust, efficient hierarchy of IP prefixes and addresses.
Licensing	
Apache	

OSM (Open Source MANO)

(Click to View More Details Online)
https://osm.etsi.org/wikipub/index.php/OSM_Release_TWO

ETSI
PUBLIC | **PRIVATE**
http://www.etsi.org/

Description of Project: Open Source MANO (OSM) aims to deliver a production-quality MANO stack for NFV, capable of consuming openly published information models, available to everyone, suitable for all VNFs, operationally significant and VIM-independent. OSM is aligned to NFV ISG information models while providing first-hand feedback based on its implementation experience.

Relevant Categories	Launch Date
NFV MANO - NFV Management and Orchestration	2016-05
Value Proposition	**Use Cases**
As an operator-led community, OSM is offering a production-quality open source MANO stack that meets the requirements of commercial NFV networks. OSM can manage external SDN controllers to perform the dataplane underlay network connectivity on behalf of the VIM. OSM is aligned to NFV ISG information models while providing first-hand feedback based on its implementation experience.	1. Multi-controller enablement - Release ONE substantially enhances interoperability with other components (VNFs, VIMs, SDN controllers) and creates a plugin framework to make platform maintenance and extensions easier to provide and support. 2. Reduced Operational Costs - Release ONE improves administrator and developer experience, both in terms of usability and installation procedure as well as enhances modelling of VNFs and network services. 3. Support for OpenVIM - OpenVIM code, included as part of the OSM install, providing a reference VIM for all-in-one installations with full support of Enhanced Platform Awareness.
Licensing	
Apache	

Facebook Open Switching System (FBOSS)

(Click to View More Details Online)
http://github.com/facebook/fboss

Facebook
PUBLIC | PRIVATE
http://www.facebook.com

Description of Project: Despite having "OS" in its name, FBOSS is not a full operating system. Facebook Open Switching System (FBOSS) is a set of applications that can be run on a standard Linux OS. The initial FBOSS release consists primarily of the FBOSS agent, a daemon that programs and controls the ASIC. This process runs on each switch and manages the hardware forwarding ASIC.

Relevant Categories
NFVI - Switching and Routing, NFVI - Network Operating Systems
Value Proposition
The actual design of and FBOSS switch and switch software is different from most existing network switches. Facebook and the OCP networking project have been explicitly driving change in that monolithic model by separating or "disaggregating" the network hardware and software. In FBOSS, switches feel like servers. For example, admins can deploy a set of FBOSS set of packages/applications to perform very specific switching tasks.
Licensing
BSD

Launch Date
2015-03
Use Cases
1. App acceleration - Low-level apps such as the FBOSS agent that deal directly with the forwarding ASIC. We also have OpenBMC, low-level software that provides management functions for power, environmentals, and other system-level modules. 2. App development - Automation apps for configuration, monitoring, and troubleshooting. FBOSS is designed to be controlled primarily via software, rather than by humans. Instead of providing a command line interface to configure it. 3. Network control - Control apps that use the FBOSS agent to implement forwarding/routing protocols or support centralized decision-making. These apps, such as the one that implements BGP, can be developed independently of the actual hardware.

Faucet SDN Controller

(Click to View More Details Online)
https://github.com/faucetsdn/faucet

Faucet
PUBLIC | **PRIVATE**
http://faucetsdn.org

Description of Project: FAUCET is an OpenFlow controller for multi table OpenFlow 1.3 switches, that implements layer 2 switching, VLANs, ACLs, and layer 3 IPv4 and IPv6 routing, static and via BGP. It is based on Waikato University's Valve and the Ryu OpenFlow Controller.

Relevant Categories
NFVI - Control
Value Proposition
Open Source SDN controller for production networks. Faucet is supported for many commercially available Openflow based switches. Faucet's design principle is to be as hardware agnostic as possible and not require Table Type Patterns.
Licensing
Apache

Launch Date
2015-03
Use Cases
1. L2, L3 Switching 2. Routing and Monitoring

GoBGP

(Click to View More Details Online)
https://github.com/osrg/gobgp

GoBGP
PUBLIC | **PRIVATE**
http://osrg.github.io/gobgp/

Description of Project: GoBGP is an open source BGP implementation designed from scratch for modern environment and implemented in a modern programming language, the Go Programming Language.

Relevant Categories
NFVI - Switching and Routing
Value Proposition
GoBGP is designed to exploit todays multicore processors. Golang makes it extremely easy and simple to write concurrent code. GoBGP is designed to be easily integrated with other software with its RPC APIs instead of manually changing its config via CLI. GoBGP also supports its CLI though. GoBGP is implemented based on an IETF draft, vendor-neutral model for network configuration and policy. Imagine that you can configure BGP across multiple vendors' gear with the single model.
Licensing
Apache

Launch Date
2017-02
Use Cases
1. Enable control plane / BGP routing - L3 obviously scales further and is more reliable then large L2 domains. This supports routing closer to the edge of the network.

