



Máster Universitario en Software Libre

Curso Académico 2014/2015

Proyecto Fin de Máster

The OpenDaylight Open Source Project

Autor: Sergio Najib Arrouibi Braojos

Tutor: Dr. Gregorio Robles

Agradecimientos

A mi familia y a mi pareja, por su apoyo incondicional

*Al equipo de Libresoft de la Universidad Rey Juan Carlos, por su afán en enseñar el qué y el
porqué del Software Libre*

Dedicatoria

Para todos aquéllos que hacen posible el fenómeno del Software Libre

(C) 2014 Sergio Najib Arroudji Braojos. Some rights reserved.

This document is distributed under the Creative Commons Attribution-ShareAlike 3.0 license,

available in <http://creativecommons.org/licenses/by-sa/3.0/>

Source files for this document are available at

<http://github.com/sarroudji/MFP/opendaylight/>

Contents

1	Introduction	19
1.1	Terminology	19
1.1.1	Open Source Programmable Networking	19
1.2	About this document	20
1.2.1	Document structure	20
1.2.2	Scope	21
1.2.3	Methodology	21
2	Goals and Objectives	23
2.1	General Objectives	23
2.2	Subobjectives	23
2.2.1	Acquire competence on OpenDaylight project	23
2.2.2	Analyze OpenDaylight project from an Open Source perspective	24
2.2.3	Statistics and measures of the OpenDaylight project	24
3	OpenDaylight: A first view	25
3.1	OpenDaylight Project	25
3.2	SDN	29
3.2.1	What is SDN?	29
3.2.2	SDN: Market share and expectations	32
3.3	NFV	34
3.3.1	What is NFV?	35
3.3.2	SDN/NFV relationship	36
3.3.3	NFV benefits	37

4 OpenDaylight: Economic Aspects	39
4.1 SDN and NFV: A new market	40
4.2 SDN/NFV with OpenDaylight: Software Cost Reduction	43
4.3 SDN/NFV technology: Cost Reduction	45
4.4 OpenDaylight: Economic Aspects to go Open Source	49
4.4.1 Technology Adoption	49
4.4.2 The Operator's View	51
5 OpenDaylight: Legal Aspects	57
5.1 OpenDaylight License: EPLv1.0	57
5.1.1 History Of The Eclipse Public License	60
5.1.2 Main Features of EPLv1.0	61
5.1.3 Other Software Development Considerations	61
5.2 Other Licensing Considerations	63
5.3 Bylaws	65
6 OpenDaylight: Governance and Community Management	67
6.1 OpenDaylight Project Governance	67
6.1.1 Technical Steering Committee (TSC)	68
6.1.2 Board of Directors	70
6.2 OpenDaylight Community Management	71
6.2.1 Communication Channels	72
6.2.2 Community Tools	73
6.2.3 Marketing and Community Building	76
7 OpenDaylight: Technical Aspects	81
7.1 Technical Overview	81
7.2 OpenDaylight Projects	84
7.3 OpenDaylight Source Code	91
7.4 Release Roadmap	96
8 OpenDaylight: Project Evaluation	103
8.1 Metrics	104

8.2 Tools and Data Sources	105
8.3 Metrics Analysis	107
8.3.1 Source Code Metrics	108
8.3.2 Mailing List Activity	117
8.3.3 Q&A Activity	125
8.3.4 Bug discussion and close efficiency	131
9 Conclusions	145
9.1 Evaluation	145
9.2 Lessons learned	146
9.2.1 Lesson 1: Why Open Source?	146
9.2.2 What I learned	147
9.2.3 Knowledge and skills acquired in the M.Sc. studies that helped me in this work	147
9.3 Future work	148
9.3.1 More on...	148
9.3.2 Other aspects	149
9.3.3 Other focuses	149
A EPLv1.0	151
B Bylaws Summary	157
C TSC Charter	173
D TSC Policy	179
E Mail Lists	181
Bibliography	187

List of Figures

3.1	SDN 3 Layer Architecture	31
3.2	SDN Market: 2018 SDNCentral estimations	33
3.3	Vision of Network Functions Virtualization	36
3.4	Network Functions Virtualization relationship with SDN	37
4.1	Gartner Hype Cycle	40
4.2	SDN Hype Cycle	42
4.3	Open Source Software: Service Revenue Increase	44
4.4	Open Source Software: More potential customers	45
4.5	Mobile Backhaul Network Invest Expected	46
4.6	2017 SDN CAPEX Reduction Cost	47
4.7	2017 SDN OPEX Reduction Cost	48
4.8	NFV Reduction Cost	49
4.9	Open Source on SDN/NFV - A requirement analysis	54
4.10	Open Systems and Open Source: The SDN linkage	55
7.1	OpenDaylight Architecture Diagram	83
7.2	OpenDaylight Controller	86
7.3	OpenDaylight Programming Language Distribution	95
7.4	OpenDaylight LOC based Project Distribution	98
7.5	OpenDaylight Hydrogen Architecture Diagram	99
7.6	OpenDaylight Helium New Features Diagram	100
7.7	OpenDaylight Projects Life Cycle	101
8.1	OpenDaylight Project Onion Model	114

8.2 OpenDaylight Project Commits Per Month	117
8.3 OpenDaylight Project Mail Messages Per Month	124
8.4 OpenDaylight Project Q&A Questions Per Month	130
8.5 OpenDaylight Project Q&A Answers Per Month	131
8.6 OpenDaylight Bug Status Percentage	135
8.7 OpenDaylight Project Bugs Opened Per Month	139
8.8 OpenDaylight Project Bugs Resolved Per Month	141
8.9 OpenDaylight Project BMI (Resolved vs. Opened)	142

List of Tables

1.1	Structure of this thesis.	20
3.1	SDN 2018 Sub-Market Expectations	32
4.1	SDN 2017 OPEX Savings Estimations	48
6.1	Example of OpenDaylight’s OpenDove Project mail lists	73
7.1	OpenDaylight projects’ repositories	92
7.2	OpenDaylight projects’ LOC distribution	97
8.1	OpenDaylight top 20 committers	110
8.2	OpenDaylight top 20 committer companies	111
8.3	OpenDaylight top 20 most active mail lists	120
8.4	OpenDaylight top 20 mail lists with the most users involved	121
8.5	OpenDaylight top 10 people asking on Q&A OpenDaylight project	128
8.6	OpenDaylight top 10 answering people on Q&A OpenDaylight project	129
8.7	OpenDaylight top 10 people with higher reputation on Q&A OpenDaylight project	129
8.8	OpenDaylight top 10 most BUG resolute people	136
B.1	OpenDaylight Bylaws Summary: Article I	157
B.2	OpenDaylight Bylaws Summary: Article II	158
B.3	OpenDaylight Bylaws Summary: Article III	158
B.4	OpenDaylight Bylaws Summary: Article IV	159
B.5	OpenDaylight Bylaws Summary: Article IV	160
B.6	OpenDaylight Bylaws Summary: Article V	161
B.7	OpenDaylight Bylaws Summary: Article VI	162

B.8 OpenDaylight Bylaws Summary: Article VI	163
B.9 OpenDaylight Bylaws Summary: Article VII	164
B.10 OpenDaylight Bylaws Summary: Article VIII	165
B.11 OpenDaylight Bylaws Summary: Article VIII	166
B.12 OpenDaylight Bylaws Summary: Article IX	167
B.13 OpenDaylight Bylaws Summary: Article X	168
B.14 OpenDaylight Bylaws Summary: Article XI	169
B.15 OpenDaylight Bylaws Summary: Article XII	170
B.16 OpenDaylight Bylaws Summary: Article XIII	171
B.17 OpenDaylight Bylaws Summary: Article XIV	172
E.1 OpenDaylight’s OpenDove Project mail lists	186

Summary

The main goal of this work is to perform a comprehensive analysis of the OpenDaylight Open Source Project. This collaborative project, hosted by the Linux Foundation, has been created in order to achieve following goal: **Develop an Open Source Programmable Networking Platform.**

In this document, a detailed study of many different aspects having to do with a project of this type of characteristics will be analyzed. Examples of these aspects are, for instance, the licensing mechanisms adopted by the project, the community behind the project, descriptive statistics about the project, the economic aspects around the project, the technical state of the project, and how OpenDaylight has progressed from its foundation date until today.

Resumen

El principal objetivo de este trabajo es realizar un análisis detallado del proyecto de código abierto OpenDaylight. Este proyecto colaborativo, perteneciente a la Linux Foundation, ha sido creado para una misión principal: **Desarrollar una plataforma de Código Abierto para Redes Programables.**

En este documento, un estudio pormenorizado de los distintos aspectos asociados al Código Abierto en un proyecto de estas características serán analizados. Un ejemplo de los aspectos a estudiar será el mecanismo de sistema de licencias, la comunidad que reside detrás del proyecto, estadísticas descriptivas del proyecto, los aspectos económicos, si los hubiera, alrededor del proyecto, así como el estado a nivel técnico o cómo ha evolucionado OpenDaylight desde la fecha de su fundación.

Chapter 1

Introduction

1.1 Terminology

1.1.1 Open Source Programmable Networking

In the same way Cloud Computing means a revolution in Computer Science, where computing resources are considered as flexible facilities that provide different kinds of services, Networking is evolving into a new era. Networks hardware is considered to be a resource that is flexible and easily programmable in order to adapt to the specific Networking necessities. **SDN** [Fou14g] (Software Defined Networking) and **NFV** [ETS13] (Network Functions Virtualization) technologies have been strongly brought to foreground in Networking Science, in order to provide on the one hand management of the network services through abstraction of lower level functionality and characteristics, and on the other hand to provide a network architecture virtualization technologies to simulate nodes existing on a network.

Around these technologies, The OpenDaylight Open Source Project, hosted by the Linux Foundation, has appeared to provide mechanisms not only to use the previously described technology, but also to guarantee to all the potential users of this technology the freedoms that Open Source mean, i.e.:

- The freedom to use the program, for any purpose.
- The freedom to study and adapt the programs (modify).

- The freedom to distribute the program to others.
- The freedom to distribute to others the modified versions of the program.

Analyzing the OpenDaylight Open Source Project is a good opportunity to investigate, on the one hand, an incipient technology and how an also incipient Open Source Project can influence not only that technology, but also the different aspects around a technology, as people involved in the development of the technology (the community), the impact on the economic aspects around the technology, and, above all, the aspects that Open Source itself supposes for this kind of projects.

1.2 About this document

1.2.1 Document structure

In order to provide a detailed analysis of The OpenDaylight Open Source Project, this work contains different chapters that describe some important aspects around the project from an Open Source perspective:

Chapter Name	Description
Introduction	A complete introductory overview of The OpenDaylight Open Source Project
OpenDaylight Economic Aspects	Detailed study of economic aspects around the project
OpenDaylight Legal Aspects	Complete analysis of the license or licenses used in OpenDaylight and the different advantages and disadvantages of this kind of licensing scheme
OpenDaylight Governance and Community Management	Analysis of the community and its governance, its organization, communication channels, politics and mission
OpenDaylight Technical Aspects	Study of the technology behind the project and its scope
OpenDaylight Project Evaluation	Graphics and statistics around the OpenDaylight project, such as number of committers, number of open/closed issues, mail lists statistics, activity and in general all statistics to perform an evaluation of the project from a statistical perspective

Table 1.1: Structure of this thesis.

1.2.2 Scope

This document is not entitled to perform a complete description of the protocols and technologies that OpenDaylight uses in order to achieve the programmability network platform it pretends to provide. They will be smoothly analyzed in order to clarify how OpenDaylight works, but not all the technologies will be described, and those described will not be done in deep.

Beyond the purely technical aspects, this work pretends to focus on analyzing OpenDaylight project from an Open Source perspective, analyzing the pros and cons of Open Source, and the different aspects that an Open Source Project faces.

1.2.3 Methodology

Different tools and documentation have been used in order to perform this work. The documentation used for this purpose has been basically web pages available around The OpenDaylight Open Source Project [Pro14d]. A complete description of the used documentation will be provided in the Bibliography.

Regarding tools, apart from web browsers used to navigate through the project documentation, the different metrics obtained around OpenDaylight project have been obtained through the MetricsGrimoire metrics toolset [Car14a]. In particular, among the MetricsGrimoire tools, following have been used: CVSanaly [Car14c], Bicho [Car14b] and MailingListStats [Car14d].

Apart from that, the SLOCCount [Whe14] tool has been used in order to detect the amount of lines of code that exist on the project, as well as the different programming languages used in the project implementation and its respective distribution inside the project. SLOCCount output has been processed with a text processing tool, gawk [Sys89]. Other utility, Python matplotlib [mat14a], has been used to generate graphics for certain metrics.

Chapter 2

Goals and Objectives

2.1 General Objectives

The general objectives of this work are, basically, on the one hand, acquiring knowledge, competence and skills around OpenDaylight, while, on the other hand, analyzing the project from an Open Source perspective.

2.2 Subobjectives

In order to achieve the main goals this work pursues, the following operative objectives have been identified:

1. Acquire competence on OpenDaylight project.
2. Analyze OpenDaylight project from an Open Source perspective.
3. Extract the most significant statistics in OpenDaylight project to determine its state of the art.

2.2.1 Acquire competence on OpenDaylight project

- Perform an overall description of the OpenDaylight project.
- Acquire competence on OpenDaylight documentation, and synthesize the most important aspects of the project.

- Analyze the technical aspects of the project.

2.2.2 Analyze OpenDaylight project from an Open Source perspective

- Analyze the licensing model followed by the project
- Study the economic aspects behind this project
- Determine the different aspects behind the project's community

2.2.3 Statistics and measures of the OpenDaylight project

- Perform a complete measures compilation of the project
- Evaluate the State of the Art of the project based on its measures

Chapter 3

OpenDaylight: A first view

3.1 OpenDaylight Project

OpenDaylight is a collaborative project, started in April 8, 2013 [Ann14b], and developed inside the Linux Foundation Collaborative Projects ecosystem. This fact is the first one to remark, due to the fact that, normally, projects under the Linux Foundation are supposed to have strong economic and infrastructure support, due to the following reasons:

- **Linux Foundation** is one of the most profitable Open Source organizations, mostly due to the importance of its main project, **the Linux Kernel Project**. In 2012, The Linux Foundation obtained a **total revenue of \$17,123,662**, according to [Aud12]. Among the collaborators of the Linux Foundation Collaborative Projects, the following are remarkable [Fou14e].
 1. Cisco
 2. Google
 3. HP
 4. IBM
 5. Intel
 6. Qualcomm Innovation Center
 7. Samsung
 8. New York Stock Exchange Technologies

- **Collaborative Projects** under the Linux Foundation are only a few, in a more focused strategy compared to other organizations hosted in other organizations such as Mozilla Foundation, the Apache Software Foundation or the Free Software Foundation, that fund more projects. Examples of other projects hosted by the Linux Foundation Collaborative Projects are [Fou14e]:

1. Allseen Alliance
2. Code Aurora
3. MeeGo
4. OpenVirtualization Alliance
5. The Bel Language
6. OpenMama
7. Tizen
8. Xen Project
9. Yocto Project

Being a Linux Foundation Collaborative Project, OpenDaylight is part of a technology that is booming in the last years, having to do with **SDN** and, to a lesser extent, with NFV, in particular in the Networking Industry. According to OpenDaylight project [Pro14e]:

OpenDaylight is a community-led, open, industry-supported framework, for accelerating adoption, fostering new innovation, reducing risk and creating a more transparent approach to Software-Defined Networking.

OpenDaylight is a Collaborative Project at The Linux Foundation. It is structured using Open Source development best practices, and is comprised of the leading organizations in the technology industry.

In particular, as stated in the “About” section of OpenDaylight project [Abo14], the adoption of new technologies and pursuit of programmable networks has the potential to significantly improve levels of functionality, flexibility and adaptability of mainstream datacenter architectures.

Leveraging this abstraction to its fullest requires some adaptation from the network perspective, as well as adaptation and evolution to a Software-Defined architecture. One of the most important architectural elements required to achieve this goal is SDN and NFV platform, to enable network control and programmability from a centralized point.

SDN, and NFV to a lesser extent, are a hotbed of innovation nowadays, with a bunch of vendors bringing products and technologies to market. Ironically, the myriad options may prove counterproductive to SDN and NFV adoption. At this early stage of SDN and NFV adoption, the industry acknowledges the benefits of establishing an open, reference framework for programmability and control through an Open Source SDN and NFV solution.

Such a framework maintains the flexibility and choice for organizations to deploy SDN and NFV as they still mitigate many of the risks of adopting early stage technologies and integrating with existing infrastructure investments.

With OpenDaylight, a community with some of the most important companies involved in Computer Science, Virtualization and Networking Industry, has come together to fill this need through the combination of open community developers and Open Source code and project governance that guarantees an open, community decision making process on business and technical issues. **Establishing an Open Source project in this way is designed to help accelerate the development of technology available to users** and enable widespread adoption of SDN and create a solid foundation for NFV.

OpenDaylight can be a **core component** within any SDN architecture, providing an Open Source SDN and NFV controller that enables users to reduce operational complexity, extend the life of their existing infrastructure hardware and enable new services and capabilities only available with SDN.

Whether an organization is an enterprise IT provider, a network service provider or a Cloud services provider, it can begin taking advantage of SDN and NFV using a community-driven, Open Source controller framework available today.

The first Open Source software release by the OpenDayLight Open Source Project is known as “**Hydrogen**”. This release, Hydrogen, is the first simultaneous release of OpenDaylight, delivering three different editions to help a wide array of users get up and running as quickly as possible. The three options available are a Base Edition, a Virtualization Edition and a Service Provider Edition, and all of them are already available for download.

But, **how important is this project from the Neworking Technology Industry main companies perspective?**. A view to the main members board of the OpenDaylight project is a clear idea of the importance of this project from the industry perspective, as well as how strategic is this technology perceived from the most important market leaders from not only the Networking Market but also from Computer Science and Computing Virtualization Market leaders.

Among the **platinum members of the project**, following are noteworthy [Mem14]:

- Brocade
- Cisco
- Citrix
- Ericsson
- HP
- IBM
- Juniper Networks
- Microsoft
- RedHat

Meanwhile, the **golden members of the project**, are:

- NEC
- vmWare

Last, but not least, there is a huge number of members (more than 25) considered as **silver members of the project**. Among these members, some of the most important are:

- Intel
- Oracle
- Dell
- Huawei
- Fujitsu
- Avaya
- H3C
- Plexxi
- Zte
- ... and many others

So, to summarize, The OpenDaylight Project is a collaborative Open Source project that aims to accelerate adoption of Software-Defined Networking (SDN) as well as Network Functions Virtualization (NFV) for a more transparent approach that fosters new innovation and reduces risk.

As shown before, there is a huge and increasing interest in this kind of technology by the most important companies. Why this interest in these technologies? Why and how are they changing the market in this kind of industry? Why is there a so huge interest in SDN from the industry main actors?

In next sections a quick view of these technologies is provided, as well as some indicators of how quick this kind of technology is increasing inside the networking market.

3.2 SDN

3.2.1 What is SDN?

SDN technology allows network administrators to manage network resources as network services through abstraction of lower level provided functionality, by the physical separation of

the network control plane from the forwarding plane, and where a control plane controls several devices [Fou14g].

Software-Defined Networking (SDN) has appeared as a dynamic, manageable, cost-effective, and adaptable emerging architecture, ideal for the high-bandwidth, dynamic nature of today's applications.

This architecture decouples the network control and forwarding functions, enabling the first one to become directly programmable and the underlying infrastructure to be abstracted for applications and network services.

Related to this, The OpenFlow™ protocol appears as a foundational element for building SDN solutions.

The SDN architecture, depicted in Figure 3.1, is characterized by the following aspects:

- **Open standards-based and vendor-neutral:** Being implemented through open standards, SDN simplifies network design and operation. Instructions are provided by SDN controllers, which follow a particular set of open protocols and devices, instead of multiple, vendor-specific devices and protocols.
- **Directly programmable:** Network control plane is directly programmable. This aspect is possible due to the fact that control plane is decoupled from forwarding plane.
- **Agile:** Abstracting control from forwarding plane lets network administrators dynamically adjust network-wide traffic flow to meet particular needs of the network at each particular moment.
- **Centrally managed:** Network intelligence is centralized in software-based SDN controllers. This kind of elements maintain a global view of the network topology. Network itself appears to applications and policy engines as a single, logical switch.
- **Programmatically configured:** Network managers can configure, manage, secure, and optimize network resources through SDN very quickly via dynamic, automated SDN

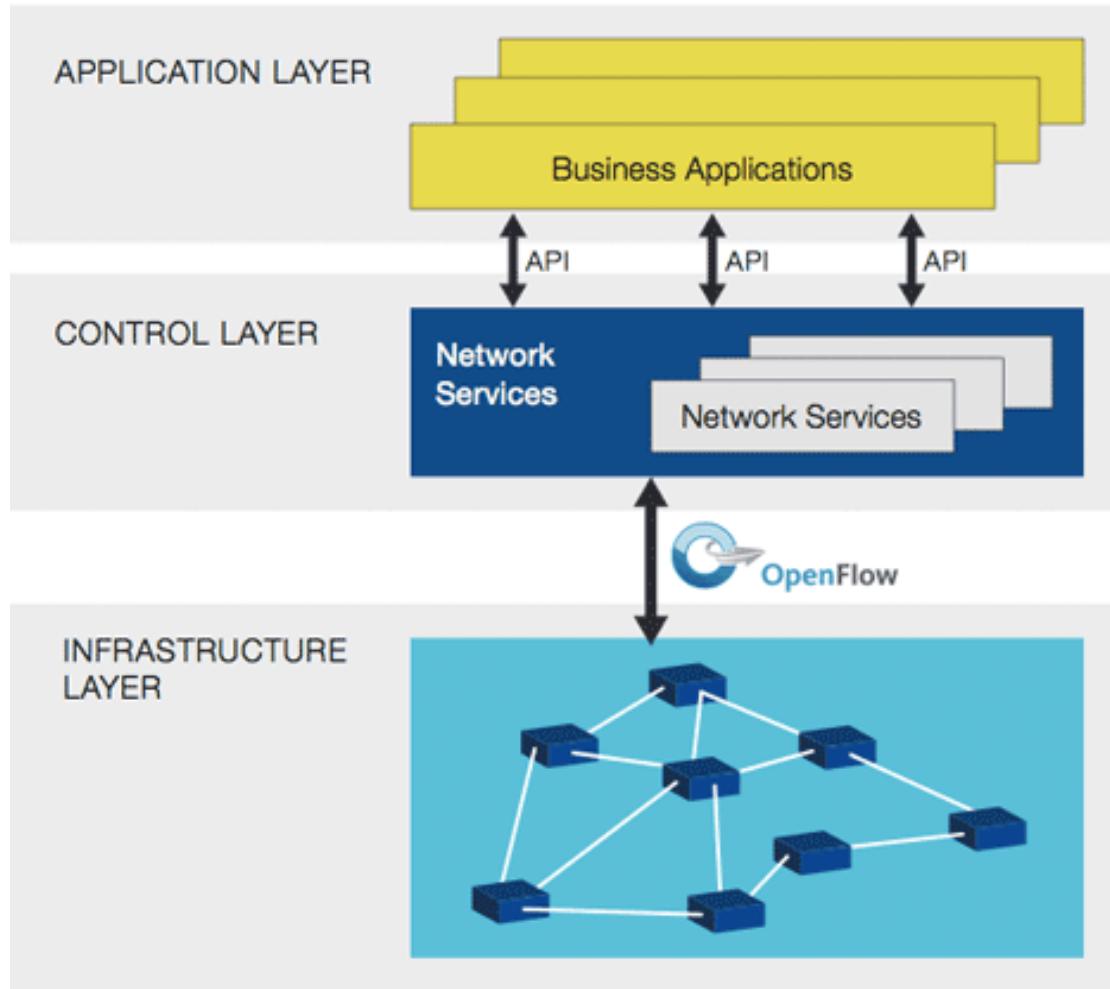


Figure 3.1: SDN 3 Layer Architecture

programs, which can be written themselves because the programs do not depend on proprietary software.

The aspects detailed previously make the SDN technology one of the most exciting technologies currently, together with Cloud Computing and Big Data. Indeed, the increasing interests in all of them has also to do with the fact that all these technologies, together with Virtualization, are necessarily linked among them.

Next section shows how the market has increased in the last years, together with the big expectations that SDN technology is supposed to have in the next years.

3.2.2 SDN: Market share and expectations

SDN is considered a boosting technology. In year 2013, the service-provider SDN market —hardware and software combined — was about **\$840 million in 2013** and is considered to grow to **\$15.6 billion in 2018**, according to a report released by ACG Research [Mat14b].

In this research, numbers don't include the enterprise market, and apply only to service providers, including large-data-center owners such as Google and the big data centers of carriers such as NTT or AT&T.

Here's how ACG's 2018 forecast breaks down into the different sub-markets taking into account the different main areas of SDN deployment:

SDN Submarket	Market expectation
Data center	\$3.3 billion
IP services	\$4.2 billion
Metro networks	\$4.3 billion
Core networks	\$3.8 billion

Table 3.1: SDN 2018 Sub-Market Expectations

The numbers reflect only “live” SDN deployments, i.e.: cases where service providers would use the technology, and not just installing it for future use.

However, it must be remarked that expectations are even higher considering other reports. By mid 2013, market considered for year 2013 was about \$1.5 billion, and expectations for year 2018 was about \$35.5 billion according to a research by SDNCentral [PP14].

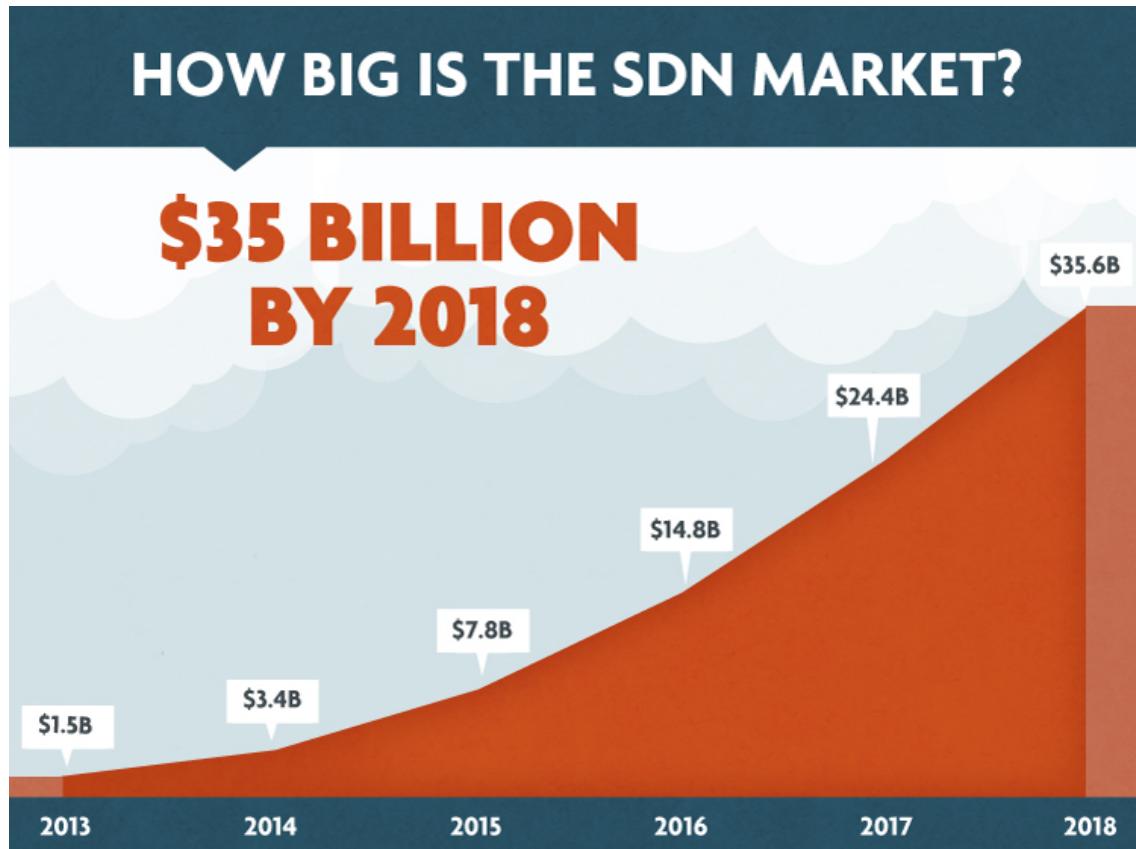


Figure 3.2: SDN Market: 2018 SDNCentral estimations

This last analysis and infographics doesn't compare directly with the previous one, as it wasn't limited to service providers, and included SDN-ready equipment, sales that represent customers preparing for SDN even if they wouldn't be using it immediately.

Analyzing the data of this last report, a number of key takeaways must be highlighted:

1. The software-defined networking SDN market is expected to surpass \$35 billion in 2018.
2. Adoption of SDN technology has accelerated in recent years from sales of \$10 million in 2007 to \$252 million in 2012.
3. The emergence of the software-defined networking market is supported by growth in venture capital investment in SDN-focused companies. Venture capital funding of SDN-related companies rose from \$10 million in 2007 to \$454 million in 2012.

According to this report, the main three areas driving the rise in SDN are:

1. Cloud Computing

2. Big Data

3. Mobility

Apart from pure SDN Market numbers, there are other aspects to consider. For example, the VC(Venture Capital) investment has grown, according to this report, from \$10 million in 2007 to \$454million in 2012.

This report also considers that Networking Industry spending Percentage on SDN will rise from 2% in 2013 to 40% in 2014, with more than 220 companies by mid 2013, compared to the zero existing in 2009.

Last, but not least, it must be remarked that up to more than \$1.5 billion has been invested in acquisitions of SDN related companies up to 2013, including the most remarkable one, the acquisition of Nicira Inc. by VMWare by July 2012, for approximately \$1.05 billion in cash plus approximately \$210 million of assumed unvested equity awards [VMW12].

Far from giving a very detailed information on SDN market, this section has shown through all the numbers above how important is the SDN market and how important could be an Open Source Software Project such as OpenDaylight, which is in the end a key component in SDN deployment.

3.3 NFV

Network Functions Virtualization is a concept directly tied to Network Operators, as, normally, their networks consist of a large and increasing variety of proprietary hardware and software applications that implement different functionality.

To launch a new service on a Network Operator usually requires yet another different kind of hardware, meaning finding the space and power to accommodate these new equipment. As time goes by and Network Operator infrastructure grows, adding new appliances becomes more difficult, by the increasing costs of energy, capital investment challenges and, of course the rarity of skills necessary to design, integrate and operate these new equipment.

Apart from that, end of life for hardware-based appliances is reached rapidly, requiring much of the integration and deploy cycle to be repeated with no revenue benefit in most of the cases. Besides this, hardware life cycles are becoming shorter, due to the fact that technology and services innovation accelerates quicker up on time.

3.3.1 What is NFV?

As defined by ETSI in [ETSI13] defines **Network Functions Virtualization** as follows:

Network Functions Virtualization aims to transform the way that network operators architect networks by evolving standard IT virtualization technology to consolidate many network equipment types onto industry standard high volume servers, switches and storage, which could be located in Datacentres, Network Nodes and in the end user premises.

What NFV aims to get is, basically, homogenize the hardware of a Network Operator to basically three kind of standard generic use hardware. In particular, by the use of **High Volume Servers**, **High Volume Storage** and **High Volume Ethernet Switches**. Accomplishing this homogenization involves the implementation of **Network Functions** in software. This Network Functions must run on a range of industry standard server hardware, and can be moved to, or instantiated in, various locations in the network as required, without the need for installation of new equipment, at least, from a functional perspective.

In the end, for Network Operators, NFV means migration of the different hardware specific nodes implementing different functionalities, such as DPI (Deep Packet Inspection), Firewalls, CDN (Content Delivery Network), Radio Access Network Nodes, Message Router, Carrier Grade NAT and/or any other network specific functionality, which involve **Fragmented non-commodity hardware**, to Network Functions (also known as **Virtual Appliances**) running on the previously defined Standard High Volume hardware.

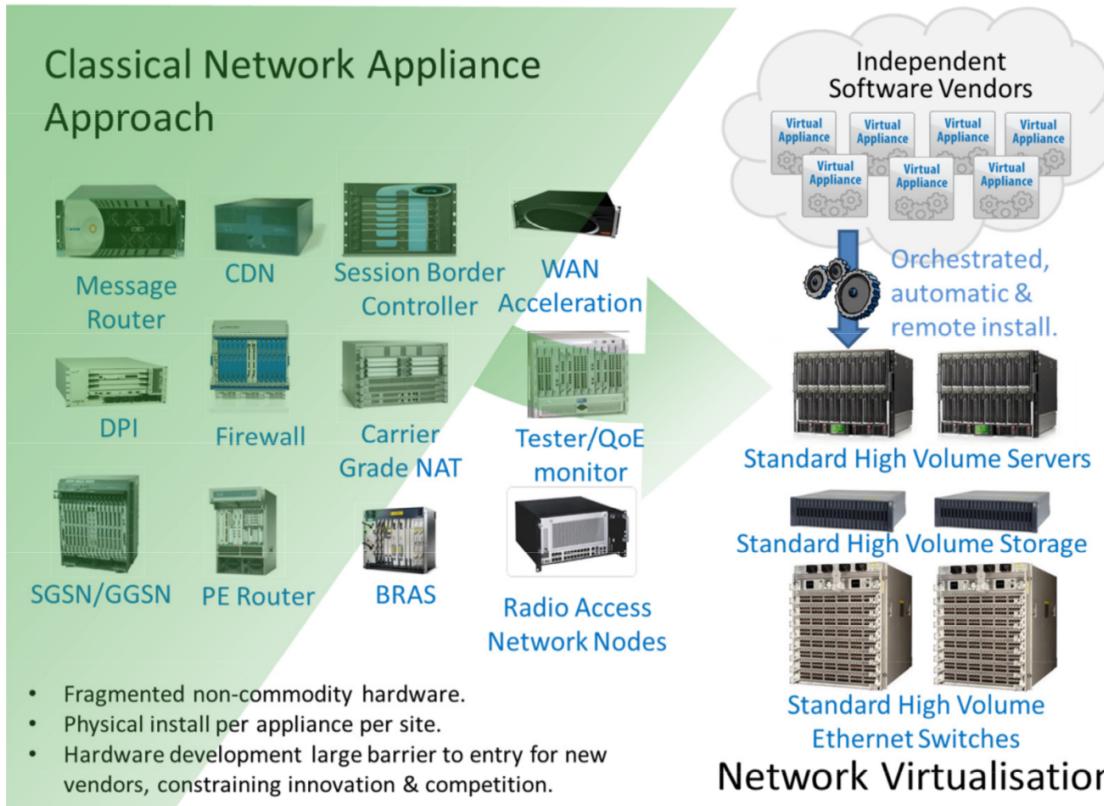


Figure 3.3: Vision of Network Functions Virtualization

3.3.2 SDN/NFV relationship

Network Functions Virtualization is highly complementary to Software Defined Networking (SDN), as recognized by ETSI in [ETSI13]. However NFV is not dependent on SDN (or vice-versa). NFV can be implemented without a SDN being required, although the **both solutions can be combined and potentially greater value accrued**.

Network Functions Virtualization goals can be achieved using non-SDN mechanisms, relying on the techniques currently in use in many operators. But approaches relying on the separation of the control and data forwarding planes as proposed by SDN can enhance performance, simplify compatibility with existing deployments, and facilitate operation and maintenance procedures.

Network Functions Virtualization is able to support SDN by providing the infrastructure upon which the SDN software can be run. Furthermore, Network Functions Virtualization aligns

closely with the SDN objectives to use commodity servers and switches.

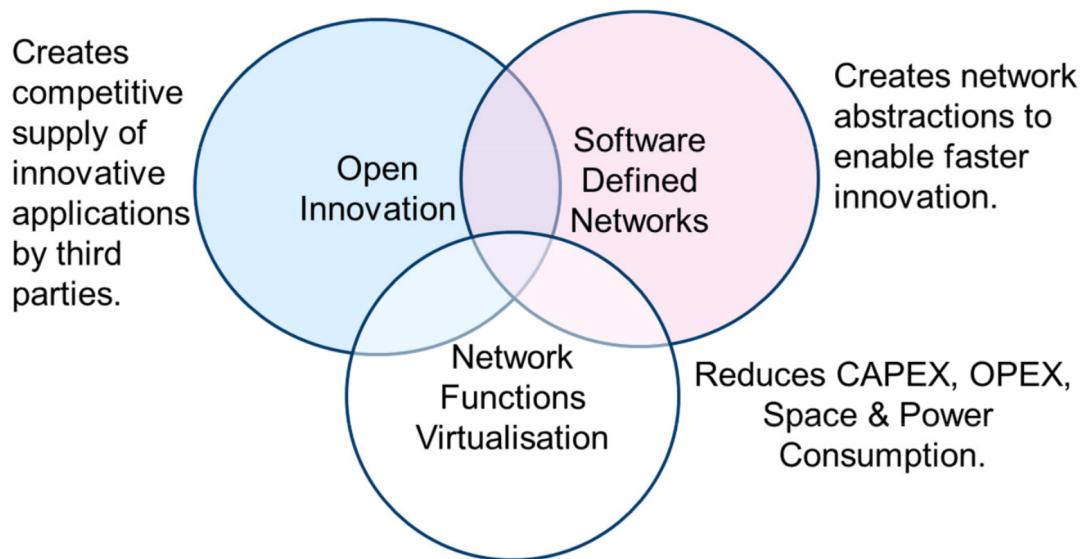


Figure 3.4: Network Functions Virtualization relationship with SDN

ETSI intends to work closely with organizations progressing work on SDN such as the Open Networking Foundation (ONF) whose work is specifically taken into account by SDN to achieve common goals.

