# Why do innovation specialist firms engage with Open Source projects?

Master Thesis MSc. Business Administration


**Date:** 03-01-2017

**Author:** Jeroen Maarten Boon

S0192562

**Contact:** jeroenboon@gmail.com


UNIVERSITY OF TWENTE.

**Supervisors University of Twente:**
Dr. R. van Reekum
Dr. Ir. S.J.A. Löwik

**Supervisors Technical University Berlin:**
Dipl.-Kfm. M. Böhm
Prof. Dr. K. Blind

ERICSSON

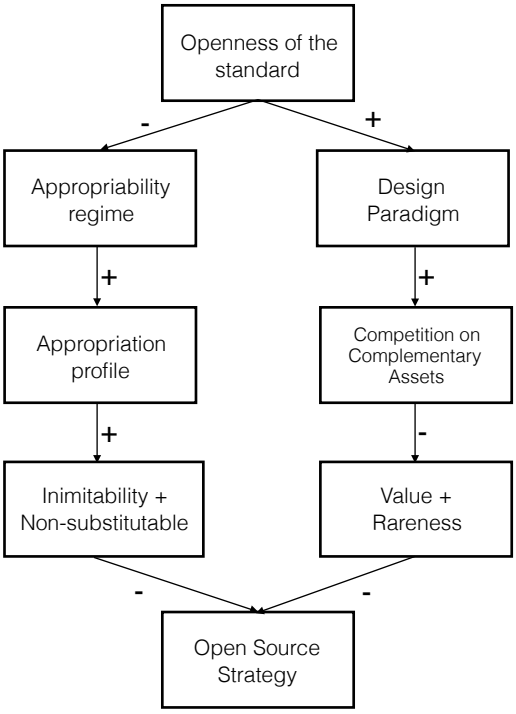
**Supervisors Ericsson AB**
Dr. C. Tapia Garcia
J. Ahlberg

# Summary

Open Source Software (OSS) is increasing in popularity. Proponents argue that OSS leads to a higher rate of innovation, is less prone to bugs, and has a faster adoption rate than proprietary software. However, within OSS a company cannot rely on traditional means of exclusive ownership of particular resources in order to gain a competitive advantage. Therefore, it is important for a firm to have a good understanding of what resources it can afford to share to Open Source projects, and which resources should be kept within the firm. The purpose of this research was to answer the following question: *"What variables influence organizations to choose Open Source activities over proprietary development?".* While Open Source is becoming a very common topic for many different technology firms, research on the topic is lacking. Existing studies primarily describe a few success stories, or in what ways companies engage with Open Source projects. However, to date no study has investigated the reasons for companies to do so and the situations in which it makes sense for a company to engage with Open Source projects.

In order to answer this question a theoretical model was created based on Teece's Profiting from Innovation (PFI) framework combined with the Resource Based View (RBV). The model provides an in-depth understanding of variables and their relationships. This model has been tested using a multiple case study approach, using five cases within the mobile telecommunications industry. Triangulation of data was used to answer the research question, including data from annual reports, Open Source projects, Press releases and semi-structured interviews.

The theoretical model was further developed to account for the in depth knowledge about the variables and relationships provided by the case studies. This has resulted in the model presented in the figure on the right. The model provides clear insights into which variables are key to making Open Source decisions within a firm. Additionally, these insights have been linked to specific Open Source monetization strategies. While such strategies were developed in earlier studies, a guideline on when to use what type of strategy was lacking. The model can be used by managers to evaluate both the technological environment of the firm, as well as how their firm is configured towards this environment. Additionally, the model provides insights into the resources of the firm, and whether these are suited for Open Source activities. Not every software resource is suitable for an Open Source approach, there are those that are simply too specialized or valuable to the firm and the model indicates when this is the case. For those resources that are suited for Open Source activities the model provides insights on what type of Open Source monetization strategy to use.



# TABLE OF CONTENTS

# 1. Introduction

The use of Open Source Software (OSS) is growing exponentially (W3Techs, 2016; Statcounter, 2016) and despite the free nature of OSS, companies have been successful in building business models around it. However, in these cases the businesses either secured complementary assets to the OSS enabling them to capture a profit (Teece, 1986) or kept the core technology proprietary and created Open Source complementers (Henkel et al., 2014).

OSS tends to use and help define standards and specifications, as the availability of their source code promotes open, democratic debate around the specifications (European Interoperability Framework, 2004, p,10). Standard Defining Organizations (SDOs) are starting to explore working with OSS in future standardizing activities. In many aspects OSS and standards are complementary rather than competitive, each providing their own strengths (ETSI, 2016, p.4). As a result, innovation specialist firms are experiencing a growing demand to participate in Open Source projects. Yet, these firms focus on building core technologies, while depending on a strong appropriability regime in order to capture profits (Teece, 1986). OSS potentially threatens the protection of the intellectual property (IP) of these firms in two ways:

(1) IP licensing: In traditional (de jure) standardization it is clear who owns IP and the IP is licensed on FRAND terms (with or without monetary compensation). However, in Open Source projects it is unclear who owns IP (in software) and in which terms it will license. Additionally, the patent holder may choose not to license its IP if the holder was not involved in the Open Source project.

(2) Appropriability: Inclusion of OSS provides threats to the appropriability regime. Open Source licenses such as GPL determine that any derivative work must be licensed under the same conditions. This raises two issues for firms. First, this prevents exclusive access to the software. Second, companies involved in Open Source projects may be forced to give out licenses to proprietary technologies which are unfriendly to business (Rosen, 2004). These licenses threaten the ability of firms to charge licensing fees for their innovations, which is for many firms a prerequisite for engaging with open innovation (Chesbrough, 2003; Laursen and Salter, 2014).

Despite these challenges, Open Source can be an effective form of open innovation (West and Gallagher, 2006; Henkel, Schubert and Alexy, 2014) and generate significant commercial benefits (Afuah and Tucci, 2012; Baldwin and von Hippel, 2011; Henkel and Baldwin, 2011). Therefore, it is important for firms to make a conscious decision on whether to participate in OSS and what parts of its IP it is willing to reveal in exchange for these benefits (Henkel, et al. 2014). Existing studies on successful firm engagement in Open Source mainly focus on firms with strong complementary assets, or product vendors (Krishnamurthy, 2003; Henkel, 2006; O'reilly, 2007).

However, the environment is different for innovation specialists. Innovation specialist firms are focused on developing new technologies, but do not necessarily develop the implementations. This results in strong patent portfolio's and underdeveloped specialized complementary assets. Their profits are secured through appropriability and they employ legal teams in charge of securing licensing deals. As a result, they experience organizational inertia regarding Open Source and will be less likely to engage in Open Source activities. Yet, growing OSS demand pressures innovation specialists to participate.

This debate boils down to an important question first posed by Teece (1986) "who profits from innovation and why?". OSS may be free, but firms engaging in OSS development are doing so from business reasons and not altruistic ones. Using the PFI framework as a foundation this study will focus on how OSS development can be used to increase a firms' profit share from an innovation.

## 1.1 Research problem

The objective of this study is to contribute to the development of an Open Source strategy framework, with the focus on identifying which variables influence organizations to choose Open Source activities over proprietary development. This goal translates into the following research question: *what variables influence organizations to choose Open Source activities over proprietary development?* This research question is investigated through the following sub questions:

(1) What is Open Source Software?

(2) What are the main differences between proprietary software development and Open Source software development?

(3) What are the differences between Open Source development and Open Innovation?

(4) How do companies benefit from Open Source projects?

Open Source is still a very new and rare topic within management studies. Therefore the first three questions are aimed at creating a thorough theoretical understanding of Open Source and how it relates to firms. The fourth and fifth question are more strategic in nature. Building on the body of knowledge developed in the literature review a causal model will be developed which will be tested in the second part of this thesis.

## 1.2 Justification

There have been some studies into the business models around OSS and how firms can capture value from OSS related activities. However, these primarily resort to describing the business models of highly successful examples such as Red Hat and Google. Additionally, they primarily focus on a description of how these specific firms benefit from Open Source, rather than what antecedents lead to these firms engaging with Open Source projects. Typically, these firms are vertically integrated owning large parts of the value chain, making it possible for these firms to rely on complementary products or services to secure profits.

For innovation specialist firms the situation is different, their focus lies primarily on the development of new technology and they are less integrated in the supply chain. As a result they are reliant on the out-licensing of technology to generate income. Open Source potentially threatens this business model, which is why many innovation-specialist firms are reluctant to engage in Open Source development. However, there is a growing demand for Open Source participation, including from the clients of these innovation specialist firms. As a reaction to this increased demand open innovation networks such as the European Telecommunications Standards Institute (ETSI) have announced their first OSS projects, placing OSS in the direct environment of innovation specialists. Innovation specialists are formulating a response to this potential threat, but currently lack a framework for doing so. This study aims to explore the different variables that influence Open Source activities and provide a framework for determining Open Source strategy.

## 1.3 Methodology

The focus of this study lies on company strategies and activities regarding innovation and OSS development. These are topics that are highly dependent on subjective interpretations of different actors in different settings, e.g. two very similar firms may have very different attitudes towards OSS development. The context of each firm plays a strong role in shaping the firms' behavior, but the firms' decisions are actively shaping its context. This creates fuzzy boundaries between phenomenon and context. Moreover, the researcher has no control over the behavior of the firms under study. Therefore, the case study method offers the best methodological fit, as it is a strong tool in understanding real-life phenomena and relating them to their context.

The research design is a multiple-case holistic case study, analyzing five firms active in the telecommunications industry. The firms are selected for their differences in patent portfolio's and complementary assets ownership. The cases are selected to provide a realistic set of units, but the selection was heavily influenced by the availability of data and was not randomized. The telecommunications industry was chosen because of its high level of standardization and because it is one of the first industries where the SSO is engaging in OSS development. This high rate of standardization ensures that the context is very similar for each firm.

The data is primarily collected from publicly available sources such as annual reports, Open Source activity and community proceedings and press releases. Additionally, several interviews were conducted to provide both a confirmation of the data from secondary sources and to gain additional insights and facts.

## 1.4 Outline

Section 1 has laid the foundations for this report, introducing the research problem and research questions. The research was justified and a brief overview of the methodology was presented. The rest of this report is outlined as follows.

Section 2 consists of a literature review and model development. The chapter begins with the introduction of different aspects such as OSS and OSS licenses, open innovation and its link to OSS, and OSS and standardization. The second part of this chapter focusses on how businesses profit while engaging with OSS, the PFI framework and how it is affected by OSS, and key differences between OSS and closed source development. The chapter concludes with an overview of OSS strategies and the development of a theoretical framework.

Section 3 outlines the methodology. The first part focuses on explaining and justifying the case study methodology, defining the research design, and describing the sample selection. The second part of this chapter focusses on the operationalization of the different variables described in the theoretical framework, and the design of the comparative case study.

Section 4 contains all the relevant data collected through the case studies. Per firm an overview is provided for each of the technologies and Open Source projects the firms are engaged with.

In Section 5 the data from the case studies is analyzed and the propositions developed in Section 2 are tested.

Finally, in Section 6 the data and analysis is linked to the theoretical model and interpreted. The Section provides an answer to both the sub research questions and the research question of this thesis: *How and why do innovation specialist's firms decide to Open Source their R&D developments?* The second part of this chapter discusses the implications of these findings to the theoretical body of knowledge and implications for business policies.

# 2. Literature review

As Open Source development and its implications are a relatively new field in management, this literature review is rather extensive. The first part of this review provides a more general introduction into Open Source, the Open Source licenses and how Open Source ties into open innovation. From Section 2.4 onwards the review is more business focused. Section 2.4 describes how three companies that are successful in Open Source have built their business models through the lens of the Profiting From Innovation Framework. Section 2.5 goes into the differences between Open Source and proprietary software development from a production perspective, and Section 2.6 provides insights into current Open Source strategies. The presented body of knowledge is summarized and linked in Section 2.7, which concludes with the proposal of a theoretical model.

## 2.1 Open Source Software

Open Source software (OSS) is software whose source code is available for use, modification, or enhancement by anyone (opensource.com, 2016). In contrast, closed source software or proprietary software is software which is licensed under the exclusive legal right of its owner. It is purchasable by users, who will only receive a binary (non-readable) version of the program: therefore, modification by users is technically impossible. The original producer of the software must address bugs or new features. Not all OSS is free of charge, however. The source code is always freely available, but installation of the software can be restricted. For example, some OSS includes trademarks, which can only be used in installations of the software that are covered by a service license from the owner of the trademark. Of course this can be circumvented by altering the source code to remove the trademarked content, but this is the other 'not free' part of OSS. OSS comes as is, without support. Hence the users must do all the work themselves, which in turn costs time and resources. OSS differs from proprietary software in two ways: in its IP philosophy and how it is produced (West and O'Mahony, 2005).

Whereas proprietary software is only distributed in its compiled form, the original source code of OSS is also made publicly available. Yet, merely publishing the source code of software does not make it Open Source. For software to be truly Open Source, it must comply with four freedoms as defined by GNU. "*Free software" means software that respects users' freedom and community. Roughly, it means that the users have the freedom to run, copy, distribute, study, change and improve the software. Thus, 'free software' is a matter of liberty, not price"* (GNU, 2015).

The other difference to proprietary software lies in its production. Proprietary software is developed within a company, whereas OSS is developed via collaboration, out in the open. In theory everybody can contribute to OSS. In practice, whether the contributions will be included in the project is dependent on the community (West and Gallagher, 2006). Many communities have a lead-team that decides which contributions are included and which are not.