HAProxy

(Click to View More Details Online)
http://www.haproxy.org/#down

HAProxy Technologies
PUBLIC | **PRIVATE**
https://www.haproxy.com/

Description of Project: HAProxy is a free, very fast and reliable solution offering high availability, load balancing, and proxying for TCP and HTTP-based applications. It is particularly suited for very high traffic web sites and powers quite a number of the world's most visited ones. Over the years it has become the de-facto standard open source load balancer, is now shipped with most mainstream Linux distributions, and is often deployed by default in cloud platforms

Relevant Categories
VNF - Layer 4-7 Security, VNF - Layer 4-7 Acceleration and Caching
Value Proposition
A load balancer's performance related to these factors is generally announced for the best case. HAProxy involves several techniques commonly found in Operating Systems architectures to achieve the absolute maximal performance. In production, HAProxy has been installed several times as an emergency solution when very expensive, high-end hardware load balancers suddenly failed on Layer 7 processing. Its mode of operation makes its integration into existing architectures very easy.
Licensing
GPLv2, LGPL

Launch Date
2001-12
Use Cases
1. Load Balancing - HAProxy powers the uptime of organizations with large infrastructures and enormous traffic demands by giving them the flexibility and confidence to deliver websites and applications with high availability, and security at any scale. 2. High availability - By controlling traffic during load spikes, crashes or server maintenance, HAProxy application load balancing solutions ensure that online services and applications are always available. 3. Application acceleration - Many of the world's most highly regarded brands trust HAProxy to give them the flexibility and confidence of highly available, secure and accelerated delivery of their websites and applications.

YANFF - Yet Another Network Function Framework

(Click to View More Details Online)
http://github.com/intel-go/yanff

Intel
PUBLIC | PRIVATE
http://www.intel.com/

Description of Project: YANFF is is a set of libraries for creating cloud-native Network Functions (NFs). It enables a rethinking NFV for cloud-native - from development to deployment and running. It's easy write, easy scale, easy change. YANFF brings DPDK to clouds and creates a new concept of network function development through high-level abstractions with no performance sacrifice.

Relevant Categories
NFVI - Infrastructure Layer, VNF - Layer 4-7 Security, VNF - Layer 4-7 Acceleration and Caching
Value Proposition
YANFF simplifies the creation of network functions without sacrificing performance. YANFF enables higher level abstractions, DPDK as a fast I/O engine, is Go-based, and has a built-in scheduler to auto-scale processing based on input traffic rate. YANFF brings direct access to Intel hardware capabilities, i.e, AES, QAT, etc.. with 10x reduction in lines of code, a lower barrier to entry, an similar performance with C/DPDK per box, in a cloud native deployment.
Licensing
BSD

Launch Date
2017-03
Use Cases
1. Data place acceleration 2. Cloud-native networking 3. Infrastructure as Code

OpenContrail

(Click to View More Details Online)
http://github.com/Juniper/contrail-build

Juniper Networks
PUBLIC | PRIVATE
http://www.juniper.net/us/en/

Description of Project: OpenContrail is an Apache 2.0-licensed project that is built using standards-based protocols and provides all the necessary components for network virtualization–SDN controller, virtual router, analytics engine, and published northbound APIs.

Relevant Categories
NFVI - Switching and Routing, NFVI - Control, NFV MANO
Value Proposition
OpenContrail can act as a fundamental network platform for cloud infrastructure. It has an extensive REST API to configure and gather operational and analytics data from the system. OpenContrail can forward traffic within and between virtual networks without traversing a gateway. It supports features such as IP address management; policy-based access control; NAT and traffic monitoring. It supports the existing BGP/MPLS L3VPN standard for network virtualization.
Licensing
Apache

Launch Date
2013-09
Use Cases
1. Cloud Networking – Private clouds for enterprises or service providers, Infrastructure as a Service (IaaS) and Virtual Private Clouds (VPCs) for Cloud Service Providers 2. Network Function Virtualization (NFV) in Service Provider Network – This provides value added services (VAS) for service provider edge networks such as business edge networks, broadband subscriber management edge networks, and mobile edge networks. 3. Successful Cloud Adoption - Introducing Contrail into a heterogeneous cloud/virtualization environment will help an organization move towards SDN adoption based on open standards.

OpenDataPlane Project

(Click to View More Details Online)
http://www.opendataplane.org/downloads/

Linaro
PUBLIC | **PRIVATE**
https://www.linaro.org

Description of Project: The OpenDataPlane project has been established to produce an open-source, cross-platform set of application programming interfaces (APIs) for the networking data plane.