3.3.3 NFV benefits

Taking into account the previous sections, where NFV concept has been detailed, an analysis of the main benefits of the migration of the Network Operators to this kind of infrastructure based on Virtual Appliances running on common High Volume Hardware. ETSI [ETSI13] defines next **benefits** for a Network Operator taking into account migration to Network Functions Virtualization:

- **Reduced equipment cost.** Basically, by exploiting the economies of scale of the IT industry.
- **Increased velocity of Time to Market.** By minimizing the network operator cycle of innovation.
- **Reduction of development costs and time to market.** Motivated by a more efficient test and integration, and the possibility of running production, test and reference facilities on the same infrastructure.

- **More innovation due to openness.** Motivated by Open Virtual Appliance market where small players and academia can bring new services.
- **Optimizing network configuration and/or topology.** This optimization can be performed in near real time based on the actual traffic/mobility patterns and service demand.
- **Reduced energy consumption.** By exploiting power management features in standard servers and storage hardware.
- **Improved Operational Efficiency.** Taking advantage of the uniformity of the physical network platform:
 - IT orchestration mechanisms provide automated installation.
 - Eliminating the need for application-specific hardware. The skills base across the industry for operating standard high volume IT servers is much larger and less fragmented.
 - Reduction in variety of equipment for planning and provisioning.
 - Option to temporarily repair failures by automated re-configuration.
 - The potential to gain more efficiency between IT and Network Operations.
 - The potential to support in-service software upgrade.

On this chapter, a first introduction to the **OpenDaylight Open Source Project** has been presented. Apart from that, main technologies involved in this project, **SDN** and **NVF** have been introduced as well. On next sections, it will be studied in depth other aspects of the project, focusing on those areas having to do with Open Source nature of the project.

Chapter 4

OpenDaylight: Economic Aspects

Economic issues around Open Source software are always there. It is important to separate Open Source software itself from its usual “costless” nature. Free Software itself has not helped to this separation, as free refers to the “Freedom” aspect, not to the costless aspect. Richard Stallman is the one how has most times clarified this:

To understand the concept, you should think of "free" as in "free speech" not as in "free beer".

This chapter will analyze the economic aspects after OpenDaylight Open Source project. First of all, the SDN and NFV market will be analyzed. Although some data on the market expectations that this technology aroused [3.1](#) on the first days. However, a more detailed analysis of the current state of SDN/NFV market will be performed to identify the accomplishment of the expectations.

Apart from that, it is important to identify that SDN/NFV technologies, as incipient technologies, could mean new business models for selling companies, as well as new cost saving methods for network operators, as was justified in [3.3](#). Both aspects will be analyzed on this chapter as well.

Last, but not least, it is important to focus on the OpenDaylight project. Why has this Open Source project been founded? Was it done as a way of altruism? Or it was rather to respond to a

certain particular need on the Networking industry? Last section of this chapter analyzes those facts that could mean the foundation of OpenDaylight project, as well as reasons to go Open Source.

4.1 SDN and NFV: A new market

As remarked in section 3.2.2, by 2013 there were big expectations in networking industry around SDN/NVF technologies. Some optimistic reports predicted a market expectations for year 2018 about \$35.5 billion, while others were more cautious and estimated \$15.6 billion by that same year.

It is usually common that real market do not follow market expectations for those cases where a technology is on its incipient state. The **hype cycle**, a branded graphical theory developed by IT research firm Gartner, do already consider that **peak of inflated expectations**.

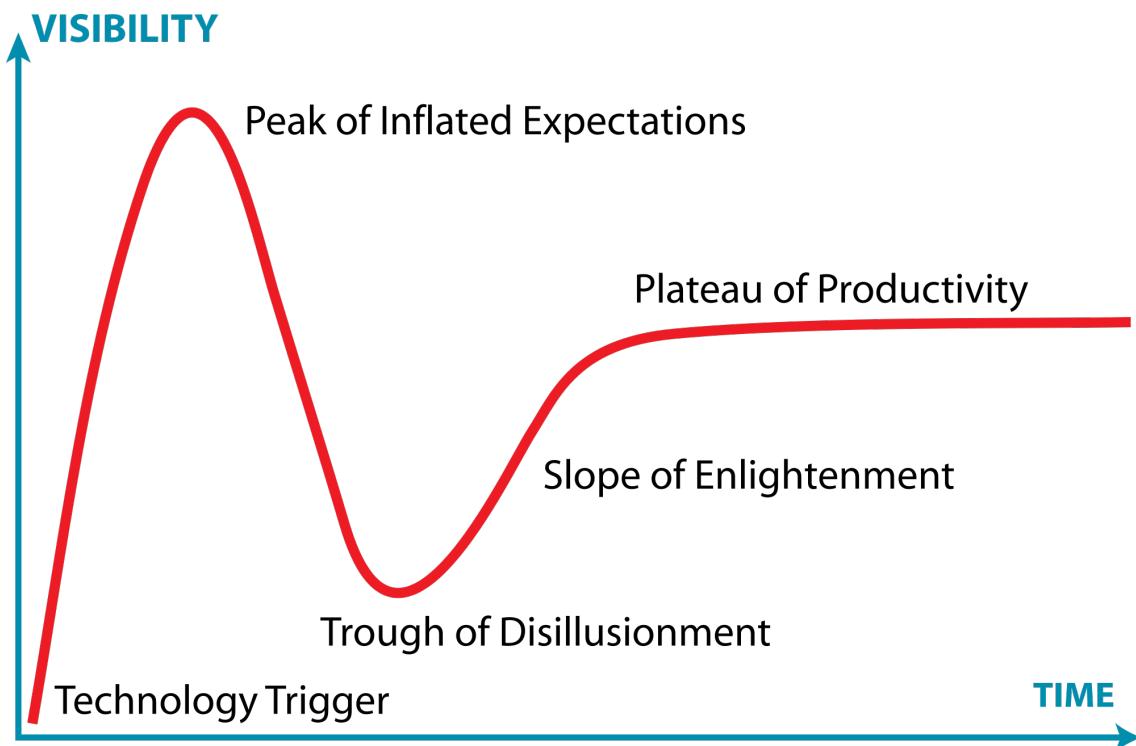


Figure 4.1: Gartner Hype Cycle

This cycle [Inc], which tries to represent the maturity, adoption and social application of specific technologies, fits well to describe what is going on in SDN/NVF technology life cycle. From

Hype Cycle's perspective, the life cycle of a technology has five different phases:

1. **Technology Trigger.** A potential technology breakthrough arises interest.
2. **Peak of Inflated Expectations.** Early publicity produces a number of success stories.
3. **Through of Disillusionment.** Interest wanes as experiments and implementations fail to deliver.
4. **Slope of Enlightenment.** More instances of how the technology can benefit the industry start to become true.
5. **Plateau of Productivity.** Mainstream adoption starts to take off.

SDN/NFV technologies seem to have crossed the Peak of Inflated Expectation and are starting down that part of the Gartner Hype Cycle [Eri14]. Second thoughts are starting to be present when talking about these technologies, where some disappointing facts have started to appear:

- **NVF seem to not going to cut for high performance applications.** NFV seems to have some hard problems related to the “state” IP networks own. NFV applications must deal with communications and therefore contain a fair amount of session state. This state must be accessed quickly to maintain performance, something which seems not easy in NFV architecture.
- **No impressive SDN/NFV IPOs companies have been performed.** After initial IPOs around Nicira or Contrail from VMWare and Juniper respectively, it seems no big action regarding companies acquisitions exist. This is another measure of how this market is losing expectations.
- **SDN seems for the near future to be confined to the data center.** Existing Internet Service Providers (ISPs) and Enterprise Networks, although evaluating SDN’s impacts, are not defining transition plans with confidence.
- **Market sales are far from expectations.** Some recent reports, such as the IDC [IDC14] one, are predicting sales of \$960M in 2014, which is far from the \$3.4B predicted in some of the most optimistic reports.

SDN/NFV technologies are, in turn, suffering the perception of being a complicated technology. In a survey performed to more than 100 IT professionals by Pica8 [Pic14], the perception of more than **50% of the professionals** was that **have no time or skills today to build the**

stack themselves. Meanwhile, 20% of the professionals thought that want an open platform, but they still need help selecting the pieces for their scenarios.

Together with all the previous statements, there is a particular problem with SDN today, which is relate to the concept of adopting a new technology. In the end, **the challenge is that existing hardware, which is in turn hardware deployed today in network operators, ISPs, etc, cannot support SDN technology** [Net14].

New standards have emerged, new Open Source Apps are incubating, and big industry heavy-weights are aligning – all to support SDN. **But the problem with SDN today is the hardware.** Industry groups have collaborated to deploy and drive Open SDN, **with OpenDaylight project leading the SDN adoption.** However, these groups do not solve today's SDN challenge – it is still the hardware deployed today. These groups are **solving today's challenge with future solutions.** There is one exception, which is the OpenDaylight project. This project provides some southbound “old” API's that today's hardware does support. But, in the end, it is not enough.

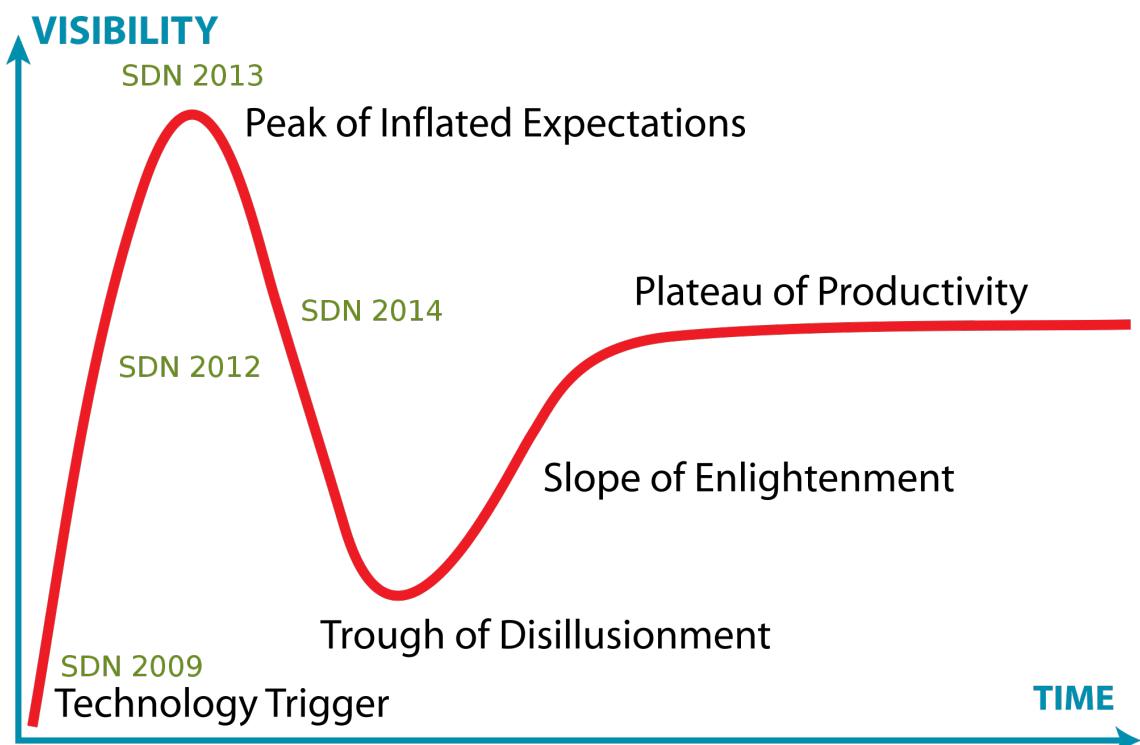


Figure 4.2: SDN Hype Cycle

The issues detailed previously are driving some **downward corrections on market expectations** regarding these technologies. Some reports, such as the already mentioned IDC [IDC14], estimate that, by 2018, SDNmarket will gain ground within the broader enterprise and cloud service provider markets for datacenter networking to over **\$8B**. Other reports contemplate even worst expectations, decreasing them to **\$3.6B** by 2019 [RNe14].

4.2 SDN/NFV with OpenDaylight: Software Cost Reduction

It is important to analyze the importance of OpenDaylight from the perspective of being Open Source. Being OpenDaylight an Open Source software solution, it has been proved that, for system integrators, using Open Source is cheaper. This implies **Software Cost Reduction** in the complete solution.

In this section, software cost reduction will be analyzed in order to clarify how important is for companies having access to Open Source projects, and incorporating this software to the complete system integrated solution.

Customers will demand SDN technology providers to deliver "SDN complete solution". OpenDaylight project, being Open and costless as well, will be a key piece for technology providers to offer a SDN solution, freeing the customer to focus on business rather than IT. Apart from software, solution comprises hardware and services.

Open Source Software is the key to construct a SDN solution which allows SDN system integrators a more flexible and profitable business case [Rie07]. Being costless, software expenses will exist (software integration, bug fixing, adaptations, etc.), but will be less compared to proprietary solutions. This will allow SDN system integrators to increase their business from two perspectives.

On the one hand, SDN system providers **will benefit from Open Source Software by increasing services revenue per customer**. If SDN provider owns only the services component, it will have to pay other companies for the software and the hardware. In a SDN system integrator's

interest, it is desirable to acquire hardware and software as cheaply as possible. Hardware costs will be somehow not flexible, being the ones cheaper in the market. However, from software perspective, if Open Source software is an option, as in OpenDaylight's example, it is typically much cheaper than closed source software, hence its use increases profits for the system integrator.

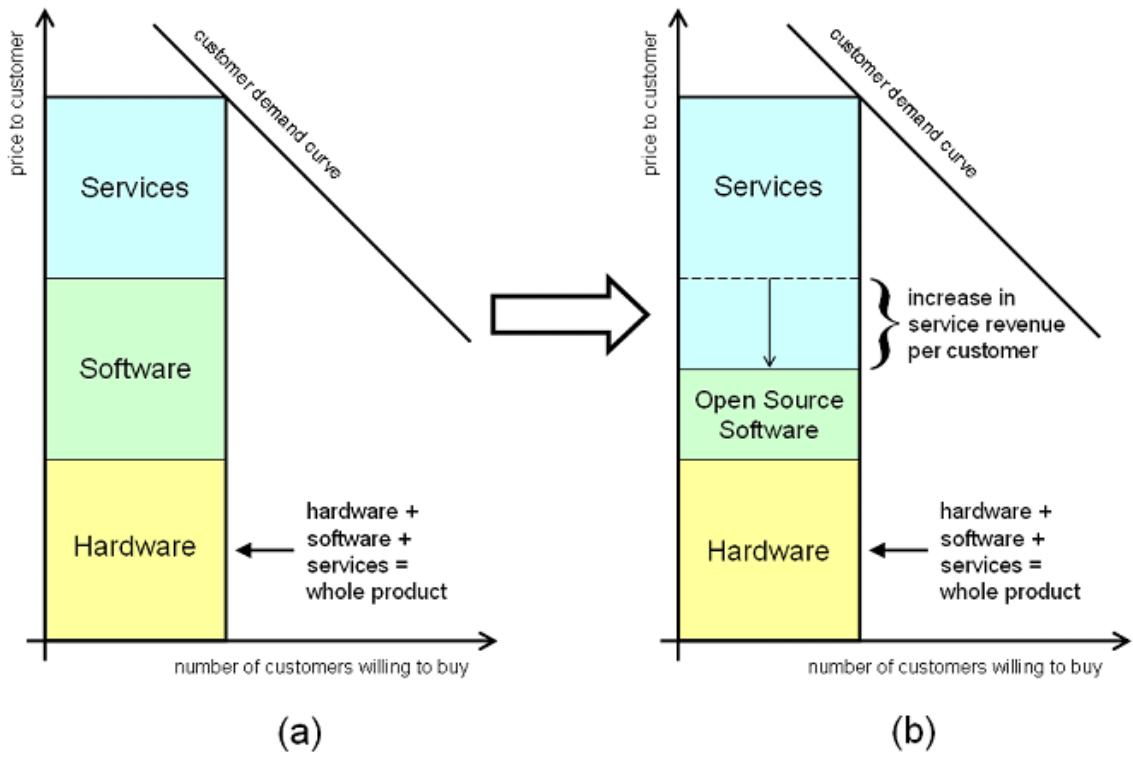


Figure 4.3: Open Source Software: Service Revenue Increase

On the other hand, a SDN system integrator can vary solution pricing to a prospective customer's problem. The reason to this, switching from closed source software, typically more expensive, to less expensive Open Source software, means also reduction of the lower price limit. This option usually means new possible contracts, with a new set of more price-sensitive customers within reach. **Not only does Open Source software improve profits on the original individual sales, it also increases the total number of potential customers.**

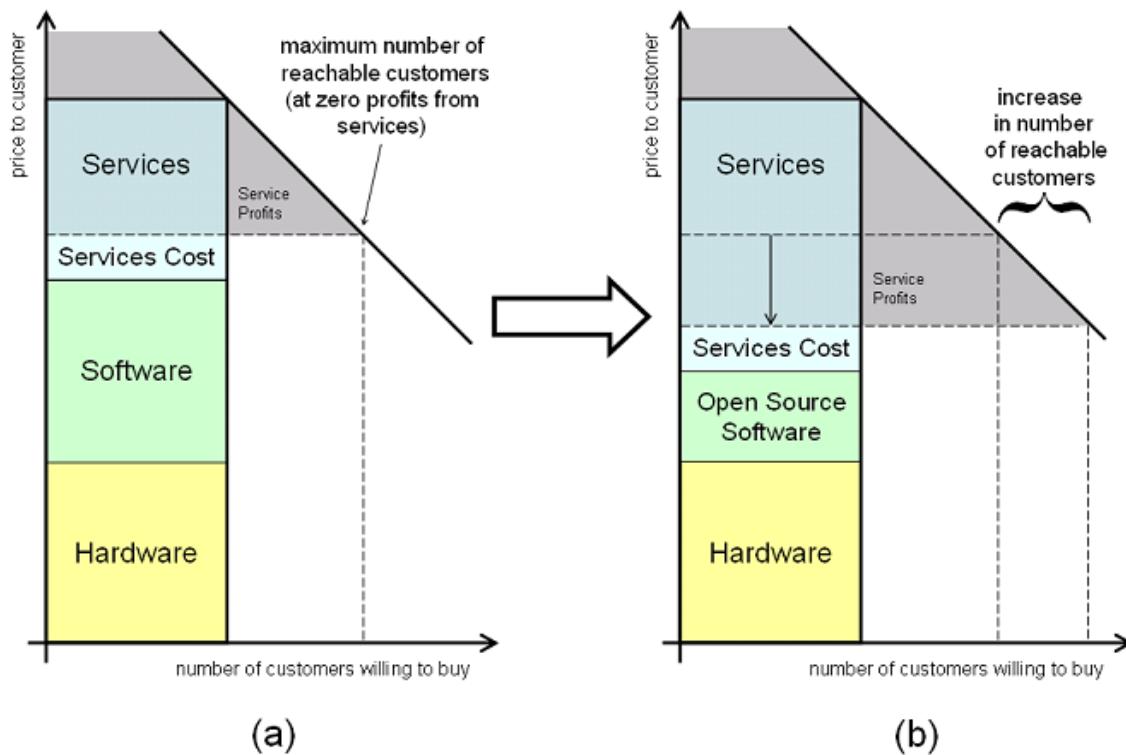


Figure 4.4: Open Source Software: More potential customers

To summarize, it is a fact that has been proven how Open Source software has changed the rules of the game. All major vendors involved in OpenDaylight project are complete solution providers in turn, and consider that, investing in production of Open Source products will benefit them in the long term, as this will mean software cost reduction that will enable other business scenarios.

4.3 SDN/NFV technology: Cost Reduction

Another important economic aspect about SDN and NFV technologies, and, to a lesser extent, to OpenDaylight project, is related to **cost reductions associated to technology**. This aspect is, indeed, the most important aspect about SDN and NFV, being the strengthen of the technology, as explained in 3.3.3.

It is important to understand, on the one hand, how important is for mobile operators to retain and gain users, by providing new services and better network access throughputs. In order to do so, LTE adoption is contemplated as the new revolution in mobile networks.

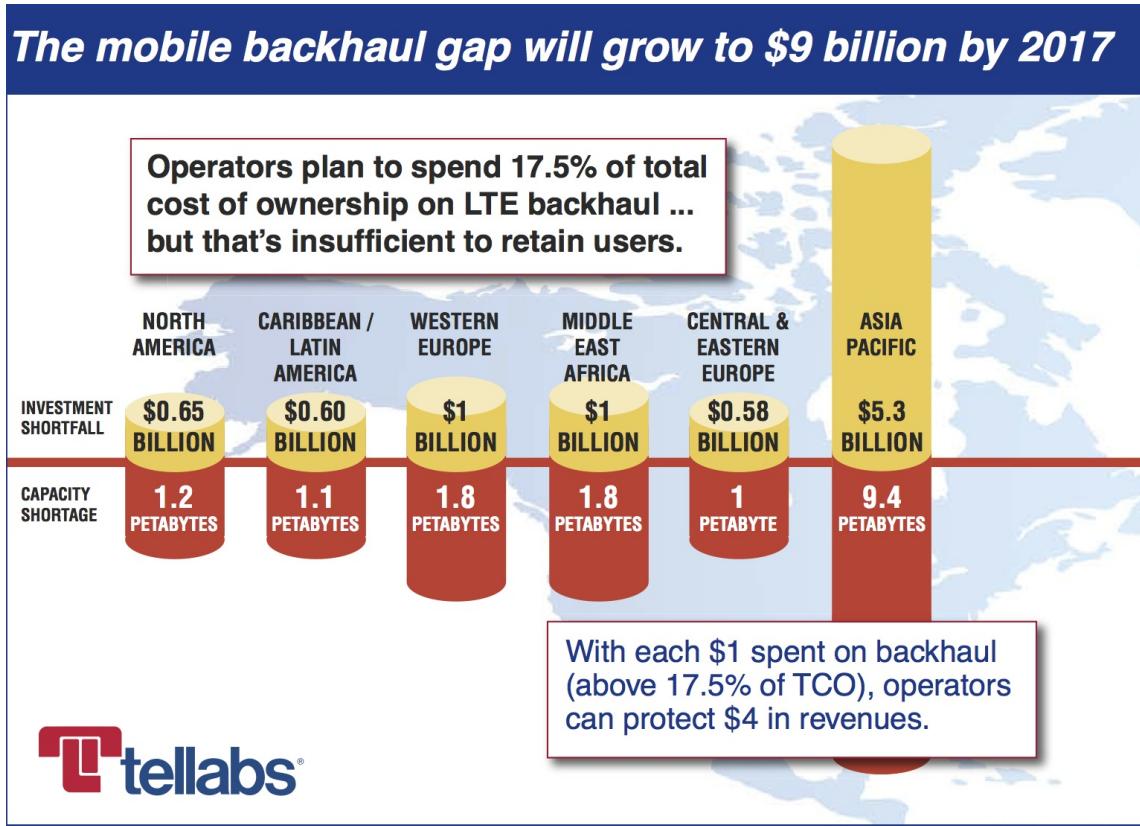


Figure 4.5: Mobile Backhaul Network Invest Expected

There are estimations [Tel14] that contemplate that up to **17.5% of total TCO** (Total Cost of Ownership) will be spent in LTE backhaul. Estimations also consider that spending \$1 in the backhaul can protect up to \$4 in revenues.

Figure 4.5 shows how backhaul inversion is considered to grow up to \$9B by 2017. Meanwhile, SDN appears as a key cost reduction technology. There are SDN based shortfall reductions estimations that vary from 15% to 91%, depending on the zone considered. This means that up to \$4B could be saved by SDN technology adoption world-wide.

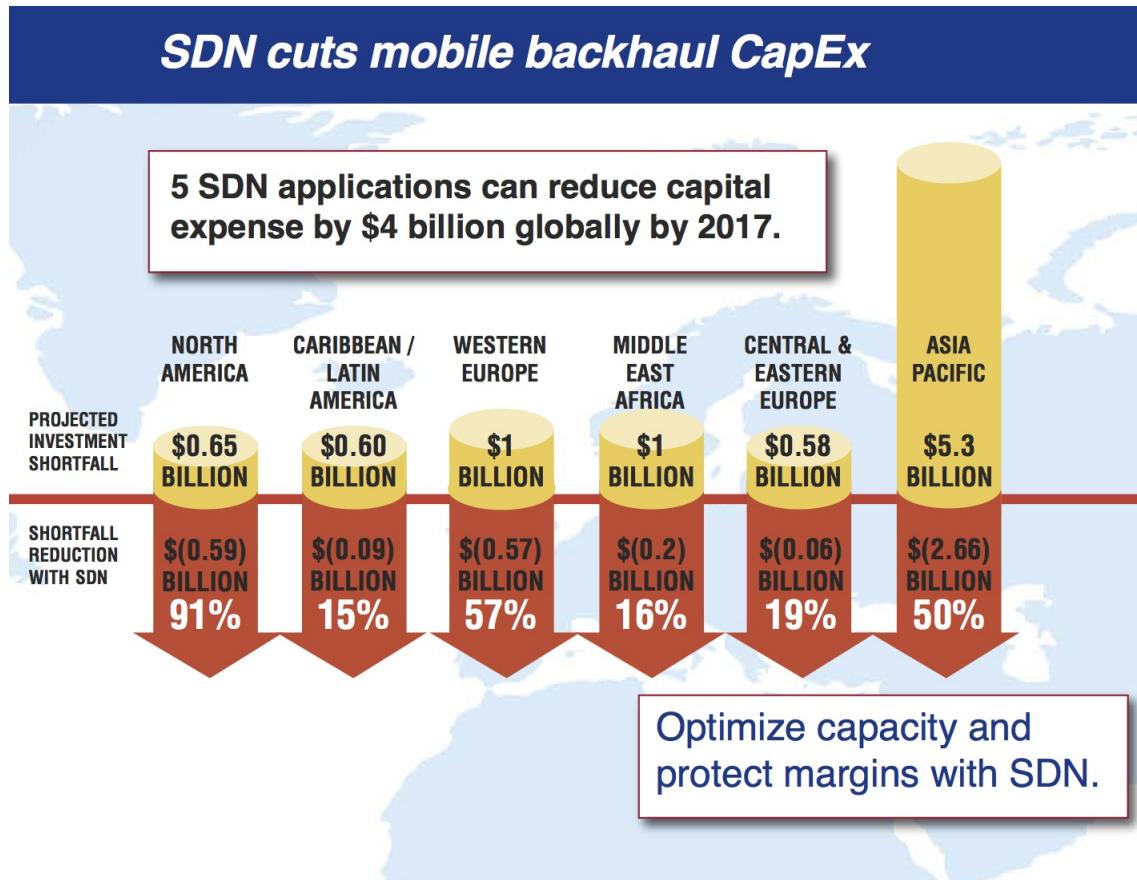


Figure 4.6: 2017 SDN CAPEX Reduction Cost

However, the SDN technology is also considered a boosting savings technology from Opex perspective. In that same report, there are estimations about total savings of \$9B in a study of 5 different applications where SDN technology fits. Stu Benington, Tellabs director of technology and strategy, asserted in that same report:

Earlier, we highlighted the key role of SDN in closing the mobile backhaul gap. Now we can show that on top of the Capex savings, SDN can cut Opex dramatically and preserve operator margins in backhaul networks.

Next figure shows, for all of the applications, the OPEX reduction in percentage.

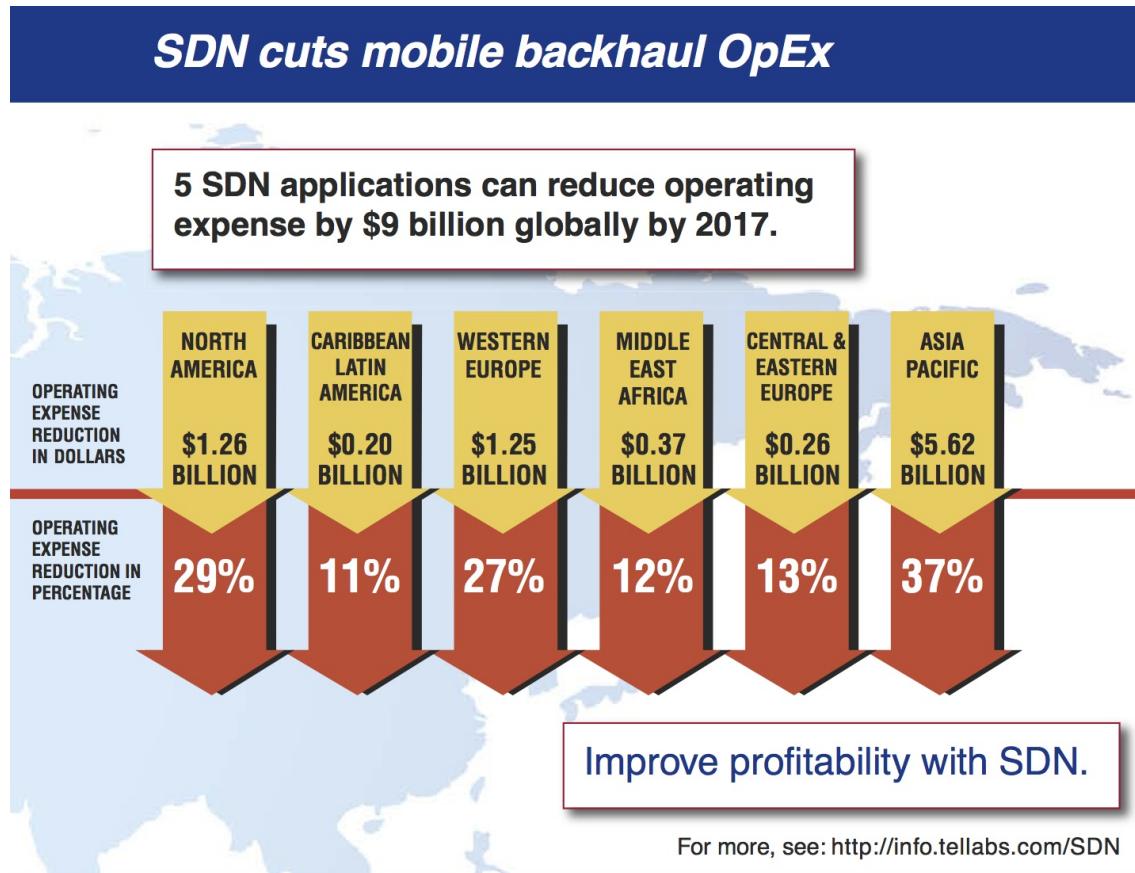


Figure 4.7: 2017 SDN OPEX Reduction Cost

5 different applications studied where Wifi video/redirect, cloud Radio Access Network (RAN), local breakout/Internet Internet Exchange Points (IXP), metro aggregation/load redistribution and small cells. On next table the different savings considered per application by 2017 are shown:

SDN Application	OpeX Savings
Wifi offloading/video redirect	\$3.12 billion
Cloud RAN	\$2.17 billion
Local breakout/Internet IXP	\$1.83 billion
Metro Aggregation/Load Redistribution	\$1.22 billion
Small Cells	\$591 million

Table 4.1: SDN 2017 OPEX Savings Estimations

SDN technology is not the unique one regarding OPEX/CAPEX savings, and NFV follows in this line at par. Another report by Juniper Networks [fJN14]. This report, among other interest-

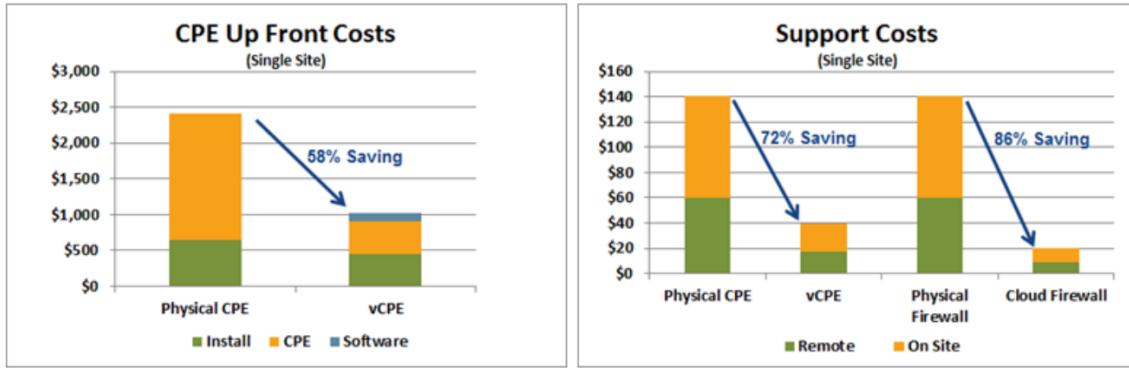


Figure 4.8: NFV Reduction Cost

ing conclusion, estimates that, by migrating to a virtualized CPE, up-front costs can be reduced by 58%, while support costs can be reduced to 72%, as shown in Figure 4.8.

Obviously, estimations are usually not correct, and information provided must be considered with the correspondent mistrust that any estimation report propitiates. The most relevant aspect of the previous data, far from estimations data or other considerations, is the high expectations that SDN/NFV technology imply from customers' perspective. This is always exciting from SDN providers and integrators, and of course, for Open Source projects around this technology, such as OpenDaylight.

4.4 OpenDaylight: Economic Aspects to go Open Source

Apart from the previous considerations regarding economic aspects behind SDN/NFV technologies, it is **important to analyze the economic reasons of the OpenDaylight project**.

4.4.1 Technology Adoption

The first factor to remark about SDN/NFV in general, and OpenDaylight project in particular, is **how impressive is to see major heavyweight users of networking like Ericsson, Cisco, or Juniper, together with heavyweights of IT and Operating System industries, such as HP, IBM, Microsoft, Red Hat or Citrix**.

What are the reasons of different competitor companies to cooperate among them on an Open Source project? It is not the first time when this situation takes place. Other similar cases, such as

Webkit Open Source Project, made competitors such as Google and Apple cooperate around development of a particular technology, in this case a Web Browser Engine.

In the OpenDaylight Open Source project, the key words of big competitors cooperating are **“help accelerate the development of technology”**. In the project website [Abo14], this seems to be the main reason:

With OpenDaylight, a community has come together to fill this need through the combination of open community developers and Open Source code and project governance that guarantees an open, community decision making process on business and technical issues. Establishing an Open Source project in this way is designed to help accelerate the development of technology available to users and enable widespread adoption of SDN and create a solid foundation for NFV.

Neela Jacques, executive director of OpenDaylight, has pointed out in more than one occasion [Ber14]:

The industry needs to create a platform that encourages both innovation for the common good while not disturbing the ability of members to create and sustain their own differentiated value for their customers.

It is also a remarkable fact how important is, in the history of communications, the adoption of standards and terms and conditions for interoperability. OpenDaylight is an effort of the industry to produce Open Source software and standards on SDN/NFV networking technologies, so that this technologies adoption is faster. From Neela Jacques perspective, **Open Source** has some disadvantages as well, but it is, no doubt, **the best way to reach interoperability**:

Open Source maybe imperfect as a way to move forward but it beats the tyranny of looking for an authoritarian dictator in the market and possible interoperability issues.

Islands of communications were never good for anyone long-term and such will be the case with SDN and NFV. Meanwhile, over the years, **Open Source** efforts have shown that communal efforts based on freely contributed intellectual property that is tested and honed **does not foreclose the ability to make money.**

It historically can be argued **what it does is to expand the opportunities for success, as Open Source has long ago shed its pejorative connotations, and rightfully can be seen as being open for business.** The project should be seen as the chance for the industry to break through and score [Ber14], as **Open Source, due to its transparent nature, usually acts as a demonstration laboratory to show the technology, “AS IS”.**

4.4.2 The Operator's View

In the end, there is a mandatory perspective to take into account, which is **customer opinion.** SDN/NFV technologies will be in the end new technologies to adopt, and adopters' view of drivers, barriers, time lines and targets for both technologies, as well as the role that open systems and Open Source play in its advancement and adoption need to be studied.

In this sense, **OpenDaylight is well aware of the industry requirements.** To investigate issues that arouse, **SDN, NFV and Open Source Report** [Gig14b] was driven by both **OpenDaylight Open Source project together with Gigaom Research** [Gig14a]. This report surveyed 600 IT decision makers and technologists in medium to large organizations within enterprise (300) and service provider (300) organizations in North America [Gig14b].

The survey report covered several main sections:

1. **Demographic Questions.** Data regarding number of employees, location servers, SDN adoption, etc.
2. **Current State of The Network.** Questions about cost of networking equipment and operations, end user satisfaction, business management satisfaction or open systems consideration.
3. **SDN Benefits and Challenges.** This section asked for ratings on SDN benefits and challenges, from lower capital expenses (CAPEX) and operating expense (OPEX), to the biggest impacts or the current SDN vendor offerings satisfaction.
4. **SDN/NFV and Open Source Technology.** Obviously, **the main section to analyze from Open Source perspective.** On this part of the survey, the view of the surveyed professional related with the Open Source Technology was investigated through next questions:
 - **What is your view of the role of industry standards and open systems in the development and deployment of SDN/NFV solutions?.** Available answers were from “Critically important” to “Non important at all”.
 - **What is your view of the role of Open Source technology in the development and deployment of SDN/NFV solutions?.** Available answers were from “Critically important” to “Non important at all”.
 - **What are the biggest advantages of applying Open Source technology to SDN/NFV solutions and deployments? (Select three.).** Different advantages were proposed:
 - (a) Avoiding vendor lock-in.
 - (b) Lower acquisition and maintenance costs.
 - (c) Gain access to large library of networking software systems.
 - (d) Boost networking software capabilities.
 - (e) Accelerate networking software enhancements.
 - (f) Assure software systems interoperability.
 - (g) Ease management of network – and networking software.
 - (h) Leverage top developer talent and software best practices.
 - (i) Other (Please specify.)
 - **What are the biggest drawbacks of applying Open Source technology to SDN/NFV solutions and deployments? (Select three.).** Different drawbacks were proposed:

- (a) Lack of commercial vendor support.
 - (b) Incompatible with existing network
 - (c) Performance shortcomings
 - (d) Feature shortcomings
 - (e) Reliability concerns
 - (f) Security concerns
 - (g) Deployment challenges – e.g., increased integration and test burden
 - (h) Operational complexity – e.g., non-standard management interfaces and tools
 - (i) Legal concerns – e.g., license management and compliance
 - (j) Other (Please specify.)
- **Rate 1 to 5, your intention to apply Open Source technology to the following SDN/NFV functions (1 being least likely to apply and 5 being most likely to apply).**
 - (a) SDN controller
 - (b) Network automation
 - (c) Network monitoring and analytics
 - (d) Northbound APIs
 - (e) Southbound APIs
 - (f) Policy management
 - (g) Security services – e.g., firewall, intrusion prevention...
 - (h) Network optimization – e.g., application delivery controller, WAN optimization...
 - (i) Orchestrator – e.g., OpenStack and associated plug-ins
 - (j) Hypervisor – e.g., Kernel-based Virtual Machine (KVM)
 - (k) Other (Please specify.)