Closed source development is done through a specialized process, managed by a localized team of qualified developers, careful project management and occasional enhancements through updates and new releases. In contrast, Open Source is based on continuous improvement, collaboration among developers and users irrespective of geography and adherence to open standards (Raghunath an, Prasad, Misha and Chang, 2005, p.903). To facilitate collaborative development OSS is highly modular, making it easier for contributors to determine on that part of the project they can have a meaningful contribution. To illustrate, a Linux distribution is an operating system composed of a Linux kernel, GNU tools and libraries, a window system, a window manager, and a desktop environment. As such an Open Source end-product is often a combination or 'stack' of different other OSS.

The Open Source freedoms are enforced through licenses. A license is a legal term used to describe the way a copyright or patent owner grants permission to others to use its intellectual property (Raysman, Pisacreta & Adler, 2008). Software is covered by two types of intellectual property, copyright for the expression and patents for the underlying technological innovation. Not all software is patented, as this must be filed for specifically, but all software is automatically protected by copyright. OSS licenses give automatic permission to anyone who adheres to the terms to use the software as described by the license. While there are many variations of these copyright licenses, generally OSS licenses can be divided into two categories:

- Academic licenses: these licenses allow the software to be used for any purpose, with no obligation on part of the licensee to distribute the derivative works as Open Source (Rosen, 2004, p.70).
- Reciprocal licenses: these licenses also allow the software to be used for any purpose, under the requirement that any derivative work is distributed under the same Open Source license (Rosen, 2004, p.71).

The academic license is more friendly to business, allowing businesses to build proprietary software upon the OSS. Reciprocal licenses are considered more business unfriendly, excluding the sale of any derivative work and even requiring any derivatives to be made publicly available. under the same Open Source license. While some companies have found ways to construct a business model around Open Source projects with reciprocal licenses, this is quite rare. A more detailed discussion of these business models can be found in Section 2.4.

When a company contributes to OSS covered by a reciprocal license it is not allowed to charge a fee for distribution of the derivative (Rosen, 2004). This can be interpreted to cover the licensing of any relevant software patents as well. However, if a company owns patents covering parts of the software and it did not contribute to the software, it can seek licensing fees for its patents, as it did not enter the license agreement. Each license deals with patents in a different way. Some licenses only enforce inclusion of patents covering the contributions made by the firm, while others enforce a complete defensive suspension.

The defensive suspension implies that when a company using the OSS starts patent litigation against another user of the software (over functionality contained in the OSS), it will lose its license to the OSS. Take for example section 3 of the Apache v2.0 license: "*If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed*" (Apache Foundation, 2004).

To date none of the OSS licenses have been tested in court, as such their legal enforceability is hypothetical, especially concerning the different patent claims. Some of the OSS licenses stem from before software patents were possible and do not make any direct statements regarding patents, but they could be interpreted to include patents (Ahlberg, 2014). Therefore, firms must be very cautious about contributing to OSS and be prepared to also license their relevant patents for free when contributing to OSS.

### 2.1.1 Open Source Governance

On the one hand there are licenses which strongly influence the rights to the source code and determine how open the software is, but on the other hand there is a strong influence of the Open Source governance model on the creation of the software. The governance model used by a OSS project determines who influences the roadmap; how transparent the decision making process is; and how compliance requirements are enforced.

Currently the governance of many of the larger Open Source projects is being unified under foundations such as the Linux Foundation. However, there are still many differences in governance between Open Source projects. Some projects are tightly orchestrated by a small group of people, which is the case for Linux. The original creator and a select group of confidants determine which changes get eventually accepted in the new version of the kernel. Other (primarily smaller) projects are maintained by one person, who decides what is accepted and what not. Under the Linux Foundation the Open Source projects are managed much like SDOs are organized. Usually there is a democratically elected Technical Steering Committee, a Board of Directors and different Working Groups. Module contributions are discussed in the relevant Working Groups and when consensus is reached the contribution is submitted to the software.

The governance of a project determines how much influence firms can assert on the direction of the software, whether they have the same access as other contributing firms, if trademarks are used to enforce compliance, and much more. In order to fully understand the ramifications of contributing to OSS a firm must understand the governance structure. In some cases, a board position can be 'bought' by becoming a sponsor to the project, while in other cases all decisions are made by a single firm (e.g. Android).

## 2.2 Businesses and Open Source Software

The goal of a firm is to create a profit. As such, a firm's contributions to OSS are not made for altruistic reasons, rather there is some benefit justifying the companies' support. There have been various studies on the business models firms have created around OSS (e.g. Lerner and Tirole, 2002; Krishnamurthy, 2003; Goldman and Gabriel, 2005). Notable firms operating Open Source business models are Red Hat, Google, and Oracle. The Profiting from Innovation framework (Teece, 1986) provides insights in the workings of these business models.

### 2.2.1 Profiting from Innovation

The Profiting from Innovation (PFI) framework was introduced by David Teece in 1986. The framework was the first to combine perspectives from economic organization, business strategy, technology, and innovation. Teece's article pioneered the question "*under what conditions do firms profit from innovation?*" and provided an answer as well. The framework can be used to understand what conditions lead to an innovator profiting from its innovations, as in many situations this is not the case. In the PFI framework Teece introduces three main influencers on a firm's ability to profit from its innovations: the appropriability regime, the dominant design paradigm, and complementary assets.

#### Appropriability regime

Teece introduces the appropriability regime as a measure for the imitability of a technology. This measure is a function of both legal impediments (intellectual property rights) and inherent replicability of the technology. Appropriability and commercialization strategies are directly influenced by both the nature of the product and the intellectual property. A tight appropriability regime prevents imitation of the firm's products and creates a (temporary) monopoly allowing the firm to create a profit. A weak appropriability regime allows for imitation to take place and will shift profits away from the innovator.

Teece considered the appropriability regime as a 'given' factor — either weak or strong — that govern an innovator's ability to capture the profits generated by an innovation. This regime is contingent of the nature of the technology and the efficacy of legal mechanisms of protection. More recent discussions (e.g. Pisano, 2006) of PFI have highlighted that appropriability regimes are increasingly becoming the focus of strategies of firms. Phenomena such as OSS and the deliberate sharing of IP are ways for firms to alter the appropriability regime for a specific area of interest. Moreover, studies have shown that the strength of a patent or copyright can change (Sherry and Teece, 2003) and even the legal context can be redefined through jurisprudence.

## Design paradigm

It is commonly recognized that science and technology develops in an evolutionary way towards a dominant design. There are two stages in this evolution: the pre-paradigmatic stage, when there is no single accepted concept of the phenomena, and the paradigmatic stage which begins when one of the concepts becomes more widely adapted and takes a lead over the other concepts. The second stage signals scientific maturity and the acceptance of a dominant design, which will remain in force until the paradigm is overturned. In the first stage competition is focused on designs, with manufacturing being loosely and adaptively organized. Once a dominant design appears, competition shifts to price and manufacturing processes. Investments are focused towards creating economies of scale and learning, creating a lock-in for manufacturers. The dynamics of the design paradigm influence the way technologies compete with one another. "*at some point in time, and after considerable trial and error in the marketplace, one design or a narrow class of designs begins to emerge as more promising. Such a design must be able to meet a whole set of user needs in a relatively complete fashion"* (Teece, 1986, p.288). The timing of this process should influence a firms' decisions.

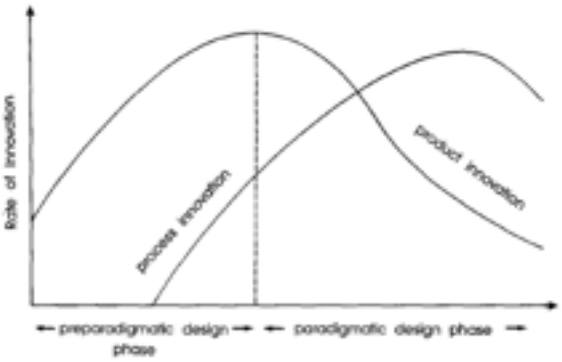



**FIGURE 1: INNOVATION OVER THE PRODUCT/ INDUSTRY LIFE CYCLE (TEECE, 1986, P.289)**

In the pre-paradigmatic phase there is strong competition among technologies, with no technology having a significant advantage over the others. As such substitutability is high and firms risk investing in technologies that will not amount to anything. Modern innovation is generally not the result of a single firm's effort. Many technologies are systemic, requiring the combination of multiple complementary technologies and patents. In sectors with an active SDO the pre-paradigmatic phase is the phase during which the standards are being developed. During this process different innovation specialists usually compete to have their technology included in the standard, as this will help them secure ownership of a portion of the dominating design.

At some point certain designs will become dominant over the others and emerge as the leading technology. In many cases this is caused by dynamically increasing returns, innovations often have a (temporarily) exponential return and those that get ahead tend to stay ahead. However, superior marketing or sales can also cause a technology to become more adapted than others, the more a technology is employed, the greater its attraction relative to the alternatives. This phase is not static though. After the emergence of a dominant design, alternatives do not disappear and it is possible for alternative technologies to overtake the dominant position. Yet, in industries with active SDOs their standards often define the dominant design and create a lock-in for a certain amount of time.

## Complementary assets

Schumpeter (1950) argued that there is something about large enterprises that help it capture profits from their innovations, basing his arguments on a market level monopoly power issue. PFI changes this scope to the firm level and its asset structure, measuring market power based on complementary asset heterogeneity and imitability. These complementary assets represent investments a firm must make to reach the market. Services such as marketing, competitive manufacturing, and after-sales support are almost always needed. Every innovation will require additional complementary assets, like computer hardware needing software or electric cars needing specialized charging stations. These complementary assets can be completely related to the innovation, like software and hardware, but this is not a necessity. The same complementary asset can be crucial for multiple innovations

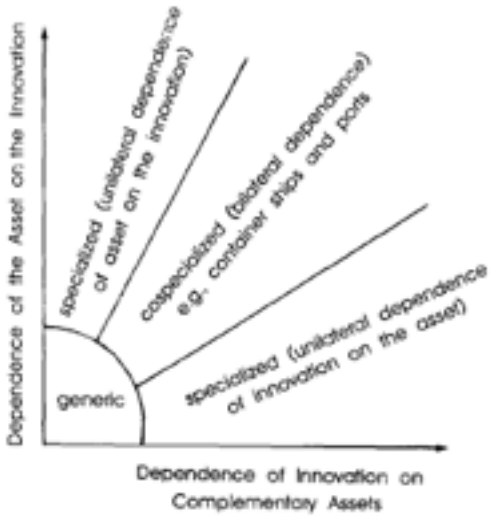



**FIGURE 2: COMPLEMENTARY ASSETS: GENERIC, SPECIALIZED, AND CO-SPECIALIZED (TEECE, 1986, P.289)**

The PFI taxonomy around complementary assets and technology is: specialized, co-specialized and generic. This taxonomy is based on the (interdependence) between the asset and the core innovation, as depicted in Figure 2. A specialized asset is characterized by a unilateral dependence from asset on innovation, or from innovation on asset. A co-specialized asset is characterized by a bilateral dependence between the asset and the innovation. A generic asset is no dependence between the asset and the innovation, and is likely to be also used by other innovations. Teece (1986) argues that control of an asset not necessarily implies control of a market, unless the asset defines a relevant market. Complementary assets often not only play a role in appropriability, but also in shaping the future strategy of the enterprise. This effect can be both positive or negative, due to the path dependency of the complementary asset. The asset requires a certain amount of investments to be developed, and shifts in the core technology can potentially render the asset obsolete.

In a more recent paper Teece (2006) relates this view of complementary assets to the resource based view (RBV). Resources have been defined as stocks of available factors that are owned or controlled by the firm (Amit and Schoemaker, 1993, p. 35). The RBV describes resources in terms of Value, Rareness, Imitability, and Non-substitutability (Barney, 1991), which can be applied to the complementary assets in the same way. Generic assets are likely easy to imitate or substitute, therefore not rare and will not be very valuable in terms of monitoring the innovation. On the other hand, specialized and co-specialized assets will be harder to imitate or substitute as they require more investments, and they are more likely to be valuable in capturing profits.

## Profiting from innovation

The interaction between these three building blocks determines in large part who can capture profits from an innovation. Under a strong appropriability regime an innovator can protect its IP and capture at least a portion of the profits related to its innovations, that is until patents expire or the IP is invented around. If appropriability is weak and imitation is easy, markets do not work that well and the profits from innovation may accrue to the owners of certain complementary assets (Teece, 2006).

IP appropriability weakens over time (patents expire, are invented around, etc.), complementary assets are a strong long-term channel for profiting of an innovation, especially when complementary assets are specialized (unilateral dependence of asset on innovation or vice-versa) or co-specialized (mutual dependence between innovation and asset). In many cases the investments required to develop these complementary assets create strong barriers to entry, like in the biomedical industry. Important specialized complementary assets in this industry are the abilities to test new drugs, getting approvals and market the drugs to doctors and insurance companies. Developing these assets is very costly, which is why incumbent firms usually collaborate with large established firms to bring their new drugs to the market. Because of the strong role the complementary assets play most profits from the drugs will go to the established firms, rather than the innovating firm.

### 2.2.2 Profiting from Open Source Software

In the case of OSS, the IP is given away for free, resulting in a non-existent appropriability regime and full imitability. Following the PFI framework, companies will need to focus on complementary assets, which is exactly what all OSS business models are about. Red Hat can profit from providing service, consulting, and training for Linux. Linux has a steep learning curve requiring very specific knowledge and companies find it more convenient to hire support from Red Hat rather than employing their own specialists. This is a complementary asset Red Hat has specialized in. Even though no money can be made from the innovation itself, Red Hat can create a turnover of about $1.7 billion on their complementary assets (Red Hat, 2016).