Relevant Categories
NFVI - Infrastructure Layer
Value Proposition
The primary goal of ODP is to provide a common set of APIs for application portability across a diverse range of networking platforms (SoCs and servers) that offer various types of hardware acceleration. By decoupling the API definition from its implementation, APIs can be defined to address data plane application needs rather than to expose specific platform capabilities and allows vendors full freedom to implement these APIs in a manner optimized to the capabilities of each platform.
Licensing
BSD

Launch Date
2015-02
Use Cases
1. Support for multiple NICs - ODP will provide the first truly open source, cross platform solution designed and developed by a significant cross section of networking leaders, all of whom are members of the Linaro Networking Group. 2. Cross platform support - Implementations of ODP currently exist for a wide range of platforms spanning diverse instruction set architectures including ARM, MIPS, Power, x86, as well as proprietary SoC architectures, and general-purpose servers.

OpenSwitch

(Click to View More Details Online)
http://www.openswitch.net

Linux Foundation
PUBLIC | **PRIVATE**
http://www.linuxfoundation.org/

Description of Project: The OpenSwitch Platform is an open source, Linux-based network operating system (NOS) platform. Built under the open source model, OpenSwitch offers the freedom of innovation while maintaining stability and limiting vulnerability. OpenSwtich provides a fully-featured L2/L3 control plane stack, traditional and programmatic, declarative control plane.

Relevant Categories
NFVI - Infrastructure Layer, NFVI - Switching and Routing, NFVI - Network Operating Systems
Value Proposition
OpenSwitch open-sourced NOS passes the ability to scale downstream disaggregated networking solutions from the technologically astute high-end resources-rich “Networking Elite” hyperscalers to the mainstream market. OpenSwitch has the potential to foster disaggregated open-networking solutions adoption, unlocking enormous opportunities.
Licensing
Apache

Launch Date
2016
Use Cases
1. Mainstream networking gear deployments that are looking for a turnkey solution to achieve the agility and the efficiency of DC technologies adopting an open source network 2. Networking engineering capable organizations, including MSDC, that are looking for an open-source NOS to use as a foundation to develop their own networking gear. 3. Embedded / custom designs, that are looking for unambitious networking gear for connectivity and virtualization services.

OPNFV

(Click to View More Details Online)
https://www.opnfv.org/software/downloads

Linux Foundation
PUBLIC | **PRIVATE**
http://www.linuxfoundation.org/

Description of Project: Open Platform for NFV (OPNFV) facilitates the development and evolution of NFV components across various open source ecosystems. Through system level integration, deployment and testing, OPNFV constructs a reference NFV platform to accelerate the transformation of enterprise and service provider networks. Goals include accelerating time to market for NFV solutions, ensuring the platform meets the industry's needs, and enabling end user choice in specific technology components.

Relevant Categories
NFVI - Hardware, NFVI - Infrastructure Layer, NFVI - Switching and Routing, NFVI - Network Operating Systems, NFVI - Control
Value Proposition
OPNFV facilitates the development and evolution of NFV components across various open source ecosystems. Through system level integration, deployment and testing, OPNFV creates a reference NFV platform to accelerate the transformation of enterprise and service provider networks. OPNFV is uniquely positioned to bring together the work of standards bodies, open source communities, and commercial suppliers to deliver a de facto NFV platform for the industry.
Licensing
Apache

Launch Date
2017-09
Use Cases
1. vCO: Virtual Central Office 2. vCPE: Virtual Customer Premise Equipment 3. vEPC: Virtual Evolved Packet Core

FD.io

(Click to View More Details Online)
https://fd.io/resources/

Linux Foundation
PUBLIC | **PRIVATE**
http://www.linuxfoundation.org/

Description of Project: At the heart of FD.io is Vector Packet Processing (VPP) technology. FD.io provides a modular, extensible user space IO services framework that supports rapid development of high-throughput, low-latency and resource-efficient IO services. The design of FD.io is hardware, kernel, and deployment (bare metal, VM, container) agnostic.

Relevant Categories
NFVI - Infrastructure Layer, NFVI - Switching and Routing, VNF - Layer 4-7 Acceleration and Caching
Value Proposition
FD.io is a collection of several projects & libraries to amplify the transformation that began with DPDK to support flexible, programmable & composable services on a generic hardware platform. FD.io offers the Software Defined Infrastructure developer community a landing site with multiple projects fostering innovations in software-based packet processing towards the creation of high-throughput, low-latency & resource-efficient IO services.
Licensing
Apache

Launch Date
2016-02
Use Cases
1. VPP Router or Virtual Switch: VPP can be used as a fully featured high performance multi-layer virtual router & virtual switch. VPP can be deployed as an IPv4/IPv6 software router, delivering consistent throughput performance 2. Replacing a Hardware Router or Switch: VPP is a fully functional router, albeit with the expectation of an external control plane. It is possible to run VPP as a CPE or branch router or whatever location where a 1Tbps router is sufficient. 3. Network Functions: VPP supports a set of Virtual Network Functions (VNFs), either one of many "IPv4 as a service" mechanisms (MAP-E, MAP-T, LW46, SIIT-DC), or IPv6 as a service (6RD), Load-balancing, IPFIX/Netflow probe, NAT44 / CGN.