Analyzing the results of the report is critical to understand **how important is Open Source and Open Systems perceived from IT professionals**. Indeed, the results were incredibly in favour of Open Source and Open System when talking about SDN/NFV adoption.

Survey results (see Figure 4.10) further raise the importance of Open Source in considering

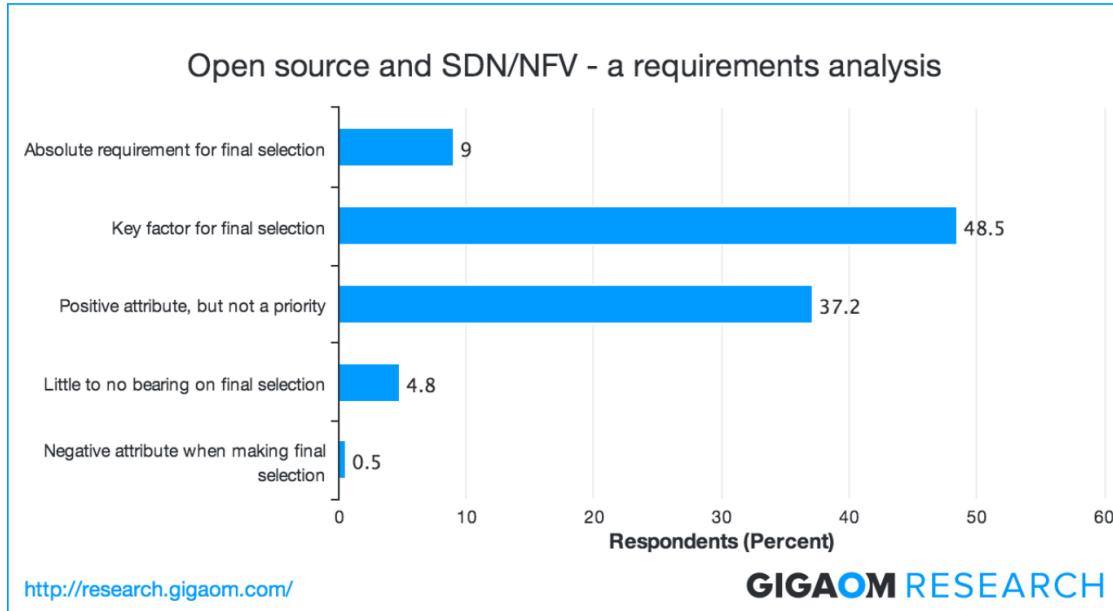


Figure 4.9: Open Source on SDN/NFV - A requirement analysis

SDN and NFV solutions. **94.7% look at Open Source as a positive attribute of any SDN or NFV solutions.** In particular, 9% consider Open Source an absolute requirement for final selection, while 48.5% consider it a key factor for final selection. 37.2% believe it is a positive attribute, although it is not a priority for the selection. Meanwhile, only 0.5% consider Open Source to be a negative attribute when making final selection.

The other important conclusion reached from the report is the **perception of the importance of Open Source for SDN and NFV success:**

- 76 percent of respondents consider open systems as critically or very important to SDN and NFV.
 - 18.3 percent consider it critical.
 - 57.8 percent consider it very important.
- An almost matching 68.5 percent of respondents view Open Source as critically or very important to SDN and NFV.
 - 13.5 percent consider it critical.
 - 55 percent consider it very important.

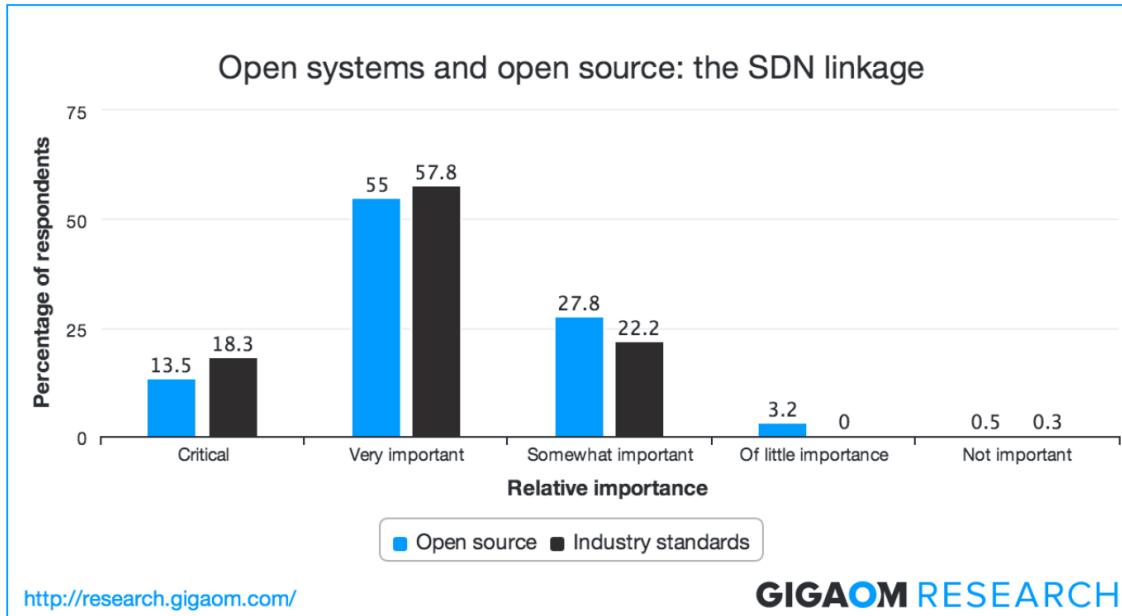


Figure 4.10: Open Systems and Open Source: The SDN linkage

To summarize, the conclusion drawn is clear (see Figure 4.10): “**Open Source serves as the principal delivery mechanism for SDN and NFV solutions based on industry standards and open systems**”.

To summarize, Open Source has indeed become mainstream, representing what IT users have come to expect from new technologies solutions over Open Source and Open System on last years: **maturity, robustness, and reliability**.

It must be highlighted, as final conclusion, that, **from an economic perspective, going Open Source is not only desirable, but rather be mandatory to avoid future customer pejorative perception for being a closed solution**.

Chapter 5

OpenDaylight: Legal Aspects

5.1 OpenDaylight License: EPLv1.0

OpenDaylight project has been released under EPL (Eclipse Public License)-v1.0 [Fou04]. This license, detailed in Appendix A, that was launched in February 2004, is considered an **Open Source Software License** due to basically two reasons:

1. It guarantees the “Four Freedoms” that Open Source must guarantee:
 - Freedom to use.
 - Freedom to inspect.
 - Freedom to copy and redistribute copies.
 - Freedom to modify and redistribute modified versions.
2. It is considered also **Open Source/Free Software** due to the fact that the two main groups having to do with this kind of software recognizes the license to be Open Source/Free Software:
 - **Open Source Initiative** [Ini98a]. This corporation recognizes EPLv1.0 not only among the ones considered to be an **Open Source license**, but also **one of the most popular ones** [Ini98b].
 - **Free Software Foundation** [Fou14b]. This nonprofit organization recognizes this license to be a **Free Software license**, although **EPLv1.0 is not compatible with GNU GPL** [Fou14c].

Taking into consideration that EPLv1.0 is Open Source, a deeper analysis of this license must be performed. Before analyzing which kind of Open Source software license is EPLv1.0, a classification of the different kind of Open Source licenses should be performed. However, before doing such analysis, a concept related to Open Source license, which is **Copyleft**, must be introduced. According to FSF [Fou14d]:

Copyleft is a general method for making a program (or other work) free, and requiring all modified and extended versions of the program to be free as well.

However, this definition is not as concrete as it should. Although the term “Copyleft” was defined by FSF, OSI defines the concept in a much more understandable way [Ini14]:

“Copyleft” refers to licenses that allow derivative works but require them to use the same license as the original work. For example, if you write some software and release it under the GNU General Public License (a widely-used copyleft license), and then someone else modifies that software and distributes their modified version, the modified version must be licensed under the GNU GPL too.

Taking into consideration the previous concept, according to the kind of licenses and if they are copyleft licenses or not, it is usual to follow this **classification regarding Open Source Software Licenses**:

- **Minimalistic/Academic Licenses:** These kind of **No Copyleft** licenses are the most simple ones. **None of them has Patent Software clauses**, and they **rarely mean incompatibilities** with other kind of licenses. This kind of license is normally very well considered (for both parts, Open Source and private industry). They usually contain a **No warranty clause**, and normally have **no restrictions**. They usually suppose **no legal complexity**. The most popular of this kind of licenses are:
 - BSD License.
 - ISC License.
 - MIT License.

- **Permissive Licenses:** This kind of **No Copyleft** licenses are similar to the academic ones, but they normally include some kind of restriction. This kind of restriction is normally related to **Patent Software clauses**. They **rarely mean incompatibilities** with other kind of licenses, **although there are exceptions**. Among this kind of licenses, **the most popular one is Apache License**.
- **Weak Copyleft Licenses:** This kind of **Weak Copyleft** licenses contain copyleft clauses with certain limitations. They normally define that **not all derivative works must inherit the copyleft license**. Derivative works inherits or not the license often depending on the manner in which it was derived. They are normally considered **Copyleft oriented to just source files**. The most popular of this kind of licenses are [Fou14c]:
 - Mozilla Public License.
 - Common Public License.
 - **Eclipse Public License**.
- **Copyleft Licenses:** This kind of **Copyleft** licenses contain **strong clauses related to redistribution of derivative works**. This kind of licensing was first introduce by FSF, and is not well considered because of its “**viral character**”. The most popular license of this type is **GNU Generic Public License (GPL)**.

Once described the classification of Open Source licenses, it has been asserted that EPLv1.0 is classified as a **Weak Copyleft** license, as recognized by FSF [Fou14a]. But, apart from that, inspecting the EPLv1.0 license, it can be found that **there is a possibility to distribute software under on its own license agreement** [Fou04], when distributing “object code”:

3. REQUIREMENTS

A Contributor may choose to distribute the Program in object code form under its own license agreement,

....

When the Program is made available in source code form:

- a) it must be made available under this Agreement; and
- b) a copy of this Agreement must be included with each copy of the Program.

As demonstrated above, there are some conditions in the license that allow this software to be distributed under other kind of license. However, if distributing source code, distribution must be performed under same agreement. **This kind of clauses are the ones that make this kind of licenses to be known “Copyleft oriented to source files”.**

A more detailed description of the EPLv1.0 is detailed below [Wil09].

5.1.1 History Of The Eclipse Public License

Born in 1999, The Eclipse Public License began life within the IBM Corporation as the **IBM Public License (IPL)**. IBM was willing to release Open Source code, but felt that needed to draft their own new license to meet their specific needs. IPL named IBM Corporation as the licensor of code that it covered, unfortunately. That issue meant that the license could not easily be reused by others to cover code their own code.

As a result, when IBM came to create a revised version of their license in 2001 they generalized the terms to remove direct reference to themselves and renamed it the **Common Public License (CPL)**. In 2001, IBM formed a consortium of interested technology companies around the platform including themselves, Borland, SuSE and Red Hat, and, at that same time, **IBM released their software development platform Eclipse under the CPL**.

By 2003 it was decided that the Eclipse platform needed its own legal entity to manage code contributions coming from so many disparate sources. At the same time, the consortium had expanded to include over 50 members. Besides this, the CPL was revised in two ways to ease the establishment of the Foundation.

On the one hand, the ‘steward’ organization for the license was changed from being IBM to being the Foundation itself. **This helped cohesion of the Foundation’s members** by ensuring that authority to issue revised versions of the license in the future.

On the other hand, it was decided that one term of the so-called **“patent-retaliation” clause within the CPL should be revised**. The CPL states that a licensee will lose their license to use

adapt and distribute the code if they start litigation alleging infringement of a patent of theirs by the CPL-covered code. This clause was retained. However the CPL also states that if a licensee starts any software-patent-related litigation against any contributor to the code, then they also lose their license. This would be true even if the litigation related to some entirely different piece of software..

The CPL has now been retired by IBM, and the Open Source Initiative recommend the use of the EPL in its place.

5.1.2 Main Features of EPLv1.0

The **EPL grants the “Four Freedom” rights**, as shown before. However, the license enumerates the rights granted as follows:

- Licensee has the right to copy, adapt and distribute the program in source or object code form.
- Licensee has the right to distribute the code in object code form alone under a different license, provided that license is compatible with the EPL.
- Licensee has the patent rights from all contributors to use and make available the code.
- Licensee has the right to distribute works which contain the code in combination with new code modules, and to license the new code modules in any way the distributor wishes.

To summarize the most important parts of the EPL, it must be highlighted that EPL:

- **Explicitly grants patent rights** where necessary to operate the software.
- **Keeps the covered code itself Open Source**, due to its “Weak Copyleft” nature.
- **Allows expansion of the code via new modules that can be licensed in non-open ways.**

5.1.3 Other Software Development Considerations

Some considerations must be taken into account regarding the consideration that the Eclipse Foundation provides about the EPLv1.0 license:

1. The Eclipse Foundation makes clear that, in their opinion, “merely interfacing or interoperating” with an Eclipse plugin does not make your code a derivative work of the plugin.
2. Therefore original software developed using the Eclipse platform can bear any license its author chooses when distributed.

Once EPLv1.0 license has been analyzed and classified, it must be analyzed **which reasons make OpenDaylight Open Source project** to use this kind of license. In the FAQ section of OpenDaylight Open Source project, next statement appears [Lic14]:

5. What license will the project use? Why was that decision made?

OpenDaylight is structured and governed using Open Source best practices and is licensed under the Eclipse Public License-v1.0 (EPL), which is a common choice for Java-based projects. The EPL is an approved Open Source license by the Open Source Initiative and considered a free software license by the Free Software Foundation. The license choice of EPL maximizes OpenDaylight's license compatibility with the large ecosystem of libraries and 3rd party components that have already been released under the EPL license. Where necessary the Board may approve exceptions for other licenses.

Basically, the reasons for using this kind of license are basically three:

- **Open Source/Free Software license.** Being recognized by both OSI and FSF, this license is suitable into the structured and governance bylaws of the project having to do with its Open nature and commitment to the Open Source movement.
- **Java-based project.** EPLv1.0 is a common license for Java-based projects. Being **developed in Java programming language primarily**, OpenDaylight Open Source project fits quite well into this kind of license, as other projects written in Java do, as, for instance, Eclipse IDE.
- **3rd party libraries/components compatibility.** Last, but not least, Board in OpenDaylight decided that **EPLv1.0 license fits better in terms of compatibility with 3rd party**

components and libraries used inside the project compared to other kind of licenses. Examples of 3rd parties used in OpenDaylight Open Source project are Java programming language, Maven, Open vSwitch or Open Flow Java implementation.

- **Avoid licensing fragmentation.** Using this kind of “Weak Copyleft” license helps on avoiding licensing fragmentation for derivative works/forks/spinoffs. A more detailed information on how fragmentation is avoided appears in Section 5.2.

Up to this point, EPLv1.0 and OpenDaylight Open Source project to use this license have been analyzed. In next section, other legal considerations and Bylaws will be studied in order to clarify other important issues that are considered inside OpenDaylight Open Source project to be important as well.

5.2 Other Licensing Considerations

In the previous section EPLv1.0 license and its application to the OpenDaylight Open Source project were studied. However, there are **other licensing considerations** to take into account by the project and its Governance Board.

For this reason, having a look at the FAQ of OpenDaylight Open Source Project [FAQ14], other legal considerations must be taken into account:

8. Will all development be done in the open?

All OpenDaylight source code development will be done in the open. Some contributors or companies may choose to incubate and develop code internally and then contribute it openly later, which is fine as well, but the core project code will all be developed in the open for everyone to see and use under a standard OSI-approved EPL Open Source license.

The previous statement tries to perform as a **declaration of intent** by the project Governance. In particular, it tries to clarify the intention of perform development of the core project with an Open Source licensing scheme, **clarifying the intention to use EPL Open Source license** as a general rule to guarantee open development.

17. Will the products developed on the framework also be Open Source?

This is the sole discretion of each company. The intent of OpenDaylight is to employ collaborative, Open Source development to build and support an Open Source framework and codebase on top of which commercial products and services can be built. How each vendor adopts OpenDaylight and commercializes their own products will be an individual business decision for them, subject to the Eclipse Public License.

With the previous statement, the OpenDaylight Open Source Project Governance tries to clarify the discretion of each company to perform **whichever commercial actions they consider**, taking into account the Eclipse Public License and the different statements appearing on it. Far from being commercially agnostic, this statement is a nod to empower companies to use this Open Source project as a base to commercialize products and services.

19. How will OpenDaylight prevent fragmentation?

Many Open Source projects eventually see forks or spinoffs from the original codebase. Experimentation and innovation off of an Open Source codebase is common and encouraged. However, long term the open governance and open technical decision making structure employed by the OpenDaylight project should help minimize the need to do any significant fragmentation from the core project codebase.

Further, OpenDaylight employs an OSI approved Open Source license. The EPL requires any distributed modifications to the program also be licensed under the EPL. For more information on the EPL, see the Eclipse EPL FAQ.

The previous statement is a clarification about the **decision to use EPL** to minimize forks of the original codebase. This license usage does not guarantee forks/spinoffs of the project are not taking place, but, at least, guarantees them to be Open Source if the source code is modified and wants to be distributed. In next section, **Bylaws of OpenDaylight Open Source project**

will be summarized. Far from being analyzed in a deep basis, the most important statements in the project bylaws are to be enumerated in order to **clarify those legal statements** that must be taken into consideration.

5.3 Bylaws

Adopted by July 28, 2014, OpenDaylight Project Inc. contains a set of thirteen articles to clear up all the different issues from a Legal and/or Governance perspective. The document, known as “Second Amended And Restated By-Laws” [Pro14c], is summarized in Appendix B, and define a complete legal statement that considers several aspects having to do with:

1. **Name, Purpose and Offices.** This Article focuses on some legal aspects such as the name of the company, the offices or the purpose and non-profit status of the project.
2. **Members.** This Article summarizes the different kind of members and the conditions of membership of the OpenDaylight project.
3. **Actions of Members.** Different kind of actions that members can carry out are collected on this Article.
4. **Directors.** This Article focuses on the different responsibilities and rights that directors own.
5. **Executive Committee and Other Committees.** On this Article a clarification of how an Executive Committee can be created or modified, as well as how other Committees can be composed in order to perform general or special duties or how meetings of the Executive Committee will take place.
6. **Officers.** A complete description of the Officers, their role in OpenDaylight Open Source project, vacancies, their election, tenure, executive role, vice-president roles, secretary role, treasurer role and compensation fees is provided in this Article.
7. **Notices.** Information regarding notifications, communication and electronic transmission to any Director or Member of OpenDaylight Open Source project.
8. **Indemnification.** Actions regarding the Rights of the ODP, Success on the merits, Authorization, Advance Payments, Non-Exclusivity clauses, Jurisdiction, Insurance or Severability are some of the aspects covered in this Article.

9. **Books And Records.** Stuff related to Books and Records of Account are handled on this Article.
10. **Certain Transactions.** A complete analysis of the conditions of the transactions with interested parties and OpenDaylight members (Directors, Officers, etc.) are cleared up on this article.
11. **Grants, Contracts and Loans.** Information about grants and contribution authorization, contract execution, and finance products such as checks, drafts or deposits are handled in this Article.
12. **General Provisions.** Fiscal Year, Reserves, corporate Seal adoption or Proprietary Rights are discussed on this Article.
13. **Antitrust Compliance.** A complete and detailed Article about antitrust and competition laws, Availability of Intellectual Property and Obligations to Endorse.
14. **Amendments.** Conditions of alterations and amendments to bylaws.

Chapter 6

OpenDaylight: Governance and Community Management

This chapter summarizes the state of the OpenDaylight project from management perspective. Different important aspects for an Open Source project, related to the community that makes it up and its governance will be analyzed, such as:

- **Project Governance.** Aspects such as decision taking, how board of directors is established and different aspects having to do with the way the project is governed are handled on this subchapter.
- **Community Management.** An enumeration of the different tools available for the community to contribute to the project will be analyzed. Besides this, communication channels and strategy, marketing campaigns of the project, community project rules, organization politics, entry barriers and other similar points in OpenDaylight project are studied in this chapter.

6.1 OpenDaylight Project Governance

OpenDaylight project intention from a governance perspective is to ease an open ecosystem, not only from the software perspective, as it is provided with an Open Source license, but also from other areas that are as important as software development. OpenDaylight is structured to be **Open**:

- **Open Design.**
- **Open Development.**
- **Open Contribution.**
- **Open Governance Model.**

Being a Linux Foundation Collaborative Project, OpenDaylight implements many Open Source best practices that are familiar to other leading projects of that same organization. Besides this, any individual, company or organization can engage directly and immediately to begin shaping the future of the project, as OpenDaylight is **open to anyone**.

Anyone can develop and collaborate, whether code, documentation or any other kind of contribution. Anybody can also be elected to be part of the governance of the project, turning into a project leader and providing leadership regarding the technical direction of OpenDaylight. There are basically two groups that compose the governance of the OpenDaylight project:

6.1.1 Technical Steering Committee (TSC)

Among the responsibilities of the TSC, tasks as technical best practices (including the Development Process), simultaneous release dates, quality standards maintainence, technical progress monitoring, technical conflicts mediation and resolution between committers and project leads or inter-project collaboration organization. The TSC oversees the design and development activities carried out in the OpenDaylight project, as outlined in the **TSC Charter** [[TSC14a](#)]. This document, appearing in Appendix C, contains a collection of sections defining the different considerations that must rule the OpenDaylight project from a technical view:

- **Section 1. Guiding Principle.** Defines the project operation principles. It defines that project proposals, time lines and status **must not merely be open, but also easily visible to outsiders**. To do so, the project **will operate transparently, openly, collaboratively, and ethically**.
- **Section 2. Evolution of ODP Governance.** This section defines the existence of two areas of governance. **TSC performs Technical leadership**, while **Board of Directors** is responsible of **business leadership**.

- **Section 3. Board's role.** Clarification on the Board's role is described in this section. Typically, **the board have no say in technical issues**, although it sets overall project policy and scope.
- **Section 4. TSC Establishment.** Mechanisms to elect TSC members are summarized on this section. Initially, **TSC shall be formed by representatives designated by Platinum members**. However, active Committers will elect in the future a number of TSC members, still to be determined. Platinum members may designate a TSC, although this is expected to be a temporary measure.
- **Section 5. Responsibilities of the TSC.** Definition of the different task where TSC is responsible appear on this section of the Charter. **Technical best practices, release dates and quality or technical conflict mediation are the main responsibilities** of this committee.
- **Section 6. Operations.** Description of the **Development Process** is described on this section. Life-cycle process, Core projects and changes approval criteria (cleanliness of code base, stability, predictability, etc.), are defined.
- **Section 7. Project Roles.** Definition of the roles to consider on the project are defined under this section. Basically, three kind of roles are defined:
 1. Committers. People who has right to commit to one or several projects. Rights and obligations are defined.
 2. Project Leads. Project Lead, which is **unique per project**, are Committers selected by vote, unless only one committer exist.
 3. Contributors. People contributing to the project but without commit permission.

This committee is also responsible of defining the guidelines of the **TSC Policy [TSC14b]**. This document, which appears in Appendix D, has been **set by the Board of Directors**, defines policies to govern the operation of the TSC, **related to release process, technical scope, business goals, etc**). TSC must adhere to these policies to the best of its abilities.

In the end, these policies are a set of **guidelines that apply to the projects existing under OpenDaylight, the artifacts created, its operation and scope**. TSC Policies emphasize as-

pects that must cover the previous statements, such as:

- **Singularity.** To avoid overlapping goals between projects.
- **Cohesiveness.** To form a cohesive system where all projects connect between them.
- **Simultaneous Release.** To ensure release at regular intervals in a “release often, release early” basis.
- **Communication.** To emphasize on open, fair and consistent communication within the project.
- **Openness.** To ensure Technical decisions are made in an open fashion.

6.1.2 Board of Directors

As described before in TSC Charter, **Board of Directors is responsible of business leadership**. This group will not participate on technical issues or individual project scope, as long as each project remain within the direction of the policies defined by the Board of Directors. Board of directors will be rather responsible of:

- **Set overall project policy.** In consultation with TSC, policies will be decided.
- **Describe aggregate scope of projects.** Always, without interfere on each particular project scope.
- **Drive a technical vision and direction.** By setting the tendencies and technologies driving the market and SDN/NFV state of art.
- **Define a release guidance to the TSC.** In order to provide guidelines and approximate dates of releases.

The Board of Directors is composed of a group of highly qualified members, with extensive knowledge on Cloud, Virtualization, Routing, Internet Technologies and Networking in general, SDN and NFV in particular, of the most important member companies involved in the project.

- **Neela Jacques (OpenDaylight Executive Director).** He works with the community to advance SDN and NFV. Formerly at VMware, developed and took to market the company’s software-defined data center vision and strategy including VMware’s vCloud Suite.

- Abhishek Chauhan. CTO and vice-president for the Cloud Networking Group at Citrix.
- Chris Wright. Technical Director of SDN at Red Hat.
- David Meyer. CTO and Chief Scientist at Brocade Communications.
- David Ward. CTO and Chief Architect of Engineering at Cisco Systems.
- Francois LeMarchand. Head of Ericsson SDN product & solutions group.
- Inder Gopal. Responsible for all hardware and software networking products in IBM.
- Jennifer Lin. Jennifer was a member of the founding team and VP, Marketing at Contrail Systems, where she led the product management and marketing activities, until Juniper acquisition of the company.
- Marc Cohn. Senior Director at Ciena Corporation, has been a major contributor to Ciena's Software Defined Networking (SDN) initiative, and chairs the Committee (MEC) for the Open Networking Foundation (ONF).
- Rajeev Nagar. Oversees the Core Networking team, responsible for infrastructural networking/connectivity technologies for Microsoft.
- Sarwar Raza. Director of Cloud Networking and Software Defined Networking (SDN) in the Advanced Technology Group at HP Networking.
- Terry Nakajima. Senior Manager in NEC Corporation of America, leading business strategy and engineering for NEC's Software defined networking products.

6.2 OpenDaylight Community Management

This chapter will analyze aspects having to do with community management. Different tools and communication channels that the project provides to its community in order to facilitate community involvement in the project will be studied.

Besides this, Marketing strategies will be analyzed, as well as community building and new volunteers attraction strategies. Entry barriers will be analyzed as well, by studying the different mentorship and internship programs or the available documentation for new members.

6.2.1 Communication Channels

Apart from tools related to software development and other technical stuff, OpenDaylight project provides a set of tools for community members to keep in touch and communicate each other. Apart of the aforementioned tools, there is a set of channels to help community communication:

- **Social Media.** Far from criticism, and with emphasis on its pragmatic view, the OpenDaylight project is conscious of the **importance of the Social Media**. For this reason, main Social Networks accounts have been created for communication with the community:
 - **Twitter.** OpenDaylight Twitter account [[Twi14](#)], with nearly 5300 followers, important information and events are published through this tool.
 - **Facebook.** Not as important compared to Twitter, OpenDaylight Project Facebook page [[Fac14](#)], with 1360 Likes, is another channel where links to interviews and important events are published.
 - **Google+.** OpenDaylight Google+ profile [[Plu14](#)], with nearly 700 Google+ users on its circles, provide similar information about the community of OpenDaylight project, such as events holding, community involvement presentations or technical overview videos.
 - **LinkedIn.** OpenDaylight LinkedIn profile [[Lin14](#)], with more than 360 followers, is a tool to communicate with this Social Network oriented to job environment. New software releases, man events to occur, new members in the project or presentation of members of the community are examples of the kind of information which is broadcasted in LinkedIn's OpenDaylight Project page.
 - **Youtube.** OpenDaylight's Youtube channel [[You14](#)] has more than 800 subscribers, and owns more than 60 videos, from the introduction of the project to different HackFest speaks, tutorials and round tables.
- **IRC Channel.** Hosted on freenode.net, the #OpenDaylight channel hosts around some dozens of members willing to help and connect with other members of the OpenDaylight developer community.
- **Mail Lists.** Apart from the previous channels, more than 100 mail lists exist [[Lis14](#)] to assist on different question and issues regarding a particular aspect of a particular project.

Mail List	Discussion
opendove-announce	Open DOVE Announcements (low volume)
opendove-bugs	Notifications from Bugzilla for Open DOVE
opendove-dev	Developer discussions for Open DOVE
opendove-gerrit	Gerrit automated notices about Open DOVE
opendove-jenkins	Jenkins CI notifications for Open DOVE
opendove-users	Community driven support alias for Open DOVE

Table 6.1: Example of OpenDaylight’s OpenDove Project mail lists

There is normally one mail list per aspect of the project considered to be under discussion.

For instance, “OpenDove” project hosts the mailing lists listed in Table E.1.

A complete list of OpenDaylight Mail List can be obtained in on Appendix E.

- **Blogs.** Blogs in OpenDaylight are a powerful and extensively used tool to communicate latest updates in the project. Two blogs stand out among the rest:
 1. Neela Jacques’ blog. Neela Jacques, executive director of OpenDaylight, shares his view on the different threats and opportunities around OpenDaylight project, with high focus on the Open nature of the project and technologies involved.
 2. OpenDaylight’s blog. This generic blog is more focused on broadcasting general information related to the project, from event celebration, to community member’s spotlights, it also covers important update information, such as the announcement of the OpenDaylight Ambassador program, or the different new tutorials available.
- **Contact Information.** Apart from the previous communication channels, OpenDaylight project provide a Contact Information section with the most important contact mail addresses, such as the ones for Press and Analyst Inquiries, Event Inquiries, Website Questions or other Information Inquiries. It also provide a form to address general inquiries.

6.2.2 Community Tools

As defined previously, OpenDaylight project is defined to be **open**, highlighting the project wish for **everybody to participate**. Every Open Source project that emphasizes on community needs to provide an **infrastructure that ease all different kind of activities involved on**

collaboration, not just software development, but also other very important activities such as documentation, Quality Assurance (QA), etc.

In this sense, OpenDaylight project is not an exception, and provides several tools [Too14], available to help OpenDaylight contributors and collaborators on the different tasks they want to carry out. Among the different utilities, next ones must be remarked:

- **Wiki** [Wik14x]. For an Open Source project, **it is very important to own a well structured and simple Wiki**. A project wiki is, normally, together with the FAQs, the entry point for starting contributions and potential community members. In OpenDaylight Wiki, a simple main page appears, with several important aspects such as:
 1. **Welcome message**. An introductory welcome message with sign up information needed for contribution.
 2. **OpenDaylight Projects**. With different important introductory documentation, such as **Getting Started for Developers** [Gui14a] guide, the **Getting Started for Projects** [Gui14c] guide, or the **Code of conduct**. This section includes as well infrastructure guides, Core Architectural Principles and a link to the different sub-projects existing under OpenDaylight project. Another important reference linked from this section is the **Bugs Wiki page** [Bug14a], with information to help on bug resolution, fields used on but reporting, etc.
 3. **Technical Collaborations**. This section of the Wiki covers different technical collaboration aspects having to do with Developer Events, Work Streams, Meetings and Committees, Intern Project Ideas and Cross Project Collaborative Groups. Other aspects, such as Simultaneous Release information for each of the upcoming releases, are also handled in this section of the Wiki.
 4. **Community**. Last, but not least, some information regarding the Community, such as the Ambassador Program or the different presentations [Pre14] of the OpenDaylight project available.
- **Ask** [Ask14]. This **AskBot** [TM14a] based **Q&A Forum** provides a mechanism for the community to ask their questions for other members of the community to provide help on that particular aspect. Questions appearing on this tool are usually of an **advanced**

technical level, i.e.: “how we could use an ADSAL plugin service inside of an MDSAL plugin ?”. This tool provide a **Tagging Mechanism**, where questions are classified normally by its technical area. Tags such as **REST**, **openflow**, **vlan** or **GUI** or **neutron** are examples of the tags under which the different questions on Ask are classified.

- **Identity** [Ide14]. Identity utility is just a **WSO2 Identity Server** [WSO14] based tool to create and manage an OpenDaylight user accounts. Contributors to the OpenDaylight tools must have signed-up in Identity to publish code, review code, publish Wiki pages or perform any other contribution task. Apart from being a user account management tool, **Identity also manages different applications a particular contributor has access too**.
- **Bugzilla** [Bug14b]. A **Bugzilla** [bug14c] based **Bug Tracking** web tool. In this sense, OpenDaylight project just takes advantage of this **widely used** Bug Tracking system. OpenDaylight Bugs are classified by Product (ovsdbd, for instance) and Component inside the product (API, for instance). Nearly 500 Bugs are, up to date, registered on OpenDaylight Bugzilla.
- **Gerrit** [Ger14b]. A **Gerrit** [Ger14a] based **Reviewing Code** web tool. In this sense, OpenDaylight project just takes advantage of this **widely used** reviewing code system, that easily integrates with other tools such as Jenkins and Git Source Control system.
- **Jenkins** [Jen14]. A **Jenkins** [CI14] **Continuous integration** web tool. In this sense, OpenDaylight project just takes advantage of this **widely used** continuous integration system, that easily integrates with other tools such as Gerrit and Git Source Control system, to perform automatic Quality Assurance in Continuous Integration.
- **Nexus** [Nex14]. A **Nexus** [Son14c] **Repository Manager** web tool. This tool allows proxying remote repositories as well as sharing software artifacts, organize artifacts, improve collaboration across teams and source the highest quality components for use in their applications.
- **Sonar** [Son14a]. A **SonarQube** [Son14b] **technical debt** web tool. With a very efficient way of navigating, this tool provides a balance between high-level view, dashboard, TimeMachine and defect hunting tools. OpenDaylight tracks in this tool some of the different projects, such as yangtools, OpenFlow or osvdb, to quickly uncover components that are in Technical Debt and establish action plans to fix them.

6.2.3 Marketing and Community Building

OpenDaylight project is, no doubt, following an Open Source strategy more aligned to the pragmatic view of Open Source. With a high commercial view, caused mainly by the amount of big and important companies involved, and a well driven marketing strategy, this Open Source community is more focused on the openness aspect of the project, rather than on the idea of the Open Source Software aspect.

Taking the previous statement into consideration, it must be remarked that OpenDaylight project **has invested a big effort on project broadcasting and community involvement**. In this section, the most important aspects having to do with this aspects are analyzed. Questions such as “How can a contributor participate in the project?” or “Which are the main community driven events?” are answered in this section.

To start with, OpenDaylight project provides some guidelines on **How to participate** [Par14]. From the project’s perspective, the best entry point to the project is the “Getting Started Guide”. Apart from that, there is also an “Events” entry, where the most important meetings are notified. Both aspects are analyzed in this section.

Having to do also with Marketing and Community building, some other sections from the OpenDaylight project web page, such as “News” or “Resources”, are part of the marketing strategy of the project. To end this section, other groups and resources that OpenDaylight provides to achieve Community Building will be mentioned.

1. **Getting Started Guide.** This introductory guide [Gui14b] plays the role of starting point for any contributor, collaborator or company which is willing to participate in the community. This guide provides different starting information classified by the different kind of community roles considered, i.e.:

- **For Users.** Download software releases information and software editions are provided for the user to start playing with OpenDaylight project software. Apart from that, reference to previously described mail lists is suggested for further investigation.

- **For Developers.** A list of the tools and resources available for developers to start contributing are also described. Examples of pointed resources, the previously described tools (remarking Wiki as the most important), as well as communication channels such as mail lists.
 - **For Membership.** Not only personal contribution, but also companies and organizations, are welcome to join at any time and to start participating, as OpenDaylight is an Open Source software project. Companies and organizations interested in formal membership in the project can contact submit a request for information on the membership page [Joi14].
2. **Main Events.** The most important marketing tool available for Open Source project are Events. This kind of meetups are very important for people involved in the community, members, companies and in general any person, organization or community involved int the project to know each other.

Main event regarding OpenDaylight Open Source project is **OpenDaylight Summit**. This **annual** event will hold its second edition in 2015 in Santa Clara, where the first edition, in 2014, was also held. First edition was a great success, with **over 600 attendees and almost 50 speakers** [Fou14f]. OpenDaylight events are directly driven by the Linux Foundation.

OpenDaylight Summit first edition was a two day event, with more than 40 speaks, covering different aspects about the OpenDaylight project in particular, but also, from a more general perspective, about all the different technical aspects, technologies and protocols involved in OpenDaylight, such as SDN, NVF, OpenFlow, Open vSwitch, Policy and Orchestration, etc, where a complete set of videos and presentations result from.

Apart from main event, there is a set of **secondary events** organized by OpenDaylight project. Among the most important, next ones are found:

- **HackFests.** This kind of events focus on **introducing new community members**, providing hands on workshops aimed at building, through OpenDaylight platform,

SDN applications. Other aspects, such as **bugfix coaching sessions**, also take place into this kind of event, to help people willing to contribute and ease the entry barriers to start. Düsseldorf or Tokyo are among the cities that have hosted this kind of OpenDaylight events.