Google does the same with Android. While Android is free and Open Source, Google owns many complementing services such as Search, News, Email, and the Play store. All these services provide Google with advertising space to generate revenue. Next to that, the Play store is the main application provider for Android. Through the Play store users can buy and install applications on their devices, Google takes a percentage of all sales. While there are alternatives available, the Play store runs on about 90% of the Android devices. The Amazon branded version of Android for instance, does not ship with the Google Play store but has an Amazon based alternative (Amazon, 2016).

Oracle's business model around MySQL is different. Oracle offers a dual license for its database software. MySQL is released under the GPL license and under a commercial license. Under the GPL users can create proprietary apps which connect to MySQL, which is enough for most users. However, if you want to include MySQL or adjust the software without releasing it under the GPL you must buy a commercial license from Oracle. to contribute to the original MySQL software, you must grant Oracle (joint) copyright ownership of your work (Oracle, 2016). This allows Oracle to relicense the software under their commercial license. In addition, Oracle also exploits complementary assets to secure profits such as annual support subscriptions, training, and consultancy services.

There are also examples of less obvious complementary assets for OSS. In these cases, the core technology is again given away for free, so the firm should develop complementary assets to capture profits. Under the academic license firms can extend or alter existing OSS and sell or license the new product for a profit. Apple's osX is an example of such development. The core of the operating system is a version of FreeBSD Unix on which Apple build its own proprietary layers to create a differentiated version.

Concluding, a firm is not an altruistic institution, its goal is to make a profit. As such any contribution to OSS development must be justified by being able to generate a revenue or cost-saving based on OSS. As the core technology is given away for free, the PFI framework constitutes that the business' best option is to develop complementary assets. The most successful examples of OSS-based businesses have all succeeded in doing so. However, to be able to develop such complementary assets a firm must be somewhat vertically integrated in its supply chain. The firm should be more than just an innovator as it cannot rely on the appropriability regime to capture its share of the profits.

## 2.3 Open Source versus Closed Source Production

Section 2.2 describes various advantages of open innovation, which also apply to Open Source development. However, there are several other advantages to Open Source stemming from the way Open Source is organized compared to how closed source production is organized. The first section of this chapter describes production related advantages and disadvantages to Open Source development compared to closed source development. The second section of this chapter goes deeper into specific software characteristics and how they relate to Open Source and closed source development.

Even when a firm can create the necessary complementary assets to profit from OSS and is confident the benefits will outweigh the costs compared to closed source production, not all types of software are viable to be an Open Source project. While some people seem to think that OSS developers are always waiting for new projects to pick up, this is certainly not the case. Numerous Open Source projects have failed due to not gaining enough traction, or failing to govern the community in the right way. OpenHub, an organization analyzing different Open Source projects estimates that for the projects it tracks (345,045) 80.4% is inactive, and 13.9% has very low activity (Openhub, 2016). Only the top 3% of OSS projects has received an update in the past 2 years.

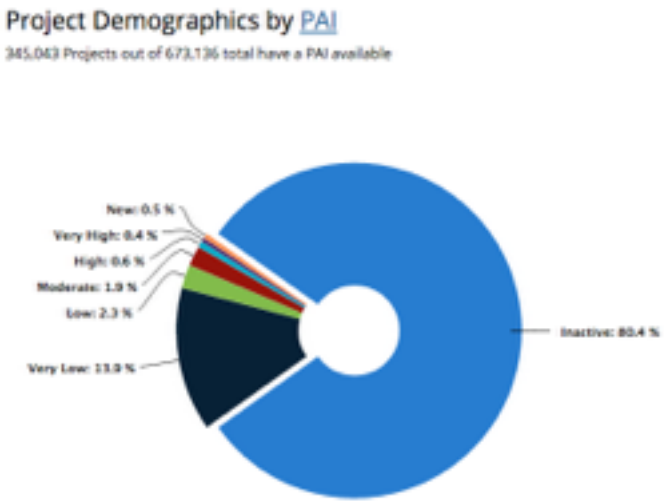



**FIGURE 3: PROJECT ACTIVITY ICON (PAI) ESTIMATION (OPENHUB, 2016A)**

It is to be expected that the majority of projects are not maintained, as there will be competing projects that resolve in a dominant design, pet projects that are not interesting to the entire community or projects that are simply abandoned. Moreover, this figure does show that open sourcing a project is not a guarantee for success of the project. Some companies and developers believe that if they Open Source their software, external developers will automatically appear. In reality this is not the case. More importantly, the most popular projects receive more contributions from company-funded developers than from individuals (Linux Foundation, 2014). Therefore, firms should not expect Open Source development to be a magically appearing free source of labor.

Next to that, not all types of software are suited to access the benefits Open Source can offer. Open Source advocate Eric Raymond (1999) describes five software discriminators which push towards Open Source. He indicates that Open Source software will be more suitable than closed source software in cases where:

(1) Reliability/stability/scalability are critical

The argument for this criteria is that in every case, many heads are better than one and no company would be able to hire and coordinate the amount of developers that are working on OSS. However, as this is highly dependent on the popularity of the project, not all OSS will have this benefit. The graph above indicates this argument would hold true for only the top 1% to 2.9% of all Open Source projects.

(2) Correctness of design and implementation cannot readily be verified by means other than independent peer review

Raymond argues that decentralized peer review trumps all the conventional methods for trying to ensure the correctness of the software. However, there are numerous examples of companies being brought in to vet OSS before it is implemented in an enterprise.

(3) The software is critical to the user's control of his/her business

Raymond argues that to avoid a vendor lock-in of critical processes, users should not use closed source software for such applications. However, Open Source does not necessarily mean that there is no lock-in, the investments that go into installation, training and servicing are often stronger forces of lock-in than licensing fees.

(4) The software establishes or enables a common computing and communications infrastructure. Software that is dependent on interacting with other software, using standardized APIs and that is used by many different parties will attract more developers.

(5) Key methods (or functional equivalents of them) are part of common engineering knowledge
This argument is a more pragmatic approach to IP, if there is no exclusive IP it is probably more efficient to have Open Source and avoid duplicate efforts.

Not all these criteria are specific to software characteristics or provide a strong distinction between OSS and closed source software. The most defining traits stem from criteria 3 and 4. These criteria form the bases for a software classification. Raymond (1999) distinguishes between categories as infrastructure, middleware, and application type software. These categories have a distinct hierarch, however, products are not fixed to one category. For example, early database software could be considered as an application, primarily used within single companies. After decoupling the database infrastructure from the front end interface, it evolved into middleware. Commoditization of middleware will push it towards infrastructure, which is currently happening to operating software.

It can be argued that OSS will be more prevalent in infrastructure (internet protocols, operating systems, applications that benefit from network interaction) than in applications, while middleware (databases, development tools) is more of a hybrid category. Hence closed source software innovation occurs in niche markets at the upper end (application) and there is a limited lifespan for closed-source monopolies as the products gradually evolves into open-source infrastructure. This lifecycle view of software suggests that closed source software innovation occurs in niche markets at the upper end (application) and that there is a limited lifespan for closed-source monopolies as the products will gradually evolve into open-source infrastructure due to increased benefits from OSS.

## 2.4 Open Innovation

The term open innovation was pioneered by Chesbrough in 2003 and has been the subject of many studies since. Over the past decade the concept was further defined and conceptualized, expanding the definition to acknowledge the intentionality of knowledge flows (Chesbrough, 2006) and to include unprotected knowledge flows: "*We define open innovation as a distributed innovation process based on purposively managed knowledge flows across organizational boundaries, using pecuniary and non-pecuniary mechanisms in line with the organization's*

*business model"* (Chesbrough and Bogers, 2014, p.15). Open innovation can be seen as a collaboration between different firms, sharing their knowledge to achieve a higher rate of innovation. These firms make conscious decisions on what knowledge they share and what not. Sharing knowledge does not require giving up ownership nor giving it away for free. On the contrary, open innovation often goes hand in hand with high levels of appropriability (Laursen and Salter, 2014). Open innovation assumes that internal ideas can be taken to the market through external channels, outside the current boundaries of the firm, to generate additional value (Chesbrough, 2006, p.1).

### 2.4.1 Open Source and Open Innovation

West and Gallagher (2006) argue that Open Source can only be called open innovation if it has a business model. Innovation is often defined as the successful commercialization of an invention (Business Dictionary, 2016) and as such the business model part is included in Chesbrough's definition (2014). Whether Open Source is consistent with open innovation depends on the ability of a firm to capitalize on the products created in the Open Source projects. However, from an Open Source community perspective this is not the case. While Open Source projects do not necessarily lead to a commercialization of an invention, it does lead to new inventions being used in the market. For many Open Source projects there will be a mix of firms and individuals contributing, hence not every participant will be driven by a commercial goal. Additionally, to say that a firm needs a precise business model to participate in OSS development is perhaps a too narrow scope. Many firms will participate in Open Source projects to save money, rather than to create new commercial activities. While it is safe to say that some firms will create business models based on OSS, for many firms having a good business reason to contribute is enough.

In open innovation firms can disclose knowledge while protecting their IP through patents or non-disclosure agreements (NDA's). As the participants in the open innovation network are known, this is a relative easy expedition. Studies have shown that open innovation is supported by strong appropriability regimes (e.g. Cassiman and Veugelers, 2002; Heiman and Nickerson, 2004; West, 2005; Chesbrough, 2006). For firms the ability to protect their knowledge reduces fears of opportunistic behavior from external actors (Henkel & Baldwin, 2011). The presence of strong appropriability ensures firms ownership of their IP, even when shared. Contrasting studies suggest that being overly protective might result in firms missing out on opportunities to exchange knowledge with others and will direct firms away from discovering new products and services (von Hippel, 2005; Bessen and Maskin, 2009). This suggests that there exists a fine balance between openness and appropriability. Emphasizing on appropriability too strong will weaken the benefits gained from open innovation (Laursen and Salter, 2014), yet firms need a level of appropriability in order to open up.

This balance becomes even more important when engaging in Open Source projects. When firms disclose knowledge to OSS, the IP will become publicly available. Moreover, a firm should be very aware of the type of license attached to the OSS and the implications of this license on the IP protection. The OSS license potentially weakens or removes the IP protection. Therefore, firms participating in Open Source projects should make a conscious decision on whether to participate; what part of its IP it will reveal; and how the project can be monetized. Since the firm will be giving up its valuable IP on one side, it must make up for this loss somewhere else. One way to protect the firm's interest is to conduct in "selective revealing" (Henkel, 2006).

Selective revealing is a form of open innovation in which innovators waive, or not establish in the first place, legal exclusion rights to specific parts of their technology (Henkel, 2006), while maintaining exclusivity for core technologies. Selective revealing implies that the actor does not reveal out of principle, but rather because of weighting commercial pros and cons (Henkel, Schubert and Alexy, 2014). Such commercial pros can consist of positive marketing effects, technical benefits, and access to new markets (Afuah and Tucci, 2012; Baldwin and von Hippel, 2011; Henkel and Baldwin, 2011). However, revealing is not without risk. Firms have obvious concerns regarding the loss of competitive advantage through imitation, reduced compatibility, reduced safety, and security, and rising maintenance costs. Since the process of revealing is irreversible, a firm should select on a case-by-case basis if the benefits of revealing outweigh the disadvantages.

The main influence on the net benefits of selective revealing appears to be an increasing customer demand, making Open Source part of the competitive advantage and marketing tool. Selective revealing can also benefit markets by significantly lowering transaction costs (Henkel and Baldwin, 2011; Henkel et al., 2014), since there is no contracting involved. Finding this balance and determining what to reveal and what not is a difficult endeavor. Many firms have difficulties to do so. The older a firm is the stronger the organizational inertia to shift from a closed to an Open Source development process (Henkel, 2006). This organizational inertia will be even stronger for firms with large patent portfolio's. As appropriability is an important revenue stream, opening up their IP is a direct threat to operations (Teece, 1986).

Innovation specialist firms are focused on developing new technologies, but rely on other firms to develop the implementations. These firms have strong patent portfolio's, but are less integrated in the vertical supply chain. As a result, their profits are secured through strong IP licensing and they employ legal teams in charge of securing licensing deals. As discussed, many of the OSS licenses can be interpreted to indicate that by contributing to the OSS, the firm also licenses all its patents that cover parts of the existing software under the OSS license. Since anyone is permitted to use OSS, the OSS provides the opportunity for any firm to obtain a royalty free license to the innovator's patents.

Such a direct threat to the profit mechanisms results in strong inertia regarding opening their firms to OSS development. Studies show that once engaging with Open Source projects, firms learn how and what parts of their knowledge to share and reveal more after having gone through a learning process (Alexy et al., 2013; Henkel et al., 2014). This is dependent on the amount of interaction the firm has with the community. A firm that merely posts its source code on their website without the ability for interaction with customers will experience far smaller benefits from Open Source development and will not experience the same learning curve as a firm that does open itself up for community interaction (West and O'Mahony, 2008; Henkel et al., 2014). This is similar to the coupled mode of open innovation, mixing both inbound and outbound knowledge transfers. Overcoming the organizational inertia remains a challenge. Case studies suggest that the main reason for overcoming this inertia is a growing customer demand. The majority of innovation specialists starts to contribute to Open Source projects to satisfy a key customers' demand (Henkel, 2006).

### 2.4.2 Open Source and Standards

Compatibility standards are crucial in a technological system; they provide the shared language that technologies use to communicate with one another. Especially in markets with large numbers of interdependent suppliers and a rapid technology pace, standards are important (Simcoe, 2006). The open process of OSS development and the fast adoption rate of some projects have led to some confusion about the difference between open standards and OSS. The main difference

between a standard and OSS is that a standard is a specification of functionality and not necessarily an implementation thereof. OSS always begins as an implementation, which over time can be adopted as a standard.