OpenDaylight

(Click to View More Details Online)
https://www.opendaylight.org/what-we-do/current-release

Linux Foundation
PUBLIC | **PRIVATE**
http://www.linuxfoundation.org/

Description of Project: OpenDaylight is leading the transformation to Open SDN. By uniting the industry around a common SDN platform, the ODL community is helping to make interoperable, programmable networks a reality. This is why so many regard OpenDaylight as the industry's de facto standard.

Relevant Categories
NFVI - Control
Value Proposition
OpenDaylight (ODL) is a modular open platform for customizing and automating networks of any size and scale. With over 1000 developers, 50 member organizations and supporting approximately 1 billion subscribers around the world, OpenDaylight has developed integrated toolchains for leading use cases. OpenDaylight code has been integrated or embedded in more than 50 vendor solutions and apps. It is also at the core of broader open source frameworks, including ONAP, OpenStack, and OPNFV.
Licensing
Eclipse Public License Version 1.0

Launch Date
2013-03
Use Cases
1. Virtual Central Office (vCO) 2. Network resource optimization 3. Fine-grained control for NFV/cloud

Open vSwitch

(Click to View More Details Online)
http://openvswitch.org/download/

Linux Foundation
PUBLIC | **PRIVATE**
http://www.linuxfoundation.org/

Description of Project: Open vSwitch is a production quality, multilayer virtual switch licensed under the open source Apache 2.0 license. It is designed to enable massive network automation through programmatic extension, while still supporting standard management interfaces and protocols (e.g. NetFlow, sFlow, IPFIX, RSPAN, CLI, LACP, 802.1ag). In addition, it is designed to support distribution across multiple physical servers similar to VMware's vNetwork distributed vswitch or Cisco's Nexus 1000V.

Relevant Categories
NFVI - Switching and Routing
Value Proposition
Open vSwitch is targeted at multi-server virtualization deployments, a landscape for which the previous stack is not well suited. These environments are often characterized by highly dynamic end-points, the maintenance of logical abstractions, and (sometimes) integration with or offloading to special purpose switching hardware.
Licensing
Apache

Launch Date
2009-07
Use Cases
1. Open vSwitch has support for both configuring and migrating network state between instances. If a VM migrates between end-hosts, it is possible to not only migrate associated configuration (SPAN rules, ACLs, QoS) but any live network state. 2. Supports features that allow a network control system to respond and adapt as the environment changes. This includes support such as NetFlow, IPFIX, and sFlow an a network state database (OVSDB) that supports remote triggers. 3. HWintegration - Open vSwitch's forwarding path (the in-kernel datapath) is designed to be amenable to "offloading" packet processing to hardware chipsets, whether housed in a classic hardware switch chassis or in an end-host NIC.

Open Network Automation Platform (ONAP)

(Click to View More Details Online)
https://gerrit.onap.org/r/#/q/status:open

Linux Foundation
PUBLIC | **PRIVATE**
http://www.linuxfoundation.org/

Description of Project: ONAP is an open source project backed by the Linux Foundation that enables telecommunications and cable operators to effectively deliver end-to-end services across vendor-agnostic Network Functions Virtualization (NFV) Infrastructure, as well as Software Defined Network (SDN) and legacy network services. The more than 50 member organizations include the top 10 networking vendors and global carriers representing 55% of the world's mobile subscribers.

Relevant Categories
NFVI - Control, NFV MANO, VNF - Layer 4-7 Security, VNF - Layer 4-7 Acceleration and Caching
Value Proposition
ONAP provides a comprehensive platform for real-time, policy-driven orchestration and automation of physical and virtual network functions that will enable software, network, IT and cloud providers and developers to design and automate new services, and support complete lifecycle management.
Licensing
Apache

Launch Date
2017-03
Use Cases
1. Residential vCPE 2. Voice over LTE 3. Virtual firewall

Data Plane Development Kit (DPDK)

(Click to View More Details Online)
http://dpdk.org/download

Linux Foundation
PUBLIC | **PRIVATE**
http://www.linuxfoundation.org/

Description of Project: The Data Plane Development Kit (DPDK) is a set of data plane libraries and network interface controller drivers for fast packet processing, currently managed as an open-source project under the Linux Foundation.

Relevant Categories
NFVI - Infrastructure Layer, NFVI - Switching and Routing
Value Proposition
The DPDK provides a programming framework for x86, ARM, and PowerPC processors and enables faster development of high speed data packet networking applications. It scales from low-end (Intel Atom) processors to high-end (Intel Xeon) processors. It supports instruction set architectures such as Intel, IBM POWER8, EZchip, and ARM. It is provided and supported under the open-source BSD license.
Licensing
BSD

Launch Date
2012-09
Use Cases
1. Routers/Switches/Network Appliances, including bare metal, virtualized and in containers. 2. Wireless core, edge and access. 3. Public and private cloud.