- **Developer Design Forums.** This kind of event, **focused on the developer community**, try to engage in a range of hot topics geared towards **networking**. Additionally, this kind of forum help to **set the stage for future OpenDaylight releases**.
- **Other events.** Not directly associated or organized by the OpenDaylight project, the community can also receive speaks on this project as part of other events. Examples of events where OpenDaylight appeared are the **OpenStack Summit**, that will be hosted in Paris in November 2014, or the **Broadband World Forum**, that will be hosted in October 2014. Being part of such important events gives a very wide idea of the importance of the OpenDaylight project and how crucial is the role that this project can play looking ahead.

3. **News.** OpenDaylight web page is usually communicating **Announcements** [[Ann14a](#)] considered to be remarked due to their importance for the community. This is, no doubt, together with Social Media, a good mechanism to communicate about the small achievements that the community accomplishes along the time. The kind of information announced varies, but has to do with aspects such as:

- **New companies joining the project.**
- **Community members upgrade membership processes.**
- **Awards, recognition and prizes.**
- **New programs oriented to community building.**

In general, all kind of important stuff from community building perspective are communicated through this section. Examples of announcements are shown below:

- Infinera and KEMP Join the OpenDaylight Project to Enable Open Software-Defined Networking [[Ann14c](#)].
- OpenDaylight Announces HP Is Now a Platinum Member [[Ann14d](#)].
- OpenDaylight Announces Summer Internship Program for Student Developers [[Ann14e](#)].

- OpenDaylight is the First Open Source Project to Receive Interop Event's Grand Prize Award [Ann14f].
4. **Resources.** Apart from the already described sections, where resources like the “Getting Started Guide” appear, OpenDaylight provides a **Resources** [Ann14g] section, where important **Publications** [Pub14] and other **Collateral** [Col14] documents are collected.

But OpenDaylight available resources are not limited to just publications and other kind of documentation. There are also available some **Community Labs** [Lab14], publicly available, to allow developers to become familiar with the OpenDaylight platform, on the one hand, and help support the integration and testing of SDN and NFV innovations and applications, on the other hand. These labs are important, taking into account that, **as the OpenDaylight community continue to grow**, including platforms, protocols and technologies, **so does the need for testing and integration to ensure interoperability between all the components.**

5. **Community Programs and User Groups.** Last, but not least, community involvement programs play a prominent role in community building, and Open Source governance members are aware of it. For this reason, there are different groups and programs to help community building. Among the different programs, next ones must be highlighted:

- **Summer Internship Program** [Pro14b]. OpenDaylight has accepted five student interns during the summer of 2014, to work in the OpenDaylight development community. Each intern was assigned a mentor to work closely with for the duration of his/her internship. To select interns, Internship proposals from student were sent to OpenDaylight governance, with a project proposal related to the project, together with academic experience and Open Source project experience of the candidates.
- **The OpenDaylight Ambassador Program** [Pro14a]. Interest in the OpenDaylight project is growing day by day. This growth, which is taking place quickly, has driven to OpenDaylight governance to initiate the ambassador program, which is intended to help sustain and grow the worldwide community. **OpenDaylight experts will be part of this program**, as no one knows and use the code better, and are willing to share that knowledge with others around the world, with the project providing

support and resources to enable that.

- **OpenDaylight User Groups** [Gro14]. This kind of groups are **regional, self-organized, informal associations** that perform global meetings to discuss about OpenDaylight project. The assistants to this kind of meetings share knowledge, recruit new developers on board, discuss best practices and technical challenges, as well as create awareness. These groups also run small hackathons and host small social community events. These groups do not have a formal relationship with the OpenDaylight Project, but are considered to be a key part of the vast and growing ecosystem of the community. Up to date, there are three groups in USA (Austin, TX; Baltimore, MD; Silicon Valley, CA), one group in Canada(Montréal, QC) and also a group in Europe(München, Germany).

Apart from the previous remarks, OpenDaylight project also emphasizes on the use of the communication channels, such as mailing list or IRC channel, remarking as well the existence of tools, such as the Developer Wiki, described also in the previous subsections, as components available for the community to start contributing and involving in the OpenDaylight project.

Chapter 7

OpenDaylight: Technical Aspects

Technical aspects around OpenDaylight project will be analyzed in this chapter, and how the project has been born inside Networking industry to empower and accelerate technologies involved, among other tasks. This fact, being based on some emerging technologies, but also, in which there is a high expectation too, is one of the most important challenges within the project.

This chapter will analyze the state of art of the project and the different protocols used in the project, together with other aspects that are also remarkable from the technical perspective, such as programming languages used.

7.1 Technical Overview

As described in the introductory chapter 3, SDN and NFV technologies are the most important ones in OpenDaylight project. SDN was born as a mechanism to separate the control plane from the data plane within the network, allowing centralization of the intelligence and state of the network while abstracting the complexity of the underlying physical network.

In order to achieve, as summarized by the OpenDaylight Technical Overview [Ove14], **great strides regarding SDN goals have been made within the industry toward with standardized protocols such as OpenFlow**. However, OpenDaylight mission perception is that **greater collaboration leveraging Open Source development best practices will significantly accelerate real, deployable solutions for the industry at large**.

On the other hand, by evolving network services from an appliance model to one that leverages virtual compute, storage, and networking, **NFV** empowers network functions definition to **drastically improve the cost of deploying networking components as well as the agility of when and where to run.**

Both technologies, SDN and NFV are a new way of deploying network infrastructures. A software-defined network adapts to the requirements of applications deployed on the network, to improve **current generation of networks and architectures, which are statically configured and vertically integrated, has the limitation of scaling up difficultly**. However, **new generation applications such as Big Data or Video Delivery require networks to be agile and to flexibly adapt to application requirements.**

From a high level view, software defined networking is commonly described in layers.

- **Network Apps & Orchestration:** The top layer consists of business and network logic applications that control and monitor network behavior. In addition, more complex solution orchestration applications needed for cloud and NFV thread services together and engineer network traffic in accordance with the needs of those environments.
- **Controller Platform:** The middle layer is the framework in which the SDN abstractions can manifest, providing a set of common APIs to the application layer (commonly referred to as the northbound interface) while implementing one or more protocols for command and control of the physical hardware within the network (typically referred to as the southbound interface).
- **Physical & Virtual Network Devices:** The bottom layer consists of the physical & virtual devices, switches, routers, etc., that make up the connective fabric between all endpoints within the network.

OpenDaylight, as shown in Figure 7.5, has been designed on a modular and pluggable basis. Main component of the project is a flexible controller platform, which is considered to be the core component. **This controller is implemented strictly in software and is contained within**

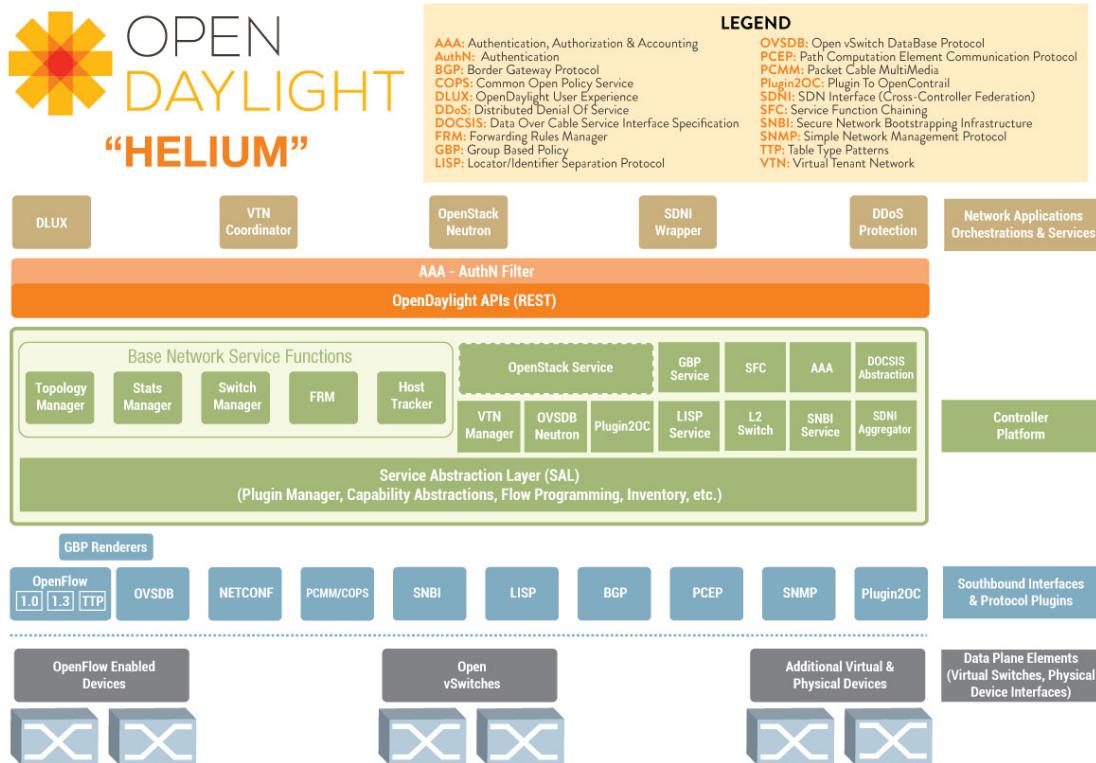


Figure 7.1: OpenDaylight Architecture Diagram

its own Java Virtual Machine (JVM) [TM14b]. Having some disadvantages, **Java has been selected as the core component programming language to achieve the wider target users**, as it is known that Java can be deployed on a wide hardware and operating system platforms.

The controller exposes open northbound APIs which are used by applications. OpenDaylight supports the **OSGi framework** and bidirectional **Representational State Transfer (REST)** [W3C14] for the northbound API. By using both technologies a **complete scope is provided**, as, while the OSGi framework is used for applications that will run in the same address space as the controller, the REST (web based) API is used for applications that do not run in the same address space (or even necessarily on the same machine) as the controller.

The **business logic and algorithms reside in the applications**. Controller is used by the application to **gather network intelligence, run algorithms and perform analytics**. Apart from this, **controller is operated by applications to orchestrate the new rules, if any, throughout the network**.

Apart from being multi-platform designed and being easily scalable, **extensibility** is one of the main characteristics of OpenDaylight project. **Controller** has been implemented **to be upgraded by dynamically pluggable modules** that can be added to perform required network tasks. There are a series of base network services to perform different basic tasks, as understanding what devices are contained within the network, capabilities of each of the components, monitoring or statistics retrieval. Besides this, for enhanced SDN functionality, platform oriented services and other extensions can also be inserted into the controller platform.

The **southbound interface is capable of supporting multiple protocols (as separate plugins)**, e.g. **OpenFlow 1.0, OpenFlow 1.3, BGP-LS, etc.** Service Abstraction Layer (SAL), has been designed for these modules to be dynamically linked into. The **SAL exposes device services to which the modules north of it are written, determining how to fulfill the requested service**, irrespective of the underlying protocol used between the controller and the network devices.

One of the most exciting aspects of the OpenDaylight Project is that its components and technologies continue evolving based on the contributions of its developer community, favored by its extensible and modular design.

7.2 OpenDaylight Projects

After technical overview presentation 7.1, an analysis of the different technologies involved on OpenDaylight project development is performed on this chapter. To start with, it has already been mentioned that **OpenDaylight is structured in different projects that handle different software features provided**. Being the controller the main project of OpenDaylight, there is a complete set of projects to perform different tasks, provide different features to applications through Northbound API and communicate over different protocols through the Southbound (SB) API.

OpenDaylight project is in turn a set of different projects that have been contributed to a com-

mon platform, in order to achieve an homogeneous ecosystem around SDN and NFV technologies. Some of the projects are focused on a particular feature (i.e.:security). Meanwhile, other projects are based on developing or enhancing a Network Protocol (i.e.:OpenFlow). There are also another kind of projects that focuses on actions to accomplish for the correct evolution of the project, such as Documentation or Integration projects. In this section, OpenDaylight's project will be enumerated and briefly described, in order to increase the knowledge of each of the different pillars on which the project is sustained. It must be remarked that all of the information is available on the OpenDaylight Wiki [Wik14x]:

1. **AAA Service** [Wik14a]. AAA project is aimed to provide **Authentication, Authorization and Accounting in OpenDaylight project**. On the one hand, Authentication allows managing users permissions to access the platform in a token/claim-basis, with support of authentication methods such as OAuth, SAML or Keystone, among others. On the other hand, Authorization mechanism allows determining which part of the SDN controllers can be used and/or administered by a particular user. Last, but not least, authentication provide features for all accesses to be recorded and searchable for various purposes such as billing, analysis, diagnostics or security audit.
2. **Affinity Metadata Service** [Wik14b]. The Affinity service provides an API to allow applications and controller itself to create and share an abstract networking topology, independent to the infrastructure needs and preferences. The term **affinity describes the nature of “conversations” across the networking infrastructure and their participants**. The Affinity API allows intent to be specified in application and service terms independent of how and where the communicating workloads attach to the network, with no details of forwarding devices, vendors or technologies included. **This can enable users to, for example, customize the behavior of virtual networks to better support their applications without knowing anything about bridges, routers, VLANs or tunnels**. In the end, the **affinity** concept is based on a layered architectural model in which intelligent controllers dynamically provision data center network infrastructure to satisfy workload affinities, as directed by external applications.
3. **BGP-LS/PCEP** [Wik14c]. BGP/PCEP protocol project aims to provide Java-based implementation of Border Gateway Protocol (BGP) [For06] and Path Computation Element

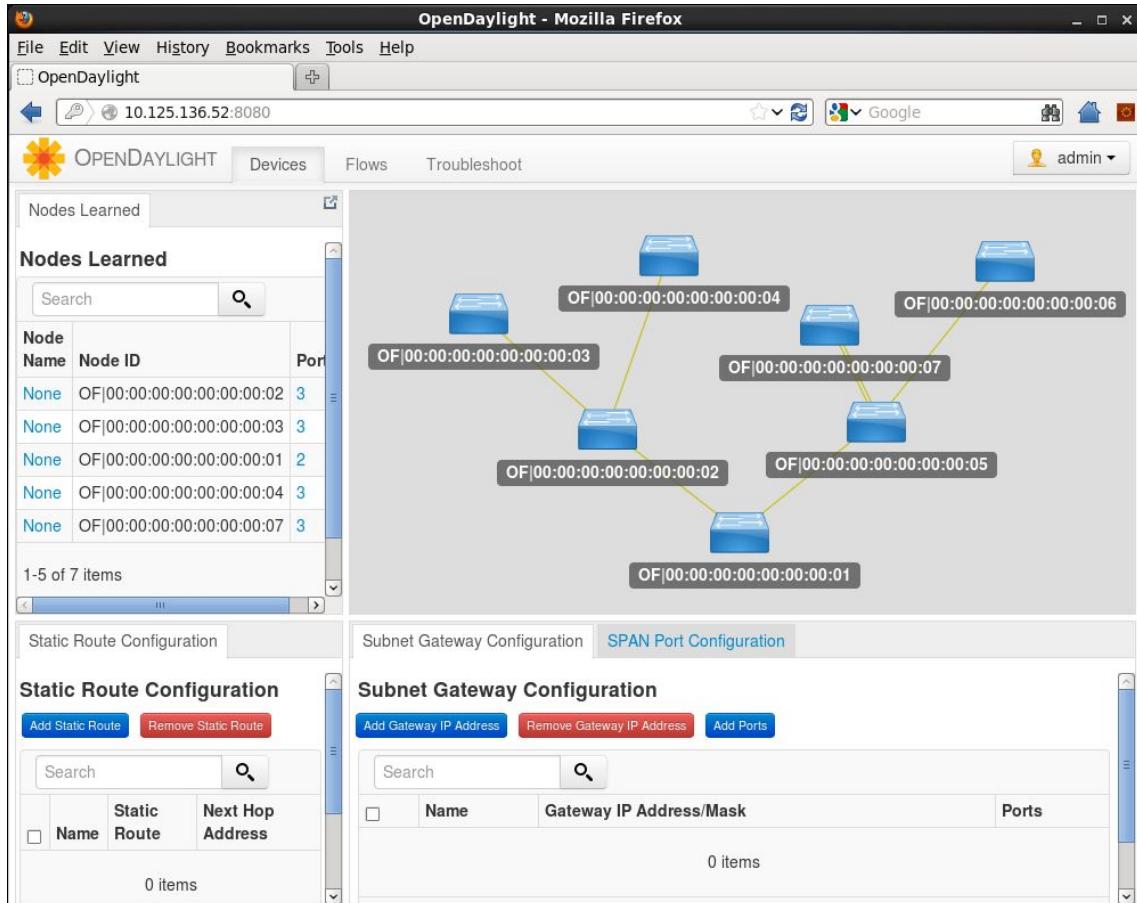


Figure 7.2: OpenDaylight Controller

Protocol (PCEP) [For09]. By enabling OpenDaylight platform to utilize more standardized ways of talking to the underlying network, it will be deployed in a wider variety of scenarios. **BGP is the core protocol holding together the Internet in its current shape and form**, while PCEP was designed for offloading optimal path computation. It is also remarkable the fact of the availability of Open Source Network Protocols software implementation, and BGP-LS/PCEP is aimed also to empower this fact.

4. **Controller** [Wik14d]. The OpenDaylight Controller –depicted in Figure 7.5– is without doubt the **main project** inside project’s ecosystem. OpenDaylight’s SDN Controller provides the ability to **deploy software to control the network gear and redeploy as needed**. The main goal within this project is to have a modular Controller with a well published Northbound API for network Applications to write towards while utilizing SB protocols such as OpenFlow to communicate with supported downstream network nodes.

Apart from detailed characteristics, the Controller provides interfaces to third party applications, such as **OpenStack Neutron** module, or advanced features such as **clustering, auditing and monitoring**.

5. **dlux - openDayLight User eXperience** [Wik14f]. This project focuses on OpenDaylights user experience. The latest (current) version uses a JavaScript based Single-Page-Application using AngularJS, running on NodeJS.
6. **Documentation Project** [Wik14g]. This vertical group aims to build the content infrastructure, improve the user content for the various OpenDaylight projects and ensure continued quality in the content.
7. **Defense4All** [Wik14e]. SDN application for detecting and mitigating Distributed Denial of Service (DDoS) attacks. By communicating with OpenDaylight Controller via north-bound REST API, performs tasks such as **monitoring behavior of protected traffic** and **diverting attacked traffic to selected Attack Mitigation Systems (AMSS)**.
8. **Dynamic Resource Reservation** [Wik14h]. Reservation application provides end to end multi-layer provisioning. **Reservation application focuses on flexible provisioning based on information such as latency, bandwidth or any other networking parameter involved in an end to end communication**.
9. **Group Policy Plugin Project** [Wik14i]. Currently, the project is in the early stages. The main aim of this project is to **separate application connectivity requirements and underlying details of network infrastructure** via an application-centric policy model.
10. **Integration Group** [Wik14j]. This project is more a infrastructure project that tries to coordinate and drive **integration and test efforts to achieve a successful OpenDaylight release**. To achieve this mission, tasks such as developing a test strategy, perform continuous system integration test plan or develop and maintain an automated test framework are responsibilities of this project.
11. **L2 Switch** [Wik14k]. This project tries to separate out the Layer 2 specific handling code into a separate layer, creating several reusable services, by providing some generic L2 services, such as **Packet Handling, Address Tracking or Path Computation Services** . De-

pending on the Plugins loaded, Controller can contain certain packet handling/forwarding behavior with the ARPHandlers, Host Trackers, Simple Forwarders, etc.

12. **LISP Flow Mapping** [Wik14l]. The aim of this project is to enable LISP protocol [For13b] inside OpenDaylight. LISP is a network-layer-based protocol that enables separation of IP addresses into two new numbering spaces: Endpoint Identifiers (EIDs) and Routing Locators (RLOCs). This technology provides a flexible map-and-encap framework that **can be used for overlay network applications, such as data center network virtualization, and Network Function Virtualization (NFV)**.
13. **Open DOVE** [Wik14m]. Open Distributed Overlay Virtual Ethernet (OpenDOVE) project is a network virtualization platform with a full control plane implementation for OpenDaylight and **data plane based on OpenvSwitch** [For13a]. DOVE is, in turn, a network virtualization platform that provides isolated multi-tenant networks on any IP network in a virtualized data center, providing each tenant with a virtual network abstraction providing layer-2 or layer-3 connectivity and the ability to control communication using access control policies.
14. **OpenFlow Plugin** [Wik14n]. OpenFlow [Fou12] is a vendor-neutral standard communications interface defined to enable interaction between the control and forwarding layers of an SDN architecture. **The OpenFlow plugin project intends to develop a plugin to support implementations of the OpenFlow specification** as it develops and evolves. Specifically the project will continue to provide support for the existing OpenFlow 1.0 implementation, developing a plugin aiming to support OpenFlow 1.3.x, and further supporting subsequent OpenFlow specifications.
15. **OpenFlow Protocol Library** [Wik14o]. This project provides a library that mediates communication between OpenDaylight controller and hardware devices supporting **OpenFlow** [Fou12]. Primary goal is to **provide a communication channel to the user in order to provide a mechanism for managing network hardware devices**.
16. **OpFlex Implementation Project** [Wik14o]. This project consist of the implementation of the OpFlex protocol, together with an OpenFlex SB plugin and Policy Agent to implement functionality of OpenFlex [For14a], an architecture that provides a distributed

control system based on a declarative policy information model.

17. **OVSDB Open vSwitch Database Integration Project** [Wik14p]. The OVSDB Plugin project tries to accomplish the implementation of the **Open vSwitch Database management protocol** [For13c]. The project consists of a library, along with various plugin usages, that allows configuration of vSwitches through southbound API.
18. **OSCP Project** [Wik14q]. The OpenDaylight SDN Controller Platform (OSCP) is the **network application platform that provides unified network intelligence, enterprise-class scalability and high availability, and a platform to deploy a wide range of network applications**. The OSCP controller provides the centralized control plane tier in the three-tier Open SDN architecture. While OSCP is logically centralized, the controller is installed on multiple controller-nodes for redundancy and scale.
19. **PacketCable PCMM Project** [Wik14r]. Packet Cable MultiMedia (PCMM) [For14b] provides an **interface to control and management service flow for Cable Modem Termination System (CMTS) network elements**. A service flow constitute a data path between a CMTS and a subscriber's cable modem (CM) guaranteed application specific quality of service (QoS), known as Dynamic Quality of Service (DQoS). PCMM offers the ability to deliver new services using existing cable infrastructure. **DOCSIS** [CTL09] encapsulation according to the standard has been introduced in OpenDaylight to achieve abstraction of this encapsulation type.
20. **Secure Network Bootstrapping Infrastructure (SNBI) project** [Wik14s]. The Secure Network Bootstrapping Infrastructure (SNBI) project is aimed to **perform securely and automatically the bring up of an integrated set of network devices and controllers**. SNBI devices and controllers automatically discover each other, get an IP address assigned, and establish secure IP connectivity.
21. **Service Function Chaining** [Wik14t]. Service Function Chaining provides the ability to define an ordered list of a network services (firewalls, load balancers, routers). These service are then linked together in the network to create a service chain. **This project provides the infrastructure (chaining logic, APIs) needed to provision a service chain in the network and an end-user application for defining such chains**.

22. **SNMP4SDN** [Wik14u]. This project has been proposed to develop a Simple Network Management Protocol (**SNMP**) **southbound plugin to control underlying devices supporting SNMP**, using off-the-shelf commodity Ethernet switch. In addition to SNMP support, this plugin will provide certain capabilities to manage configurations that can only be accessed via CLI, e.g. ACL or disabling flooding, since such configurations are necessary for using Ethernet switches for SDN.
23. **Table Type Patterns (TTPs)/Negotiable Datapath Models (NDMs)** [Wik14x]. Table Type Patterns (TTPs) are the first tangible output from the ONF's Forwarding Abstractions Working Group (FAWG). The goal is to **allow for an OpenFlow controller and OpenFlow switch to agree on a set of functionality**, and help managing the increased diversity made possible with OpenFlow versions 1.1+.
24. **Toolkit Project** [Wik14v]. One of the concerns in OpenDaylight was the consistent feedback received from developers about the complexity in developing Apps for OpenDaylight. OpenDaylight Toolkit is a **framework to enable developers to easily develop applications on top of the OpenDaylight controller**. The main mission of the project is to be infrastructure independent and allow developers to develop application on top of multiple platforms.
25. **Virtual Tenant Network (VTN)** [Wik14w]. OpenDaylight Virtual Tenant Network (VTN) is an **application that provides multi-tenant virtual network on an SDN controller**. The network is designed on VTN, as **VTN basically allows the users to define the network with L2/L3 network look and feel**. Once network is designed, it will automatically be mapped into underlying physical network, and then configured on the individual switch leveraging SDN control protocol.
26. **YANG Tools** [Wik14y]. Last, but not least, YANG Tools is a infrastructure project aiming to develop **necessary tooling and libraries providing support of NETCONF** [For14d] and **YANG** [For14c] for Java (JVM-language based) projects and applications. The project aims generic functionality related to the YANG, that should not be a part of Controller project, and should live in separate project to be reused by other applications and parties without directly depending on Controller.

7.3 OpenDaylight Source Code

It is important to analyze **programming languages involved**, in order to identify which are the main software development languages used in both the core controller and the different plugins and other subprojects available on the project, as programming languages usage is a key aspect to analyze the technical constraints.

As previously described, OpenDaylight project is composed in turn of a subset of projects that allow achieving a complete platform to act as SDN Controller as well as provide NFV functionalities through an Open Source software basis.

OpenDaylight's software source code is splitted taking into account different previously described projects existence, and can be retrieved from different GIT repositories (one per project), described in Table 7.1.

The collection of repositories in Table 7.1 contains all the source code composing OpenDaylight software. It has been also clarified that the core component, the **controller, has been implemented in Java**. But, regarding source code, programming languages used and other software stuff, is worth considering a set of questions: how wide is OpenDaylight software? Are there other programming languages used in the project apart from Java? Which is the most used programming language?

To start with, it must be clarified that SLOCCount [Whe14] has been used in order to perform different measures across the different repositories. By using this tool, next measures have been obtained:

- **Source Lines Of Code (SLOC)**. This tool allows counting the total amount of lines of code existing on the project. This is a first approach to measure the complexity of the project and different subprojects.
- **Programming Languages Usage and Classification**. SLOCCount also provides a mechanism to classify the total percent usage of each of the programming languages in the whole source code of the project.

Project	Git Repository
AAA Service	https://git.opendaylight.org/gerrit/p/aaa.git
Affinity Metadata Service	https://git.opendaylight.org/gerrit/p/affinity.git
BGP-LS/PCEP	https://git.opendaylight.org/gerrit/p/bgpcep.git
Controller	https://git.opendaylight.org/gerrit/p/controller.git
dlux - openDayLight User eXperience	https://git.opendaylight.org/gerrit/p/dlux.git
Documentation Project	https://git.opendaylight.org/gerrit/p/docs.git
Defense4All	https://git.opendaylight.org/gerrit/p/defense4all.git
Dynamic Resource Reservation	https://git.opendaylight.org/gerrit/p/reservation.git
Group Policy Plugin Project	https://git.opendaylight.org/gerrit/p/groupbasedpolicy.git
Integration Group	https://git.opendaylight.org/gerrit/p/integration.git
L2 Switch	https://git.opendaylight.org/gerrit/p/l2switch.git
LISP Flow Mapping	https://git.opendaylight.org/gerrit/p/lispflowmapping.git
Open DOVE	https://git.opendaylight.org/gerrit/opensdove.git
OpenFlow Plugin	https://git.opendaylight.org/gerrit/p/openflowjava.git
OpenFlow Protocol Library	https://git.opendaylight.org/gerrit/p/openflowplugin.git
OpFlex Implementation Project	https://git.opendaylight.org/gerrit/p/opflex.git
OVSDDB Open vSwitch Database Integration Project	https://git.opendaylight.org/gerrit/p/ovsdb.git
OSCP Project	https://git.opendaylight.org/gerrit/p/net-virt-platform.git
PacketCable PCMM Project	https://git.opendaylight.org/gerrit/p/packetcable.git
Secure Network Bootstrapping Infrastructure (SNBI) project	https://git.opendaylight.org/gerrit/p/snbi.git
Service Function Chaining	https://git.opendaylight.org/gerrit/p/sfc.git
SNMP4SDN	https://git.opendaylight.org/gerrit/p/snmp4sdn.git
Table Type Patterns (TTPs)/Negotiable Datapath Models (NDMs)	https://git.opendaylight.org/gerrit/p/ttp.git
Toolkit Project	https://git.opendaylight.org/gerrit/p/toolkit.git
Virtual Tenant Network (VTN)	https://git.opendaylight.org/gerrit/p/vtn.git
YANG Tools	https://git.opendaylight.org/gerrit/p/yangtools.git

Table 7.1: OpenDaylight projects' repositories

Using SLOCCount is frankly simple, as it only has to be executed on the directory where all the code has been retrieved:

```
$ slccount .

Creating filelist for aaa
Creating filelist for affinity
Creating filelist for bgpcep
Creating filelist for controller
...
Creating filelist for yangtools
```

SLOC	Directory	SLOC-by-Language (Sorted)
661838	vtn	cpp=349960, java=192289, ansic=80272, python=17402, perl=11928, xml=8904, sh=964, asm=119
289437	controller	java=250524, xml=37816, sh=772, jsp=325
142681	oscp	java=98499, python=42727, xml=1075, sh=380
99718	opendove	ansic=67487, python=23998, java=7260, xml=817, sh=156
82770	yangtools	java=75792, xml=6978
59026	defense4all	java=56122, xml=2010, sh=461, perl=356, python=77
55431	bgpcep	java=43460, xml=7586, python=4385
47232	openflowplugin	java=37604, xml=7486, python=2140, sh=2
37232	sdninterfaceapp	java=33954, xml=3278
33577	ovsdb	java=28162, xml=4840, sh=421, python=154
32348	dlux	xml=32086, java=262
27003	openflowjava	java=25704, xml=1299
23443	snbi	ansic=18796, java=3357, xml=1168, sh=72, python=50
22672	opflex	ansic=19412, python=2844, sh=416
20032	snmp4sdn	java=18171, xml=1861

```

16138 groupbasedpolicy java=14532,xml=1228,python=378
15627 packetcable      java=13780,xml=1248,python=455,sh=144
15436 integration      xml=9377,python=3851,sh=2208
14479 lispflowmapping  java=11384,xml=3095
11415 aaa              java=7977,xml=2925,sh=513
11311 toolkit          java=5365,xml=5254,jsp=499,sh=193
11167 l2switch         java=8190,xml=2977
9567 sfc              java=5902,xml=3223,python=417,sh=25
8375 docs             xml=8375
8206 affinity         java=5128,xml=1949,python=1085,sh=44
3659 tcpmd5           java=2161,xml=1205,ansic=293
1752 reservation      xml=1752
1567 ttp               xml=892,java=437,python=238
910 odlparent         xml=910

```

Totals grouped by language (dominant language first) :

java:	946016 (53.63%)
cpp:	349960 (19.84%)
ansic:	186260 (10.56%)
xml:	161614 (9.16%)
python:	100201 (5.68%)
perl:	12284 (0.70%)
sh:	6771 (0.38%)
jsp:	824 (0.05%)
asm:	119 (0.01%)

Total Physical Source Lines of Code (SLOC) = 1,764,049

Development Effort Estimate, Person-Years = 512.70

Basic COCOMO model, Person-Months = 2.4 * (KSLOC**1.05))

Schedule Estimate, Years (Months) = 5.74 (68.83)

(Basic COCOMO model, Months = 2.5 * (person-months**0.38))

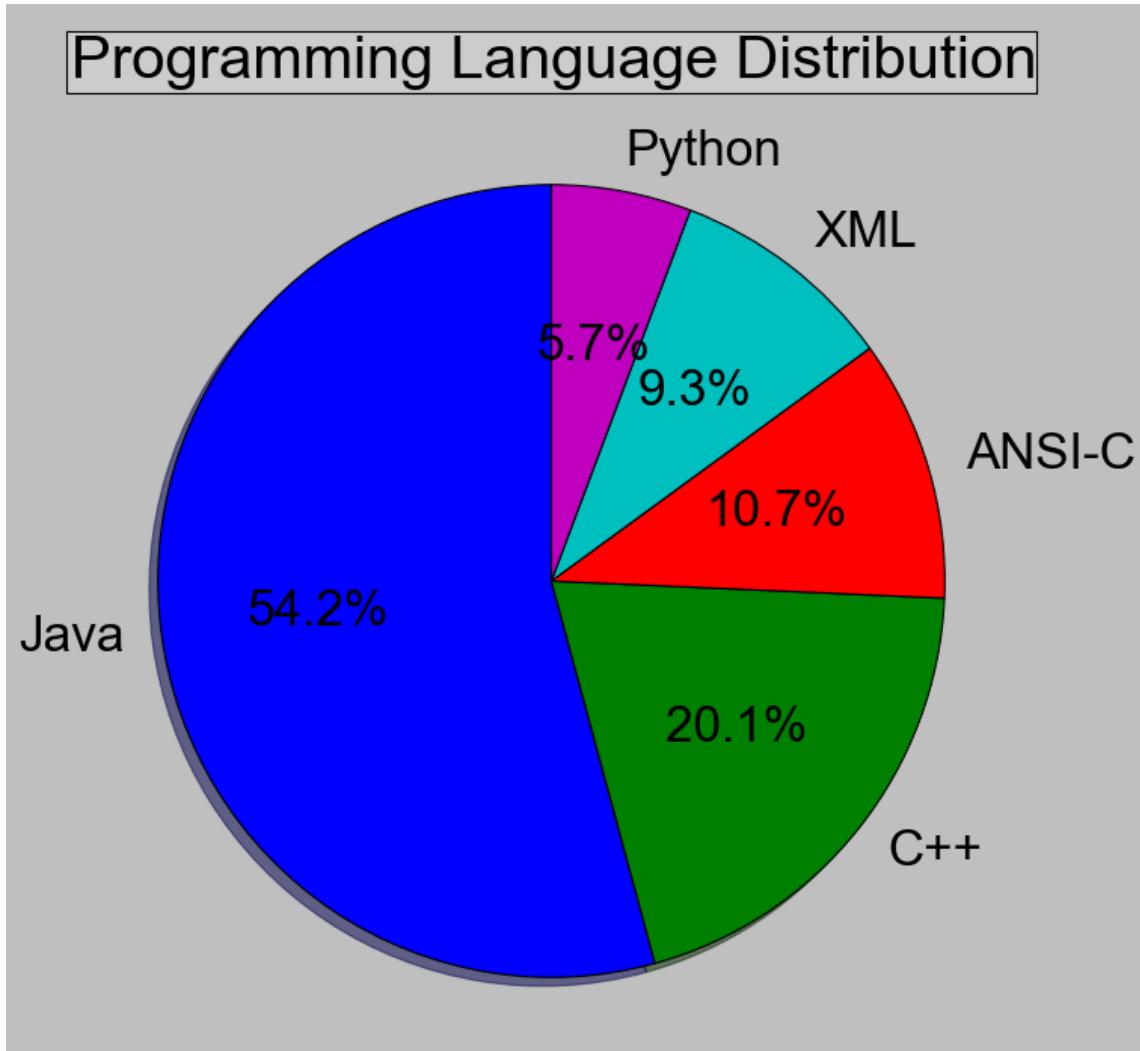


Figure 7.3: OpenDaylight Programming Language Distribution

Estimated Average Number of Developers = 89.39

Total Estimated Cost to Develop = \$ 69,259,253

(average salary = \$56,286/year, overhead = 2.40).

SLOCCount, Copyright (C) 2001-2004 David A. Wheeler

Taking the previous data into consideration, some graphics can be created to show the different measures performed ¹ on a more representative way. To start with, Figure 7.3 shows the distribution percentage from programming languages perspective:

As shown before, **Java, C++ and ANSI-C** are predominant programming languages used in-

¹Software Source Code retrieval performed on October 4th, 2014

side OpenDaylight project. 85% of the code is written in one of these programming languages, while **15% of the remaining code is written in well-known programming languages, such as Python or XML²**. It must be highlighted that **C, Java and C++ are among the most used five programming languages in 2014, while Python holds the eighth place**, according to the TIOBE Index [Sof14].

It is also interesting to classify the different existing projects distribution taking into account Lines Of Code (LOC). First, with a very simple calculation of the total lines of code in the project, it results on OpenDaylight project to have a **total of 1764049 LOCs**. This way, calculation of the percentage of the total code used by each project can be performed easily. Table 7.2 summarizes the output of the previous command.

Table 7.2 has been obtained by using some advance text processing command, such as gawk [Sys89], and dumping the output of the lines of code distribution to a file (i.e.:loc.txt), it is quite easy to calculate percent of lines of code used by each of the subprojects, as shows next command:

```
$ cat loc.txt | grep ^[0-9] | awk {'print $1'} |  
while read line; do echo "scale=2; ($line*100)/1764049" | bc
```

Data retrieved can be plotted in a more representative way through a percent distribution Pie chart, as shown in Figure 7.4.