Standards can be created through two distinct processes. *De facto* standards are standards which are based on popularity. When a certain technology claims dominance over a market, this often drives remaining actors to switch to the dominant technology as well. For example, after a period of strong competition, VHS eventually became the *de facto* standard in videotape technology and Betamax was withdrawn from the market. Even though Betamax was superior, it could not compete with the marketing tactics of the VHS team.

*De jure* standards are standards which are artificially created through Standard Defining Organizations (SDOs). These standards are agreed upon voluntarily by the members of the SDO, which are usually the major players in a certain industry. The SDO provides a forum where firms collaborate in the design and promotion of the standard. Especially in the information and communications industry *de jure* standards are important, as it allows devices and software to communicate and interact with one another. Global communications standards such as GSM and 3G are what allows us to use our cell phones all around the world.

Besides the distinction between *de facto* and *de jure* standards, standards can be either open or proprietary. The documentation or specification of proprietary standards are not available to the public. Its owners can decide to maintain a monopoly over the standard, or charge fees for the access to the standard. Open standards are not necessarily free to implement, but the specifications are freely available and the standardizing process is open. This means that anyone may participate in the standards development process, standards are created through democratic processes and all documentation is made public. If the standard is covered by patents, these patents are declared Standard Essential Patents (SEP) and the owners are bound by the SDO agreement to license these under Fair, Reasonable and Non Discriminatory (FRAND) conditions (Lewis, 1986). If the standard contains patented technology which cannot be licensed under FRAND conditions, the specification is changed to exclude those patents. Licensing under FRAND conditions aims to make commercial implementation of the standard available to anyone who wishes to do so. Some open standards will be free to implement, while for others licensing fees must be paid to the holders of the SEP.

Some Open Source proponents have argued that an open standard is only open if it can be freely adopted, implemented, and extended (Perens, 2002; Digistan, n.d.). However, most scholars (e.g. Simcoe, 2006; Kretchmer, 2005) and SDOs do not place this condition on open standards. While all forms of standards created in Open Source communities are open standards, not all open standards adhere to the Open Source ideology.

**TABLE 1: EXAMPLES OF DIFFERENT TYPES OF STANDARDS**

| | **Open Standard** | **Proprietary standard** |
|---|---|---|
| **De facto** | ZIP, HTML | VHS, FAT |
| **De jure** | GSM, 3G, 4G | IrDA |

### Openness of Standards

In Section 2.3.1 a distinction was made between open standards and proprietary standards. However, the difference between these standards is not black and white. A standard can have some open characteristics, while still being a proprietary standard. Krechmer (2005) defines ten rights on which openness of a standard can be measured, based on views from the different stakeholders to a standard: creators, implementers, and users. These rights play different roles for each group of stakeholders and Krechmer argues that for a standard to be truly open for all, it should support open rights for all stakeholders involved. These three stakeholders have varying views on what constitutes an open standard to them, which Krechmer summarizes as follows:

(1) The SDO represents the standards creators. They value openness primarily in the process and organization of the standard such as openness of meetings, consensus, and due process.

(2) Implementers value openness primarily in rights to use the standard, preferable at no-cost. Implementers want the right to improve upon the standard and the standard should benefit all competitors equally.

(3) For a user the availability of multiple implementations of the standard is important. Users value competition, long term support, and backwards compatibility.

While these views are quite distinct from one another, they can be translated into the set of ten rights that enable Open Standards which can be found in Table 2.

**TABLE 2: TEN REQUIREMENTS OF OPEN STANDARDS ADOPTED FROM KRECHMER (2005, P.3)**

| Requirement | Stakeholder(s) |
|---|---|
| Open Meetings | Creator |
| Consensus based decision making | Creator |
| Due process | Creator |
| Globalised standard | Creator, Implementer, User |
| Open IPR | Creator, Implementer, User |
| Open documents | Implementer, User |
| Open changes | Implementer, User |
| Open interfaces | Implementer, User |
| Open use | Implementer, User |
| On-going support | User |

Krechmer has applied this classification to seven different SDOs and found that while many organizations present their work as “open standards”, they often just meet a few of the ten criteria. While the process is often considered open from the creators point of view, this is not necessarily the case for implementers and users.

## 2.5 Open Source Strategy

Classical organizational strategy views such as Porter's Value Chain, the Resource Based View (RBV) and Teece's PFI framework are primarily based on ownership and control as the key levers in achieving success. However, a large part of the potential value in Open Source development lies in external resources that are not owned by the firm in question, e.g., volunteer contributions, innovation communities and ecosystems, and surrounding networks. While firms have access to these resources, they cannot exert exclusive rights over the resulting innovations. Thus, traditional strategic terms such as ownership, entry barriers, switching costs and intra-industry rivalry are of secondary importance when dealing with Open Source strategy (Chesbrough & Appleyard, 2007).

The main question companies should ask themselves when developing Open Source strategy is "*who captures the value created by the open invention and coordination?*" and "*how are they doing it?*" (Chesbrough & Appleyard, 2007). Firm's engaging with Open Source development have to identify how to capture and sustain the created value. Capturing at least some part of the value is important, as it creates the necessary incentive for the firm to sustain continued participation and support of Open Source initiatives. Perr, Sullivan, and Appleyard (2006) have identified four primary categories of value capturing strategies:

(1) Self-service: The user community creates a software application to serve its own needs. While this does not incorporate an element of value capture, it could translate into cost savings for a firm.

(2) Deployment: The innovation activities by the company heighten the user experience, which are subsequently sold, e.g. Red Hat's Enterprise Linux is available for free but companies are willing to pay service fees for the increased reliability of the software.

(3) Hybridization: Proprietary extensions are made to the OSS to differentiate from original works, e.g. Apple's OS X consists of a combination of FreeBSD with multiple proprietary add-ons.

(4) Complements: Hardware or appliances are sold that incorporate the OSS. The package price decreases for the customer, while the hardware price remains the same, thus demand for the device increases.

The strategies are ranked based on the extent to which it engages the firm with Open Source development. The self-service strategy is rated most engaging, as it embeds the firm completely within an Open Source community contributing to- and using OSS for their own benefits such as cost reductions. Deployment is rated slightly less engaging as it primarily focusses on the sale of support or consultancy, while this is often closely linked to strong engagement and contributions to the OSS this is not a necessity. Hybridization is ranked third as it primarily focusses on creating proprietary works extending the OSS. Finally, there is the complements strategy, which is regarded as least engaging as it does not necessarily require any contributions to be made to the OSS. However, in many cases the firm makes some contributions to the OSS to ensure compatibility with the proprietary hardware.

These strategies are not exclusive and larger firms will use a combination of these strategies for different products or technologies. While these strategies move beyond the scope of traditional strategy views such as the value chain, they are not unrelated, e.g. the notion of the PFI framework can be used to identify what assets are key in capturing profits from an innovation and how a decision to Open Source certain aspects of this asset would change the status quo. Taking the server and datacenter example, Intel and IBM primarily own IP in the hardware technology rather than the software technology. Additionally, the firm's own important complementary assets such as computer chipsets manufacturing and the knowledge on creating and servicing datacenters, while other firms such as Microsoft owned the server software and the related IP. By contributing to Linux and other OSS related to server software the IP and complementary asset ownership of Microsoft decreases and the relative value of Intel's and IBM's products increases.

## 2.6 An Open Source decision model

There are multiple models covering the strategic decisions involved with innovation. The aim of this study is to determine how to make decisions regarding innovation in Open Source settings. The technology portfolios of innovation specialist firms are large and diverse and cannot be covered with a single analysis when considering OSS. Therefore, a model has been adopted which regards single technological innovations. Teece's PFI "*isolates a set of strategic issues associated with commercializing technological innovation. It is innovation specific"* (Teece, 2006, p.1144).

Almost no technology is developed in a vacuum. Often a technological advancement is the result of a combination of many different smaller technologies, developed by different companies. Many of these developments are organized through standardization efforts. However, a *de jure* standard is not necessarily widely adopted in the market. For a technology to become the dominant design takes time, and even than there is no guarantee. However, the more open the standard is, the faster the standard will be adopted as the dominant design.

**Proposition 1**: A more open standard will lead to a faster adoption of a dominant design.

However, as described in Section 2.3.3, this openness comes at a price. The more open a standard is the weaker the appropriability regime around that standard becomes. Openness of a technology standard leads to an increased documentation about the standard, potentially increasing the prior art in the market and diminishing the possibilities to patent certain innovations. Moreover, open standards are either licensed under FRAND conditions, or are free to implement for anyone, without having to license patents, which results in either a limited appropriability regime, or in the last case a non-existent appropriability.

**Proposition 2:** A more open standard will have a weaker appropriability regime than a less open standard.

Teece describes two stages of the design paradigm: the pre-dominant phase and the dominant phase. In the pre-dominant phase there is strong competition between different technologies and companies will not invest a lot of resources in technology specific resources, as these might become obsolete after the adoption of a different standard. Once a dominant design emerges competition shifts to price and quality and companies will start investing in technology specific assets to help them compete. This relationship between the dominant design and complementary assets is described as:

**Proposition 3:** An adoption of a dominant design will lead to an increase in complementary assets.

The appropriation profile of a firm can be seen as its attitude towards appropriability. Firms with a strong appropriation profile take full advantage of the appropriability regime and employ legal

teams to secure patents portfolio's and pursue an offensive patent strategy. However, this attitude is strongly influenced by the presence of an appropriability regime. There is no point in having a strong appropriation profile in an environment with a weak appropriability regime.

**Proposition 4:** A stronger appropriability regime leads to a stronger appropriation profile of the firm.

Under the current IP protection laws software is in a unique position. As software is protected by both copyrights and patents, it can be simultaneously regarded as both the technology and an asset, i.e. a decision to make a certain software package Open Source is both a decision on the appropriability of the innovation and the complementary assets within the PFI framework. Due to the nature of most OSS licenses, contributions to OSS often result in contributing both the copyright and patents involved. Therefore, decisions regarding Open Source contributions always should encompass both appropriability considerations and asset considerations.

To capture this duality of software as both a technology and asset it will be regarded as a resource as defined in the RBV. The RBV regards resources "[…] *to include all assets, capabilities, organizational processes, firm attributes, information, knowledge, etc. controlled by a firm that enables the firm to conceive of and implement strategies that improve its efficiency and effectiveness"* (Barney, 1991, p.101). This definition allows the resource to attribute both the asset and the knowledge embedded in the software to the resource.

The RBV describes resources based on their capacity to contribute to a firm's competitive advantage using the characteristics Value, Rareness, Inimitability, and Non-substitutability (VRIN). Of these, Value is considered as the most important attribute. The asset is only considered a resource if it helps the firm exploit an opportunity or neutralize a threat in the firm's environment (Value). The Rareness, Inimitability and Non-substitutability are characteristics that help the firm maintain the competitive advantage for a period of time.

When regarding the VRIN characteristics of software within the perspective of the PFI framework the Value and Rareness attributes correspond to the characteristics of complementary assets, whereas the Inimitability and Non-substitutability correspond with the appropriability regime. In the case of complementary assets generic assets will certainly not play a key role in creating a competitive advantage and due to their nature will not be rare. Co-specialized and specialized complementary assets on the other hand, require investments and are dependent on the innovation (and or, vice-versa) and will therefore be more likely a valuable resource.

As discussed in Section 2.4.1 the appropriability regime is a function of both legal impediments (intellectual property rights) and inherent replicability of the technology. The replicability depends on the nature of the product, such as whether the knowledge is tacit or codified, or whether it is a product or process. This corresponds directly to the Non-substitutable attribute of the RBV. Because of the nature of programming, no software engineers will write the code for an identical function in the same way, hence every implementation of the software functionality can be considered a potential substitute. Legal impediments in the form of patents protect the resource from such imitations or substitutions as it covers the technical innovation rather than the software implementation.

**Proposition 5:** The value and rareness of a software resource is higher for specialized and co-specialized assets than for generic assets.

**Proposition 6:** The inimitability and non-substitutability of a software resource is higher for companies with a strong appropriation profile than for companies with weaker appropriation profiles.

The value of resources is a contextual construct, which depends on both corporate strategy and the environment of the company (Talaja, 2012, p.12). Moreover, valuable resources that are not rare cannot be the sources of the competitive advantage, as they are available to everyone. However, this does not mean that valuable resources that are not rare are irrelevant to a company. These resources ensure the survival of the company and enable it to achieve competitive parity in the industry in which it operates (Talaja, 2012, p.13). Hence, it can make sense to place resources that are valuable, but not rare in the Open Source domain. But resources that have strong VRIN attributes should be protected at all times, as this is the primary source of competitive advantage. Therefore, VRIN software should not be shared with Open Source communities. This relation leads to the following proposition:

**Proposition 7:** Companies will adopt more engaging Open Source strategies for resources with low VRIN, than for resources with high VRIN.

The theoretical model based representing these propositions is illustrated in Figure 4.