FRRouting (FRR)

(Click to View More Details Online)
https://github.com/FRRouting/frr

Linux Foundation
PUBLIC | **PRIVATE**
http://www.linuxfoundation.org/

Description of Project: FRRouting (FRR) is an IP routing protocol suite for Linux and Unix platforms which includes protocol daemons for BGP, IS-IS, LDP, OSPF, PIM, and RIP. FRR's seamless integration with the native Linux/Unix IP networking stacks makes it applicable to a wide variety of use cases including connecting hosts/VMs/containers to the network, advertising network services, LAN switching and routing, Internet access routers, and Internet peering.

Relevant Categories
NFVI - Switching and Routing
Value Proposition
FRR was created to provide layer 3 connectivity throughout a data center, from the spine switches and leaf switches all the way down to hosts, virtual machines and containers. It is designed to streamline the routing protocol stack. FRR can connect connecting hosts, virtual machines, and containers to the network; network switching and routing; and Internet access/peering routers. FRR provides a simplified data center design subnet freedom, enhanced redundancy and stateless load balancing.
Licensing
GPLv2

Launch Date
2017-10
Use Cases
1. Routing - FRR is a routing software package that provides TCP/IP based routing services with routing protocols support such as RIPv1, RIPv2, RIPng, OSPFv2, OSPFv3, IS-IS, BGP-4, and BGP-4+.

OpenLSO

(Click to View More Details Online)
https://wiki.mef.net/display/CESG/OpenLSO

MEF
PUBLIC | **PRIVATE**
https://www.mef.net

Description of Project: OpenLSO is a MEF-facilitated service orchestration ecosystem that demonstrates integrated use of Open Source solutions and interfaces that adhere to MEF-defined LSO specifications. OpenLSO is primarily aimed at Service Providers that wish to accelerate their adoption of LSO as defined by MEF in order to achieve fully-featured end to end service orchestration of MEF-defined service lifecycles.

Relevant Categories
NFV MANO
Value Proposition
OpenLSO aids service providers that implement service orchestration as per the MEF LSO specifications across multiple networks and domains. OpenLSO enables service providers to improve interoperability, scalability, testability, reusability, and security. OpenLSO also serves equipment manufacturers, SDOs, Open Source communities and MEF projects. For equipment manufacturers, they can use OpenLSO to repackage their solutions and bring them to market with support for their value add features.
Licensing
OpenLSO components have individual licenses.

Launch Date
2016-03
Use Cases
1. Control - Control, in the context of LSO (Lifecycle Service Orchestration), allows service subscribers to actively make changes to their service(s) within the constraints specified by the Service Level Agreement in place for that service. 2. Fulfillment - Fulfillment, in the context of LSO (Lifecycle Service Orchestration), refers to the set of service provider or operator activities and assets used to fulfill an order for a service provider from a customer. 3. Assurance - In the context of LSO (Lifecycle Service Orchestration), refers to the set of capabilities that provide service providers with trouble-related information so that they may track service impact and status of trouble resolution.

NGINX Open Source (OSS)

(Click to View More Details Online)
https://www.nginx.com/resources/wiki/start/topics/tutorials/install/

NGINX, Inc.
PUBLIC | **PRIVATE**
http://nginx.com/

Description of Project: NGINX is a free, open-source, high-performance HTTP server and reverse proxy, as well as an IMAP/POP3 proxy server. NGINX is known for its high performance, stability, rich feature set, simple configuration, and low resource consumption.

Relevant Categories
VNF - Layer 4-7 Security, VNF - Layer 4-7 Acceleration and Caching
Value Proposition
The nginx web server can act as a very capable software load balancer, in addition to its more traditional roles serving static content over HTTP and dynamic HTTP content on the network using FastCGI, SCGI handlers for scripts, WSGI application servers or Phusion Passenger modules. Nginx uses an asynchronous event-driven approach to handling requests. Nginx's modular event-driven architecture can provide more predictable performance under high loads.
Licensing
BSD

Launch Date
2011-07
Use Cases
1. Front end proxy - Because NGINX uses a non-threaded, event-driven architecture, it is able to outperform web servers like Apache. This is particularly true in deployments that receive heavy loads. 2. Mail proxy - Mail proxy features include TLS/SSL, STARTTLS, SMTP, POP3, and IMAP proxy authentication using an external HTTP server. 3. Reverse proxy - The key advantage with NGINX is its nominal RAM and CPU usage under heavy load. Squid is best applied to cache dynamic content for applications that cannot do it themselves.

Ryu Network Operating System

(Click to View More Details Online)
https://github.com/osrg/ryu

NTT Communications
PUBLIC | PRIVATE
http://www.ntt.com

Description of Project: Ryu is a network OS that integrates with OpenStack and supports OpenFlow. It provides a logically centralized controller and well-defined API that make it easy for operators to create new network management and control applications.