7.4 Release Roadmap

Up to date, only two official releases of OpenDaylight have been released. Releases were named “Hydrogen” and “Helium”. Initial version, called “Hydrogen”, was released February 4th, 2014. This release was a first try by OpenDaylight group to achieve a SDN controller that could act as SDN/NFV controller. To do, not all the projects already available were included, but rather some of the most important which were, in turn, with such a maturity level to be included in a official release. The table in Figure 7.5 shows the modules in the first releases:

²XML is rather a markup language, but has been considered a programming language to homogenize data dumped by SLOCCount tool

Project	LOCs	LOCs percentage
vtn	661838	37.51%
controller	289437	16.40%
oscp	142681	8.08%
opendove	99718	5.65%
yangtools	82770	4.69%
defense4all	59026	3.34%
bgpcep	55431	3.14%
openflowplugin	47232	2.67%
sdninterfaceapp	37232	2.11%
ovsdb	33577	1.90%
dlux	32348	1.83%
openflowjava	27003	1.53%
snbi	23443	1.32%
oflex	22672	1.28%
snmp4sdn	20032	1.13%
groupbasedpolicy	16138	0.91%
packetcable	15627	0.88%
integration	15436	0.87%
lispflowmapping	14479	0.82%
aaa	11415	0.64%
toolkit	11311	0.64%
l2switch	11167	0.63%
sfc	9567	0.54%
docs	8375	0.47%
affinity	8206	0.46%
reservation	1752	0.09%
ttp	1567	0.08%

Table 7.2: OpenDaylight projects' LOC distribution

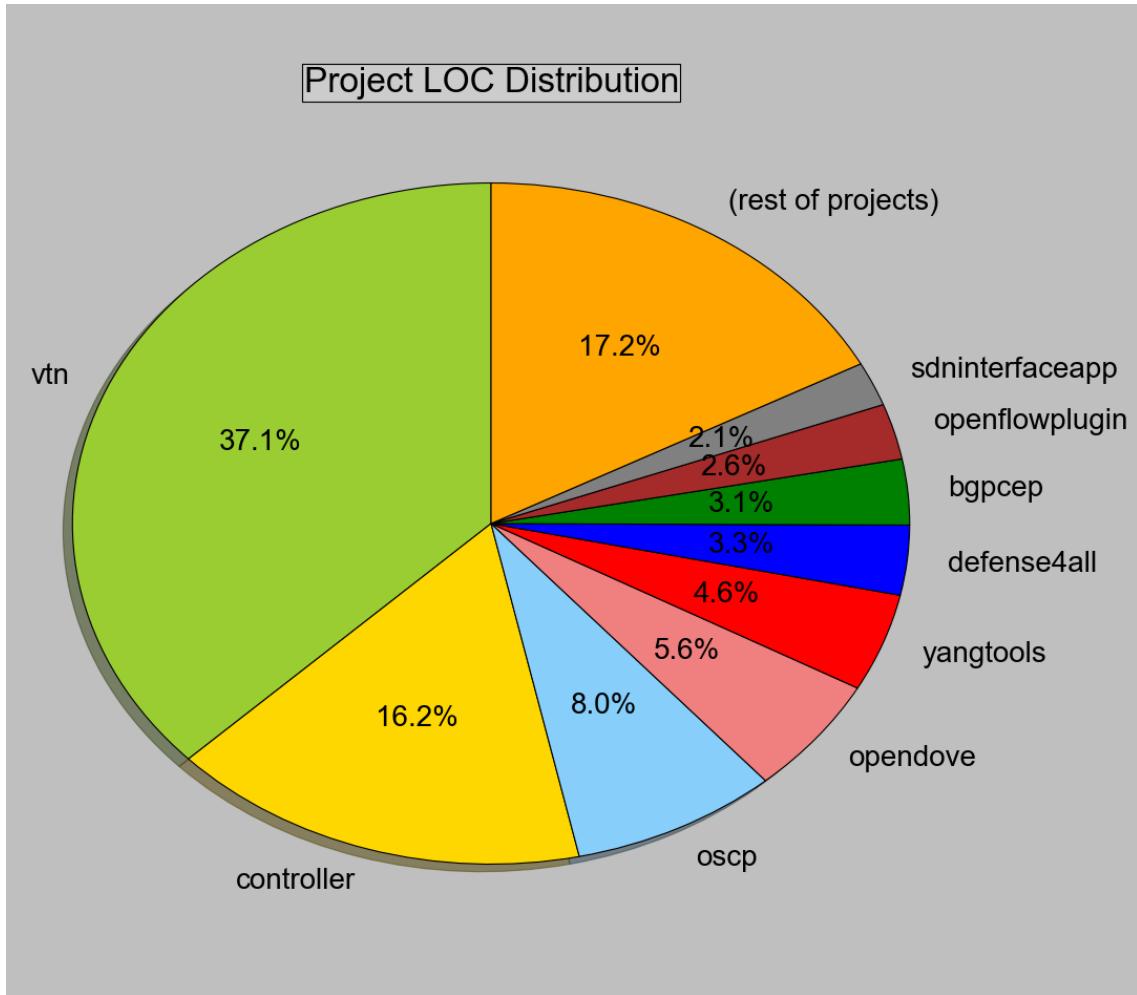


Figure 7.4: OpenDaylight LOC based Project Distribution

Meanwhile, “Helium”, the second final version of OpenDaylight, which was released September 29th, 2014, brought up to ten improvements and new features. Among them, next ones are remarkable:

1. **New SB plugins.** This way, enhanced communication to underlying hardware was introduced. Among the most important protocols introduced, next ones are to highlight:
 - OpenFlow TTP
 - PCMM
 - SNBI
2. **New features and services.** Among new features and services introduced, next ones are found:

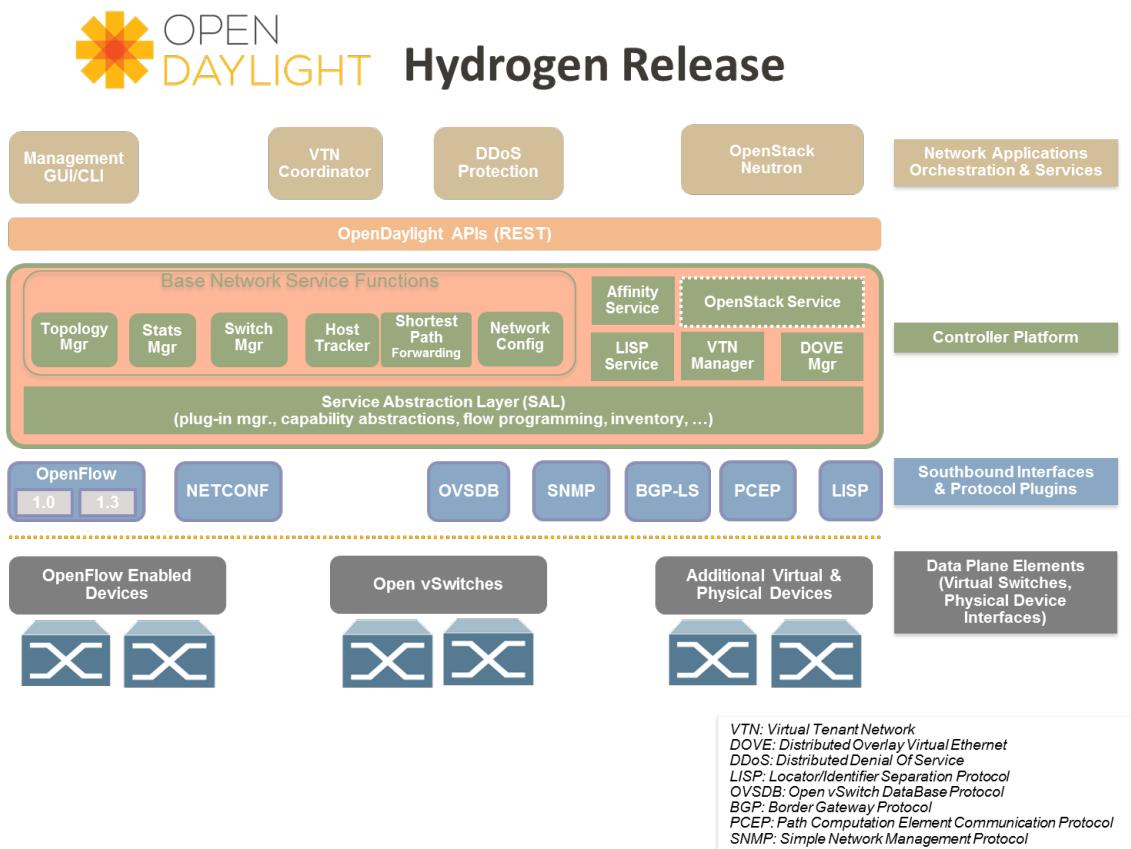


Figure 7.5: OpenDaylight Hydrogen Architecture Diagram

- AAA
- DOCSIS abstraction
- SFC
- L2 Switch
- OVSDB Neutron

3. **New NB API.** Last but not least, new NB API has also been added:

- AuthN API

Figure 7.6 shows information about new features, services and APIs that have been incorporated in “Helium” release.

There is no available information about next releases to be considered. However, it is important to know about **Project Life Cycle** [Lif14] inside OpenDaylight. How are new projects introduced? Can anybody propose a project? What are steps and times needed for a project to be

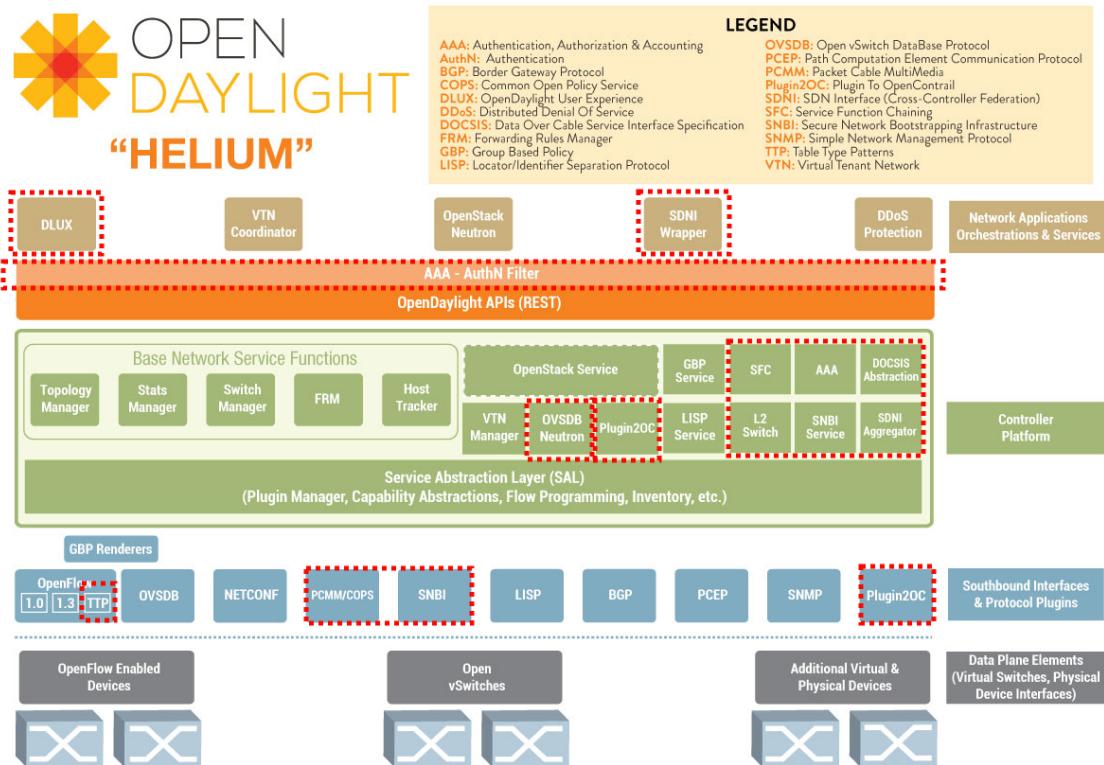


Figure 7.6: OpenDaylight Helium New Features Diagram

included in an official release?

The most important aspect regarding new projects proposal is that **everybody can propose a new project**. Obviously, it has to be a project that results valid to the OpenDaylight project, and must be related to the technologies and features provided by it.

Figure 7.7 shows the steps followed by a new project from the time it is proposed until it is included in the Core or Top Level projects, or it is Archived due to decisions taken inside the Community to make that project out of the scope of OpenDaylight.

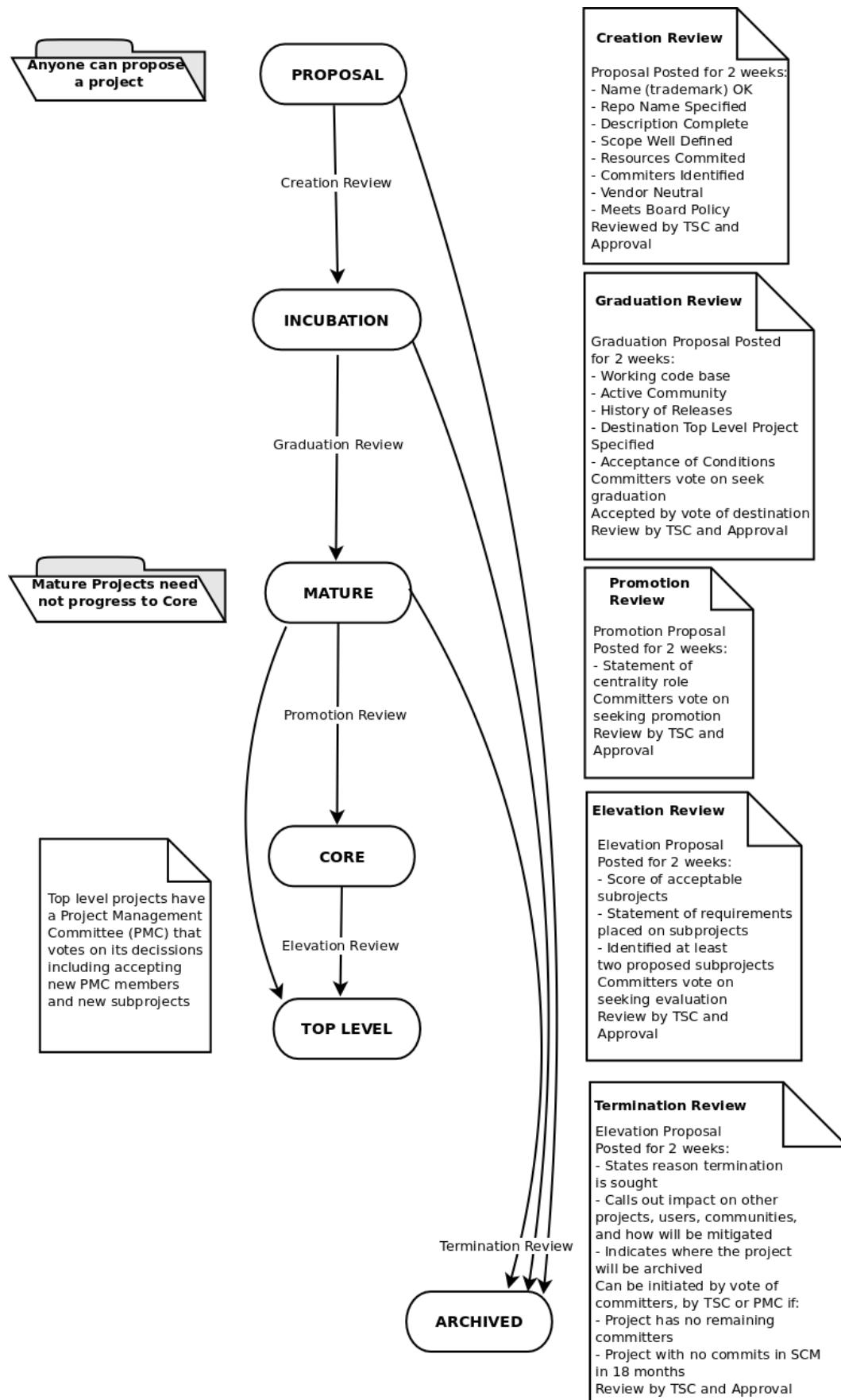


Figure 7.7: OpenDaylight Projects Life Cycle

Chapter 8

OpenDaylight: Project Evaluation

On this chapter, an analysis of the most representative metrics of the OpenDaylight project will be performed. Metrics of a particular Open Source project can help to determine its state of art, its maturity, its community involvement and also to perform the evaluation of the project's quality, above all if the project can be compared to other Open Source project existing on the same category. In the end, success for an Open Source project depends, among other factors, on its community involvement, and community involvement can be measured taking into account the **activity** of its members.

However, OpenDaylight project, being a recently created project, with a short path traversed, is in turn the only Open Source project related to SDN/NFV technology. From this perspective, metrics analyzed will focus rather on how the project is progressing after its foundation, which particularities are observed on particular dates (e.g.: major version releases), the activity observed from the community perspective (e.g.: code commits, Bug resolution, Mail Lists activity, etc.) and, in general, any metric that can help on profiling the community involvement. It is also important to analyze the “Onion Model” [CH14] that characterizes the project, in order to classify the proportion between Release Coordinators/Core Developers, Regular Developers and Occasional Developers. According to this model, approximately 80 percent of the changes are proposed by 20 percent of the top committers, and the other way round. An analysis of the proportion of the number of total committers that are needed to contribute 80% of the total number of commits will be analyzed, to identify to which degree OpenDaylight project follows this kind of model.

Another important comparison for an Open Source project is to analyze differences in previously described metrics between releases. How much code was contributed on last release, how much was contributed to the previous release, which are the most relevant differences between releases, if any.

Before accomplishing the analysis of the metrics, this chapter will describe, on the one hand, which metrics are to be calculated, and the reason to do so. On the other hand, a description of the tools and data sources used to perform the metrics obtaining. Later, metrics obtained will be explained and analyzed. Last, but not least, a short summary together with the conclusions of the metrics analyzed.

8.1 Metrics

Different metrics will be obtained to measure the activity of its members, associated to the different issues that an Open Source project faces. Basically, an Open Source project activity can be determined by:

- **Source Code commits and reviewing.**
- **Bug Resolution.**
- **Mail List activity.**
- **Q&A/Forum activity.**

Each of the previous areas can help to identify different metrics that can be determined on each of the areas:

- Source Code commits and reviewing. No doubt, source code commits can help to determine a lot of different aspects associated to demographics of the community, among which the following stand out:
 1. Most active committers and institutions.
 2. Average age¹ of committers in the project.

¹Taking into account *age* as period since a committer performed the first code contribution to the project

3. New vs. retained developers.
- Bug Resolution. Bug resolution metrics are a good way of understanding the responsiveness of a community to the issues that appear on the project. Next measures are remarkable related to this fact:
 1. Resolved vs. Remaining Bugs Distribution.
 2. Most active bug resolvers.
 3. BMI (Backlog Management Index) Progression: : $BMI = (\text{CLOSED BUGS}/\text{OPEN BUGS}) * 100$.
 - Mail List activity. Mail Lists provide a good measure of the activity and discussions going on in the project, as well as a good thermometer to compare activity between releases, with measures such as:
 1. Number of mails sent.
 2. Mail List participants.
 3. Most active mail lists and participants.
 - Q&A/Forum activity. This kind of Q&A based tools allow to perform certain measures that provide information on activity related to support inside the project, with measures such as:
 1. Number of questions asked and answered.
 2. Users asking and answering questions.

The previous kind of metrics will help to determine the activity of different aspects of OpenDaylight Open Source project, and will also help determining the differences between up to date releases, i.e., Hydrogen and Helium.

8.2 Tools and Data Sources

Once metrics have been identified, it is worth explaining the different methods accomplished to retrieve metrics of OpenDaylight project. On the one hand, a set of **tools** have been used to perform data-mining of the project. The collection of tools used to retrieve data were basically

those hosted under the MetricsGrimoire [Car14a] toolset collection. Among all of the toolset, next ones have been used:

1. **CVSanaly** [Car14c]. This tool, written in Python, allows obtaining and analyzing a particular software among different kind of them (CVS, as well as Subversion and GIT) and stores analyzed information on a SQL Database. This information can be retrieved later from the Database in order to measure, for example, aspects such as:
 - Number of commits in a period of time, or a particular time or date.
 - Most active committers.
 - Territoriality of a project, by calculation of different people modifying a particular file.
 - And, in general, all the information that can be retrieved from that particular Software Configuration version system.
2. **Bicho** [Car14b]. This tool, written in Python, retrieves and organizes information from issue tracking systems such as Bugzilla, which is the one used inside OpenDaylight Open Source project, but also from Jira, SourceForge, Allura, GitHub, Google Code, Launchpad or Redmine. This table allows obtaining metrics such as:
 - Bugs Opened.
 - Bugs Closed.
 - Resolution time.
 - Most active bug closers.
3. **MailingListStats** [Car14d]. This tool, written in Python, retrieves and organizes information from mailman based mail lists, which is the mailing system used in OpenDaylight. This information can help on data mining information such as:
 - Mails sent in a particular period of time.
 - Mails answered in a particular period of time.
 - Top people sending mails to the mail lists.
 - Top people answering mails in the mail lists.

4. **Siby1** [Bit14]. This tool, written in Python, retrieves and organizes information from question and answer sites based on the Ask.org framework, which is the system used in OpenDaylight. This information can help on data mining information such as:

- Questions proposed in a particular time range.
- Answers performed in a particular time range.
- Top people sending questions to the Q&A system.
- Top people answering questions to the Q&A system.

Meanwhile, on the other hand, the previous tools have been used to retrieve data from different OpenDaylight project data sources. In particular, OpenDaylight **data sources** that have been used to retrieve information where the next ones:

- **OpenDaylight GIT repositories**. As stated previously on table 7.1, different existing OpenDaylight Open Source project's GIT repositories have been used to determine metrics associated to Source Code.
- **OpenDaylight mail lists** [Lis14]. As described in table E.1, different existing OpenDaylight Open Source project's mail lists have been used to determine metrics associated to activity on them.
- **OpenDaylight Ask** [Ask14]. OpenDaylight's Q&A system, based on AskBot [TM14a], is used as well to determine activity regarding Q&A.
- **OpenDaylight Bug Tracker** [Bug14b]. OpenDaylight's Bug Tracker system, based on BugZilla, has also been used as data source to provide to Bicho, and obtain metrics related to Issues resolution.

8.3 Metrics Analysis

Once tools to be used and data sources have been detailed, metrics obtained on each of the different areas described on previous section will be explained and analyzed, together with the method used to follow it.

8.3.1 Source Code Metrics

This subsection summarizes the demographics associated to OpenDaylight Open Source project. By studying different metrics, some of the most relevant facts associated to the community, such as top committers, top committing companies, committers involvement to identify if the project follows the “Onion Model” or projects with a higher activity are examples of metrics that can help on identifying activity or community involvement among other aspects.

In order to parse source code metrics, CVSanalyY [Car14c] tool has been installed in order to be used to obtain metrics associated to Source Code metrics. After cloning all repositories of OpenDaylight project 7.1, and installation of MySQL server, creation of a Database to store CVSanalyY parsing information has been performed through next SQL commands:

```
mysql> CREATE DATABASE odl CHARACTER SET utf8
COLLATE utf8_unicode_ci;
```

After Database creation, CVSAnaly2 tool has been executed, providing information of the Database (name, username and password) and directories where repositories have been cloned, in order to perform the data mining to the SQL Database. Assuming all projects are cloned on a particular directory, executing CVSAnaly2 tool indicating all the sub-directories containing the different OpenDaylight projects can be done as described below²:

```
$ for i in *;
> do CVSAnaly2 -fdbuser -fdbpwd -dodl $i;
> done
```

After the previous command, data is parsed and downloaded to the Database. Available tables after performing information retrieval are next ones:

```
mysql> show tables;
+-----+
```

²Software Source Code retrieval performed on October 4th, 2014.

```

| Tables_in_odl      |
+-----+
| action_files      |
| actions           |
| actions_file_names|
| branches          |
| commit_graph      |
| file_copies       |
| file_links        |
| files              |
| people             |
| repositories       |
| scmlog             |
| tag_revisions      |
| tags               |
+-----+

```

A complete reference of the Database schema is available in [Met14b]. Once information is retrieved, collected and organized into the different tables, the most representative metrics can be analyzed, considered most representative from the community perspective.

First data to be retrieved from Source Code data is the **top committers from both people and company perspective**. To identify top 20 committers, next SQL statement can be performed:

```

mysql> select p.name, count(distinct(s.id)) as "Number of commits"
from people p, scmlog s where p.id=s.author_id group by p.name
order by count(distinct(s.id)) desc LIMIT 20;

```

The previous SQL command show next table, identifying **top 20 committers** in the project:

Name	Number of commits
Robert Varga	1951
Tony Tkacik	1361
Ed Warnicke	1007
Dana Kutenicsova	537
Madhu Venugopal	497
Martin Vitez	416
Giovanni Meo	409
Alessandro Boch	272
Shigeru Yasuda	247
Maros Marsalek	230
Moiz Raja	207
Michal Rehak	202
Luis Gomez	180
Michal Polkorab	174
Milos Fabian	168
Tomas Olvecky	158
Ryan Moats	153
David Goldberg	151
Mathieu Lemay	132
Dave Tucker	122

Table 8.1: OpenDaylight top 20 committers

Meanwhile, top companies contributing to the project can be obtained as well. Although **Cisco** seems to stand out over the rest of companies, it is also important to perform a classification of the **top 20 contributing companies**. In order to identify each of the companies, mail account server has been used. Taking the previous factors into account, next SQL statement can be used to obtain top contributing companies:

```
mysql> select SUBSTRING_INDEX(p.email, '@', -1) as "Company
Commits", count(distinct(s.id)) as "Number of commits" from
people p, scmlog s where p.id=s.author_id group by
SUBSTRING_INDEX(p.email, '@', -1) order by
count(distinct(s.id)) desc LIMIT 20;
```

Assuming “gmail.com”, “yahoo.com” or other user mail addresses correspond to non company users, and merging some of the results corresponding to different domains of a same company (for instance, us.ibm.com/in.ibm.com), the results obtained on a per company basis is as follows:

Company Commits	Number of commits
cisco.com	8349
non-company contributors	1060
redhat.com	323
nec.com	307
ibm.com	306
ericsson.com	302
pantheon.sk	282
contextream.com	192
inocybe.com	181
plexxi.com	172
brocade.com	156
hcl.com	146
itri.org.tw	118
.opendaylight.org	96
colindixon.com	81
daylight6.linux-foundation.org	78
hp.com	72
noironetworks.com	49
linuxfoundation.org	37
readams.net	37
necam.com	29

Table 8.2: OpenDaylight top 20 committer companies

In order to identify the “Onion Model” that this community follows, it must be clarified that those considered as **Core developers** are those in charge of **80%** of the total commits. Meanwhile, **Regular developers** are responsible of **80% to 95%** of the code changes. Meanwhile, **Occasional Committers** are the ones in charge of the rest **5% of the commits**.

Taking into account that, within existing data, the total number of commits, as well as the total number of committers, can be obtained with next SQL statements:

```
mysql> select count(distinct(s.id)) as "Number of commits"
from scmlog as s LIMIT 10;
+-----+
| Number of commits |
+-----+
|          12587 |
+-----+  
  
mysql> select count(distinct(name)) as "Committers" from people;
+-----+
| Committers |
+-----+
|         286 |
+-----+
```

So, with up to **12587 total commits** and a total of **286 distinct committers**, in order to identify core developers proportion, the number of top committers that have uploaded up to **80%** of the commits, that is, **10070** commits, must be calculated. By using SQL statements, limiting the number of commits to a particular number of top committers, it is found that to perform that number of commits up to **38 committers** are needed:

```
mysql> select SUM(commits) as "Number of Commits" FROM
(select p.name, count(distinct(s.id)) as "commits"
from people p, scmlog s where p.id=s.author_id group by
p.name order by count(distinct(s.id)) desc LIMIT 37) AS commits;
+-----+
| Number of Commits |
+-----+
```

```
|          10048 |
+-----+
mysql> select SUM(commits) as "Number of Commits" FROM
(select p.name, count(distinct(s.id)) as "commits"
from people p, scmlog s where p.id=s.author_id group by
p.name order by count(distinct(s.id)) desc LIMIT 38) AS commits;
+-----+
| Number of Commits |
+-----+
|          10104 |
+-----+
```

With the previous information, it can be ensured that **top 38 committers** are in charge of the top 80% of the commits. This means **13% of the total committers** number. Meanwhile, the number of occasional committers that contribute the last **5% of the total commits**. In order to do so, it must be identified, among people contributing to a lesser way, how many contributors are needed to obtain **630 commits**. That can be done with next SQL statements:

```
mysql> select SUM(commits) as "Number of Commits" FROM
(select p.name, count(distinct(s.id)) as "commits" from
people p, scmlog s where p.id=s.author_id group by p.name
order by count(distinct(s.id)) LIMIT 167) AS commits;
+-----+
| Number of Commits |
+-----+
|          628 |
+-----+
```

```
mysql> select SUM(commits) as "Number of Commits" FROM
(select p.name, count(distinct(s.id)) as "commits" from
```

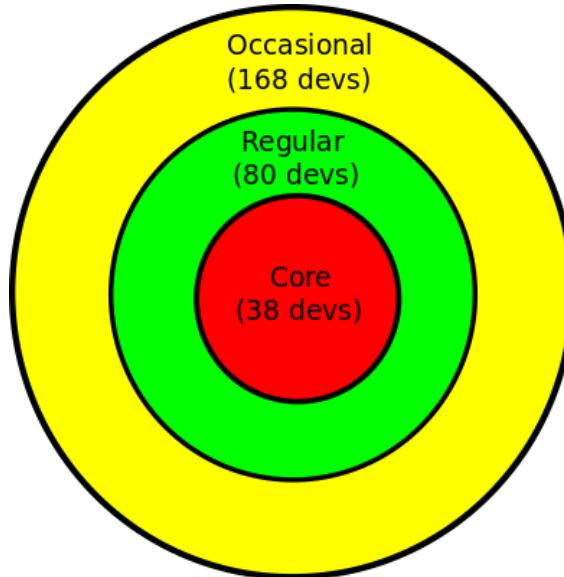


Figure 8.1: OpenDaylight Project Onion Model

```
people p, scmlog s where p.id=s.author_id group by p.name
order by count(distinct(s.id)) LIMIT 168) AS commits;
+-----+
| Number of Commits |
+-----+
|          639 |
+-----+
```

With the previous data, it can be asserted that up to **168 occasional committers** are needed to provide the last 5% of the total commits, while the rest of the commits is provided by the remaining **80 regular committers**. With the previous information, the “Onion Model” representing OpenDaylight project remains as given in Figure 8.1.

As final conclusion, it has been demonstrated that OpenDaylight is not a exception, and follows the Onion Model to a higher scale, as **13% of the total number of committers are responsible of the 80% of the commits up to date**.

Another important metrics to be considered regarding source code commits has to do with the ammount of commits that appear near release dates. Considering release dates, February 4th and September 29th for existing releases [7.4](#), Hydrogen and Helium respectively, the amount of

commits can be analyzed for previous dates (one month, for instance), to compare to existing mean per month of commits previous to that last release month. On the one hand, it must be considered that first commits appear on March 22th, 2013.

To find out the average commits per month, it can be considered that initial moments of the project also have a big amount of commits, so later dates will be considered. For instance, considering last 2013 six months commits, the number of commits can be obtained with next SQL statement:

```
mysql> select COUNT(id) from scmlog  
WHERE date BETWEEN '2013-06-01' AND '2013-12-01';  
+-----+  
| COUNT(id) |  
+-----+  
| 3031 |  
+-----+
```

3031 commits in six months shows a mean of approximately **500 commits per month**. However, the number of commits on last month before OpenDaylight Hydrogen release, shows a remarkable increase on the number of commits:

```
mysql> select COUNT(id) from scmlog  
WHERE date BETWEEN '2014-01-01' AND '2014-02-01';  
+-----+  
| COUNT(id) |  
+-----+  
| 1360 |  
+-----+
```

With the previous data it can be demonstrated that **the last month before Hydrogen release, number of commits was 2.5 times bigger compared to the mean of the previous six months**.

This same analysis can be applied to study Helium release, whose release date was September 24th, 2014:

```
mysql> select COUNT(id) from scmlog
WHERE date BETWEEN '2014-02-24' AND '2014-08-24';
+-----+
| COUNT(id) |
+-----+
|      4799 |
+-----+
```

4799 commits in six months shows a mean of approximately **800 commits per month**. However, the number of commits on last month before OpenDaylight Helium release, shows a remarkable increase on the number of commits:

```
mysql> select COUNT(id) from scmlog
WHERE date BETWEEN '2014-08-24' AND '2014-09-24';
+-----+
| COUNT(id) |
+-----+
|      1538 |
+-----+
```

The previous numbers show that **the last month before Helium release, number of commits was almost two times bigger compared to the mean of the previous six months**. With a further processing of each of the commits that occur per month, it can be shown the fact that, the closer the release date is, the highest number of commits occur as can be seen in Figure 8.2.

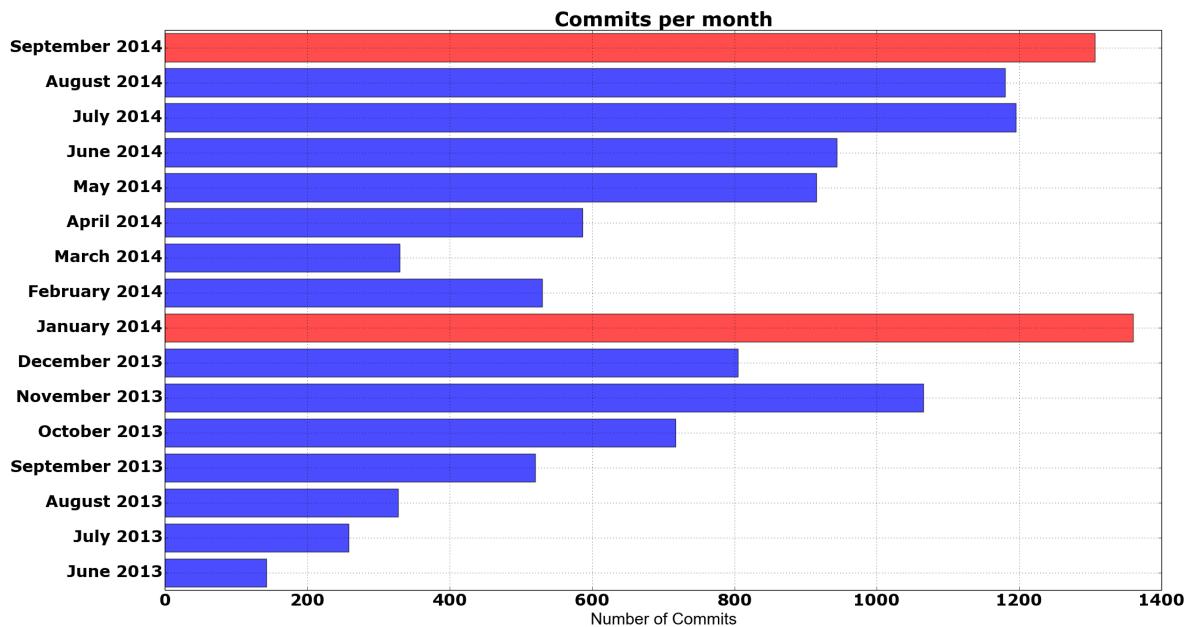


Figure 8.2: OpenDaylight Project Commits Per Month

8.3.2 Mailing List Activity

Analyzing up to date message exchange on OpenDaylight’s project mailing lists can give an idea on the activity of the project according to this particular aspect. Mail Lists is always one of the main communication channels in Open Source project, and OpenDaylight is not an exception.

MailingListStats [Car14d] tool, written in Python, can be used to parse the different messages that have been sent to a particular mail lists, and dump different information to a Database that allow later performing some metrics associated to them, such as the number of messages sent to a particular date, people participating, etc.

The first action to perform is the creation of a Database to retrieve all the information from the mail lists. This can be done with next SQL statement:

```
mysql> CREATE DATABASE odl_maillists CHARACTER SET utf8
COLLATE utf8_unicode_ci;
```

In order to use the tool, the first items to be found out are the URLs containing the archives of the mail lists used. To do so, archive URLs of all of the lists described in Appendix E have been identified and dumped to a file. To do so, OpenDaylight Mail List [Lis14] page's HTML has been parsed to obtain different mail list archive URLs and dumped to file:

```
$ for MAIL_LIST in
$(wget -O - https://lists.opendaylight.org/mailman/listinfo
| grep listinfo | sed -e 's/<[^>]*>//g');
do echo "https://lists.opendaylight.org/pipermail/\${MAIL_LIST}";
done >> mail_archive.txt
```

Iterating each of the lines within that file, **mlstats** tool has been used to dump all the mail lists archives to the Database:

```
$ for archive in $(cat mail_archive.txt);
do mlstats --db-user=dbuser --db-password=dbpasswd
--db-name=odl_maillists ${archive}; done
```

Available tables after performing information retrieval are next ones:

```
mysql> show tables;
+-----+
| Tables_in_odl_maillists |
+-----+
| compressed_files          |
| mailing_lists              |
| mailing_lists_people       |
| messages                   |
| messages_people            |
| people                     |
+-----+
```

```
+-----+
```

A complete reference of the Database schema is available in [Met14c]. The previous tables will contain the processed information of the mail lists retrieved from the tool. SQL statements can be used in order to perform the different metrics obtaining:

```
mysql> select COUNT(distinct(message_id)) as
  "Number of messages" from messages_people;
+-----+
| Number of messages |
+-----+
|          151285   |
+-----+
```

The previous command will allow dumping the total of the archives to the Database³. Once that is done, some metrics can be obtained from the Database. To start with, up to date messages sent to all of the different OpenDaylight mail lists can be obtained with next SQL statement:

```
mysql> select m.mailing_list_url, COUNT(DISTINCT(message_id))
  AS "Messages" FROM messages as m GROUP by mailing_list_url
  ORDER by COUNT(DISTINCT(message_id)) DESC LIMIT 20;
```

The previous command will show the 20 most active mail lists up to date. Gerrit and Jenkins mail lists are the top most used mail lists. Regarding projects, Controller, as the core component, is also an active project regarding mail list message exchange as it can be seen in Table 8.4. Meanwhile, total number of users involved in all mail lists can be obtained through next SQL statement:

```
mysql> select COUNT(distinct(email_address))
  as "Users" from messages_people;
```

³Mail Lists Archives have been retrieved on October 26th, 2014

Mail List	Number of Messages
controller-gerrit	36463
yangtools-gerrit	14263
bgpcep-gerrit	13287
controller-jenkins	8954
jenkins	8645
bugs	8051
integration-jenkins	7781
controller-dev	7061
openflowplugin-gerrit	6527
controller-bugs	6223
ovsdb-gerrit	6045
integration-gerrit	5798
vtn-gerrit	3390
openflowplugin-jenkins	3281
openflowjava-gerrit	3118
lispflowmapping-gerrit	2937
openflowplugin-dev	2097
integration-dev	1925
yangtools-jenkins	1916
snmp4sdn-gerrit	1357

Table 8.3: OpenDaylight top 20 most active mail lists

```
+-----+
| Users |
+-----+
|     873 |
+-----+
```

Another interesting classification is the activity of each of the mail lists. This can be obtained with next SQL statement:

```
mysql> select m.mailng_list_url as "Mail List",
```

```
COUNT(DISTINCT(email_address)) AS "Users Involved"
FROM messages_people as m GROUP by mailing_list_url
ORDER by COUNT(DISTINCT(email_address)) DESC LIMIT 20;
```

Opposite to the number of mail messages, it seems that “dev” mail lists, as well as “users” mail lists to a lesser extent, are the ones with most users involved. This seems reasonable, as Gerrit/Jenkins mail lists are more related to automatic mail lists sending information, normally from Gerrit/Jenkins users, with automatic resulting test/integration information.