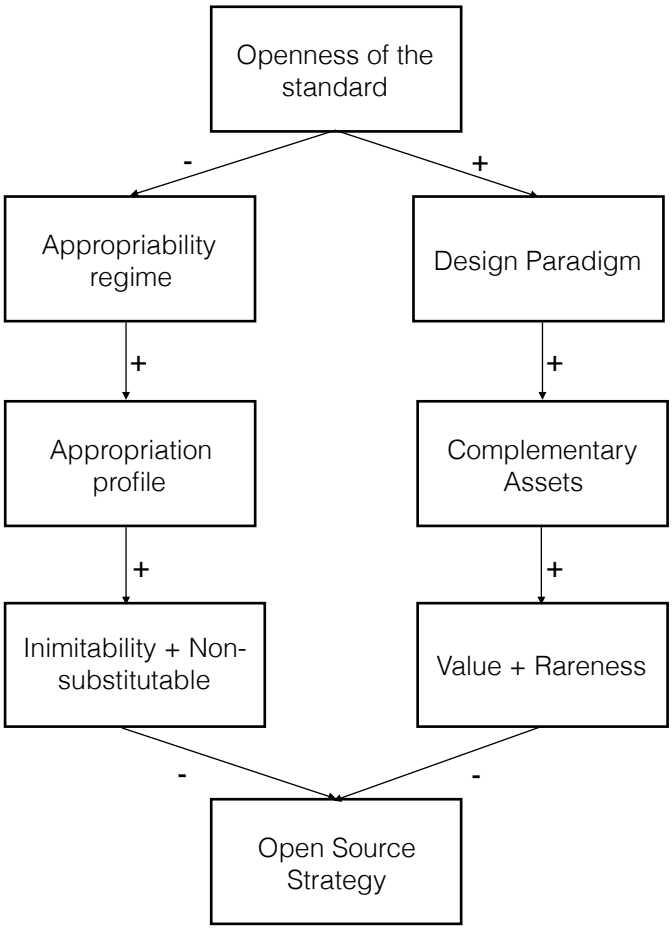



**FIGURE 4: THEORETICAL MODEL**

# 3. Methodology

The objective of this study is to contribute to the development of an Open Source strategy theory. To date there have been no previous studies testing propositions in this area of theory, therefore a set of theories has been developed based on the literature review in Section two. However, the original theories of the PFI framework and the resource based view are primarily based on ownership and control as the key levers in achieving success, which is not the case in Open Source development. Though the propositions reflect a specific set of deterministic relationships, it is highly likely that additional relationships exist between the variables. Moreover, constructs such as strategic value, appropriation profile and complementary assets are highly subjective in nature, as these depend strongly on what value the firm places on these resources.

Usually, with such predefined relationships a survey would provide the best methodological fit. However, due to the relative low amount of preexisting studies into Open Source strategy, and the subjective nature of the constructs a case study is a better fit in this scenario. The case study will allow for a more in depth discovery of additional variables and relationships associated to Open Source strategy making. The interviews allow for a more in-depth discovery of the role of specific resources within a company which is not possible using a survey. Moreover, the survey requires a large sample to be able to conduct the required statistical analysis. The number of participants was too small for a meaningful statistical analysis in this study, therefore the comparative case study was chosen.

An often heard critique of the case study is that it is primarily an explorative method, aimed at discovering preliminary concepts which require further explanatory research not provided by the case study method. However, when following systematic procedures and guiding the case study with literature based propositions the case study is just as suited for explanatory studies as other methods (Yin, 2009).

## 3.1 Research design: comparative case study

The research design is a logical plan of getting from the initial set of research questions to answers. The research design "*guides the investigator in the process of collecting, analyzing, and interpreting observations. It is a logical model of proof that allows the researcher to draw inferences concerning causal relations among the variables under investigation"* (Nachmias & Nachmias, 1992, p. 77-78).

This study follows a multiple-case holistic design. The cases all have strong similarities in their context, as all firms are set within the same industry. However, the context cannot be regarded completely similar as the firms are located on different levels in the supply chain and each case actively shapes the context of the other cases. Therefore, this cannot be regarded a single-case embedded design.

The telecommunications industry is characterized by a high level of shared innovation and standardization. Through SDO activities global standards for communication have been created based on the innovations of many different firms. The SDO activities play an active role in shaping the future of the industry by setting the technological roadmaps for innovation. The industry is characterized by a large amount of patents, owned by many different firms. As a result a lot of data about the processes, patents and licensing deals is available.

Because the telecom standards are globally created by a consortium of related SDOs it is possible to determine the openness of the standard for all players in the market. Yet, depending on the amount of SEP a firm owns the standard can be more, or less open for the individual case. The design paradigm will be completely similar for each firm, and the appropriability regime will only vary slightly based on the patents owned by the firm.

Telecom standards create a fixed time-frame for which the design paradigm is set. Additionally, patent licensing deals are often an elaborate process which can take several years to be agreed upon. Therefore the case studies will be done during a fixed timeframe. OSS activities in wireless network operations are relatively new and are primarily possible due to the introduction of software defined networking around 2010-2011. The most current standard is 4G, which was first released in 2010. The timeframe under study will be 2010 - 2015.

Selecting cases with a very similar context allows for an analysis and comparison of company specific characteristics and differences. The goal of the case study is to provide generalization to expand and generalize the theoretical propositions as stated in Section 2.6, and not to enumerate frequencies for a statistical generalization (Yin, 2009). Therefore, the cases will be selected to reflect significant differences in characteristics, while having a highly similar context.

### 3.1.1 Case selection

The units of analysis are the individual firms within the telecommunications industry. In selecting the units to be studied a number of factors have to be taken into account. The first one is the access to potential data, as you cannot study a case for which no data is available to you. The second aspect to consider is the likelihood of the case illuminating the research questions, when multiple cases are available this is the determining factor. When no case can be found which can shed light onto the research questions a researcher should consider changing the questions (Yin, 2009).

The cases were selected based on a number of criteria related to the research model as described in Section 2.6. In order to create a complete view of Open Source behavior in the telecommunications industry firms have been selected based on differences in the complementary assets they own, SEP ownership, and data availability. The first thing considered was the complementary assets owned by the cases to ensure that firms from all levels in the supply chain were represented. Additionally, contrasting cases in terms of SEP ownership were selected. For network providers this was not possible as none of the large network providers owns a significant patent portfolio directly related to the telecom standards.

For technology innovators Ericsson and Qualcomm have been selected. Studies have shown that these companies with Nokia own over 70% of all SEP to 3G technology. Nokia has been excluded from analysis because it underwent a series of acquisitions, mergers and corporate restructuring over the past years making the firm impossible to compare year-on-year.

While Ericsson also specializes in equipment manufacturing, studying a firm in this area that does not own SEP can offer interesting insights. Cisco is a large manufacturer of internet related hardware, including carrier-grade equipment. According to the ETSI IPR database Cisco owns only 1 patent family relevant to 3G. Compared to the over 800 patent families relevant to 3G, the effect of this single family on Cisco's position can be neglected.

The selected network operator is Telefónica. This firm is the seventh largest operator (in terms of subscribers) active in multiple distinct regions with standardized income statements available from 2000-2015. While there are larger operators such as China Mobile and Vodafone, these firms are either only active in limited regions, or have no distinct Open Source strategy. From the top 10 largest operators Telefónica is the first firm on the list to have both.

Apple was selected as a contrasting case to Qualcomm, while Qualcomm is not producing their own smartphones, it is producing the chipsets used in these phones for the majority of Android based smartphones. Whereas Ericsson and Cisco are more focused on the business side of the telecom industry, Qualcomm and Apple are building the consumer products.

The selected firms provide a mix of innovation specialist firms located in the top of the supply chain, more vertically integrated firms, and firms focused on the final stages of the supply chain. Naturally the innovation specialist firms will have a larger amount of SEP than the firms downstream of the supply chain, while the downstream firms are stronger positioned in regard to complementary assets.

5 Companies have been selected for the case studies, active in 7 underlying technology standards. Some of these standards are de jure standards, such as those developed at ETSI and 3GPP, while others are software packages which have become a *de facto* standard. Not every company has a stake in each of the standards, but for some standards such as Network Functions Virtualization both Ericsson and Telefonica own a portion of the IP.

The five different companies present very different cases in terms of SEP ownership and the complementary assets each firm deploys. Table 6 provides an overview of the complementary assets owned by each firm. Further discussion of complementary assets per firm can be found in the respective case studies.

While Ericsson owns a significant stake in the GMS/3G/4G standards, and Android at some level is using these standards, Ericsson is not involved with this project. So even though Ericsson owns IP covering these standards, the standard will not be observed for Ericsson. Moreover, Ericsson's patents primarily focus on the network provider side of the technology, whereas Qualcomm is more specialized on modems for mobile devices. Qualcomm is actively contributing some of its IP related to GSM/3G/4G to Android, which is why the combination of these two will be analyzed for this company.

**TABLE 3: OWNERSHIP OF COMPLEMENTARY ASSETS**

| Standard | ERICSSON | QUALCOMM | CISCO | TELEFONICA | APPLE |
|---|---|---|---|---|---|
| Software Defined Networking | x | - | x | - | - |
| OpenStack | x | - | x | - | - |
| Web standards | - | x | - | - | x |
| GSM/3G/4G | x | x | - | - | - |
| Network Functions Virtualization | x | - | - | x | - |
| iOS | - | - | - | - | x |
| SWIFT | - | - | - | - | x |

## 3.1.2 Data collection

"*The case study inquiry copes with the technical distinctive situation in which there will be many more variables of interest than data points, and as a result relies on multiple sources of evidence, with data needing to converge in a triangulating fashion, and as another result benefits from the prior development of theoretical propositions to guide data collection and analysis"* (Yin, 2009, p. 66).

While the case study is often regarded as a qualitative research method (Cresswell, 2007), it is not limited to this. Case studies do not necessarily rely on the direct and detailed observational evidence marked by other forms of qualitative research, and some case studies use a mix of both qualitative and quantitative evidence. The main point is to use multiple sources of evidence and creating a chain of evidence in order to provide a reliable source of information (Yin, 2009).

As the context was very similar for the cases the first step of the research was to analyze this context. This process consisted of analyzing existing studies on the telecommunications industry, market analysis reports, proceedings and publications by the relevant SDOs and patent analysis studies. For the individual cases the measurement process consisted of an analysis of data stemming from four sources:

(1) Annual reports: the annual reports for each company of 2010 - 2015 have been analyzed to gain insights into active business segments, patent activities and strategic goals.

(2) Open Source projects: a number of Open Source projects have been analyzed in order to determine the participation of each company and its engagement strategy.

(3) Press releases: for each firm press releases from both the firm, and journalistic inquiries were collected and analyzed.

(4) Semi-structured interviews. The goal of the interviews was twofold: at first the interviews were used to confirm facts about the companies already discovered during the analysis of the other sources. The second goal of the interviews was to uncover new facts specifically regarding the Open Source strategy of the firm. An overview of the interview subjects can be found in Table 3. The transcripts of the interviews can be found in Appendix B.

During the research the researcher was embedded within the IP licensing department at Ericsson, which provided generous access to various key decision makers. The majority of interviewees at Ericsson were a part of the Open Source Core Team, a team of various professionals in charge of any decisions made regarding Open Source within the company. The respondents from Qualcomm and Telefonica were chosen based on both accessibility and their position within the company. Both respondents were highly involved with any Open Source decision making happening within each firm.

Additionally, during his internship at Ericsson the researcher had the opportunity to participate in various strategic meetings from the Open Source Core Team. While the contents of these meetings were covered by an NDA it did help the researcher with validating his interpretations of the various case studies.

**TABLE 3: OVERVIEW OF INTERVIEWS**

| Interviewed | Position | Date | Duration |
|---|---|---|---|
| **Gunnar Nilsson (1)** | Open Source Strategies, Ericsson | 27-06-2016 | 60 minutes |
| **Magnus Karlsson** | Expert Open systems software architecture, Ericsson | 07-07-2016 | 80 minutes |
| **Christian Hoymann** | Senior Research Engineer, Ericsson | 29-07-2016 | 45 minutes |
| **Gunnar Nilsson (2)** | Open Source Strategies, Ericsson | 21-09-2016 | 30 minutes |
| **Jimmy Ahlberg** | Defensive IPR Strategist, Ericsson | 11-10-2016 | 50 minutes |
| **Eduardo Fullea** | Intellectual Property & Standards Specialist, Telefonica | 20-10-2016 | 50 minutes |

For each case study a similar case protocol is used. The protocol is designed to move from the top technology level to the environmental level, down to the firm level, the resource level and finally to the Open Source project level.

## 3.2 Measurements

Statistical analysis offers explicit criteria for the interpretations of a study's findings, however this is not the case for the case study. In order to increase reliability of case study research a case study protocol has to be developed. The protocol is intended to guide the researcher in carrying out the data collection (Yin, 2009). Because of the wide variety of data and sources the researcher has to have a specific predetermined method to interpret the collected data. The model developed in Section 2.7 describes the different variables under study and is the guideline for interpreting the data from each case study. This chapter describes the methods for measuring and interpreting each variable.

### Openness of the standard

Following the classification by Krechmer (2005) as discussed in Section 2.3.4 the openness of a standard is measured on ten different characteristics. The complete list of measurement items and their values can be found in Appendix A.

The original measure defined by Krechmer (2005) also included a measure for the IPR policy. However, this measure does not result in a strong differentiation between royalty-free and FRAND based standards, as the overall difference in score can be as small as only 1 out of 18 points. In reality this has a very strong impact on how the openness of the standard is perceived, especially from an implementer's and user's perspective.

Because of this reason, and the fact that the appropriability regime will be measured as an individual variable I have decided to exclude this measure from the original list and not regard it here.

### Design paradigm

The design paradigm is an ongoing temporal construct. Over time a dominant design emerges and gains increasing popularity, until a new design emerges which gradually cannibalizes on the existing paradigm. The majority of technology standards in the Telecommunications industry are defined by either ETSI or 3GPP. Their standardizations processes ensure that both the pre-dominant and the dominant phase are very structured. During the pre-dominant phase companies compete with one another to contribute to the standard. The 'winner' is chosen by the members of the SDO through consensus, and once a 'winner' is chosen that technology is accepted into the standard. Once the specification for the standard is finished it is published, and because of the global dominance of ETSI and 3GPP this new standard is generally accepted as the new dominant design paradigm. Commonly these design paradigms are locked in place for anywhere from 5 to 10 years, e.g. we are still using the GSM standard which was created in 1991.