Relevant Categories
NFVI - Network Operating Systems, NFVI - Control
Value Proposition
Ryu supports various protocols for managing network devices, such as OpenFlow, Netconf, OF-config, etc.The framework facilitates development by providing the basic functions for controlling the data plane and the functions that are common to SDN applications. Ryu has the OpenFlow controller function, so SDN applications that use OpenFlow switches can easily be developed using Ryu.
Licensing
Apache

Launch Date
2011-12
Use Cases
1. L2 network architecture - Flat L2 networks regardless of the underlying physical network 2. Multi-tenant networks - Scalable multi-tenant isolations 3. Network control - Logically centralized controller for thousands of switches (OVS, openRlow switch)

Open Network Linux

(Click to View More Details Online)
http://github.com/opennetworklinux/ONL

Open Compute Project (OCP)
PUBLIC | **PRIVATE**
http://www.opencompute.org/

Description of Project: Open Network Linux is a Linux distribution for "bare metal" switches, that is, network forwarding devices built from commodity components. ONL uses ONIE to install onto on-board flash memory. Open Network Linux is a part of the Open Compute Project and is a component in a growing collection of open source and commercial projects.

Relevant Categories
NFVI - Network Operating Systems
Value Proposition
Open Network Linux (ONL) is the Linux distribution for bare-metal switches that runs underneath Big Switch's commercial Switch Light OS. Open Network Linux supports multiple switch fabric APIs including: OF-DPA, OpenNSL and SAI. Specific information about hardware support for the different APIs can be found in the HCL. Open Network Linux is compatible with most forwarding agents including: FRR, BIRD, Facebook FBOSS and Azure SONiC.
Licensing
GPLv2, GPLv3, Eclipse Public License 1.0

Launch Date
2014-01
Use Cases
1. Routing and Switching - ONL supports the Facebook FBOSS fowarding agent. FBOSS currently only runs on the Wedge with other systems in the works. 2. APIs - ONL provides multiple options for forwarding agents and APIs to control forwarding on switches. The options go from direct (OpenNSL, SAI, OF-DPA) to full systems (FBOSS). Each option is available via installable packages.

Open Network Install Environment (ONIE)

(Click to View More Details Online)
https://github.com/opencomputeproject/onie

Open Compute Project (OCP)
PUBLIC | **PRIVATE**
http://www.opencompute.org/

Description of Project: Created by Cumulus Networks, Inc. in 2012, the Open Network Install Environment (ONIE) Project is a small operating system, pre-installed as firmware on bare metal network switches, that provides an environment for automated operating system provisioning.The ONIE project enables a bare metal network switch ecosystem where end users can choose among different network operating systems.

Relevant Categories
NFVI - Hardware, Installation Environment
Value Proposition
ONIE is an open source “install environment” that runs on a management subsystem utilizing facilities in a Linux/BusyBox environment. This environment allows end-users and channel partners to install the target network OS as part of data center provisioning, in the fashion that servers are provisioned. ONIE enables switch hardware suppliers, distributors and resellers to manage their operations based on a small number of hardware SKUs.
Licensing
GPLv2

Launch Date
2013-06
Use Cases
1. Reduce costs - ONIE enables switch hardware suppliers, distributors and resellers to manage their operations based on a small number of hardware SKUs. This in turn creates economies of scale in manufacturing, distribution, and stocking.

SONiC

(Click to View More Details Online)
http://github.com/Azure/SONiC/projects

Open Compute Project (OCP)
PUBLIC | **PRIVATE**
http://www.opencompute.org/

Description of Project: SONiC is a collection of networking software components required to have a fully functional L3 device. It is designed meet the requirements of a cloud data center. It is fully open-sourced at OCP.

Relevant Categories
NFVI - Switching and Routing, NFVI - Network Operating Systems
Value Proposition
True to its name, SONiC is about cloud speed and scale – and because it can easily extend with other open source, third party or proprietary software components, it can ultimately speed time to market and greatly improve datacenter efficiencies. SONiC is a uniquely extensible platform, with a large and growing ecosystem of hardware and software partners, that offers multiple switching platforms and various software components.
Licensing
Apache, GPLv2, Multiple different licenses for components.

Launch Date
2016-03
Use Cases
1. Hardware development - SONiC is built on the Switch Abstraction Interface (SAI), which defines a standardized API. Network hardware vendors can use it to develop innovative hardware platforms keeping the programming interface to ASIC consistent. 2. In-service upgrades - It does this in conjunction with Switch State Service (SWSS), a service that takes advantage of open source key-value pair stores to manage all switch state requirements and drives the switch toward its goal state. 3. Network management - Monitoring and diagnostic capabilities are also key for large-scale network management. Microsoft continuously innovates in areas such as early detection of failure, fault correlation, and automated recovery mechanisms.

OpenConfig Project

(Click to View More Details Online)
http://openconfig.net/software/

OpenConfig
PUBLIC | **PRIVATE**
http://www.openconfig.net/

Description of Project: OpenConfig is a collaborative effort by network operators to develop program interfaces and tools for managing networks in a more dynamic, vendor-neutral way. OpenConfig's initial focus is on compiling a consistent set of vendor-neutral data models (written in YANG) based on actual operational needs from use cases from multiple network operators. These models may be developed directly by OpenConfig, or compiled from 3rd party modules that conform with the OpenConfig requirements.