Mail List	Users Involved
controller-dev	527
openflowplugin-dev	176
controller-users	156
integration-dev	147
ovsdb-dev	127
vtn-dev	101
.opendaylight-users	81
yangtools-dev	76
release	71
project-proposals	70
infrastructure	69
openflowjava-dev	67
groupbasedpolicy-dev	58
dev	54
bgpcep-dev	41
bgpcep-users	38
lispflowmapping-dev	33
affinity-dev	32
dlux-dev	32
opendove-dev	31

Table 8.4: OpenDaylight top 20 mail lists with the most users involved

To finish the analysis of the metrics related to mail lists, it is important to analyze the impact on a new release in the mailing list system, and if there is an impact on the number of mails sent. Considering release dates, February 4th and September 29th for existing releases [7.4](#), Hydrogen

and Helium respectively, the amount of mail list messages can be analyzed for previous dates (messages sent to the mail lists on last month, for instance), to compare to existing mean per month of commits previous to that last release month. On the one hand, it must be considered that first mails appear on March 22th, 2013.

To find out the average commits per month, it can be considered that initial moments of the project also have a big amount of messages, so later dates will be considered. For instance, Considering last 2013 six months mail list activity, the number of mails can be obtained with next SQL statement:

```
mysql> select COUNT(message_id) as "Number of messages"
from messages WHERE arrival_date
BETWEEN '2013-06-01' AND '2013-12-01';
+-----+
| Number of messages |
+-----+
| 29479 |
+-----+
```

This number shows that there is, approximately, a mean number of **4913 messages per month** before Hydrogen release. However, on Hydrogen release date, the number of message increased dramatically, as follows:

```
mysql> select COUNT(message_id) from messages
WHERE arrival_date BETWEEN '2014-01-01' AND '2014-02-01';
+-----+
| COUNT(message_id) |
+-----+
| 13168 |
+-----+
```

As shown previously, there is a high increase on the number of mails messages sent to the different mail lists. **Number of mails exchanged before the Hydrogen release multiplied by more than 2.5**, compared to the mean of mails sent on previous months. This behavior can be also analyzed on Helium release:

```
mysql> select COUNT(message_id) as "Number of Messages"
from messages WHERE arrival_date BETWEEN '2014-02-24'
AND '2014-08-24';
+-----+
| Number of Messages |
+-----+
| 73591 |
+-----+
```

73591 messages in six months shows a mean of approximately **12265 commits per month**. However, the number of commits on last month before OpenDaylight Helium release, shows a remarkable increase on the number of mails sent:

```
mysql> select COUNT(message_id) as "Number of Messages"
from messages WHERE arrival_date BETWEEN '2014-08-24'
AND '2014-09-24';
+-----+
| Number of Messages |
+-----+
| 22850 |
+-----+
```

Previous numbers show that **the last month before Helium release, number of mails exchanged on lists was almost two times bigger compared to the mean of the previous six months**. With a further processing of each of the number of mails that occur per month, it can be shown the fact that, the nearest the release date is, the highest number of mail messages is

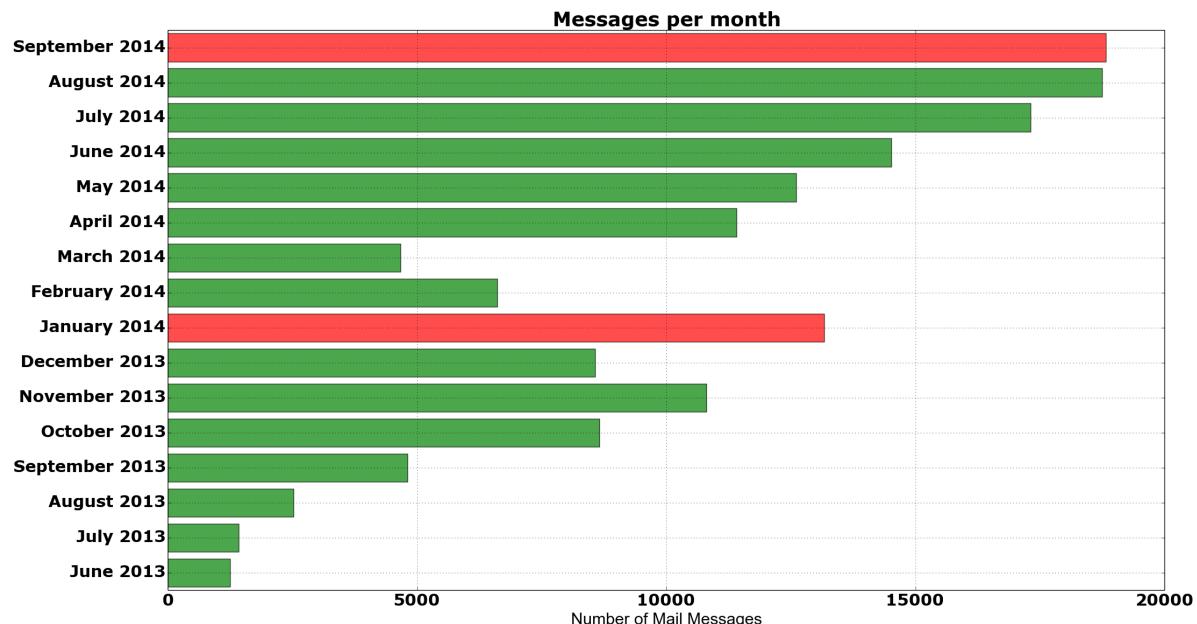


Figure 8.3: OpenDaylight Project Mail Messages Per Month

sent. Figure 8.3 provides further details.

To summarize mail list analysis, it can be ensured that, as release dates approaches, the activity around mail lists increases dramatically. In the case of mail lists, the number of mails sent to mail list increased on last month dramatically for Hydrogen release. However, regarding Helium release, the number increased for each of the last three months, with a higher increase in both July and August, and not so on September, where the number was very similar to August. This is somehow reasonable, as the release date was September, 23th.

This section has shown how mail system can be analyzed to take some important metrics having to do with support aspects existing on a particular Open Source project. Metrics such as total number of mails sent or the different mail lists activity can determine how sane is the communication mechanism used.

8.3.3 Q&A Activity

Another important source of the activity of a project is the Q&A forum. Fortunately, among powerful tools found in MetricsGrimoire [Car14a] toolset collection, **Sibyl** [Bit14] allows performing data-mining of OpenDaylight’s Q&A system, based on AskBot [TM14a].

The first action to perform is the creation of a Database to retrieve all the information from the Q&A web page. This can be done with next SQL statement:

```
mysql> CREATE DATABASE odl_ask CHARACTER SET utf8
COLLATE utf8_unicode_ci;
```

Once Database is created, Sibyl tool has to be provided with the Database name, Database username and password, type of Q&A system (up to date, “ab”, standing for AskBot, is the only one supported) and the URL to be parsed. Taking previous considerations into account, the tool can be executed as follows to performed data mining to the Database⁴:

```
$ sibyl.py -t ab -l https://ask.opendaylight.org
-d odl_ask -u dbuser -p dbpasswd
```

Previous command will allow performing some processing of the existing information in OpenDaylight’s Q&A system. Available tables after performing information retrieval are next ones:

```
mysql> show tables;
+-----+
| Tables_in_odl_ask |
+-----+
| answers           |
| comments          |
| people            |
| questions         |
+-----+
```

⁴Q&A Archives have been retrieved on November 1st, 2014

```
| questionstags      |
| tags              |
+-----+
```

Sibyl tool still has not documentation on the Database schema, although with a sort inspection on its tables and their respective columns information can be easily retrieved from each of the available tables. On the one hand, the total amount of questions and answers in the system can be retrieved with next SQL statements:

```
mysql> select COUNT(*) as "Questions" from questions;
+-----+
| Questions |
+-----+
|      303 |
+-----+
```

```
mysql> select COUNT(*) as "Answers" from answers;
+-----+
| Answers |
+-----+
|     269 |
+-----+
```

On the other hand, the total amount of people asking and answering questions in OpenDaylight's project AskBot site:

```
mysql> select COUNT(DISTINCT(author_identifier))
as "People Asking" from questions;
+-----+
| People Asking |
+-----+
|      137 |
+-----+
```

```
+-----+  
  
mysql> select COUNT(DISTINCT(user_identifier))  
as "People Answering" from answers;  
+-----+  
| People Answering |  
+-----+  
| 78 |  
+-----+
```

Last, but not least, some information about the percentage of questions that have been answered can also be obtained. Next SQL statement allow obtaining those questions that have been provided with at least one answer:

```
mysql> select COUNT(*) AS "Answered Questions"  
from questions WHERE answer_count > 0;  
+-----+  
| Answered Questions |  
+-----+  
| 213 |  
+-----+
```

Previous statement allows to ensure that **70%** (213 out of 303) of the questions are answered. Meanwhile, another important rate to obtain is top people asking and answering on the Open-Daylight's Q&A system. Both rates can be obtained with next commands:

```
mysql> select COUNT(q.id) as "Questions Answered",  
p.username from questions as q, people as p  
where p.identifier=q.author_identifier group by  
q.author_identifier order by COUNT(q.id) DESC LIMIT 10;
```

Users	Questions Asked
CobyS	30
csadi	24
chandralekha	14
zedyuang	14
subhash	12
Janki	12
JYOTI111111	10
suba	10
rmohamed	10
milstein	10

Table 8.5: OpenDaylight top 10 people asking on Q&A OpenDaylight project

Previous SQL statements allow to identify to people asking question on Q&A OpenDaylight's project system. Top 10 people regarding this aspect are shown in Table 8.5:

Meanwhile, top 10 most active people answering questions on OpenDaylight project can be obtained with next statement:

```
mysql> select COUNT(a.id) as "Messages Answered",
p.username from answers as a, people as p
where p.identifier=a.user_identifier group by a.user_identifier
order by COUNT(a.id) DESC LIMIT 10;
```

Data obtained for previous statement shows next set of top 10 answering people in Table 8.6.

Meanwhile, Sibyl tool can also be used to obtain top users in terms of reputation, considering reputation as the “Karma” that the user owns in the system, measuring the quality of the responses that users asking questions obtain. This information can be obtained through next command:

```
mysql> select username, reputation from people
order by reputation DESC LIMIT 10;
```

Username	Questions Answered
Madhu	45
Chris O'Shea	40
dfarrell07	36
kanika28	28
Devin Avery	20
Tony Tkacik	14
Robert Varga	13
milstein	10
agrimberg	8
Janki	8

Table 8.6: OpenDaylight top 10 answering people on Q&A OpenDaylight project

Username	Reputation
Madhu	366
Chris O'Shea	171
Tony Tkacik	141
dfarrell07	126
Anil Vishnoi	116
Robert Varga	81
Devin Avery	78
michal.rehak	66
kanika28	63
eclgp	61

Table 8.7: OpenDaylight top 10 people with higher reputation on Q&A OpenDaylight project

The previous statement shows top 10 people with more karma in the system. An important consequence that can be observed is the fact that reputation rank is not directly associated to the number of answers, and opinions of the answers provided also play a role (see Table 8.7). To finish with Q&A answer analysis, it must be studied how different is the number of question compared to the date, and if the dates around release dates have impact on them. To start with, it must be remarked that first messages on the Q&A forum were added on May 2014. This means that no analysis can be performed on Hydrogen release, but rather it can be observed if the number increases in September or October 2014, where Helium release have been performed.

Next figure shows the number of questions in the six months of existence of the Q&A forum, up to Helium release date:

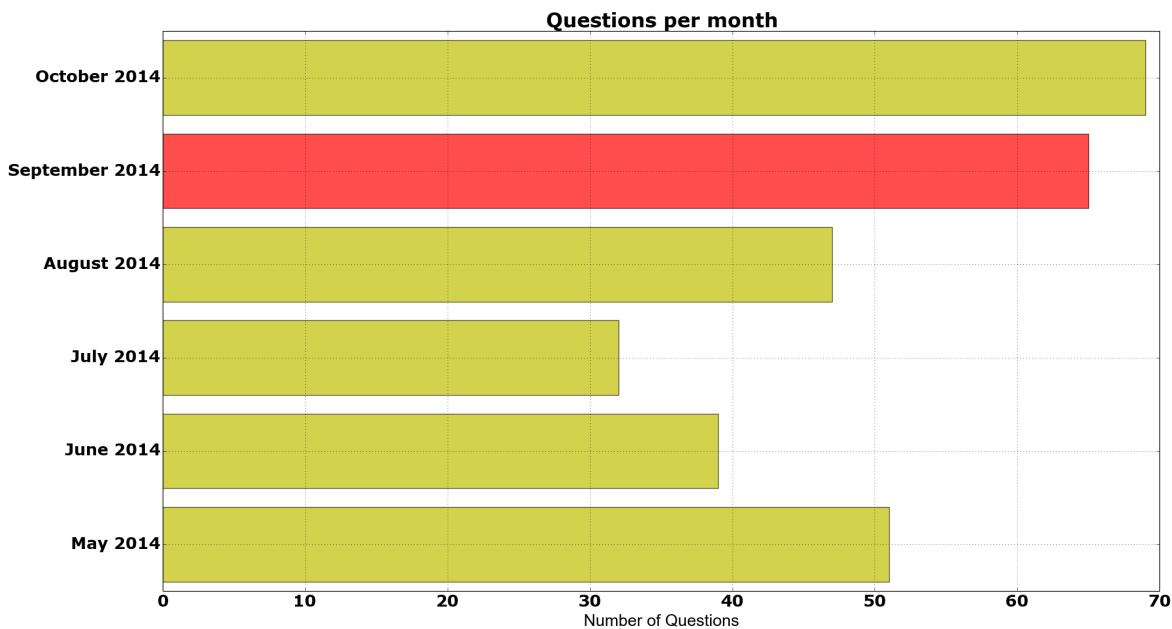


Figure 8.4: OpenDaylight Project Q&A Questions Per Month

Number of answers classified by date can be obtained as well:

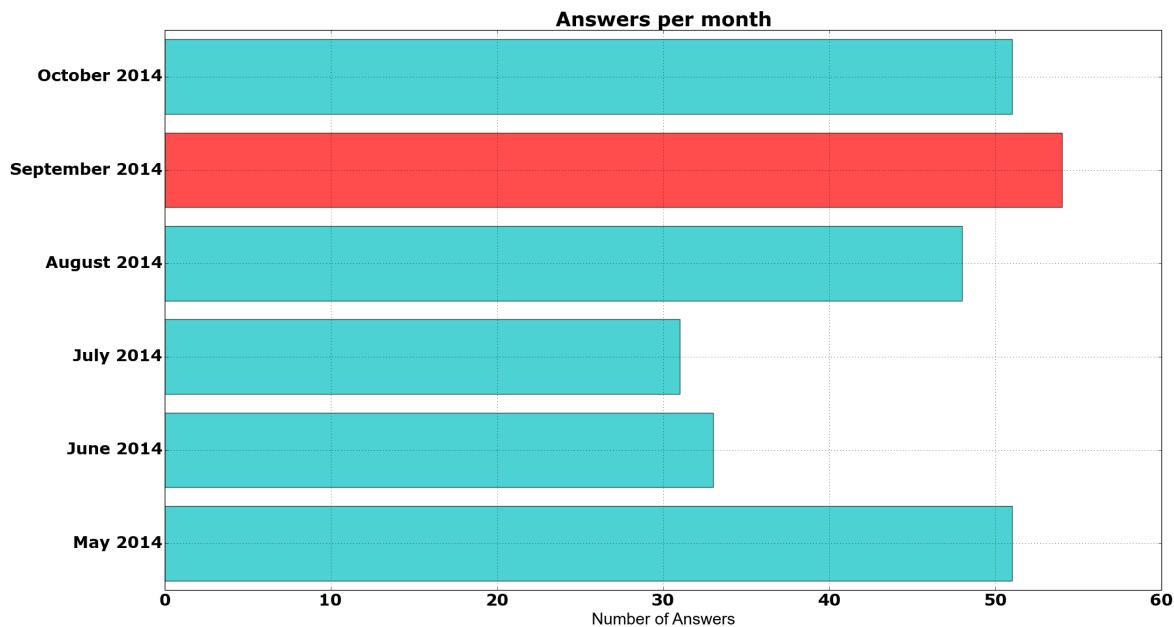


Figure 8.5: OpenDaylight Project Q&A Answers Per Month

As the number of Q&A answers is still very low on the Ask system, no representative information can be extracted from existing data. However, it is important to demonstrate how this system can be parsed in order to analyze future releases impact, Q&A system usage progress and other important stuff.

This section has shown how Q&A system can be analyzed to take some important metrics having to do with support aspects existing on a particular Open Source project. Metrics such as total number of answered questions or its percentage can determine how sane is the support mechanism used. On the other hand, meritocracy can also be measured, so that top people answering or top people with a higher reputation on the system can get recognition they deserve.

8.3.4 Bug discussion and close efficiency

Last, but not least, the OpenDaylight's Bug Tracking system is going to be analyzed. Fortunately, among powerful tools found in MetricsGrimoire [Car14a] toolset collection, **Bicho** [Car14b] allows performing data-mining of Bugzilla [bug14c] based Bug Tracker Systems, as it is the case of OpenDaylight's.

Different metrics regarding Bug Tracking tools allows to perform evaluation of the contributors and core developers, and recognize merits regarding this very important aspect of an Open Source project, as project quality is one of the key aspects for any project's success in general, and for an Open Source project in particular.

It is important to keep a reduced number of Bugs as fast as possible, but it is also important for a community to understand where are the most troubled parts of the project, to perform those actions that are necessary to improve the overall quality of the whole system. But it is also important to understand which is the incoming bug rate, the time that a bug needs to be resolved, release times where incoming bug rate is higher or the amount of bugs that are closed without action or the BMI rate that owns the community, among other metrics.

This chapter will show some examples on how to perform some of the metrics that can be retrieved and analyzed with `Bicho` tool. To start with, as required by the rest of the tools that compose `MetricsGrimoire`, the first action to perform before using `Bicho` tool is the creation of a Database that will collect all the bug information retrieved by this tool:

```
mysql> CREATE DATABASE odl_bugs CHARACTER SET utf8
COLLATE utf8_unicode_ci;
```

Once Database is created, `Bicho` tool can provided with the Database name, username and password, together with the URL of the Bug Tracker System where data will be retrieved from. Apart from that, the backend name must be provided as well, “bg” in this case (what allows indicating Bugzilla backend). Base URL of OpenDaylight’s Bug Tracker is not valid for the tool. Instead, the “Search All” URL has been provided for `Bicho` to work properly⁵:

```
$ bicho --db-user-out=dbuser --db-password-out=dbpwd
--db-database-out=odl_bugs -d 1 -b bg
--backend-user=beuser --backend-password=be-pwd
```

⁵Bug Tracker statistics have been retrieved on November 2nd, 2014

```
-u=https://bugs.opendaylight.org/buglist.cgi?  
bug_status=__all__&query_format=specific
```

Available tables after performing information retrieval are next ones:

```
mysql> show tables;  
+-----+  
| Tables_in_odl_bugs |  
+-----+  
| attachments       |  
| changes           |  
| comments          |  
| issues            |  
| issues_ext_bugzilla |  
| issues_watchers   |  
| people            |  
| related_to        |  
| supported_trackers |  
| trackers          |  
+-----+
```

A complete reference of the Database schema is available in [Met14a]. Taking into account this schema, together with the information that has been parsed, it has been found out that up to **2291** has been opened up to date in the Bug Tracker system:

```
mysql> select count(*) as "Number of Bugs" from issues;  
+-----+  
| Number of Bugs |  
+-----+  
|      2291     |  
+-----+
```

On the other hand, the available status, he total up to date number of bugs for each status and the percentage of each of the types of the status can be obtained as well with next SQL statement:

```
mysql> select count(id) as "Number of Bugs",
       (count(id)/2291)*100 as "%", status as "Status"
  from issues group by status order by count(id) DESC;
+-----+-----+-----+
| Number of Bugs | %      | Status        |
+-----+-----+-----+
|      1720 | 75.0764 | RESOLVED      |
|      438  | 19.1183 | CONFIRMED     |
|       59  | 2.5753  | VERIFIED      |
|       49  | 2.1388  | WAITING_FOR REVIEW |
|       25  | 1.0912  | IN_PROGRESS   |
+-----+-----+-----+
```

The previous data show how there is an important number of bugs which has been resolved and verified, with only a percentage of **20.2%** of the Bugs being in Confirmed/Waiting for Review/In Progress state. However, there is a huge number of Resolved bugs that still are to be verified, up to **75%** of the total number of bugs are in a Resolved state, but in a **still to verify** state.

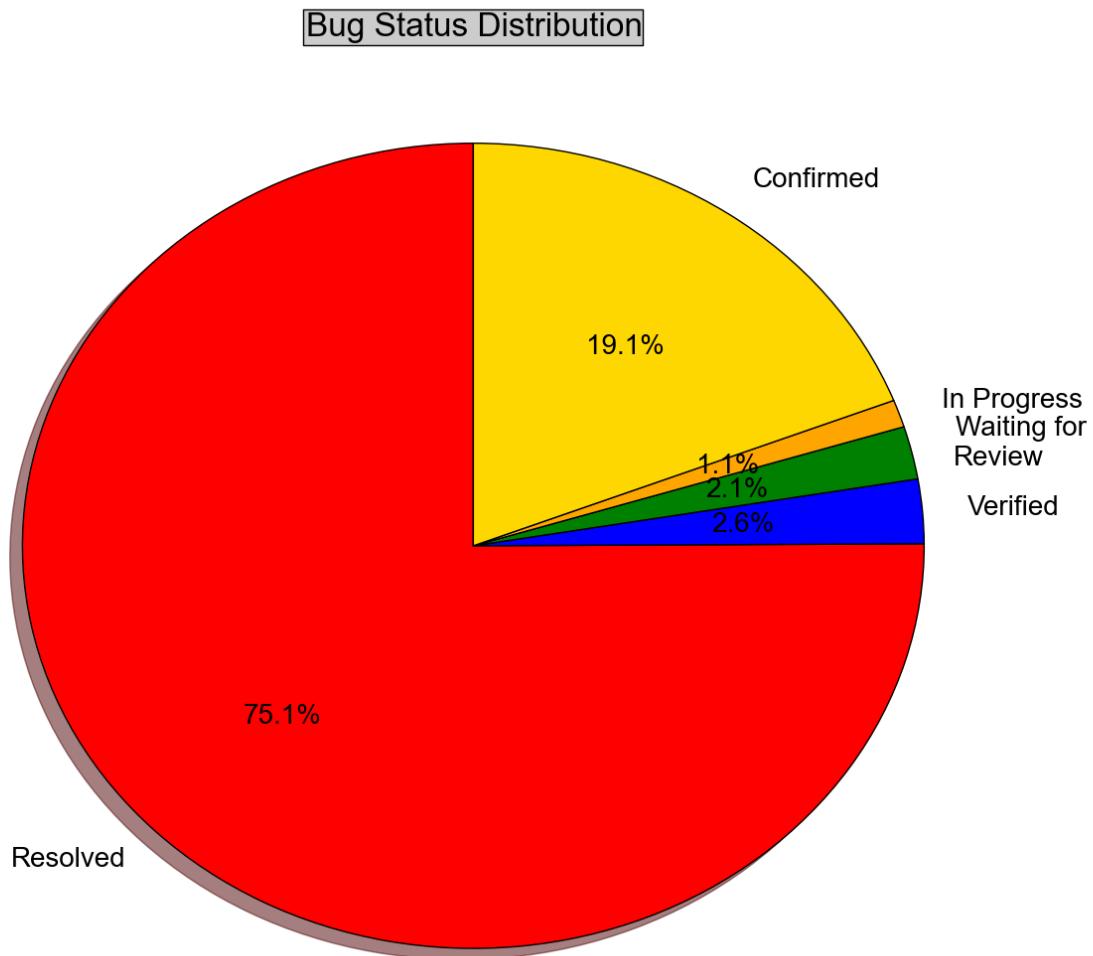


Figure 8.6: OpenDaylight Bug Status Percentage

The previous graph show that OpenDaylight's community has a very well behavior on resolution of bugs, as the number of non resolved issues is not that high. However, **there is an important issue on bug verification. Only 2.5% of the total amount of bugs are Verified, while 75% are Resolved.** There is an important opened action point to be implemented by government of the project to enforce the community resources on bug verification area.

Most active users in the Bug Tracker system can also be retrieved from the Database. This allows to perform some rankings on the most active users regarding this area. Performing some filtering, taking into account that some of the assignee of the bugs that are generic users associ-

ated to a particular area of the project have been pulled out of the classification (i.e.: controller-bugs, yangtools-bugs, openflowplugin-blugs, vtn-bugs, bugs, etc.), the most active users on bug resolution can be obtained with next SQL statement:

```
mysql> select p.user_id as "User", count(i.id) as "Bugs Resolved"
from people p, issues i where p.id=i.assigned_to and
(i.status='RESOLVED' or i.status='VERIFIED')
group by p.user_id order by count(i.id) desc limit 10;
```

The following tables show the top 10 people results obtained with the previous SQL statement, once generic users have been removed:

Username	Bugs Resolved
rovarga	109
ttkacik	103
mmarsale	82
mvitez	50
dkutenic	49
tolvecky	48
mirehak	42
jgloncak	41
tpanteli	40
milfibia	39

Table 8.8: OpenDaylight top 10 most BUG resolute people

Another important metrics to be considered regarding bug handling and resolution has to do with the amount of bugs opened and resolved that appear near release dates. Considering release dates, February 4th and September 29th for existing releases [7.4](#), Hydrogen and Helium respectively, the amount of bugs can be analyzed for previous dates (one month, for instance), to compare to existing mean per month of bugs previous to that last release month. On the one hand, it must be considered that first appear on March 23th, 2013.

To find out the average bugs opened per month, a period of six months will be considered.

The number of bugs opened can be obtained with next SQL statement:

```
mysql> select COUNT(id) as "Number of issues" from issues
where submitted_on BETWEEN '2013-06-01' AND '2013-12-01';
+-----+
| Number of issues |
+-----+
|          161   |
+-----+
```

161 bugs opened in six months shows a mean of approximately **26 opened bugs per month**. However, the number of bugs opened on last month before OpenDaylight Hydrogen release, shows a remarkable increase on the number of bugs opened:

```
mysql> select COUNT(id) as "Number of issues" from issues
where submitted_on BETWEEN '2014-01-01' AND '2014-02-01';
+-----+
| Number of issues |
+-----+
|          134   |
+-----+
```

With previous data it can be demonstrated that **the last month before Hydrogen release, number of issues increased dramatically**. The number of bugs opened was 134, while 161 were opened in six months in 2013. Regarding bugs opened on six months after Hydrogen release, the data obtained is as follows:

```
mysql> select COUNT(id) as "Number of issues" from issues
where submitted_on BETWEEN '2014-02-24' AND '2014-08-24';
+-----+
| Number of issues |
+-----+
```

```
+-----+
|      1166 |
+-----+
```

1166 Bugs opened in six months shows a mean of approximately **194 bugs opened per month**. It also shows an increase compared to the previous six months period studied, on what can be considered that, as time goes by, OpenDaylight project consist of more projects and more lines of code, meaning also more bugs opened. Apart from this consideration, the number of bugs on last month before OpenDaylight Helium release, shows a remarkable increase on the number of bugs opened:

```
mysql> select COUNT(id) as "Number of issues"
from issues where submitted_on
BETWEEN '2014-08-24' AND '2014-09-24';
+-----+
| Number of issues |
+-----+
|      449 |
+-----+
```

Previous numbers show that **the last month before Helium release, number of bugs opened was more than two times bigger compared to the mean of the previous six months**. With a further processing of each of the bugs opened per month, it can be shown the fact that, the nearest the release date is, the highest number of bugs are opened:

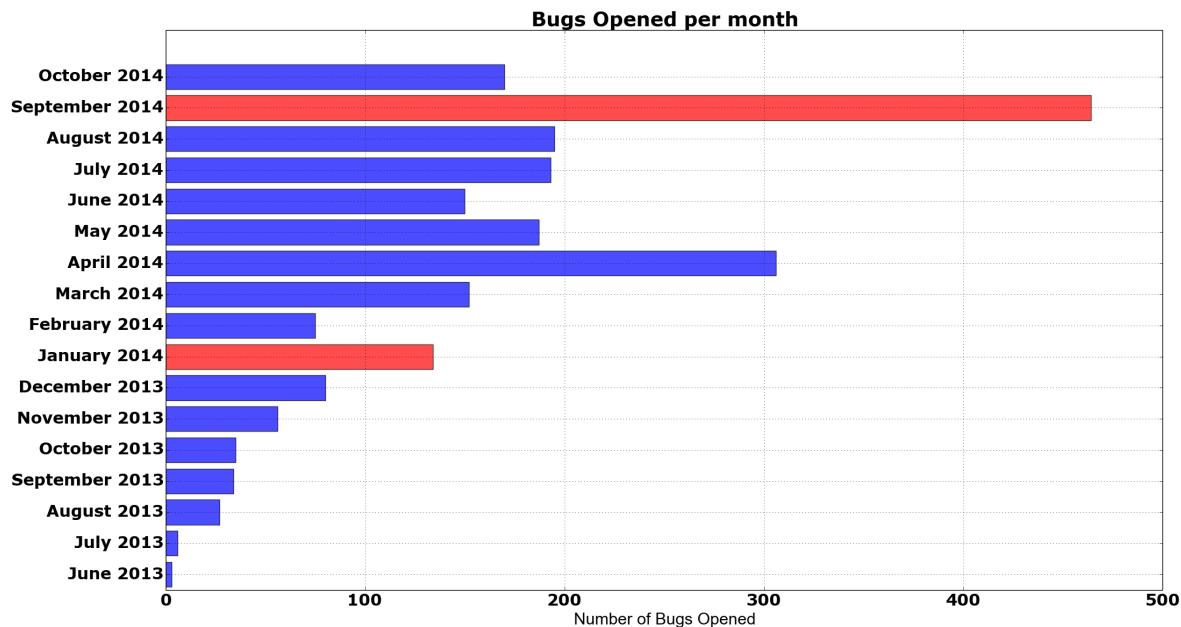


Figure 8.7: OpenDaylight Project Bugs Opened Per Month

It is also important to understand not just the number of bugs opened in that period, but also the ones that have been resolved. Taking into consideration the same six months period before Hydrogen release, the number of resolved bugs can be obtained with next SQL statement:

```
mysql> select COUNT(DISTINCT(issue_id)) as "Bugs Resolved"
from changes where new_value like 'RESOLVED' AND changed_on
BETWEEN '2013-06-01' AND '2013-12-01';
+-----+
| Bugs Resolved |
+-----+
|      106      |
+-----+
```

106 bugs resolved in six months shows a mean of approximately **18 resolved bugs per month**. However, the number of bugs resolved on last month before OpenDaylight Hydrogen release, shows a remarkable increase on the number of bugs opened:

```
mysql> select COUNT(DISTINCT(issue_id)) as "Bugs Resolved"
from changes where new_value like 'RESOLVED' AND changed_on
BETWEEN '2014-01-01' AND '2014-02-01';
+-----+
| Bugs Resolved |
+-----+
|          115 |
+-----+
```

With previous data it can be demonstrated that **the last month before Hydrogen release, number of issues resolved increased dramatically**. The number of bugs resolved was 115, while 106 were opened in six months in 2013. Regarding bugs opened on six months after Hydrogen release, the data obtained is as follows:

```
mysql> select COUNT(DISTINCT(issue_id)) as "Bugs Resolved"
from changes where new_value like 'RESOLVED' AND changed_on
BETWEEN '2014-02-24' AND '2014-08-24';
+-----+
| Bugs Resolved |
+-----+
|          896 |
+-----+
```

896 Bugs resolved in six months shows a mean of approximately **149 bugs resolved per month**. It also shows an increase compared to the previous six months period studied, on what can be considered that, as time goes by, OpenDaylight project consist of more projects and more lines of code, meaning also more bugs opened and closed as well. Apart from this consideration, the number of bugs on last month before OpenDaylight Helium release, shows a remarkable increase on the number of bugs closed:

```
mysql> select COUNT(DISTINCT(issue_id)) as "Bugs Resolved"
from changes where new_value like 'RESOLVED' AND changed_on
BETWEEN '2014-08-24' AND '2014-09-24';
+-----+
| Bugs Resolved |
+-----+
|      370      |
+-----+
1 row in set (0.00 sec)
```

Previous numbers show that **the last month before Helium release, number of bugs resolved was more than two times bigger compared to the mean of the previous six months**. With a further processing of each of the bugs resolved per month, it can be shown the fact that, the nearest the release date is, the highest number of bugs are resolved:

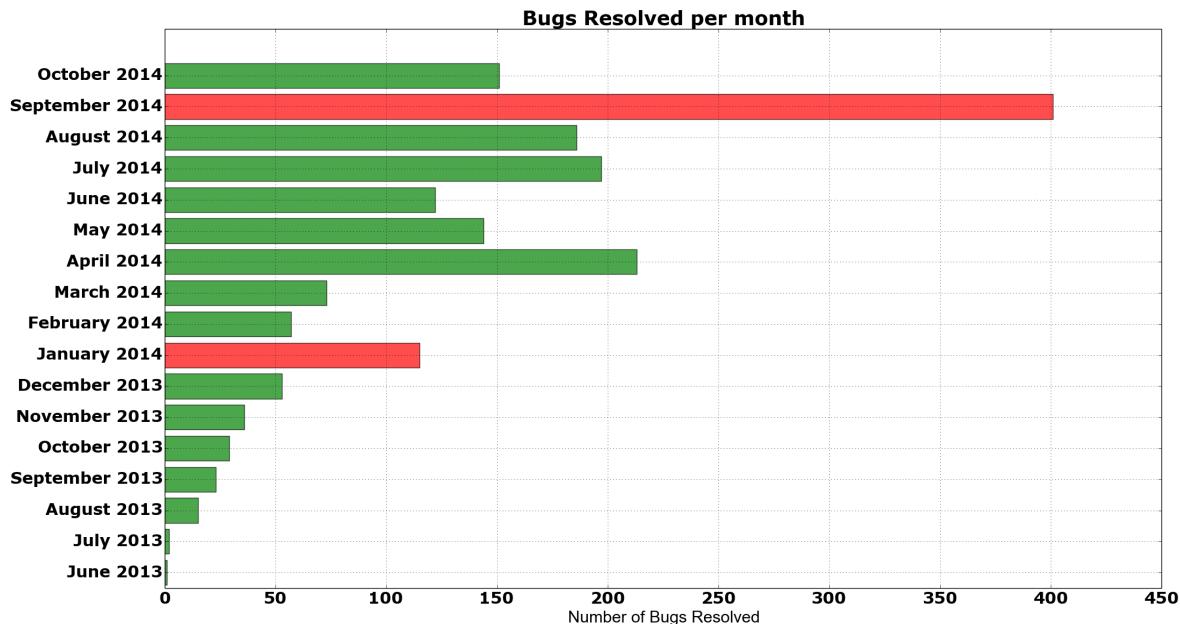


Figure 8.8: OpenDaylight Project Bugs Resolved Per Month

Previous information allows to calculate BMI, as this measure is defined as follows:

$$BMI = 100 \times \frac{ClosedBugs}{OpenedBugs}$$

BMI has a meaning when analyzing a particular date or period. A number under 100 means that the number of opened bugs in that period is superior to the number of bugs closed in that same period, while, the other way round, a number bigger than 100 allows to identify a period where resolved number of bugs is higher.

However, it has been shown that the number of bugs closed is very low, compared to the number of bugs resolved 8.6. In this case, BMI will be considered for the bugs resolved, as it is also a way of measuring how the community respond to the incoming amount of bugs opened on a particular period.