However, not all technologies used in the Telecommunications industry are standardized through the SDOs. Others are under development by SDOs but have not been published, while there are already different implementations available in the market. For completed standards created by 3GPP and ETSI we can assume that these are the dominant design, as nearly all market players in the telecom industry are a member of these SDOs. However, measuring the adoption of a dominant design for the other technologies is fairly difficult.

For example, the Software Defined Networking standard developed at the Open Networking Foundation is being adopted by some players in the market, but the foundation does not have nearly as many members as ETSI and 3GPP. As this is a technology that is situated at the core of a network this is not something you can determine by looking at the network from an outside perspective.

For the majority of technologies it is possible to determine if it currently is a dominant standard, or if it is in the predominant phase. As a rule of thumb, a technology is in its predominant phase as long as there are still other technologies being used in the market in significant numbers. However, not for every technology statistics can be found for its market share over the past years. Even then these statistics are a rough estimate often made by interest groups lobbying for one of the technologies.

### Appropriability Regime

The appropriability regime is primarily dependent on the legal protection mechanisms in place. These mechanisms are dependent on the jurisdiction of the operations. When Teece first introduced his PFI framework in 1986 there were a lot of differences between the jurisdictions worldwide. However, through the establishment of the European Union and the emergence of many global trade deals there has been a convergence of appropriability regimes. Yet, today there are still differences in what types of software patents can be registered in the EU and the US. In general the US allows for a wider registration of patents than the EU. Because all the selected cases are active globally, and the relevant OSS projects are either hosted in the US or the EU with very similar legal appropriability regimes this will be considered as a constant factor for all cases.

However, the technology standards embedded in the case studies are all developed in different SDOs or consortia. Each of these organizations has their own IPR policy, outlining the extent to which contributors can protect their IP. Drawing from the measures presented by Krechmer (2005). we can define the IPR policy as follows:

Open IPR: the extend to which IPR related to the standard is available to implementers. This practice defines the cost of the IPR to the implementer. While commercial licensing is the least open process, it may not be more costly than the RAND approach.

1. *Commercial licensing with non-disclosed licensing agreements between IPR holder and implementer.*
2. *Commercial licensing on an open basis for complementing products*
3. *RAND based licensing on specifications of both the technology and the interface*
4. *RAND based licensing on the technology but not on the interface*
5. *Fully open, non-protected specifications.*

Even though these commitments are not mandated by law, they represent legal binding contracts and have been considered as such in various court cases. The IPR policies are ranked from 'weak' to 'tight' in accordance with Teece's classification of the appropriability regime.

### Complementary assets

Teece distinguishes between specialized and generic complementary assets. However, determining which specialized assets are the most influential for a technology is difficult, as not all assets are equal. Moreover, not all specialized assets are necessarily more valuable than generic assets, in some cases the specialized asset is required to partake in the market, but does not provide any added value. All generic assets can be procured by any company at a low cost and therefore do not provide any competitive advantage.

To date, not many studies have tried to present a measure for Specialized Complementary Assets (SCA) across different technologies and companies. This is extremely difficult as firms generally do not reveal their assets for individual technologies. Previous studies have distinguished between SCA's such as distribution, industry and market share (Mitchell, 1989), production (Christmann, 2000), human capital (Colombo, Grili & Piva, 2006; Rothaermel & Hill, 2005), customer relationships and market knowledge (Mitchell, 1989; Tripas, 1997; Mitchell, 1992; Rosenbloom & Christensen, 1994). However, these studies relied on surveys among managers for their data.

Chiu, Lai, Lee & Liaw (2008) tried to operationalize SCA's by defining mathematical functions describing the relative added value of marketing, production and human capital, following the same distinction as Taylor and Lowe (1997). Yet, for this operationalization they rely on a breakdown of Cost of Goods Sold which is usually not reported in the financial statements of a company such as selling expense, manufacturing overhead and payroll expense.

Teece argues that the profits a firm is able to extract from an innovation is the result of the appropriability regime and the complementary assets owned by the firm related to a specific technology. The measures described above provide an approximation of the total set of SCA's owned by a company. However, as a company often interacts with many different technologies such measures are not useful when looking at the technology level. However, it is not possible to create a more detailed measure for SCA's specifically related to the different technologies due to several reasons. For one, the majority of technologies are intertwined and SCA's are not isolated to just one of these technologies. Therefore it is also very difficult for the companies themselves to allocate specific assets to a specific technology. Even using a survey to let managers indicate the amount of SCA's owned by the firm to a specific technology will deliver skewed results, as this is a rather subjective indication.

As it is not possible to create a high-level measurement operationalization for SCA's, this variable will be measured using a simple binomial measure. For each technology a qualitative assessment will be made on whether the SCA's owned by the firm are either 'generic' or 'specialized'. This classification will be made by looking at the amount of competitors that own the same or similar assets, and the interviewees will be asked to make the same assessment for their companies.

## Appropriation profile

The appropriation profile is dependent on a firms' attitude towards appropriability. These attitudes are commonly referred to as 'defensive' or 'offensive' strategies (Arundel & Patel, 2003). These strategies are primarily reflected in the patenting activities of a firm and its subsequent use of the patents.

Firms with a defensive strategy are primarily focused on ensuring that they can do their business. This will either result in operating in a patent-free domain, licensing proprietary technologies from third parties, or filing for basic patents in order to protect the firm's products.

Firms with an offensive strategy are actively using patents to gain a competitive advantage. This results in the creation of patent thickets that prevent others from producing products related to the base technology and actively seeking out patent infringers and negotiating licensing deals. These firms often employ legal teams in charge of IP commercialization and are vocal about their patent portfolio's.

## Resource characteristics

The VRIN characteristics are measured at project level. The nature of the technology is an important factor when measuring rareness, inimitability and non-substitutability. However, as described in the RBV, the value is the most important characteristic of a resource.

### Value and Rareness

The value of a resource lies in its ability to neutralize threats and enable company to exploit opportunities that arise in a business environment, i.e. resources are valuable if they enable a company to design and implement strategies that improve its efficiency and effectiveness. In Open Source projects this value can be derived by gaining access to a necessary resource, or leveraging the OSS to either ensure a cost reduction, or increasing the value of another resource. This value has to be regarded based on an internal analysis of strengths and weaknesses of the firm. As a result this assessment is more a qualitative one than a quantitative one. The value of the resource is its most important characteristics, i.e. the other characteristics do not matter if the resource is not valuable.

The remaining characteristics are less subjective. On the subject of rareness, Barney states "*as long as the number of firms that possess a particular valuable resource (or a bundle of valuable resources) is less than the number of firms needed to generate perfect competition dynamics in an industry, that resource has the potential of generating a competitive advantage"* (Barney, 1986, p. 107). In other words, a resource can be regarded 'rare' if the resource is not easily available to all players in the market.

The rareness of the software is highly dependent on the nature of the used technology. As described in Section 2.3 software can be classified as infrastructure, middleware, or application type software. For infrastructure the underlying functionality is common engineering knowledge, as infrastructure software provides basic functions used by a wider set of other software packages. Hence, infrastructure software is very common in the market and available to all players. Middleware software is less abstract, providing more specific functions. However, the functionality provided by middleware is more of a supporting nature, such as database management. While this is not as generic as infrastructure type software, it is still not a usable product for an end user. Finally, application-type software provides the most specific functionality. The software is designed to perform a predetermined set of tasks with a clear intended goal, e.g. a calculator app on your phone, or the software driving machines in a factory. Due to the high specialization of this software it is often custom build for a single company and as a result is very rare.

The overall assessment of Value and Rareness is strongly qualitative, therefore this measure will be regarded as such. The Value and Rareness will be coded as either 'none', 'valuable', or 'valuable & rare'. For the cases where interviews are possible the respondent will be asked to rate the strategic value of the resources included in the case studies. For the Cisco and Apple cases this is not a possibility, therefore the researcher will have to make an interpretation of this value based on the presented data.

### In-imitability and Non-substitutability

Imperfect imitability is uncommon in the software industry. Reverse engineering software functionality once it has been created is a trivial exercise compared to creating it in the first place. While it is possible to treat software as a trade secret, this is not the case in the industry under study. As the software functionality is either described in the standards, or sold to multiple companies. In order to keep the software a trade secret the software should be developed and used exclusively within the same company.

The main deterrent against imitation and substitutability in software is patents. However, not al software is patentable. For software not covered by patents the imitability / substitutability varies based on the type of software as described in Section 2.3. The more is known about the underlying functionality of the software, the easier it is to substitute, hence following categories described in Section 2.3 infrastructure type software is at the higher end of substitutability, application type at the lower end, and middleware in between.

## Open Source Strategy

As described in Section 2.6, four primary Open Source strategies can be identified. For each strategy subsequent business models were defined as described in Table 5. Measuring the strategy pursued by the company is based on a qualitative assessment of their activities.

**TABLE 5: OPEN SOURCE SOFTWARE STRATEGIES**

| Strategy | Business model | Description |
|---|---|---|
| **Self-Service** | Community Source | Consortia of end user organizations or institutions jointly develops application to be used by all. |
| **Deployment** | Subscription | Revenue derived from annual service agreements bundling open source software, customer support and certified software updates delivered via Internet. |
| | Professional services / consulting | Revenue derived from professional services, training, consulting, or customization of open source software. |
| | Support | Revenue derived from sale of customer support contracts. |
| **Hybridization** | Dual license | Vendor licenses software under different licenses (free "Public" or "Community" license vs. paid "Commercial" license) based on customer intent to redistribute. |
| | Proprietary extensions | Firms broadly proliferate open source application and monetize through sale of proprietary versions or product line extensions.Variants include mixed open source/proprietary technologies or services with free trial or "community" versions. |
| **Complements** | Device | Vendor sells and supports hardware device or appliance incorporating open source software. |

Source: Adapted from Per, Sullivan and Appleyard (2006)

## 3.3 Case studies

Comparison within the case and between cases is at the heart of the comparative approach. The dimensions to be compared are defined by the theoretical model set forth in Section 2.7. In order to provide a unified case description a case protocol has been developed. The protocol outlines the structure which is used for each case description and shows analytical steps for each of the cases.

Every case study follows the same structure, starting with a general introduction to the company, followed by an in-depth description of each standard the company is engaged with following this structure:

(1) Standard description

(2) Measures describing the standard
   1. Openness of the standard
   2. Design paradigm
   3. Appropriability regime

(3) Measures describing the company's configuration towards the standard
   1. Complementary assets
   2. Appropriation profile
   3. Measures describing the specific software resource
      1. Value and Rareness
      2. Inimitability and Non-substitutability
   4. Adopted Open Source strategy

### 3.3.1 Case comparison

Cross-case analysis involves a pattern matching logic, which encompasses comparing patterns between the cases to see if they are similar or different as a step in explaining observed processes or behaviors (Yin, 2009). While case study analysis cannot be performed through statistical methods, the propositions describe what patterns are expected for each proposition. These patterns can take four different forms:

- Probabilistic: A probabilistic relationship describes a type of relation where any change in variable A will likely result in a proportional change in variable B. While this may not be true for each individual case, when measured over many instances, an increase in the average of variable A will result in a proportional change in the average of variable B.

- Deterministic: A deterministic relationship describes a type of relation where any change in variable A will result in a proportional change in variable B.

- Necessary: This is a deterministic relationship for binary variables. Variable A can either be present or absent, and the effect B can also be either absent or present. In a necessary relationship, effect B cannot occur if variable A is absent.

- Sufficient: This is also a deterministic relationship for binary variables. In a sufficient relationship, the presence of variable A will always result in the presence of effect B. While effect B cannot be absent while A is present, effect B can manifest without the presence of A.

The data analysis for deterministic relationships consists of comparing the ranking according to the values of the independent variable with the ranking according to the value of the dependent variable. If the two rank orders are exactly the same (same order of cases), the hypothesis is confirmed. If the rank orders differ, it must be determined whether the difference is such that it can be concluded that the two rank orders have no relation to each other, or that the rank orders have the same tendency (Dul & Hak, 2008).

# 4. Case studies

This chapter presents the data derived from the five case studies. As the case studies are highly dependent on their shared context this chapter starts with an analysis of these contextual factors. The first factor regards the mobile telecommunications standards, the openness of the standardization process and the dominant design over the years. The other contextual factors are the design paradigm, the complementary assets and the appropriability regime.

The second part of this chapter focusses on the five individual case studies. Each case is studied in terms of the PFI framework and its Open Source strategy. In the final section of this chapter the emerging patterns in the case studies are discussed.

## 4.1 Ericsson

Ericsson is a Swedish multinational networking and telecommunications equipment and services company. Together with Nokia, Ericsson was one of the driving forces behind the development of GSM and the firm owns about 39,000 patents mostly related to mobile broadband technology. The company's main business is in the supply and servicing of mobile broadband networks. Ericsson is the market leader in mobile network suppliers, providing the networks for about half of the world's mobile broadband operators. Up until 2012 Ericsson held 50% shares of the Sony-Ericsson joint venture which was aimed at the manufacturing of mobile phones. Currently the company is primarily focused on developing and providing network services and operations.

The company reports on three business segments: Networks, Global Services and Support Solutions. Networks represented 50% of sales in 2015 and covers all activities related to delivering the products and solutions that are needed for mobile and fixed communication. There is a shift from hardware-based networking to software defined networking, but despite a decreasing share of total revenue hardware represents about two-thirds of sales in the segment. The main competitors in this segment are Huawei, Nokia/Alcatel-Lucent and ZTE.

Global Services is focused on network rollout services and professional services. In 2015 the segment represented 44% of net sales, of which 2/3 are recurring revenues derived from services (Ericsson, 2016). Network rollout is a low margin business. Professional services deals (among other things) with the services to networks that serve more than 1 billion subscribers in approximately 100 countries. These networks are typically multi-vendor, with more than half of the equipment deriving from non-Ericsson equipment. Major competitors include Accenture, Huawei, IBM, and Nokia/Alcatel-Lucent (Ericsson, 2016).