Relevant Categories
NFV MANO
Value Proposition
OpenConfig will enable network operators who seek to address the increased complexity, poor reliability, and greater cost of today's network infrastructure. OpenConfig aims to provide a more dynamic and programmable infrastructure and common way to configure and manage network elements. Key focus areas being addressed by OpenConfig include development of vendor-neutral APIs for declarative configuration, modern, scalable monitoring solutions, and adoption and native support on vendor platforms.
Licensing
Apache

Launch Date
2014-10
Use Cases
1. The OpenConfig working group consists of technical contributors from a variety of network operators representing a broad set of use cases.

Central Office Re-architected as a Datacenter (CORD)

Open Networking Foundation
PUBLIC | **PRIVATE**
https://www.opennetworking.org/

(Click to View More Details Online)
https://opencord.org/download/

Description of Project: The edge of the operator network (such as the central office for telcos and the head-end for cable operators) is where operators connect to their customers. CORD is a project intent on transforming this edge into an agile service delivery platform enabling the operator to deliver the best end-user experience along with innovative next-generation services.

Relevant Categories
NFVI - Infrastructure Layer, NFVI - Network Operating Systems
Value Proposition
With CORD, providers can leverage a common hardware and software infrastructure to offer traditional connectivity as well as cloud services for residential, enterprise and mobile customers. It allows these customers to configure their service packages with ease and in almost real time. In addition, CORD allows third parties to offer innovative services to common customers with a variety of partnership models.
Licensing
Apache, ON.Lab Contributor License Agreement

Launch Date
Not provided
Use Cases
1. CORD is reinventing edge of operator networks. By blending best of Cloud, SDN and NFV, it provides complete integrated solutions for fixed access, mobile and wireless 5G access, mobile core and enterprise service delivery.

Open Networking Operating System (ONOS)

Open Networking Foundation (ONF)
PUBLIC | **PRIVATE**
https://www.opennetworking.org/

(Click to View More Details Online)
https://onosproject.org/software/

Description of Project: ONOS has successfully enabled the beginning of the SDN revolution. Combining ONOS with white box switches and ONOS applications enables new forms of innovation never before possible with closed legacy networks.

Relevant Categories
NFVI - Control
Value Proposition
ONOS has been designed to take a disruptive approach to networking, leveraging white box merchant silicon hardware to build carrier-grade solutions. By moving network control into the ONOS cloud controller, innovation is enabled and end-users can easily create new network applications without the need to alter the dataplane systems.
Licensing
Apache, BSD, MIT, ON.Lab Contributor License Agreement

Launch Date
2014-12
Use Cases
1. ONOS has native support for the packet/optical network. It does so by offering an innovative converged topology view, allowing rapid introduction of new services and unprecedented optimization.

OpenStack Neutron

(Click to View More Details Online)
https://github.com/openstack/neutron

OpenStack Foundation
PUBLIC | **PRIVATE**
http://www.openstack.org/

Description of Project: Neutron is an OpenStack project to provide "networking as a service" between interface devices (e.g., vNICs) managed by other Openstack services (e.g., Nova). Starting in the Folsom release, Neutron is a core and supported part of the OpenStack

Relevant Categories
NFVI - Infrastructure Layer
Value Proposition
Neutron provides a simple networking solution for the cloud. It includes a set of application program interfaces (APIs), plug-ins and authentication/authorization control software that enable interoperability and orchestration of network devices and technologies within infrastructure-as-a-service (IaaS) environments. Neutron has a component on the controller node called the neutron server, along with a number of servers including NEC OpenFlow, Open vSwitch, Cisco switches, and Linux bridging.
Licensing
Apache

Launch Date
2013-07
Use Cases
1. Multi-homed Openstack cloud - Run a routing protocol (example BGP) against each uplink network provider. By announcing floating IP prefixes to those peers, the Neutron network would be reachable by the rest of the internet via both paths. 2. Routed Model for Floating IPs on External Network - This use case will allow an external network with a large public IP space to possibly span more than one L2 network. 3. MPLS/BGP - VPNaaS currently has experimental implementations for IPSec and TLS. The VPN endpoint is inserted in to a Neutron router.

OpenStack Tacker

(Click to View More Details Online)
http://git.openstack.org/cgit/openstack/tacker/

OpenStack Foundation
PUBLIC | **PRIVATE**
http://www.openstack.org/

Description of Project: Tacker is OpenStack project building an open NFV orchestrator with in-built general purpose VNF manager to deploy and operate virtual network functions (VNFs) on an NFV platform. It is based on ETSI MANO architectural framework and provides full functional stack to orchestrate VNFs end-to-end.