Taking previous consideration into account, next figure shows next modified BMI (resolved vs. opened bugs) index:

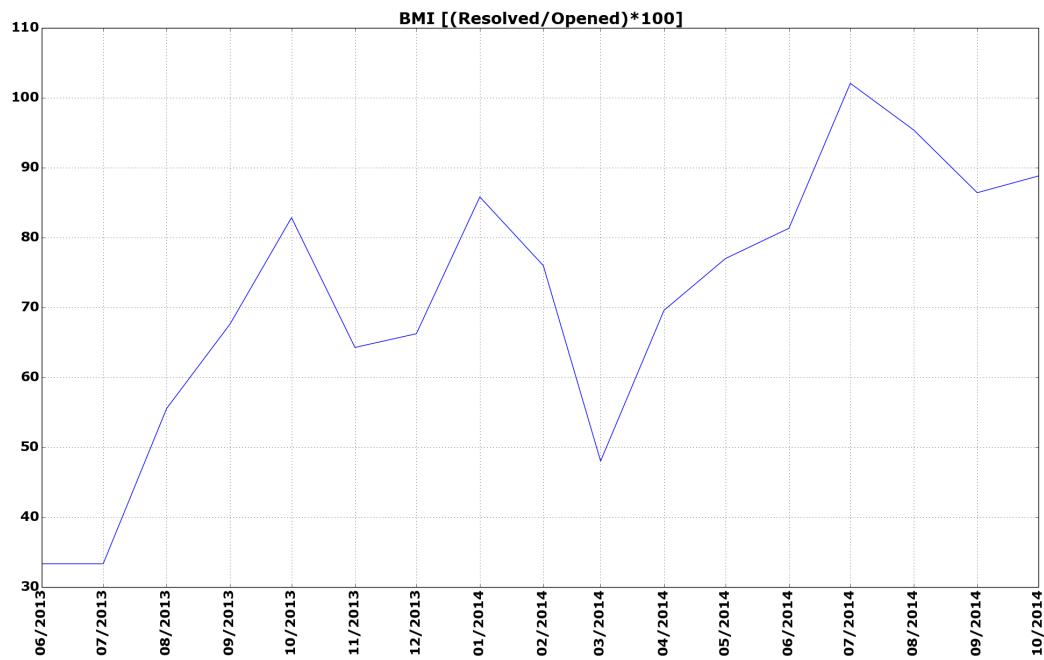


Figure 8.9: OpenDaylight Project BMI (Resolved vs. Opened)

From previous graph it can be observed how release date from Hydrogen release, February 4th, is preceded BMI peak on January 2014. Meanwhile, on Helium release date, September 24th, there is no BMI peak observed, as the maximum value was reached on July 2014. This is motivated due to the high number of bugs that was raised on that last month (494) before Helium release, despite of the fact that high amount of bugs were resolved as well (370).

This section has shown how Bug Tracker system can be analyzed to take some important metrics having to do with Bug handling and resolution aspects existing on a particular Open Source project. Metrics such as total number of bugs opened, resolved or closed, or how they are distributed in percentage can determine how sane are the different areas of the project (development, bug resolution, Quality Assurance, etc.). On the other hand, meritocracy can also be measured, so that top people resolving/verifying bugs can get recognition they deserve. Apart from that, it can also be observed how release dates impact on the different aspects regarding Bug resolution.

Chapter 9

Conclusions

9.1 Evaluation

The main goals of this document have been achieved by performing a description of the main aspects of the OpenDaylight Open Source project. To start with, a complete **Introduction** to the project have been detailed in Chapter 3, describing aspects such as which foundation the project belongs to, companies involved in the project, technologies that the project handles or which are the main objective for the OpenDaylight project itself.

On the one hand, **Economic Aspects** of the OpenDaylight project have been studied in Chapter 4. Aspects such as why NFV & SDN technologies are important from an economic perspective, what are the advantages of using Open Source, and why sometimes Open Source is perceived as the best option to innovate around an incipient technology.

On the other hand, another important aspect of the OpenDaylight project has to do with **Legal** practicalities, as defined in Chapter 5. This document has analyzed the Open Source license used in the project, which type of Open Source license it is and the reasons for using that license from the OpenDaylight Open Source project perspective. Besides this, the project has also analyzed the “Bylaws” of the project, with aspects having to do with legality around the governance of the project.

Apart from that, **Governance & Community Management** aspects have been analyzed in

Chapter 6. This chapter have analyzed, on the one hand, which is the governance of the project, based on a Technical Steering Committee and a Board of Directors. On the other hand, the document has identified main community tools, communication channels and community building mechanisms that OpenDaylight project owns to enforce the main value of an Open Source project, the community around it.

In addition to the chapters already mentioned, this document has analyzed **Technical Aspects** inside the OpenDaylight Open Source project in Chapter 7. Which programming language or languages are used, which is the architectural design of the project, which different projects compose the OpenDaylight project or which is the life cycle of a particular project are different issues covered by this section.

Last, but not least, **Project Evaluation** of the OpenDaylight Open Source project has been performed in Chapter 8. This section allows to diagnose the different practical issues an Open Source project experiences, and how by using data metrics retrieved from the project, in particular from its source code repositories, bug tracking tools, mail lists, etc., some conclusions of the project can be drawn, from which are the main contributors in a particular area to which sections of the project need improvement.

To summarize, by each of the previous chapters, this document has attempted to accomplish its main scope, which is the **analysis of an Open Source project by focusing on the main aspects that have to do with its Open Source nature**.

9.2 Lessons learned

9.2.1 Lesson 1: Why Open Source?

This document, apart from analyzing the OpenDaylight Open Source project, tries to show which are the main advantages that an Open Source project provides. Normally, **Open Source is associated to its “costless” aspect, but the rest of the aspects are forgiven**. However, a person reading this document can identify some of the other aspects meaning a competitive

advantage for Open Source projects, for instance:

- Open Source is normally the best option for different companies to start collaborating with each other.
- Some times, Open Source project is identified as the most desired kind of project by customers around an incipient technology.
- Using Open Source normally allows companies to enhance its offers or provide better support services.
- Open Source projects, such as OpenDaylight, are better to be evaluated as an option to use, as it can be analyzed due to its transparency nature.

9.2.2 What I learned

From personal perspective, this document has helped its author, mainly, to strengthen the capacity to **perform an analysis of an Open Source project** from the different perspectives that are important to take into account due to its Open Source nature. In particular, analyzing OpenDaylight Open Source project also helps to identify particularities of an incipient Open Source project around an incipient and innovative technology.

9.2.3 Knowledge and skills acquired in the M.Sc. studies that helped me in this work

In general, this document is possible due to different subjects that have been studied in the M.Sc. “Master in Software Libre” from Rey Juan Carlos University. The different subjects that have an stronger impact on the knowledge developed in this document are:

- Introduction to Free Software. This subject emphasizes on the different aspects having to do with Open Source. What is an Open Source project, how to differentiate them, and which are the main aspects that characterize this kind of projects. Knowledge within this subject have been used in all the document for different contents available.

- Legal Aspects. Different kind of Open Source licenses, legal aspects and concerns and legal constraints arising are studied in this subject, which makes it indispensable to evaluate the Legal Aspects around OpenDaylight project which has been studied in this document.
- Economic Aspects. Business Cases, economic particularities of Open Source, Business Models, Market Share reports studies and other economic aspects about Open Source projects are different aspects studied in this subject. Some of the most important knowledge acquired have been exploited to write section about Economic Aspects available on this document.
- Developers And Motivations. This subject focuses on Open Source project developers, its motivations and how they compose a community to start collaborating. Aspects such as community profiling and studying are also covered. This document takes parts of the knowledge acquired in this subject to complete Community Management chapter of this document.
- Project Management. Different aspects around Open Source management are covered in this subject, from technical aspects to Community Management, what makes it an indispensable subject for both Community Management and Technical Aspects sections available on this document.
- Project Evaluation. Last but not least, Project Evaluation subject shows how different tools can be used to extract metrics of an Open Source project and evaluate different aspects, such as the quality of the project from a particular quality model perspective, to different conclusions that apply to that particular project. Definitely, all the knowledge has been exploited in this document, in particular for the Project Evaluation section.

9.3 Future work

9.3.1 More on...

This document analyzes each of the Open Source aspects about OpenDaylight project. However, as different aspects are covered in the document, none of them is very extensive, as it has been preferred to perform a more global view of the OpenDaylight project. Any of the different

sections can be studied in a more detailed way, and would be a good point to perform a further study on a particular aspect of OpenDaylight Open Source project.

Apart from that, aspects covered in this project can be wider studied as time goes by in order to analyze both the progress of the OpenDaylight project, such as new releases, for instance, or how the community grows or decrease, but also from technologies around the project, as being incipient technologies mean also that it is still pending how they impact Networking Industry and if they are finally succeeding in that particular market.

9.3.2 Other aspects

Some aspects about OpenDaylight project have not been studied in detail. In particular, technical aspects can be studied in a more detailed way. How a fix can be submitted to the project, which is the installation process of the OpenDaylight controller, how a particular SDN Controller feature can be analyzed or tested, and, in general, any particular detail having to do with technical aspects in OpenDaylight project are not deep covered in this document, and could be also an option to study in other projects around OpenDaylight project.

9.3.3 Other focuses

This document can be an initial entry point to use as a guide to analyze Open Source projects from their different perspectives. In particular, this document could be a good initial guide to analyze other Open Source projects having to do with SDN and/or NFV. For instance, OP-NFV [OPfNP14], which is a recently created Open Source project most focused on NFV technology, whose first release is expected for January 2015.

Appendix A

EPLv1.0

Eclipse Public License - v 1.0

THE ACCOMPANYING PROGRAM IS PROVIDED UNDER THE TERMS OF THIS ECLIPSE PUBLIC LICENSE ("AGREEMENT"). ANY USE, REPRODUCTION OR DISTRIBUTION OF THE PROGRAM CONSTITUTES RECIPIENT'S ACCEPTANCE OF THIS AGREEMENT.

1. DEFINITIONS

"Contribution" means:

a) in the case of the initial Contributor, the initial code and documentation distributed under this Agreement, and

b) in the case of each subsequent Contributor:

i) changes to the Program, and ii) additions to the Program;

where such changes and/or additions to the Program originate from and are distributed by that particular Contributor. A Contribution 'originates' from a Contributor if it was added to the Program by such Contributor itself or anyone acting on such Contributor's behalf. Contributions do not include additions to the Program which: (i) are separate modules of software distributed in conjunction with the Program under their own license agreement, and (ii) are not derivative works of the Program.

"Contributor" means any person or entity that distributes the Program.

"Licensed Patents" mean patent claims licensable by a Contributor which are necessarily infringed by the use or sale of its Contribution alone or when combined with the Program.

"Program" means the Contributions distributed in accordance with this Agreement.

"Recipient" means anyone who receives the Program under this Agreement, including all Contributors.

2. GRANT OF RIGHTS

a) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free copyright license to reproduce, prepare derivative works of, publicly display, publicly perform, distribute and sublicense the Contribution of such Contributor, if any, and such derivative works, in source code and object code form.

b) Subject to the terms of this Agreement, each Contributor hereby grants Recipient a non-exclusive, worldwide, royalty-free patent license under Licensed Patents to make, use, sell, offer to sell, import and otherwise transfer the Contribution of such Contributor, if any, in source code and object code form. This patent license shall apply to the combination of the Contribution and the Program if, at the time the Contribution is added by the Contributor, such addition of the Contribution causes such combination to be covered by the Licensed Patents. The patent license shall not apply to any other combinations which include the Contribution. No hardware per se is licensed hereunder.

c) Recipient understands that although each Contributor grants the licenses to its Contributions set forth herein, no assurances are provided by any Contributor that the Program does not infringe the patent or other intellectual property rights of any other entity. Each Contributor disclaims any liability to Recipient for claims brought by any other entity based on infringement of intellectual property rights or otherwise. As a condition to exercising the rights and licenses granted hereunder, each Recipient hereby assumes sole responsibility to secure any other intellectual property rights needed, if any. For example, if a third party patent license is required to allow Recipient to distribute the Program, it is Recipient's responsibility to acquire that license before distributing the Program.

d) Each Contributor represents that to its knowledge it has sufficient copyright rights in its Contribution, if any, to grant the copyright license set forth in this Agreement.

3. REQUIREMENTS

A Contributor may choose to distribute the Program in object code form under its own license agreement, provided that:

- a) it complies with the terms and conditions of this Agreement; and
- b) its license agreement:
 - i) effectively disclaims on behalf of all Contributors all warranties and conditions, express and implied, including warranties or conditions of title and non-infringement, and implied warranties or conditions of merchantability and fitness for a particular purpose;
 - ii) effectively excludes on behalf of all Contributors all liability for damages, including direct, indirect, special, incidental and consequential damages, such as lost profits;
 - iii) states that any provisions which differ from this Agreement are offered by that Contributor alone and not by any other party; and
 - iv) states that source code for the Program is available from such Contributor, and informs licensees how to obtain it in a reasonable manner on or through a medium customarily used for software exchange.

When the Program is made available in source code form:

- a) it must be made available under this Agreement; and
- b) a copy of this Agreement must be included with each copy of the Program.

Contributors may not remove or alter any copyright notices contained within the Program.

Each Contributor must identify itself as the originator of its Contribution, if any, in a manner that reasonably allows subsequent Recipients to identify the originator of the Contribution.

4. COMMERCIAL DISTRIBUTION

Commercial distributors of software may accept certain responsibilities with respect to end users, business partners and the like. While this license is intended to facilitate the commercial use of the Program, the Contributor who includes the Program in a commercial product offering should do so in a manner which does not create potential liability for other Contributors. Therefore, if a Contributor includes the Program in a commercial product offering, such Contributor ("Commercial Contributor") hereby agrees to defend and indemnify every other Contributor ("Indemnified Contributor") against any losses, damages and costs (collectively "Losses") arising from claims, lawsuits and other legal actions brought by a third party against the Indemnified Contributor to the extent caused by the acts or omissions of such Commercial Contributor in connection with its distribution of the Program in a commercial product offering. The obligations in this section do not apply to any claims or Losses relating to any actual or alleged intellectual property infringement. In order to qualify, an Indemnified Contributor must: a) promptly notify the Commercial Contributor in writing of such claim, and b) allow the Commercial Contributor to control, and cooperate with the Commercial Contributor in, the defense and any related settlement negotiations. The Indemnified Contributor may participate in any such claim at its own expense.

For example, a Contributor might include the Program in a commercial product offering, Product X. That Contributor is then a Commercial Contributor. If that Commercial Contributor then makes performance claims, or offers warranties related to Product X, those performance claims and warranties are such Commercial Contributor's responsibility alone. Under this section, the Commercial Contributor would have to defend claims against the other Contributors related to those performance claims and warranties, and if a court requires any other Contributor to pay any damages as a result, the Commercial Contributor must pay those damages.

5. NO WARRANTY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, THE PROGRAM IS PROVIDED ON AN "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, EITHER EXPRESS OR IMPLIED INCLUDING, WITHOUT LIMITATION, ANY

WARRANTIES OR CONDITIONS OF TITLE, NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Each Recipient is solely responsible for determining the appropriateness of using and distributing the Program and assumes all risks associated with its exercise of rights under this Agreement , including but not limited to the risks and costs of program errors, compliance with applicable laws, damage to or loss of data, programs or equipment, and unavailability or interruption of operations.

6. DISCLAIMER OF LIABILITY

EXCEPT AS EXPRESSLY SET FORTH IN THIS AGREEMENT, NEITHER RECIPIENT NOR ANY CONTRIBUTORS SHALL HAVE ANY LIABILITY FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING WITHOUT LIMITATION LOST PROFITS), HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OR DISTRIBUTION OF THE PROGRAM OR THE EXERCISE OF ANY RIGHTS GRANTED HEREUNDER, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. GENERAL

If any provision of this Agreement is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this Agreement, and without further action by the parties hereto, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

If Recipient institutes patent litigation against any entity (including a cross-claim or counter-claim in a lawsuit) alleging that the Program itself (excluding combinations of the Program with other software or hardware) infringes such Recipient's patent(s), then such Recipient's rights granted under Section 2(b) shall terminate as of the date such litigation is filed.

All Recipient's rights under this Agreement shall terminate if it fails to comply with any of the material terms or conditions of this Agreement and does not cure such failure in a reasonable period of time after becoming aware of such noncompliance. If all Recipient's rights under this Agreement terminate, Recipient agrees to cease use and distribution of the Program as soon as reasonably practicable. However, Recipient's obligations under this Agreement and any licenses granted by Recipient relating to the Program shall continue and survive.

Everyone is permitted to copy and distribute copies of this Agreement, but in order to avoid inconsistency the Agreement is copyrighted and may only be modified in the following manner. The Agreement Steward reserves the right to publish new versions (including revisions) of this Agreement from time to time. No one other than the Agreement Steward has the right to modify this Agreement. The Eclipse Foundation is the initial Agreement Steward. The Eclipse Foundation may assign the responsibility to serve as the Agreement Steward to a suitable separate entity. Each new version of the Agreement will be given a distinguishing version number. The Program (including Contributions) may always be distributed subject to the version of the Agreement under which it was received. In addition, after a new version of the Agreement is published, Contributor may elect to distribute the Program (including its Contributions) under the new version. Except as expressly stated in Sections 2(a) and 2(b) above, Recipient receives no rights or licenses to the intellectual property of any Contributor under this Agreement, whether expressly, by implication, estoppel or otherwise. All rights in the Program not expressly granted under this Agreement are reserved.

This Agreement is governed by the laws of the State of New York and the intellectual property laws of the United States of America. No party to this Agreement will bring a legal action under this Agreement more than one year after the cause of action arose. Each party waives its rights to a jury trial in any resulting litigation.

Appendix B

Bylaws Summary

Article I: Name, Purpose and Offices	
Section	Description
1.1: Name	Name of the corporation will be “OpenDaylight Project, Inc.”, referred to in these By-laws as the “ODP”
1.2: Principal Office	Located at San Francisco, CA 94110. The Board of Directors of the ODP (the “Board”) is hereby granted full power and authority to change its principal office
1.3: Other Offices	Branch or subordinate offices may at any time be established by the Board at any place or places
1.4: Purpose	... The primary purpose of the ODP (collectively, “the Purpose”) is to (a) advance the creation, evolution, promotion, and support of an Open Source software defined network and network functions virtualization software platform ...
1.5: Nonprofit Status	(a) The ODP is organized and shall be operated as a non-stock, not for profit membership corporation ... (b) The Board may, in its sole discretion, elect to seek exemption from Federal taxation ...

Table B.1: OpenDaylight Bylaws Summary: Article I

Article II: Members	
Section	Description
2.1: Classes of Membership	Platinum Members, Strategic End-User Members, Gold Members, Silver Members, Individual Committer Members, and Associate Members. Additional classes of voting and non-voting members may be created in the future...
2.2: Conditions of Membership	Any association, partnership, organization, governmental agency, company, corporation, academic entity, or non-profit entity (or individual, solely with respect to Individual Committer Members or Associate Members) ...
2.3 to 2.8: Privileges	These sections reflect the privileges of being Platinum Member, Strategic End-User Member, Gold Member, Silver Member, Individual Committer or Associate Member

Table B.2: OpenDaylight Bylaws Summary: Article II

Article III: Actions of Members	
Section	Description
3.1: Action Without Meeting	Any action required or permitted to be taken by the Members, ... or at any meeting of a Member Committee, Working Group ..., may be taken without prior notice and without an in-person vote, if a consent in writing, setting forth the action to be taken, shall be signed by Members ...
3.2: Nomination and Election Procedures	Subject to the provisions of Section 4.3, the Board shall establish reasonable nomination and election procedures given the nature, size, and operations of the ODP, including a reasonable means for Members of appropriate classes to nominate a person for election as a Director.

Table B.3: OpenDaylight Bylaws Summary: Article III

Article IV: Directors	
Section	Description
4.1: Powers; Voting	... The Board may exercise all powers of the ODP and do all such lawful acts and things as are not by statute or by the Certificate of Incorporation or by these By-laws directed ...
4.2: Number of Directors	Subject to Section 4.4, the total number of Directors shall be at least one and not more than two (2) times the number of Platinum Members ...
4.3: Nomination and Election	Each Platinum Member ... shall be entitled individually to appoint one Director ... (b) Each Strategic End-User Member (while remaining in good standing) shall be entitled individually to appoint one Director ... (c) Each Gold Member ... shall have the right to vote, together with the other Gold Members as a class, to elect a number of Directors equal to the number of Gold Members then in good standing divided by three (3) ... (d) Each Silver member ... shall have the right to vote, together with the other Silver Members as a class, to elect one (1) Director.
4.4: Enlargement or Reduction	... the number of Directors, the persons eligible to become Directors and the classes of Members eligible to appoint, elect and/or nominate Directors may be amended at any time by a Super Majority Vote (as defined in Section 4.10.(b)) of the Board.
4.5: Resignation and Removal	Any Director may resign at any time upon notice to the ODP in writing or by electronic transmission ... Unless otherwise specified by law or the Certificate of Incorporation, any Director may be removed by a majority of the other Directors ...
4.6: Vacancies	Vacancies on the Board occurring as a result of ..., resignation, removal or termination of employment ..., may be filled by such Member or class of Members, as applicable.
4.7: Place of Meetings	The Board may hold meetings, both regular and special, either within or without the State of Delaware.

Table B.4: OpenDaylight Bylaws Summary: Article IV

Article IV: Directors	
Section	Description
4.8: Regular Meetings	Regular meetings of the Board may be held without notice at such time and at such place as shall from time to time be determined by the Board.
4.9: Sepcial Meetings	Special meetings of the Board may be called by the President, Secretary, or on the written request of two or more Directors, or by one Director in the event that there is only one Director in office.
4.10: Quorum, Action at Meeting, Adjournments	(a) At all meetings of the Board a majority of Directors then in office, shall constitute a quorum for the transaction of business and the act of a majority of such Directors ... (b) In order to pass a “Super Majority Vote”, a resolution must be taken at a meeting of the Board at which a quorum is present ...
4.11: Action by Consent	... any action required or permitted to be taken by the Board may be taken without a meeting and without prior notice if a majority of Directors then in office ... consent thereto in writing or by electronic transmission, and the writing or writings, or electronic transmission or transmissions, are filed with the minutes of proceedings of the Board.
4.12: Telephonic Meetings	... members of the Board or of any Board Committee may participate in a meeting of the Board or of any Board Committee, as the case may be, by means of conference telephone.
4.13: Inspection Rights	Every Director shall have the absolute right at any time to inspect, copy and make extracts of, in person or by agent or attorney, all books, records and documents of every kind, and to inspect the physical properties of the ODP.
4.14: Fees and Compensation	Directors shall not receive any stated salary or reimbursements for their services as Directors ...

Table B.5: OpenDaylight Bylaws Summary: Article IV

Article V: Executive Committee and Other Committees	
Section	Description
5.1: Executive Committee	The Board may (but shall not be required), by resolution adopted by a majority of the Directors then in office (provided a quorum is present), create an Executive Committee, consisting of one or more Directors. The Board may designate one or more Directors as alternate members of such Executive Committee, who may replace any absent member at any meeting of such Executive Committee ...
5.2: Other Committees of the Board	The Board may ... create such nominating, audit, compensation and other Board Committees, ..., to perform such general or special duties as may from time to time be delegated to any such Board Committees by the Board, subject to the limitations imposed by the Certificate of Incorporation or by these By-laws.
5.3: Meetings of Committees of the Board	... each Board Committee may adopt its own rules governing the time and place of holding and the method of calling its meetings and the conduct of its proceedings ...
5.4: Term of Office of Members of Committees of the Board	Each member of a Board Committee shall serve for such term as shall be established at the time of his or her election.
5.5: Committees of the Members	(a) From time to time, the Board may establish Member Committees in addition to the Technical Steering Committee and End-User Committee ... (b) A Technical Steering Committee (the “TSC”) of the ODP shall be established consisting of (i) a project lead from each core project, (ii) if not otherwise represented, for such period as may be established by the Board, a representative designated by each of the Platinum Members ... (c) An End-User Committee of the ODP shall be established consisting of one representative designated by each of the Strategic End-User Members ...

Table B.6: OpenDaylight Bylaws Summary: Article V

Article VI: Officers	
Section	Description
6.1: Officers	The Officers of the ODP shall be a Chairperson, a President, a Treasurer and a Secretary, each of whom shall also be a Director. The ODP may also have, at the discretion of the Board, an Executive Director, one or more Vice-Presidents, one or more Assistant Secretaries and/or Assistant Treasurers ...
6.2: Vacancies	A vacancy in any office because of ..., resignation, removal, ... or any other cause shall be filled in the manner prescribed in these By-laws for regular election
6.3: Election	The Board at its annual meeting each year shall choose a President, Chairperson, a Secretary and a Treasurer.
6.4: Tenure	Each Officer of the ODP shall hold office until his or her successor is chosen and qualifies, unless a different term is specified in the vote choosing or electing him, or until his or her earlier death, resignation or removal.
6.5: President and Executive Director	(a) The President shall have all of the powers normally associated with the role of chief executive officer and preside at all meetings of the Board (in the absence of a Chairperson) and the Members ... (b) The Executive Director (if any) shall preside over the day-to-day affairs of the ODP under the direction of the Board and the President and perform such other duties and have such other powers as the Board or the President may from time to time prescribe.
6.6: Vice-Presidents	In the absence of the President ..., a Vice-President, or if there be more than one Vice-President, the Vice-Presidents in the order designated by the Board ..., shall perform the duties of the President, and when so acting, shall have all the powers of and be subject to all the restrictions upon the President.

Table B.7: OpenDaylight Bylaws Summary: Article VI

Article VI: Officers	
Section	Description
6.7: Secretary	The Secretary shall have such powers and perform such duties as are incident to the office of Secretary ..., including without limitation a recording all the proceedings of the meetings of the ODP and of the Board.
6.8: Assistant Secretaries	Any Assistant Secretary shall, in the absence of the Secretary or in the event of his or her inability or refusal to act, perform the duties and exercise the powers of the Secretary. In the absence of the Secretary or any Assistant Secretary at any meeting of Directors, the person presiding at the meeting shall designate a temporary or acting Secretary to keep a record of the meeting.
6.9: Treasurer	The Treasurer shall perform such duties and shall have such powers as may be assigned to him or her by the Board or the President. Unless otherwise determined by the Board, the Treasurer shall chair the Audit and Finance Committees of the ODP.
6.10: Compensation	The compensation, if any, of the Officers shall be fixed from time to time by the Board, and no Officer shall be prevented from receiving such compensation by reason of the fact that the Officer is also a Director of the ODP.

Table B.8: OpenDaylight Bylaws Summary: Article VI

Article VII: Notices	
Section	Description
7.1: Delivery	(a) Whenever, under the provisions of law, ..., written notice is required to be given to any Director or Member, such notice may be given by mail, addressed to such Director or Member, at his, her or its address as it appears on the records of the ODP, with postage thereon prepaid. (b) Notice given pursuant to this Section shall be deemed given ... (i) by facsimile ... (ii) by electronic mail ... (iii) by posting on electronic network ... (c) For purposes of these By-laws, "electronic transmission" means any form of communication, not directly involving the physical transmission of paper. (d) Without limiting the foregoing, the ODP adopts electronic mail as its principal source of communication with its Members.
7.2: Waiver of Notice	Whenever any notice is required to be given under the provisions of law or of the Certificate of Incorporation or of these By-laws, a waiver thereof in writing, signed by the person or persons entitled to said notice, whether before or after the time stated therein, or a waiver by electronic transmission by the person entitled to notice, shall be deemed equivalent thereto.

Table B.9: OpenDaylight Bylaws Summary: Article VII

Article VIII: Indemnification	
Section	Description
8.1: Actions other than by or in the Right of the ODP	... the ODP shall indemnify any person who was or is a party or is threatened to be made a party to any threatened, pending or completed action, suit or proceeding, whether civil, criminal, administrative or investigative ...
8.2: Actions by or in the Right of the ODP	...the ODP shall indemnify any person who was or is a party or is threatened to be made a party to any threatened, pending or completed action or suit by or in the right of the ODP to procure a judgment in its favor by reason of the fact that he or she is or was a Director, Officer, employee or agent of the ODP...
8.3: Success on the Merits	To the extent that any person described in Section 8.1 or 8.2 of this Article VIII has been successful on the merits or otherwise in defense of any action, suit or proceeding referred to in said Sections, ...
8.4: Specific Authorization	Any indemnification under Section 8.1 or 8.2 of this Article VIII (unless ordered by a court) shall be made by the ODP only as authorized in the specific case upon a determination that indemnification of any person described in said Sections is proper in the circumstances because he or she has met the applicable standard of conduct set forth in said Sections.
8.5: Advance Payment	Expenses incurred in defending a civil or criminal action, suit or proceeding shall be paid by the ODP in advance of the final disposition of such action, suit or proceeding upon receipt of an undertaking by or on behalf of any person described in said Section to repay such amount if it shall ultimately be determined that he or she is not entitled to indemnification by the ODP as authorized in this Article VIII.

Table B.10: OpenDaylight Bylaws Summary: Article VIII

Article VIII: Indemnification	
Section	Description
8.6: Non-Exclusivity	The indemnification and advancement of expenses provided by, or granted pursuant to, the other Sections of this Article VIII shall not be deemed exclusive of any other rights ...
8.7: Jurisdiction of Delaware Court of Chancery	The Delaware Court of Chancery is vested with exclusive jurisdiction to hear and determine all actions for advancement of expenses or indemnification.
8.8: Insurance	The Board may authorize the ODP to purchase and maintain insurance on behalf of any person who is or was a Director, Officer, employee or agent of the ODP, or is or was serving at the request of the ODP as a director, officer, employee, ...
8.9: Continuation of Indemnification and Advancement of Expenses	The indemnification and advancement of expenses provided by, or granted pursuant to, this Article VIII shall continue as to a person who has ceased to be a Director, Officer, employee or agent of the ODP.
8.10: Severability	If any word, clause or provision of this Article VIII or any award made hereunder shall for any reason be determined to be invalid, the provisions hereof shall not otherwise be affected thereby but shall remain in full force and effect.
8.11: Intent of Article	The intent of this Article VIII is to provide for indemnification and advancement of expenses to the fullest extent permitted by Section 145 of the General Corporation Law of Delaware.

Table B.11: OpenDaylight Bylaws Summary: Article VIII

Article IX: Books and Records	
Section	Description
9.1: Books and Records	The ODP shall keep adequate and correct books and records of account, minutes of the proceedings of the Members, the Board and Board Committees, and a record of the Members giving their names and addresses and the class of Membership held by each.
9.2: Form of Records	Minutes shall be kept in written form. Other books and records shall be kept either in written form or in any other form capable of being converted into written form.
9.3: Reports to Directors, Members and Others	Minutes shall be kept in written form. Other books and records shall be kept either in written form or in any other form capable of being converted into written form.
9.4: Record Date	In order that the ODP may determine the Members entitled to express consent to corporate action in writing without a meeting, ..., the Board may fix, in advance, a record date, which shall not be (i) more than sixty (60) days prior to the adoption of the resolution by the Board ... (ii) later than the date upon which the Board adopts the resolution proposing the taking of such action.
9.5: Registered Members	The ODP shall be entitled to recognize the exclusive right of a person registered on its books as a Member or a representative of a Member to receive distributions, if any, and to vote, if such records indicate that such person is a Voting Member or a representative of a Voting Member, and to hold liable for Financial Obligations each Member registered on its books.

Table B.12: OpenDaylight Bylaws Summary: Article IX

Article X: Certain Transactions	
Section	Description
10.1: Transactions with Interested Parties	No contract or transaction between the ODP and one or more of its Directors or Officers, or between the ODP and any other corporation, partnership, association, or other organization in which one or more of its directors or officers are directors or officers, or have a financial interest, shall be void or voidable solely for this reason, or solely because such Director or Officer (or other director or officer) is present at or participates in the meeting of the Board or Board Committee which authorizes the contract or transaction or solely because his, her or their votes are counted for such purpose, if: (a) The material facts as to his or her relationship or interest and as to the contract or transaction are disclosed or are known to the Board or such Board Committee; or (b) The material facts as to his or her relationship or interest and as to the contract or transaction are disclosed or are known to the Voting Members entitled to vote thereon, and the contract or transaction is specifically approved in good faith by vote of the Voting Members; or (c) The contract or transaction is fair as to the ODP as of the time it is authorized, approved or ratified, by the Board, a Board Committee, or the Voting Members.

Table B.13: OpenDaylight Bylaws Summary: Article X

Article XI: Grants, Contracts, Loans	
Section	Description
11.1: Grants	The making of grants and contributions, and otherwise rendering financial assistance for the Purposes of the ODP, may be authorized by the Board. The Board may authorize any Officer or Officers, agent or agents, in the name of and on behalf of the ODP to make any such grants, contributions or assistance.
11.2: Execution of Contracts	The Board may authorize any Officer, employee or agent of the ODP, in the name and on behalf of the ODP, to enter into any contract or execute and satisfy any instrument, and any such authority may be general or confined to specific instances, or otherwise limited.
11.3: Checks, Drafts, etc.	All checks, drafts and other orders for the payment of money out of the funds of the ODP, and all notes or other evidences of indebtedness of the ODP, shall be signed on behalf of the ODP in such manner as shall from time to time be determined by resolution of the Board.
11.4: Deposits	The funds of the ODP not otherwise employed shall be deposited from time to time to the order of the ODP in such banks, trust companies, or other depositories, or shall be otherwise invested, as the Board may select or direct, or as may be selected or directed by an Officer, employee or agent of the ODP to whom such power may from time to time be specifically delegated by the Board.

Table B.14: OpenDaylight Bylaws Summary: Article XI

Article XII: General Provisions	
Section	Description
12.1: Fiscal Year	The fiscal year of the ODP shall be determined, and may be changed, by resolution of the Board.
12.2: Reserves	The Directors may set apart out of any funds of the ODP a reserve or reserves for any proper purpose and may abolish any such reserve.
12.3: Seals	The Board may, by resolution, adopt a corporate seal. The corporate seal shall have inscribed thereon the name of the ODP, the year of its organization and the word "Delaware". The seal may be used by causing it or a facsimile thereof to be impressed or affixed or reproduced or otherwise. The seal may be altered from time to time by the Board.
12.4: Proprietary Rights	(a) Except as specifically provided to the contrary in such policies and procedures as may from time to time be approved by the Board, all information disclosed by any participant during any official meeting or activity of the ODP, including but not limited to Member meetings, Member Committee Meetings, ..., electronic mail or the like, shall be deemed to have been disclosed on a non-confidential basis, but without waiver of any rights represented by valid patents, patent applications, and Federal and international statutory copyrights. (b) No express or implied right, whether by implication, ..., to any patent, copyright, trademark, trade secret, or other intellectual property right of any Member is or shall be deemed to be granted to the ODP ..., except as may be provided in a separate written agreement. (c) No Member shall at any time be required to exchange proprietary information with any other Member solely by reason of its being a Member of the ODP.

Table B.15: OpenDaylight Bylaws Summary: Article XII

Article XIII: Antitrust Compliance	
Section	Description
13.1: General	The ODP will conduct all of its activities in conformance with all international, U.S. federal and state antitrust laws and competition laws ... Each of the Members of the ODP is committed to fostering competition in the development of new products and services, and the activities of the ODP are intended to promote such competition. ... Each Member shall assume responsibility to provide appropriate legal counsel to its representatives acting under these Bylaws regarding the importance of limiting the scope of their discussions to the topics that relate to the Purposes of the ODP, whether or not such discussions take place during formal meetings, informal gatherings, or otherwise.
13.2: Availability of Intellectual Property	It is the good faith objective of the ODP (a) to make all intellectual property available as soon as its development by the ODP is complete on the same terms to all Members who have not participated in the development or determination of such intellectual property as well as to all those Members who have participated, (b) to make all such intellectual property available at the same point in time to all Members, and (c) to make all such intellectual property available to all non-Members on the same fair and reasonable terms and conditions. The Board, by Super Majority Vote, ... , may approve the distribution of intellectual property by the ODP under a license other than the Eclipse Public License, version 1.0.
13.3: No Obligation to Endorse	No Member shall, by reason of its Membership or participation in the ODP or otherwise, be obligated to license from the ODP, use or endorse any intellectual property developed or endorsed by the ODP, or to conform any of its products to any Platform developed or adopted by the ODP, nor shall any such Member be precluded from independently licensing, using or endorsing similar intellectual property, platform, software, specifications or documentation developed by it or by others.

Table B.16: OpenDaylight Bylaws Summary: Article XIII

Article XIV: Amendments	
Section	Description
14.1: Amendments	Except where such power is expressly limited by law, the Certificate of Incorporation or these By-laws as to any specific action, these By-laws may be altered, amended or repealed, and new By-laws may be adopted, in each case by an affirmative vote of a two-thirds majority of all then serving members of the Board.

Table B.17: OpenDaylight Bylaws Summary: Article XIV

Appendix C

TSC Charter

Technical Steering Committee Charter:

Section 1. Guiding Principle. The OpenDaylight Project (ODP) will operate transparently, openly, collaboratively, and ethically. Project proposals, time lines, and status must not merely be open, but also easily visible to outsiders.

Section 2. Evolution of ODP Governance. Most large, complex Open Source communities have both a business and a technical governance model. ODP's technical leadership contains both a Technical Steering Committee (TSC) and project leads for major components. ODP's business leadership is instantiated in a Board of Directors (the "Board"). This Technical Steering Committee Charter reflects a carefully constructed balanced role for the TSC and the Board in the governance of ODP. As both the Board and the TSC gain more experience with ODP's specific operations, the Board and the TSC have the ability to amend this Charter, subject to the policy and by-laws of the organization. The normal amendment process is for TSC to propose changes using simple majority of the full TSC to resolve conflicts, with these proposed changes subject to review and approval by the Board.

Section 3. Board's Role in Setting ODP Strategic Direction. The Board will set overall Project policy in consultation with the TSC. This policy will describe: ODP's scope (the aggregate scope of projects); ODP's technical vision and direction; and project release guidance to the TSC (e.g., deliver via regularly-scheduled release trains). Typically the Board will have

no say on technical issues, individual project scope & direction as long as they remain within the scope and direction of the policies established by the Board.

Section 4. Establishment of the TSC. The TSC will span the entire ODP. Initially, the TSC shall be formed from the representatives designated by the Platinum Members.