Support Solutions represented 6% of net sales in 2015. The business segment develops and delivers software-based solutions for operations support systems (OSS) and business support systems (BSS), TV and media. OSS deals with the operational management of a network, such as network configuration, fault management and network optimization. BSS deals with the business management of a network such as customer and revenue management. In TV and media, Ericsson works to enable operators and content owners to efficiently deliver and monetize video content. Up to 2012 this segment was called Multimedia. The decline in operating margin in 2014 is primarily due to increased operating expenses related to the ramp up of investments in targeted areas (Vestberg, 2015) and decreased spending by wireless operators to prepare for spectrum acquisitions.

Ericsson's first major contribution to the Open Source community was the release of Erlang in 1998 (Erlang, 2016). However, this release was more the result of inner-company politics and a decision to discontinue the use of the programming language within the company than a conscious decision to create an Open Source project (Nilsson, 2016). Between 1998 and 2012 the company has not had any real engagement with Open Source projects. While some OSS was being used within the company there was no internal policy about Open Source and the company did not make any significant contributions to Open Source projects (Ahlberg, 2016). During this period Ericsson was primarily leveraging OSS for their own products: "*Ericsson is adapting its products and work processes to take advantage of opportunities afforded by open-source software. In particular, the Linux operating system (OS) has become an important component for Ericsson*" (Mygind, Hylsberg, and Swirtun, 2006, p.1).

Over the past years Ericsson has taken a more active stance on Open Source and the number of projects the company is involved in has increased. This shift in attitude is primarily due to an increased customer demand for the firm's participation in different projects. These projects are primarily aimed at software infrastructure, and the firm has recognized that it's differentiating capabilities lie primarily in hardware products and software applications (Nilsson, 2016).

Ericsson has selected specific Open Source projects in the network infrastructure stack to engage with. Among others, these projects include Openstack, OpenDayLight, and OPNFV. "*Ericsson participates in projects really related to the NFV space: OPNFV, Openstack, OpenDaylight, OpenVswitch, the foundational projects is where we keep our focus. We also participate in ONOS, the new FDI.io the OpenO project in MANO"* (Price, 2016). Ericsson was a founding member of both OpenDayLight (in 2013) and OPNFV (in 2014). One of the reasons for joining was that the company views that the software functions will be a commodity, no matter what Ericsson does. Hence, Ericsson aims to provide input and ensure the quality of the project as many of its clients will be working with it (Karlson, 2016).

Ericsson has a large patent portfolio covering the telecom standards and litigation history of the company suggests that this portfolio contains strong SEP to the different wireless standards and is following an offensive strategy. Large players such as Samsung and Apple have settled in the past years with Ericsson over licensing fees. While the details of the settlement are not disclosed, Ericsson's shares rose significantly after both deals indicating a favorable outcome for the firm (Johnson, 2014; Nordenstam & Swahnberg, 2015). The firm reports having patent licensing agreements with over 100 players in the telecommunications industry. Average patent licensing revenue of Ericsson was $1,200 million over the past five years, which represents about 3.8% of the firm's average annual revenue. However, for Ericsson creating these licensing agreements is a difficult endeavor; the majority of manufacturers will not approach Ericsson for a licensing agreement and most of them will fight any licensing deal for as long as possible.

Analysis shows that Ericsson owns about 20% to 25% of declared SEP to GSM, 3G and 4G and a study published by the Signals Research Group suggests that this number could be higher. On behalf of Ericsson the company analyzed all contributions made by companies in the 3GPP meetings that were eventually approved and submitted to the standard. As companies do not share any contributions that are not protected by IP it can be assumed that these contributions are strongly correlated with SEP ownership. According to this research Ericsson had the highest amount of approved contributions to both 3G and 4G (Signals Research Group, 2010). However, precise numbers are not published in the report.

### 4.1.1 Network Functions Virtualization

Over the past decade development has started on separating the software functionality from the hardware in networking equipment. Traditionally software was tied in with hardware and adding functionality to a specific region meant placing additional physical hardware at the local network nodes. By separating the software from the hardware it becomes possible to operate the software in the cloud, on a centralized location. As a result adding new functionality no longer requires the placement of physical equipment, instead the functionality is run in a virtual server in a datacenter, which can be added and configured remotely. This development is called Network Functions Virtualization (NFV). The network functionality that is virtualized includes among others firewalls, switching, domain name service (DNS), content delivery networks and load balancing.

Commonly these functions are created by many different vendors and within one network it is possible to simultaneously rely on functions from different manufacturers. The software to create and manage these different NFVs is called Management and Orchestration (MANO) software. Currently there are already proprietary MANO products available in the market, but as these need to be able to interact with software from a variety of vendors this requires a certain amount of customization. Therefore, ETSI determined there was a need for a standardized interface and configuration for the different functions and MANO implementations. In November 2012 ETSI started a new project (ISG NFV) focused on developing a NFV and MANO standard.

#### Openness, design paradigm and appropriability regime

The NFV standard is being developed by ETSI, which was one of the cases studied and assessed by Krechmer (2005). Historically, ETSI has a very open approach to its standardization efforts. specifications and work-in-progress documents are published for free on their website, and the entire process is built upon democracy and consensus. However, to view all decisions, minutes, or to participate in the meetings firms have to pay to become a member. While Krechmer (2004, p8) rates ETSI at 14 out of 18 points, I have removed the IPR measure from the equation. Correcting for this alteration results in an overall openness score for ETSI of 10 out of 13 points.

The most significant development in terms of openness at ETSI is the launch of a pilot Open Source project in February 2016. This Open Source project is called ETSI Open Source Mano (OSM). While the project is closely related to the standardization activities on NFV, the work is regarded as a reference implementation or testing grounds for the standard. However, companies participating in the NFV standardization process are not obligated to participate in the Open Source project. The Open Source project is licensed under a royalty free policy, however, this policy only applies to the project, and not to the standard which is why the openness score is not adjusted.

Currently the technology is in the pre-dominant phase. While standardization development is underway, the standard is not yet completed and multiple proprietary solutions are being deployed in the market (Karlson, 2016). Additionally, there are several Open Source projects focusing on different NFV aspects, such as ETSI OSM, Open-O, and OPNFV.

Up until the start of the ETSI NFV standardization the technology was completely proprietary without any additional IPR policies limiting the appropriability regime. Companies such as Ericsson and Huawei own patents covering NFV and negotiate commercial licensing deals with infringers which are covered by an NDA. However, through the ETSI NFV standard all SEP are committed under FRAND conditions. As ETSI has a very strong global position this will have a strong influence on the overall appropriability regime once the standard is completed.

## Complementary assets and appropriation profile

Ericsson owns a number of specialized assets in the NFV space. The company has created its own NFV management and orchestration software, supports operators in integrating different NFV's from different vendors and the firm manufactures its own networking hardware to interact with the NFV functions (Ahlberg, 2016). Hence, Ericsson can be classified as having SCA's for NFV.

A large part of Ericsson's business is dependent on network rollout and services, accounting for 44% of the total revenue in 2015 (Ericsson, 2016). While network rollout is typically a low margin business, professional services deals are not. Ericsson provides professional services for networks in approximately 100 countries serving over 1 billion subscribers. As these networks are typically multi-vendor, with more than half of the equipment deriving from non-Ericsson producers, a large part of these services consists of integrating the different network functions (Ericsson, 2016).

With the development of NFV, Ericsson has created its own NFV MANO software called Ericsson Cloud Manager (Nilsson, 2016). Additionally, Ericsson owns and sells a large variety of different virtualized network functions, and the required node hardware. For Ericsson this segment represents an important part of their business, and as it was one of the early adopters developing NFV software it owns a significant IP portfolio covering the standard currently under development at ETSI. Before the standardization efforts launched Ericsson's profile was primarily defensive. The patents were primarily aimed at defending Ericsson Cloud Manager, and were not used to actively seek licensing fees (Nilsson, 2016). However, with the ETSI NFV standardization ongoing, the company is actively contributing to the standard and filing it's SEP committing it under FRAND conditions. Once the standard is finalized the IP owned by Ericsson will not be exclusively used by its clients anymore, and the company will shift towards a more offensive profile.

## Resource characteristics

### Value and Rareness

Ericsson owns both valuable IP and complementary assets in the space of NFV. The MANO software currently being sold by Ericsson is facing direct competition from ETSI OSM, and would Ericsson be contributing to this Open Source project, it would be undercutting its own position in the MANO space (Ahlberg, 2016). As a supplier of complete mobile networks and its management, MANO is a key aspect to this offering. MANO software can be consisted as middleware software, as it provides the interfaces between the different virtualized network functions and the actual infrastructure. Based on these characteristics I consider NFV/MANO to be important and valuable resources for Ericsson.

### Inimitability and Non-substitutability

Thanks to the patents owned by Ericsson there is a certain degree of inimitability and rareness to the software (Nilsson, 2016). Yet, there are multiple different MANO software vendors in the market, and as the software is not standardized, these are not always infringing upon these patents, therefore there is some substitutability to the software. The publication of the ETSI NFV standard will change this and make all NFV/MANO packages conform to the same technologies, but at the same time the ETSI IPR policy will apply to all patents, making them available to all players in the market.

I have rated the IN characteristics as medium, since the high level of customization needed for each client results in a very strong lock-in effect. While clients are browsing for solutions there are a few competing options, but after the investments has been made the lock-in effect will result in a much stronger level of IN characteristics.

### Adopted Open Source strategy

Ericsson is not engaging with the ETSI OSM project (Ahlberg, 2016). I found that the company owns a significant patent portfolio covering the technology, and is strongly dependent on licensing its patents and software to its clients. This combined with the fact that ETSI OSM is licensed under the Apache V2.0 license, makes it unfavorable for Ericsson to engage with this project. Therefore I conclude that the firm has adopted a proprietary strategy for its NFV/MANO activities.

## 4.1.2 Software Defined Networking

Software Defined Networking (SDN) is similar to NFV in that it is also a separation of software functionality from the networking hardware. However, SDN is focusing on a different layer of the architecture. While NFV is focusing on specific functions within the network, SDN is looking to virtualize the control of the network itself. The standard for SDN is called OpenFlow, which is developed by the Open Networking Foundation (ONF), additionally there is an Open Source project aiming to create an OSS implementation of the standard called OpenDayLight.

### Openness, design paradigm and appropriability regime

The OpenFlow standard is developed by the ONF which is a nonprofit funded by companies such as Deutsche Telekom, Google, Microsoft and Facebook specifically created to promote and develop SDN. Anyone can become a member of the ONF, but you have to pay to become a member (ONF, 2015). The organization is based on democratic principles and consensus, and both the standards and work in progress documents are available for free on the website (ONF, 2016a). While the ONF has a more open policy regarding its internal documents, it is not a globalized standard, which is why its overall score ties with ETSI at 10 out of 13 points.

While not all telecom players are active in the ONF, it is currently the only SDO working on SDN standardization, additionally all major networking hardware manufacturers are a member of the foundation. Hence, while not all players may be using SDN, the ONF standard is the sole standard for SDN and is thereby the dominant design.

The ONF has a royalty free licensing policy, subject to a defensive suspension (ONF, 2016b). Under these condition none of the contributing firms is able to license its patents under any other terms than royalty-free. Hence the appropriability regime can be considered completely open, scoring a 5.

### Complementary assets and appropriation profile

While Ericsson was one of the first companies to create an SDN-based switch, this was the result of a cooperation with Websprocket, with Ericsson primarily providing the hardware and Websprocket providing the enabling technology (Business Wire, 2001). Hence, Ericsson primarily owns assets related to the hardware side of SDN. However, SDN is a development that is not specific to the Telecommunications networks, rather it can be found in all different types of physical networks such as large university / corporate networks, in datacenters, and at all kinds of operators. All the major networking equipment manufacturers are members of the ONF and provide similar hardware as Ericsson for SDN. Therefore I classify these assets as generic.

SDN originates from Universities, rather than companies, and Ericsson only became involved to supply the hardware for this functionality. As a result Ericsson does not own any significant patents, and the technology is not regarded important to Ericsson's portfolio (Karlson, 2016). While the company overall has a very offensive profile, for this specific technology they follow a more defensive approach (Ahlberg, 2016).

### Resource characteristics

#### Value and Rareness

The project is of some strategic value to the firm through complementarity, e.g. Ericsson delivers both hardware and software related to the software. The company builds the switches implementing SDN, and delivers network management software which is for some part dependent on OpenDayLight. However, this service is provided by a number of large competitors such as Huawei, IBM, Accenture and Nokia/Alcatel-Lucent (Ericsson, 2016). SDN is not a technology where one firm has a strong position, rather it is a commodity (Karlsson, 2016). Therefore I have rated the value and rareness for SDN as none.

#### Inimitability and Non-substitutability

SDN software development was primarily driven at universities, putting the majority of knowledge into the public space. Currently many different firms are pursuing SDN developments, OpenDayLight alone has around many corporate members that are prominent actors in the network management space (i.e. Nokia, Huawei, Juniper, Cisco, ZTE, Brocade, etc.). The IPR policy of the Open Networking Foundation prevents any exclusivity to the technology. Moreover, the software is an infrastructure type software, therefore I have rated SDN to score low on these characteristics.