Relevant Categories
NFV MANO
Value Proposition
Tacker VNF Manager (VNFM) component manages the life-cycle of a Virtual Network Function (VNF). VNFM takes care of deployment, monitoring, scaling and removal of VNFs on a Virtual Infrastructure Manager (VIM).
Licensing
Apache

Launch Date
2015-12
Use Cases
1. vCE - Tacker API can be used by SP's OSS / BSS or an NFV Orchestrator to deploy VNFs in SP's network to deliver agile network services for remote Customer networks 2. vCPE - Tacker API can be used by SP's OSS / BSS or an NFV Orchestrator to manage OpenStack enabled remote CPE devices to deploy VNFs to provide locally network services at the customer site. 3. PE - Tacker API can be used by SP's OSS / BSS or an NFV Orchestrator to deploy VNFs within SP's network to virtualize existing network services into a Virtual Function.

P4

(Click to View More Details Online)
https://github.com/p4lang/p4factory

P4.org
PUBLIC | **PRIVATE**
http://p4.org/

Description of Project: P4 is a programming language designed to allow programming of packet forwarding planes. In contrast to a general purpose language such as C or Python, P4 is a domain-specific language with a number of constructs optimized around network data forwarding. P4 is an open-source, permissively licensed language and is maintained by a non-profit organization called the P4 Language Consortium

Relevant Categories	Launch Date
NFVI - Infrastructure Layer, NFVI - Switching and Routing	2015-02
Value Proposition	**Use Cases**
P4, the language is specifically targeted at packet forwarding applications, is designed to meet specific goals including 1) Programs created with P4 are designed to be implementation-independent 2) P4 is designed to be protocol-independent and 3) Protocol independence and the abstract language model allow for reconfigurability.	1. Network monitoring and debugging - Features such as Explicit Congestion Notification (ECN), and In-Band Network Telemetry can be used to to track the path that packets travel through the network. 2. Advanced data structures - Advanced data structures can be used to implement data-plane applications. 3. Software switching - P4 contains software tools that are essential to creating software based switches such as the P4 compiler and debugger.
Licensing	
Apache	

Project Calico

(Click to View More Details Online)
https://github.com/projectcalico/calico

Tigera
PUBLIC | **PRIVATE**
https://www.tigera.io/

Description of Project: Project Calico's mission is to provide an open pure L3 approach to virtual networking for highly scalable data centers that integrates seamlessly with could orchestration systems (such as OpenStack) to enable secure IP communication between virtual machines, containers, or bare metal workloads.

Relevant Categories	Launch Date
NFVI - Switching and Routing	2014-07
Value Proposition	**Use Cases**
Calico's simple approach to data center networking integrates seamlessly with cloud orchestration systems to enable secure IP communication between virtual machines, containers, or bare metal workloads. Based on the same scalable IP network principles as the Internet, Calico leverages the existing Linux kernel forwarding engine without the need for virtual switches or overlays. Each host propagates workload reachability information (routes) to the rest of the data center.	1. Scalable, distributed control plane - Built on proven IP routing technology to connect containers (and VMs) to one another as well as to underlying infrastructure which enables distributed security policy rules. 2. Network security - Calico contains a security layer that enables developers and operations to easily define with fine granularity which connections are allowed and which are not. These rules implement and extend the Kubernetes Network Policy API. 3. Integrated cloud platforms - Calico comes out of the box with a variety of plug-ins and recipes. Support for industry standard APIs such as Container Network Interface (CNI), Neutron, and libnetwork.
Licensing	
Apache	

Open Virtual Network (OVN)

(Click to View More Details Online)
https://github.com/openvswitch/ovs

VMware, Inc.
PUBLIC | PRIVATE
http://www.vmware.com

Description of Project: Open Virtual Network (OVN) is a new network virtualization project that brings virtual networking to the Open vSwitch user community. It is being developed by the core OVS team.

Relevant Categories
NFVI - Infrastructure Layer, NFVI - Switching and Routing
Value Proposition
OVN complements the existing capabilities of OVS to add native support for virtual network abstractions, such as virtual L2 and L3 overlays and security groups. Some high level features of OVN include 1) virtual networking abstraction for OVS, implemented using L2 and L3 overlays, 2) OVN supports flexible ACLs (security policies) implemented using flows that use OVS connection tracking, 3) OVN natively supports distributed L3 routing using OVS flows, and more.
Licensing
Apache

Launch Date
2015-01
Use Cases
1. OpenStack networking - Open vSwitch has always been to provide a production-ready low-level networking component for hypervisors that could support a diverse range of network environments. 2. L2/L3 virtual networking - OVN's design is narrowly focused on providing L2/L3 virtual networking. This distinguishes OVN from general-purpose SDN controllers or platforms. 3. Large scale deployments - Support for large scale deployments is a key goal of OVN. OVN has been deployed to of several hundred nodes. OVN has been tested by simulating deployments of thousands of nodes using the ovn-scale-test project.

SDNCentral, LLC
3511 Ringsby Ct, #101
Denver, CO 80216 USA
www.sdxcentral.com

The Trusted News and Resource Site for SDx, SDN, NFV, Cloud and Virtualization Infrastructure

Rev A