- The TSC members are the project leads from the Core projects. There is only one project lead per project. The active Committers will elect a number of TSC members; the exact number to be determined by the TSC after the organization is formed. In order to ensure a true community is in place prior to election, these TSC community seats will be elected six months after the organization forms. These elected community seats must be filled with individuals from separate, unaffiliated companies.
- If not otherwise represented, each Platinum member may designate a TSC member. This is expected to be a temporary measure and will be re-evaluated by the Board every year. If an event occurs causing a non-designated TSC member associated with a given Platinum Member to join the TSC, the designated TSC member must step down immediately, unless waived by a Super Majority Vote (as defined in the Bylaws) of the Board prior to the precipitating event and for a specified amount of time, not to exceed one year. After such specified amount of time the designated TSC member must step down, unless the waiver is renewed.
- The TSC will elect a Chair who will communicate with the Board. The Chair may attend Board meetings as non-voting observer.
- The term for TSC members, including the Chair, is one year. There are no term limits for TSC members.

Section 5. Responsibilities of the TSC. Subject to such policies as may be set by the Board, the TSC is responsible for simultaneous release dates, release quality standards, technical best practices (including the Development Process), monitoring technical progress, mediating technical conflicts between Committers and project leads, and organizing inter-project collaboration. The TSC will define ODP's release vehicles. The TSC will serve as ODP's liaisons with other consortiums and groups.

Section 6. ODP Operations. The TSC will establish the Development Process for ODP and present the Development Process to the Board for approval as the procedures are established or updated from time to time.

There will be multiple projects under ODP. Each project must be within such policies as may be set by the Board, have a well defined scope and must work within that scope. The Development Process will provide for projects to follow the life-cycle process as described in the Daylight Project Life Cycle document, the initial version of which is attached, which provides definition and context for the terms Incubator, Mature, and Core projects. The Development Process will include a process for the TSC to oversee and approve changes in the state of a project (e.g. Incubator to Mature or Mature to Core), which will include consideration of the following criteria:

- Cleanliness of code base
- Ample and diverse Contributors and Committers to assure vitality of the project
- Stability (e.g., presence of test suites and use of an appropriate source-code control system)
- Predictability of releases
- Alignment with ODP's goals

Incubator projects may be thought of as proposals for new projects. Proposals to create incubator projects may be made by anyone. Creation of incubator projects is subject to TSC approval in accordance with the Development Process: Creating a new incubator project is expected to be a lower hurdle than moving from Incubator to Core project.

The Development Process will include provision for a voting process to be implemented for decision making in accordance with the following guidelines:

- For election of persons (TSC chairs, project leaders, etc.) a multiple-candidate method should be used, e.g.:

Condorcet: http://en.wikipedia.org/wiki/Condorcet_method

or

Single Transferable Vote: http://en.wikipedia.org/wiki/Single_transferable_vote

- Multiple-candidate methods reduce to simple majority when there is only one position to be filled.
- For project internal decisions where no consensus can be reached, simple majority vote by Committers via +1 voting should be used.
- Simple majority voting should be used for decisions within the TSC, unless otherwise specified in Development Process.

The Development Process will include such processes as may be specified by the Board from time to time relating to the intake and license compliance review of contributions.

Section 7. Project Roles. Each project has one or more Contributors, who produce code, and one or more Committers, who control technical direction. Each project will be headed up by a project lead who sets overall direction for the project and reports to the TSC. Contributors and Committers, including project leads, remain in the pay of their respective employers.

Committers: For each project there is a set of people with rights to commit code to the source code management system: the Committers.

- The Committers will be the decision makers on design, code, and patches for their project. They must responsibly participate in the consensus decisions of the TSC.
- Committer rights are earned via code contribution and community trust. Committers select and vote for new Committers, subject to TSC approval. A standard meritocracy model with new Committers will be approved and implemented by the TSC which will include provision for fully open code submission, review, acceptance, build, test, delivery, and support model.
- Committer rights are per project; being a Committer on one project does not necessarily give an individual committer rights on any other project.

- Initial Committers will be specified at project creation. Additional Committers will be admitted by a vote of existing Committers with appropriate process to handle dissent.
- Committers are not necessarily from Member companies. Committers are the best available individuals, but usually full-time for any components in active development.
- The Committers will use the process established in the Daylight Project Life Cycle document maintained by the TSC in its Development Process to accept/force modifications/reject code submissions and to add/delete Committers (and other development details).
- Initial projects that form ODP's base may designate Committers. In order to preserve meritocracy in selection of Committers while ensuring diversity of Committers, each initial project will commit to taking on at least three Committers not from the company of origin within the first three months after ODP is publicly launched based upon evaluation of participation of Contributors during that time.
- A Committer who is disruptive, or has been inactive for an extended period (e.g., six or more months) may have his or her Committer status revoked by the project leads.

Contributors. Most Contributors work with their Committer and their component's sub-community. They contribute code or other artifacts, but do not have the right to commit to the code base. A Contributor may be promoted to a Committer by the projects' Committers. The Contributors should rarely be encumbered by the TSC and never by the Board.

Project Leads. The project lead is a Committer selected by vote from the Committers in the project. If there is initially only one member of the project, then that member is automatically the project lead. It is possible, and in some cases desirable, for one person to take on roles of project lead, Committer, and Contributor.

Appendix D

TSC Policy

The Board of Directors of OpenDaylight has set in place the following policies to govern the operation of the Technical Steering Committee (TSC). These policies relate to release processes, technical scope, business goals, etc.. The TSC will, to the best of its ability, adhere to these policies. In cases where the TSC makes a judgment that the goals of OpenDaylight are better served by making exceptions to these policies, it is expected that the TSC will make these exceptions and indicate their reasons to the board at the next board meeting,

The following relate to the projects initiated by the TSC and the artifacts created therein.

- Singularity: To the extent possible, there should be no overlap between the goals of the core projects and artifacts created by them
- Cohesiveness: The artifacts created within each mature project should connect appropriately to other mature/core projects to form a cohesive system. (It is understood that this will not apply to artifacts that are stand-alone by design)

The following relate to the choice of projects, assignments of tasks and delivery of code.

- Simultaneous Release: The TSC is responsible for organizing a simultaneous release of appropriate OpenDaylight Projects at regular intervals. The following relate to the operation of the TSC.
- Communication: All communication between and within projects will be in a fair, open and consistent fashion.

- Openness: The TSC should ensure that all technical decisions are made in an open and transparent fashion.

The following relates to the scope of the projects within OpenDaylight:

- Scope: Projects chosen by TSC are limited to the following areas:
 - The OpenDaylight controller
 - Software for forwarding elements
 - SB plugins to enable the controller to speak to the OpenDaylight supplied and other network elements
 - Northbound plugins to expose interfaces to those writing applications to the controller
 - Network services and applications intended to run on top of the controller, integration between the controller and other elements, and
 - Support projects such as tools, infrastructure, or testing.
 - Plugins for inter-controller communication

Appendix E

Mail Lists

Mail List	Discussion
Aaa-dev	Authentication, Authorization and Accounting
affinity-announce	Affinity Meta-Data Service Announcements (low volume)
affinity-bugs	Notifications from Bugzilla for the Affinity Meta-Data Service
affinity-dev	Developer discussions for the Affinity Meta-Data Service
affinity-gerrit	Gerrit automated notices about the Affinity Meta-Data Service
affinity-jenkins	Jenkins CI notifications for the Affinity Meta-Data Service
affinity-users	Community driven support alias for the Affinity Meta-Data Service
bgpcep-announce	BGP/PCEP protocol library Announcements (low volume)
bgpcep-bugs	Notifications from Bugzilla for the BGP/PCEP protocol library
bgpcep-dev	Developer discussions for the BGP/PCEP protocol library
bgpcep-gerrit	Gerrit automated notices about the BGP/PCEP protocol library
bgpcep-jenkins	Jenkins CI notifications for the BGP/PCEP protocol library
bgpcep-users	Community driven support alias for the BGP/PCEP protocol library

bugs	Notifications from Bugzilla (topic based)
controller-announce	OpenDaylight Controller Announcements (low volume)
controller-bugs	Notifications from Bugzilla for the OpenDaylight Controller.
controller-dev	Developer discussions for the Open Daylight Controller.
controller-gerrit	Gerrit automated notices about the OpenDaylight Controller.
controller-jenkins	Jenkins CI notifications for the OpenDaylight Controller
controller-users	Community driven support alias for the OpenDaylight Controller.
defense4all-announce	Defense4All Announcements (low volume)
defense4all-bugs	Notifications from Bugzilla for Defense4All
defense4all-dev	Developer discussions for Defense4All
defense4all-gerrit	Gerrit automated notices about Defense4All
defense4all-jenkins	Jenkins CI notifications for Defense4All
defense4all-users	Community driven support alias for Defense4All
demos-and-pocs	Demos and Proofs of Concepts
dev	OpenDaylight Development lists (very high traffic)
discovery-dev	Devloper discussions for the discovery service
Discuss	OpenDaylight cross project discussion
dlux-announce	OpenDaylight UX (dlux) Announcements (low volume)
dlux-bugs	Notifications from Bugzilla for the OpenDaylight UX (dlux)
dlux-dev	Developer discussions for the OpenDaylight UX (dlux)
dlux-gerrit	Gerrit automated notices about the OpenDaylight UX (dlux)
dlux-jenkins	Jenkins CI notifications for the OpenDaylight UX (dlux)
dlux-users	Community driven support alias for the OpenDaylight UX (dlux)
documentation	OpenDaylight Documentation Workgroup

groupbasedpolicy-announce	Group Based Policy Plugin Announcements (low volume)
groupbasedpolicy-bugs	Notifications from Bugzilla for the Group Based Policy Plugin
groupbasedpolicy-dev	Developer discussions for the Group Based Policy Plugin
groupbasedpolicy-gerrit	Gerrit automated notices about the Group Based Policy Plugin
groupbasedpolicy-jenkins	Jenkins CI notifications for the Group Based Policy Plugin
groupbasedpolicy-users	Community driven support alias for the Gropup Based Policy Plugin
incntre	[no description available]
infrastructure	Discussions about the build architecture and pipeline
integration-announce	Integration Framework Announcements (low volume)
integration-bugs	Notifications from Bugzilla for the Integration Framework
integration-dev	Developer discussions for the Integration Framework
integration-gerrit	Gerrit automated notices about the OpenDaylight Integration framework
integration-jenkins	Jenkins CI notifications for the Integration Framework
integration-users	Community driven support alias for the Integration Framework
jenkins	Jenkins automated messages [moderate - high traffic]
L2switch-dev	[no description available]
lispflowmapping-announce	OpenDaylight LISP Mapping Service Announcements (low volume)
lispflowmapping-bugs	Notifications from Bugzilla for the OpenDaylight LISP Mapping Service
lispflowmapping-dev	Developer discussions for the OpenDaylight LISP Mapping Service

lispflowmapping-gerrit	Gerrit automated notices about the OpenDaylight LISP Mapping Service
lispflowmapping-jenkins	Jenkins CI notifications for the OpenDaylight LISP Mapping Service
lispflowmapping-users	Community drive support alias for the OpenDaylight LISP Mapping Service
net-virt-platform-announce	OpenDaylight SDN Controller Platform Announcements (low volume)
net-virt-platform-bugs	Notifications from Bugzilla for the OpenDaylight SDN Controller Platform.
net-virt-platform-dev	Developer discussions for the Open Daylight SDN Controller Platform.
net-virt-platform-gerrit	Gerrit automated notices about the OpenDaylight SDN Controller Platform
net-virt-platform-jenkins	Jenkins CI notifications for the OpenDaylight SDN Controller Platform
net-virt-platform-users	Community driven support alias for the OpenDaylight SDN Controller Platform.
Odlparent-dev	[no description available]
opendaylight-announce	OpenDaylight Announcements (low volume)
opendaylight-users	OpenDaylight community support
opendove-announce	Open DOVE Announcements (low volume)
opendove-bugs	Notifications from Bugzilla for Open DOVE
opendove-dev	Developer discussions for Open DOVE
opendove-gerrit	Gerrit automated notices about Open DOVE
opendove-jenkins	Jenkins CI notifications for Open DOVE
opendove-users	Community driven support alias for Open DOVE
openflowjava-announce	OpenDaylight Openflow Protocol Library (java) Announcements (low volume)

openflowjava-bugs	Notifications from Bugzilla for the OpenDaylight Openflow Protocol Library (java)
openflowjava-dev	Developer discussions for the OpenDaylight Openflow Protocol Library (java)
openflowjava-gerrit	Gerrit automated notices about the OpenDaylight Openflow Protocol Library (java)
openflowjava-jenkins	Jenkins CI notifications for the OpenDaylight Openflow Protocol Library (java)
openflowjava-users	Community driven support alias for the OpenDaylight Openflow Protocol Library (java)
openflowplugin-announce	OpenFlow Plugin Announcements (low volume)
openflowplugin-bugs	Notifications from Bugzilla for OpenFlow Plugin
openflowplugin-dev	Developer discussions for the OpenFlow Plugin
openflowplugin-gerrit	Gerrit automated notices about the OpenFlow Plugin
openflowplugin-jenkins	Jenkins CI Notifications for the OpenFlow Plugin
openflowplugin-users	Community driven support alias for the OpenFlow Plugin
Opflex-dev	[no description available]
ovsdb-announce	OVSDB Announcements (low volume)
ovsdb-bugs	Notifications from Bugzilla for OVSDB
ovsdb-dev	Developer discussions for OVSDB
ovsdb-gerrit	Gerrit automated notices about OVSDB
ovsdb-jenkins	Jenkins CI notifications for OVSDB
ovsdb-users	Community driven support alias for OVSDB
Plugin2oc-dev	[no description available]
project-proposals	OpenDaylight project proposals to the TSC
release	Discussion of release milestones and coordination
Sdninterfaceapp-dev	[no description available]
Sfc-dev	[no description available]
Snbi-dev	[no description available]

snmp4sdn-announce	SNMP Plugin Announcements (low volume)
snmp4sdn-bugs	Notifications from Bugzilla for the SNMP Plugin
snmp4sdn-dev	Developer discussions for the SNMP Plugin
snmp4sdn-gerrit	Gerrit automated notices about the SNMP Plugin
snmp4sdn-jenkins	Jenkins CI notifications for the SNMP Plugin
snmp4sdn-users	Community driven support alias for the SNMP Plugin
Tcpmd5-dev	[no description available]
TSC	OpenDaylight Technical Steering Committee
TSC-bugs	Notifications from Bugzilla for the OpenDaylight TSC.
ttp-dev	Table Type Patterns OpenDaylight development
vtn-announce	OpenDaylight VTN Announcements (low volume)
vtn-bugs	Notifications from Bugzilla for the OpenDaylight VTN.
vtn-dev	Developer discussions for the OpenDaylight VTN.
vtn-gerrit	Gerrit automated notices about the OpenDaylight VTN
vtn-jenkins	Jenkins CI notifications for the OpenDaylight VTN
vtn-users	Community driven support alias for the OpenDaylight VTN.
yangtools-announce	OpenDaylight YANG Tools Announcements (low volume)
yangtools-bugs	Notifications from Bugzilla for the OpenDaylight YANG Tools
yangtools-dev	Developer discussions for the OpenDaylight YANG Tools
yangtools-gerrit	Gerrit automated notices about the OpenDaylight YANG Tools
yangtools-jenkins	Jenkins CI notifications for the OpenDaylight YANG Tools
yangtools-users	Community driven support alias for the OpenDaylight YANG Tools

Table E.1: OpenDaylight's OpenDove Project mail lists

Bibliography

- [Abo14] OpenDaylight Open Source Project About. Opendaylight open source project about. Website, 2014. <http://www.opendaylight.org/project/about>.
- [Ann14a] OpenDaylight Open Source Project Announcements. Announcements. Website, 2014. <http://www.opendaylight.org/news/announcements>.
- [Ann14b] OpenDaylight Open Source Project Announcements. Industry leaders collaborate on opendaylight project, donate key technologies to accelerate software-defined networking. Website, 2014. <http://www.linuxfoundation.org/news-media/announcements/2013/04/industry-leaders-collaborate-opendaylight-project-donate-key>.
- [Ann14c] OpenDaylight Open Source Project Announcements. Infinera and kemp join the opendaylight project to enable open software-defined networking. Website, 2014. <http://www.opendaylight.org/announcements/2014/09/infinera-and-kemp-join-opendaylight-project-enable-open-software-defined>.
- [Ann14d] OpenDaylight Open Source Project Announcements. Opendaylight announces hp is now a platinum member. Website, 2014. .
- [Ann14e] OpenDaylight Open Source Project Announcements. Opendaylight announces summer internship program for student developers. Website, 2014. <http://www.opendaylight.org/announcements/2014/04/opendaylight-announces-summer-internship-program-student-developers>.

- [Ann14f] OpenDaylight Open Source Project Announcements. Opendaylight is the first open source project to receive interop event grand prize award. Website, 2014. <http://www.opendaylight.org/announcements/2014/04/opendaylight-first-open-source-project-receive-interop-event-grand-prize>.
- [Ann14g] OpenDaylight Open Source Project Announcements. Resources. Website, 2014. <http://www.opendaylight.org/resources>.
- [Ask14] OpenDaylight Open Source Project Ask. Opendaylight ask. Website, 2014. <https://ask.opendaylight.org/questions>.
- [Aud12] Citizen Audit. Linux foundation 2012 return of organization exempt from income tax. Website, 2012. https://www.citizenaudit.org/460503801/#2013_12_EO/46-0503801_9900_201212.pdf.
- [Ber14] Peter Bernstein. Opendaylight means opening sdn and nfv for business. Website, 2014. <http://www.nfvzone.com/topics/nfv/articles/386695-opendaylight-means-opening-sdn-nfv-business.htm>.
- [Bit14] Bitergia. Sibyl: Question&answer analyzer. Website, 2014. <https://github.com/MetricsGrimoire/Sibyl.git>.
- [Bug14a] OpenDaylight Open Source Project Bugs. Opendaylight bugs. Website, 2014. https://wiki.opendaylight.org/view/OpenDaylight_Bugs.
- [Bug14b] OpenDaylight Open Source Project Bugzilla. Opendaylight bugzilla. Website, 2014. <https://bugs.opendaylight.org/>.
- [bug14c] bugzilla.org. Bugzilla. Website, 2014. <https://www.bugzilla.org>.
- [Car14a] GSyC LibreSoft (Universidad Rey Juan Carlos). Bitergia. Website, 2014. <http://metricsgrimoire.github.io/>.
- [Car14b] GSyC LibreSoft (Universidad Rey Juan Carlos). Bug tracker parser: Bicho. Website, 2014. <https://github.com/MetricsGrimoire/Bicho.git>.

- [Car14c] GSyC LibreSoft (Universidad Rey Juan Carlos). Data analysis: Cvsanaly. Website, 2014. <https://github.com/MetricsGrimoire/CVSAAnaly.git>.
- [Car14d] GSyC LibreSoft (Universidad Rey Juan Carlos). Mail list parser: Mailingliststats. Website, 2014. <https://github.com/MetricsGrimoire/MailingListStats.git>.
- [CH14] Kevin Crowston and James Howison. The social structure of free and open source software model. Website, 2014. <http://firstmonday.org/ojs/index.php/fm/article/view/1207/1127>.
- [CI14] Jenkins CI. Jenkins. Website, 2014. <https://www.jenkins-ci.org>.
- [Col14] OpenDaylight Open Source Project Collateral. Collateral. Website, 2014. <http://www.opendaylight.org/resources/publications>.
- [CTL09] Inc Cable Television Laboratories. Data over cable service interface specifications. Web, 2009. <http://www.cablelabs.com/wp-content/uploads/specdocs/CM-SP-PHYv3.0-I08-090121.pdf>.
- [Eri14] Ericsson. Nfv and sdn: Has the hype curve peaked? Website, 2014. <http://www.ericsson.com/research-blog/cloud/nfv-sdn-hype-curve-peaked>.
- [ETS13] ETSI. Network functions virtualisation - introductory white paper definition. Website, 2013. http://portal.etsi.org/NFV/NFV_White_Paper.pdf.
- [Fac14] OpenDaylight Open Source Project Facebook. Opendaylight project. Website, 2014. <https://www.facebook.com/pages/OpenDaylight-Project/309147425875291>.
- [FAQ14] OpenDaylight Open Source Project FAQ. Faq. Website, 2014. <http://www.opendaylight.org/project/faq>.
- [fJN14] ACG Research for Juniper Networks. Business case for nfv/sdn programmable networks. Website, 2014. <http://www.slideshare.net/junipernetworks/acg-research-paper>.

- [For06] Internet Engineering Task Force. Rfc 4271: A border gateway protocol 4 (bgp-4). Website, 2006. <https://tools.ietf.org/html/rfc4271>.
- [For09] Internet Engineering Task Force. Rfc 5540: Path computation element (pce) communication protocol (pcep). Website, 2009. <https://tools.ietf.org/html/rfc5540>.
- [For13a] Internet Engineering Task Force. Open vswitch. Website, 2013. <http://openvswitch.org>.
- [For13b] Internet Engineering Task Force. Rfc 6830: The locator/id separation protocol (lisp). Website, 2013. <http://tools.ietf.org/search/rfc6830>.
- [For13c] Internet Engineering Task Force. Rfc 7047: The open vswitch database management protocol. Web, 2013. <https://tools.ietf.org/html/rfc7047>.
- [For14a] Internet Engineering Task Force. Opflex control protocol draft. Web, 2014. <http://tools.ietf.org/html/draft-smith-opflex-00>.
- [For14b] Internet Engineering Task Force. Packetcable(tm) specification multimedia specification. Web, 2014. <http://www.cablelabs.com/wp-content/uploads/specdocs/PKT-SP-MM-I05-091029.pdf>.
- [For14c] Internet Engineering Task Force. Rfc 6020: Yang - a data modeling language for the network configuration protocol (netconf). Web, 2014. <http://tools.ietf.org/html/rfc6020>.
- [For14d] Internet Engineering Task Force. Rfc 6241: Network configuration protocol (netconf). Web, 2014. <http://tools.ietf.org/html/rfc6241>.
- [Fou04] Eclipse Foundation. Eclipse public license - v 1.0. Website, 2004. <https://www.eclipse.org/legal/epl-v10.html>.
- [Fou12] Open Networking Foundation. Openflow whitepaper. Website, 2012. <https://www.opennetworking.org/images/stories/downloads/sdn-resources/white-papers/wp-sdn-newnorm.pdf>.

- [Fou14a] Free Software Foundation. Eclipse public license version 1.0 (epl). Website, 2014. <http://www.gnu.org/licenses/license-list.html#EPL>.
- [Fou14b] Free Software Foundation. Free software foundation about. Website, 2014. <http://www.fsf.org/about>.
- [Fou14c] Free Software Foundation. Various licenses and comments about them. Website, 2014. <http://www.gnu.org/licenses/license-list.html>.
- [Fou14d] Free Software Foundation. What is copyleft. Website, 2014. <http://www.gnu.org/copyleft>.
- [Fou14e] Linux Foundation. Linux foundation collaborative projects. Website, 2014. <http://collabprojects.linuxfoundation.org>.
- [Fou14f] Linux Foundation. Opendaylight summit 2014 edition. Website, 2014. <http://events.linuxfoundation.org/events/archive/2014/opendaylight-summit>.
- [Fou14g] Open Networking Foundation. Sdn definition. Website, 2014. <https://www.opennetworking.org/sdn-resources/sdn-definition>.
- [Ger14a] Gerrit. Gerrit. Website, 2014. <https://code.google.com/p/gerrit/>.
- [Ger14b] OpenDaylight Open Source Project Gerrit. Opendaylight gerrit. Website, 2014. <https://git.opendaylight.org/gerrit>.
- [Gig14a] Gigaom. Gigaom research. Website, 2014. <http://research.gigaom.com>.
- [Gig14b] OpenDaylight Gigaom. Sdn, nfv, and open source: The operator's view. Website, 2014. <http://go.linuxfoundation.org/l/6342/pub-odp-gigaom-190314/y6dyw>.
- [Gro14] OpenDaylight Open Source Project User Groups. Opendaylight user groups. Website, 2014. <http://www.opendaylight.org/community/opendaylight-user-groups-odlug>.

- [Gui14a] OpenDaylight Open Source Project Development Getting Started Guide. Open-daylight development getting started guide. Website, 2014. https://wiki.opendaylight.org/view/GettingStarted:Developer_Main.
- [Gui14b] OpenDaylight Open Source Project Getting Started Guide. Opendaylight getting started guide. Website, 2014. <http://www.opendaylight.org/resources/getting-started-guide>.
- [Gui14c] OpenDaylight Open Source Project Getting Started Guide. Opendaylight project getting started guide. Website, 2014. https://wiki.opendaylight.org/view/GettingStarted:Project_Main.
- [IDC14] IDC. Growing sdn momentum presents fresh opportunities for datacenter networks. Website, 2014. <http://www.idc.com/getdoc.jsp?containerId=prUS25052314>.
- [Ide14] OpenDaylight Open Source Project Identity. Opendaylight identity. Website, 2014. <https://identity.opendaylight.org/carbon/admin/login.jsp>.
- [Inc] Gartner Inc. Gartner hype cycle. Website. <http://www.gartner.com/technology/research/methodologies/hype-cycle.jsp>.
- [Ini98a] Open Source Initiative. Open source initiative. Website, 1998. <http://opensource.org/about>.
- [Ini98b] Open Source Initiative. Open source licenses. Website, 1998. <http://opensource.org/licenses>.
- [Ini14] Open Source Initiative. What is copyleft? is it the same as open source? Website, 2014. <http://opensource.org/faq#copyleft>.
- [Jen14] OpenDaylight Open Source Project Jenkins. Opendaylight jenkins. Website, 2014. <https://jenkins.opendaylight.org>.
- [Joi14] OpenDaylight Open Source Project Join. Opendaylight project join. Website, 2014. <http://www.opendaylight.org/project/join>.

- [Lab14] OpenDaylight Open Source Project Community Labs. Community labs. Website, 2014. <http://www.opendaylight.org/community/community-labs>.
- [Lic14] OpenDaylight Open Source Project Licensing. 5. what license will the project use? why was that decision made? Website, 2014. <http://www.opendaylight.org/project/faq#5>.
- [Lif14] OpenDaylight Open Source Project Lifecycle. Opendaylight project lifecycle. Website, 2014. <http://www.opendaylight.org/project-lifecycle-releases>.
- [Lin14] OpenDaylight Open Source Project LinkedIn. Opendaylight project. Website, 2014. <https://www.linkedin.com/company/opendaylight-project>.
- [Lis14] OpenDaylight Open Source Project Mail Lists. Opendaylight mail lists. Website, 2014. <https://lists.opendaylight.org/mailman/listinfo>.
- [mat14a] matplotlib. matplotlib: a python 2d plotting library. Website, 2014. <http://matplotlib.org>.
- [Mat14b] Craig Matsumoto. Service-provider sdn market to approach \$16b by 2018. Website, 2014. <https://www.sdncentral.com/news/service-provider-sdn-market-16b-2018/2014/01/>.
- [Mem14] OpenDaylight Open Source Project Members. Opendaylight members. Website, 2014. <http://www.opendaylight.org/project/members>.
- [Met14a] MetricsGrimoire. Bug tracker parser: Bicho: Database schema. Website, 2014. <https://github.com/MetricsGrimoire/Bicho/wiki/Database-Schema>.
- [Met14b] MetricsGrimoire. Data analysis: Cvsanaly: Database schema. Website, 2014. <https://github.com/MetricsGrimoire/CVSAnaly/wiki/Database-Schema>.

- [Met14c] MetricsGrimoire. Mail list parser: Database schema. Website, 2014. <https://github.com/MetricsGrimoire/MailingListStats/wiki/Database-Schema>.
- [Net14] Software Defined Cloud Networking. Tag archives: Sdn. Website, 2014. <http://dcxtc.com/tag/sdn>.
- [Nex14] OpenDaylight Open Source Project Nexus. Opendaylight nexus. Website, 2014. <https://nexus.opendaylight.org>.
- [OPfNP14] Inc. Open Platform for NFV Project. Open platform for nfv. Website, 2014. <https://www.opnfv.org>.
- [Ove14] OpenDaylight Open Source Project Technical Overview. Opendaylight technical overview. Website, 2014. <http://www.opendaylight.org/project/technical-overview>.
- [Par14] OpenDaylight Open Source Project Participate. How to participate. Website, 2014. <http://www.opendaylight.org/community/how-participate>.
- [Pic14] Pica8. Sdn is hard to use / complete solution wanted. Website, 2014. <http://dcxtc.com/2014/04/05/infographic-sdn-is-hard-to-use-complete-solution-wanted>.
- [Plu14] OpenDaylight Open Source Project Google Plus. Opendaylight project. Website, 2014. <https://plus.google.com/u/0/+OpenDaylightOrg/posts>.
- [PP14] SDNCentral, Plexxi, and Lightspeed Venture Partners. Infographic: Sdn market size to reach \$35b by 2018. Website, 2014. <https://www.sdncentral.com/infographic-sdn-market-to-reach-35b-by-2018>.
- [Pre14] OpenDaylight Open Source Project Presentations. Opendaylight presentations. Website, 2014. https://wiki.opendaylight.org/view/OpenDaylight_Presentations:Main.
- [Pro14a] OpenDaylight Open Source Project Ambassador Program. Introducing the opendaylight ambassador program. Website, 2014. [http:](http://)

- //www.opendaylight.org/blogs/2014/07/introducing-opendaylight-ambassador-program.
- [Pro14b] OpenDaylight Open Source Project Internship Program. Opendaylight summer internship program. Website, 2014. <http://www.opendaylight.org/community/summer-intern-program>.
- [Pro14c] OpenDaylight Open Source Project. Bylaws. Website, 2014. <http://www.opendaylight.org/project/bylaws>.
- [Pro14d] OpenDaylight Open Source Project. Opendaylight open source project. Website, 2014. <http://www.opendaylight.org>.
- [Pro14e] OpenDaylight Open Source Project. Opendaylight open source project. Website, 2014. <http://www.opendaylight.org/project>.
- [Pub14] OpenDaylight Open Source Project Publications. Publications. Website, 2014. <http://www.opendaylight.org/resources/publications>.
- [Rie07] Dirk Riehle. The economic motivation of open source software: Stakeholder perspectives. Website, 2007. <http://dirkriehle.com/computer-science/research/2007/computer-2007-article.html>.
- [RNe14] RNewsire. Global sdn industry and sddc market forecasts in research reports. Website, 2014. <http://www.prnewswire.com/news-releases/global-sdn-industry-and-sddc-market-forecasts-in-research-reports-275885621.html>.
- [Sof14] TIOBE Software. Tiobe index for october 2014. Website, 2014. <http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html>.
- [Son14a] OpenDaylight Open Source Project Sonar. Opendaylight sonar. Website, 2014. <https://sonar.opendaylight.org>.
- [Son14b] SonarSource. Sonarqube. Website, 2014. <http://www.sonarqube.org>.
- [Son14c] Sonatype. Nexus. Website, 2014. <http://www.sonatype.com/nexus>.

- [Sys89] GNU/Linux Operating System. The gnu awk user's guide. Website, 1989. <http://www.gnu.org/software/gawk/manual/gawk.html>.
- [Tel14] Tellabs. Sdn-based backhaul applications will save \$4b in capex. Website, 2014. <http://info.tellabs.com/SDN>.
- [TM14a] AskBot TM. Askbot. Website, 2014. <https://askbot.com>.
- [TM14b] Java TM. Learn about java technology. Website, 2014. <https://www.java.com/en/about>.
- [Too14] OpenDaylight Open Source Project Tools. Opendaylight tools. Website, 2014. <http://www.opendaylight.org/community/tools>.
- [TSC14a] OpenDaylight Open Source Project TSC. Tsc charter. Website, 2014. <http://www.opendaylight.org/project/tsc/charter>.
- [TSC14b] OpenDaylight Open Source Project TSC. Tsc policy. Website, 2014. <http://www.opendaylight.org/project/tsc/tsc-policy>.
- [Twi14] OpenDaylight Open Source Project Twitter. Opendaylight project. Website, 2014. <https://twitter.com/OpenDaylightSDN>.
- [VMW12] VMWare. Vmware to acquire nicira. Website, 2012. <http://www.vmware.com/company/news/releases/vmw-nicira-07-23-12>.
- [W3C14] W3C. Rest discussion draft. Website, 2014. <http://www.w3.org/2001/sw/wiki/REST>.
- [Whe14] David A. Wheeler. A set of tools for counting physical source lines of code (sloc). Website, 2014. <http://www.dwheeler.com/sloccount>.
- [Wik14a] OpenDaylight Open Source Project Wiki. Opendaylight aaa wiki. Website, 2014. <https://wiki.opendaylight.org/view/AAA:Main>.
- [Wik14b] OpenDaylight Open Source Project Wiki. Opendaylight affinity metadata service wiki. Website, 2014. https://wiki.opendaylight.org/view/Affinity_Metadata_Service:Main.

- [Wik14c] OpenDaylight Open Source Project Wiki. Opendaylight bgp-ls/pcep wiki. Website, 2014. https://wiki.opendaylight.org/view/Main_Page.
- [Wik14d] OpenDaylight Open Source Project Wiki. Opendaylight controller wiki. Website, 2014. https://wiki.opendaylight.org/view/OpenDaylight_Controller:Main.
- [Wik14e] OpenDaylight Open Source Project Wiki. Opendaylight defense4all project wiki. Website, 2014. https://wiki.opendaylight.org/view/Main_Page.
- [Wik14f] OpenDaylight Open Source Project Wiki. Opendaylight dlux wiki. Website, 2014. https://wiki.opendaylight.org/view/OpenDaylight_dlux:Main.
- [Wik14g] OpenDaylight Open Source Project Wiki. Opendaylight documentation project wiki. Website, 2014. https://wiki.opendaylight.org/view/CrossProject:Documentation_Group.
- [Wik14h] OpenDaylight Open Source Project Wiki. Opendaylight dynamic resource reservation project wiki. Website, 2014. <https://wiki.opendaylight.org/view/Reservation:Main>.
- [Wik14i] OpenDaylight Open Source Project Wiki. Opendaylight group policy project wiki. Website, 2014. https://wiki.opendaylight.org/view/Group_Policy:Main.
- [Wik14j] OpenDaylight Open Source Project Wiki. Opendaylight integration group project wiki. Website, 2014. https://wiki.opendaylight.org/view/CrossProject:Integration_Group.
- [Wik14k] OpenDaylight Open Source Project Wiki. Opendaylight l2 switch project wiki. Website, 2014. https://wiki.opendaylight.org/view/L2_Switch:Main.
- [Wik14l] OpenDaylight Open Source Project Wiki. Opendaylight lisp flow mapping. Website, 2014. https://wiki.opendaylight.org/view/OpenDaylight_Lisp_Flow_Mapping:Main.

- [Wik14m] OpenDaylight Open Source Project Wiki. Opendaylight open dove project wiki. Website, 2014. https://wiki.opendaylight.org/view/Open_Dove:Main.
- [Wik14n] OpenDaylight Open Source Project Wiki. Opendaylight openflow plugin project wiki. Website, 2014. https://wiki.opendaylight.org/view/OpenDaylight_OpenFlow_Plugin:Main.
- [Wik14o] OpenDaylight Open Source Project Wiki. Opendaylight openflow protocol library project wiki. Website, 2014. https://wiki.opendaylight.org/view/Openflow_Protocol_Library:Main.
- [Wik14p] OpenDaylight Open Source Project Wiki. Opendaylight opflex project wiki. Website, 2014. <https://wiki.opendaylight.org/view/OpFlex:Main>.
- [Wik14q] OpenDaylight Open Source Project Wiki. Opendaylight oscp project wiki. Website, 2014. [https://wiki.opendaylight.org/view/OpenDaylight_SDN_Controller_Platform_\(OSCP\):Main](https://wiki.opendaylight.org/view/OpenDaylight_SDN_Controller_Platform_(OSCP):Main).
- [Wik14r] OpenDaylight Open Source Project Wiki. Opendaylight packetcable pcmm project wiki. Website, 2014. <https://wiki.opendaylight.org/view/PacketCablePCMM:Main>.
- [Wik14s] OpenDaylight Open Source Project Wiki. Opendaylight secure network bootstrapping infrastructure (snbi) project wiki. Website, 2014. <https://wiki.opendaylight.org/view/SecureNetworkBootstrapping:Main>.
- [Wik14t] OpenDaylight Open Source Project Wiki. Opendaylight snmp4sdn project wiki. Website, 2014. <https://wiki.opendaylight.org/view/SNMP4SDN:Main>.
- [Wik14u] OpenDaylight Open Source Project Wiki. Opendaylight table type patterns (ttps)/negotiable datapath models (ndms) project wiki. Website, 2014. https://wiki.opendaylight.org/view/Table_Type_Patterns:Main.

- [Wik14v] OpenDaylight Open Source Project Wiki. Opendaylight toolkit project wiki. Website, 2014. https://wiki.opendaylight.org/view/OpenDaylight_Toolkit:Main.
- [Wik14w] OpenDaylight Open Source Project Wiki. Opendaylight virtual tenant network (vtm) project wiki. Website, 2014. [https://wiki.opendaylight.org/view/OpenDaylight_Virtual_Tenant_Network_\(VTN\):Main](https://wiki.opendaylight.org/view/OpenDaylight_Virtual_Tenant_Network_(VTN):Main).
- [Wik14x] OpenDaylight Open Source Project Wiki. Opendaylight wiki. Website, 2014. https://wiki.opendaylight.org/view/Main_Page.
- [Wik14y] OpenDaylight Open Source Project Wiki. Opendaylight yang tools project wiki. Website, 2014. https://wiki.opendaylight.org/view/YANG_Tools:Main.
- [Wil09] Rowan Wilson. The eclipse public license - an overview. Website, October 2009. <http://oss-watch.ac.uk/resources/epl>.
- [WSO14] WSO2. Wso2 identity server. Website, 2014. <http://wso2.com/products/identity-server>.
- [You14] OpenDaylight Open Source Project Youtube. Opendaylight project. Website, 2014. <https://www.youtube.com/user/opendaylightproject>.