### Adopted Open Source strategy

Ericsson is pursuing both a hybridization and a deployment strategy for OpenDayLight. The firm incorporates some aspects of OpenDayLight in its proprietary solutions, and offers to build it into existing networks for its clients. "*Ericsson is virtualizing all of the portfolio for network functions, we're building a complete infrastructure for cloud and NFV. We're building a complete SDN portfolio based on openness, based on OpenDaylight. On top of that we're leveraging our service capabilities as a system integrator"* (Häglund, 2015). While the company is also offering hardware devices that are able to incorporate OpenDayLight, these devices are capable of running various software packages and are not dependent on OpenDayLight for their functionality. Therefore I have not regarded this as a complementing strategy but a mix of hybridization and deployment strategy.

## 4.1.3 OpenStack

OpenStack is OSS for the management of cloud-infrastructure. The software was first developed by NASA and Rackspace and was released under the Apache V2.0 license in 2010. A 'cloud' is located in a datacenter, on a large number of different servers. Each individual server runs software that allows to run multiple virtualized servers on the hardware. In the past, every single physical server had to be configured independently, making for a large overhead in datacenters. OpenStack connects to the virtualization software and allows users to configure multiple physical servers from a single interface, reducing this overhead.

While there are proprietary alternatives available, these are owned by competitors such as Microsoft (Azure) and Amazon (AWS). Cloud management software plays an integral part of transforming a datacenter containing numerous individual servers into a cloud operating as a single entity.

OpenStack is not based on a singular standardized technology. Therefore it is not possible to look at a specific technology or SDO to analyze. However, OpenStack itself can be considered a standard in the market, and development is organized through the OpenStack foundation. This foundation is supported by over 150 companies worldwide and behaves much like a SDO. It's processes are organized through a technical steering committee and different user groups.

### Openness, design paradigm and appropriability regime

While OpenStack foundation has a board of directors and a technical steering committee, the design process is very open. Every six months a design summit is organized which is open to anyone. During this summit the roadmap for the next six months is determined. All source code is Open Source and all development can be tracked through the open repository (OpenStack Foundation, 2016). Participants do not have to pay to become a member before contributing, and all proceedings and development is published openly. However, OpenStack cannot be regarded as a globalized standard, as there are multiple competitors. Hence overall the project scores 11 out of 13 points on openness.

There are many competing cloud management platforms in the market, OpenStack is but one of them. Hence the technology is still in the predominant phase. Competitors include Microsoft Azure, Google App Engine and Amazon EC2 which are running their own proprietary software in their own (for-rent) cloud data centers, and VMWare vSphere which can be licensed to run in your own datacenter.

While there are many competitors offering cloud-management software solutions, the nature of the software is such that it cannot be patented, being to abstract to offer any patentable technologies. Moreover, there are a number of Open Source implementations available. The solutions offered by Microsoft, Google and Amazon are included when renting space in their datacenters. VMware is the traditional market leader, and has been offering a mature product for the past decade, but the other implementations are catching up and VMware is losing market share. Hence, the appropriability regime can be considered open, with a large degree of knowledge diffusion and a low level of legal protection.

### Complementary assets and appropriation profile

For Ericsson OpenStack is a tool in their business offering, because of the high customization needed for every customer, an internal or open variant of cloud management software is preferred over an external proprietary solution. For Ericsson the cloud manager is only a small part in the overall system, that they do not develop themselves. Additionally, the company is not selling datacenter technology, rather datacenters are a means to deliver their other products (Nilsson, 2016). Hence the company has no SCAs in the technological area and their profile can be considered defensive.

### Resource Characteristics

#### Value and Rareness

On its own, OpenStack does not add any functionality for Ericsson, it is just an infrastructure type software that is used to manage its cloud operations. OpenStack is not a source of competitive advantage, or a necessity to operate a datacenter. There are both Open Source and proprietary alternatives available. Therefore I have rated this resource to have no strategic value for Ericsson.

#### Inimitability and Non-substitutability

OpenStack is not a patentable technology. Moreover, the software is situated between infrastructure and middleware type software. Additionally,there are both Open Source and proprietary alternatives available, therefore I have rated the inimitability and non-substitutability as low.

**Adopted Open Source strategy**

Ericsson is a member of the OpenStack community and contributes to the software. However, these contributions are primarily for their own interests, and under the Apache V2.0 license any patents owned by Ericsson on the contributions it makes are granted for free and in perpetuity to all users of OpenStack. For Ericsson OpenStack is a small software part in their overall offering, which is not offering any differentiating value. Hence, the company is primarily using the OSS and the contributions are mainly focused on bug fixing or creating interoperability. I have classified this Open Source strategy as self-service.

## 4.2 Qualcomm

Qualcomm is an American multinational semiconductor and telecommunications equipment company founded in 1985. The firm pioneered CDMA based cellular technology research in the 1990s and plays an important role in the development of WCDMA and LTE technology. Qualcomm owns a significant IP portfolio covering both technologies, the firm also develops and commercializes a number of key technologies used in mobile devices.

Qualcomm primarily focusses on the design and manufacturing of integrated circuits and system software. Qualcomm combines many different components in its chipsets such as CPU, GPU, GPS, and mobile modems for wireless communication interfaces such as CDMA, LTE, Bluetooth and WiFi. As a result their chipsets form the core of many mobile device and it is currently the preferred supplier for Android devices. Qualcomm chipsets and wireless modems can currently be found in devices from all major mobile device manufacturers.

Qualcomm divides its business in three segments, Qualcomm CDMA Technologies (QCT), Qualcomm Technology Licensing (QTL) and Qualcomm Strategic Initiatives (QSI). All of the chipset development and manufacturing activities are organized in QCT. Qualcomm does not own or operate any manufacturing themselves, instead production is outsourced to foundries such as Taiwan Semiconductor Manufacturing Company, Samsung Electronics Co. Ltd., and United Microelectronics Corporation. The main reasons for doing so stem from both technical and economic aspects. This strategy allows Qualcomm to have tight relationships with the foundries and accommodate for foundry-specific capabilities in their designs.

Additionally, because Qualcomm maintains the overall chipset design it is possible to enforce standardized interfaces and produce specific modules at the most efficient foundries for that module. Moreover, the rapid development of chipset manufacturing (Moore's Law) requires large investments in the manufacturing process which Qualcomm now is not subject to. The chipsets designed by Qualcomm also include specific software to operate different device components. The software is licensed to device manufacturers that use the chipsets in their products. QCT represents roughly 65% to 70% of Qualcomm's revenue, the average annual revenue in QCT in the past five years was $1,337 million (Qualcomm, 2016).

In general, Qualcomm has a primarily offensive appropriation profile, it has a large patent portfolio and is actively seeking patent infringers to create licensing deals. The company has the largest licensing income in the telecommunications industry. Over the past 5 years Qualcomm received an average of $6,900 million in licensing revenue, which represents about 34% of Qualcomm's total revenue (Qualcomm, 2012; 2013; 2014; 2015; 2016).

Qualcomm only owns a small portion of the patents formally declared as SEP to GSM, 3G and 4G, but the firm does own significant patents essential to CDMA. As 3G and 4G technologies in many ways incorporate CDMA technologies it can be assumed that Qualcomm has a strong patent position in these downstream technologies. Moreover, Qualcomm is the market leader in chipset development. Due to this strong position Qualcomm is able to leverage its patent portfolio to extract a large amount of licensing revenue.

Qualcomm has the largest revenue from IP licensing in the mobile wireless industry, which is why they have made a strong effort to develop an Open Source strategy that protects their IP. The firm is very active in Open Source development and is sponsoring several Open Source projects. Qualcomm is a platinum member of the Linux Foundation for example. In order to prevent granting a free patent license while contributing to these Open Source projects Qualcomm has created a 'patent firewall'.

The first step into creating this firewall was a restructuring of its financial holding and subsidiaries. In 2012 Qualcomm announced a corporate restructuring which resulted in the creation of QCT, QTL and other subsidiaries. In their press announcement the company stated "*The change in Qualcomm's corporate structure generally formalizes the way the company has been operating, and will continue to operate, its primary businesses. However, the company expects that QTI* (Qualcomm Technologies Inc.) *and its subsidiaries' product and services businesses will increase their work with open source software in the future and this restructuring will, among other things, help ensure that QTI and its subsidiaries' activities do not result in the licensing of any of Qualcomm Incorporated's patents, including its 3G and 4G patent*" (Qualcomm, 2012).

One of the projects Qualcomm is a main contributor to is the Android Open Source Project (Android). The Contributor License Agreement (CLA) of this project includes a perpetual, no-charge patent grant to any contributions: "*a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work*" of any submissions made by the contributor (Google Inc., 2014).

All Open Source contributions of Qualcomm are made by its subsidiary Qualcomm Innovation Center (QuIC), while all of the patents are owned by QTL. Moreover, the licensing agreement between QTL and the other Qualcomm subsidiaries states that these subsidiaries are not in the position to grant any licenses to the patents owned by QTL. However, the Android CLA states that:

"*For legal entities, the entity making a Contribution and all other entities that control, are controlled by, or are under common control with that entity are considered to be a single Contributor. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity*".

This definition in combination with the clause regarding the patent grant enforces that even though Qualcomm has put its IP under QTL and makes OSS contributions through its subsidiary Qualcomm Innovation Centre, the different companies are regarded as one. Effectively this would mean that Qualcomm would still be granting a free perpetual patent license to anyone using Android. To counteract this effect Qualcomm had to go a step further in order to create the patent firewall, while still being able to contribute to the Open Source projects.

In 2009 the firm founded the non-profit Code Aurora Forum which aims to contribute to both Android and the Linux Kernel. While Qualcomm has no formal control over the Code Aurora Forum, two of it's subsidiaries are members and QuIC has a permanent seat on the board. As the other members are primarily small enterprises Qualcomm has a very dominant position within Code Aurora Forum. Due to this structure the Code Aurora Forum is regarded as completely independent from the Qualcomm holding, shielding Qualcomm's patents from any contributions made to the Open Source projects by the Code Aurora Forum.

Qualcomm's contributions to any Open Source project are first made to the Code Aurora Forum, which in turn submits them to the Open Source Project. The Code Aurora Forum has also a CLA, which makes the Code Aurora Forum full owner of the copyrights to any contributions made to the Code Aurora Forum:

"***Copyright Assignment.*** *You hereby assign (or shall cause to be assigned) all Copyright rights and Copyright interests in and to Your Contributions. Such assignment is made expressly subject to any pre-existing non-exclusive licenses or other non-exclusive rights that You have granted with respect to any such Contribution. In addition, this Copyright assignment is subject to the Forum granting a Copyright grant-back license as set forth below*" (Code Aurora Forum, n.d.).

By setting up the Code Aurora Forum in such a way that it becomes the copyright owner before submitting to the Android Open Source project, while not having any formal control over the foundation, Qualcomm has created a barrier between its patents and its Open Source contributions. As a result they are free to contribute software containing patented technologies to the Code Aurora Forum, which in turn gets submitted to the Open Source projects such as Android and Webkit.

### 4.2.1 Webkit & web standards

Webkit is an Open Source web browser engine for rendering web pages using Javascript, HTML and CSS. These languages are used to create the front-end interfaces of web applications. The Webkit engine is used in (among others) Google Chrome and Apple Safari and their mobile variants (Bright, 2013). Webkit allows for more advanced rendering of Javascript and CSS elements than other browser engines (although, over the years browsers such as Internet Explorer have caught up). Webkit has been the driving force behind improving internet browsing on mobile devices. Even though Qualcomm is not directly benefiting from their involvement with Webkit, it is part of their effort to improve the usability of mobile devices:

"*Mobile browsing has become a key application offering users access from almost anywhere key services and applications on the Web. However, until recently, browsers have been optimized primarily for PCs*" (Bapts, 2010).

By creating a richer browsing experience on smartphones users have started to consume more data, and have come to require stronger processors. This change has resulted in an increasing demand for Qualcomm's chipsets.

#### Openness, design paradigm and appropriability regime

The front-end languages rendered by Webkit are defined by the World Wide Web Consortium (W3C). The W3C has implemented five principles of openness in their proceedings describing key aspects such as due process, consensus building, transparency, and commitments under FRAND conditions. However, the W3C patent policy is based on royalty free commitment and a permissive copyright license. Overall the W3C scores 12 out of 13 points on openness.

The front-end languages are the dominant design. In earlier web development there was competition from Flash, Shockwave and Microsoft Silverlight, but with the introduction of HTML 5 and CSS 3 in 201x these technologies have been replaced completely by HTML/CSS/Javascript. However, there are different competing browser engines besides Webkit.

The IPR policy of the W3C enforces a royalty-free and permissive license for their standards. Moreover, computer languages are considered an abstract idea that cannot be patented. Hence the appropriability regime for these standards is open.

### Complementary assets and appropriation profile

While Qualcomm is contributing to Webkit, it has not created any specific specialized assets to bring the technology to the market. Webkit is not mobile specific, and it not directly dependent on the chipsets created by Qualcomm. Additionally, while Webkit browsers are an addition to the mobile experience, they are not a prerequisite and alternatives are available. Moreover, Qualcomm is not involved with any actual browsers or other applications depending on Webkit. Therefore it can be concluded that the firm does not own any SCA's.

Qualcomm has a defensive profile for Webkit, as mentioned in the previous section, in many cases the programming languages and their interpreter are to abstract to be patented, and the W3C's IPR policy does not allow for rent-seeking on any patents on contributions to the standard. Hence there is no possibility for Qualcomm to create valuable patents and exploit an offensive position.

### Resource Characteristics

#### Value and Rareness

Before webkit creating mobile-friendly websites was not really possible and many website features that would work on a normal computer did not work on mobile devices during the first years. Webkit was responsible for bridging the gap between browsing on a computer and browsing on a mobile device and Qualcomm placed high value on this development for two reasons (Bapts, 2010):

1) An increased experience in mobile browsing and internet use would stimulate demand for faster and more stable wireless mobile broadband connections such as 3G and 4G.

2) An increased visual quality of apps and browsing would generate more interest in smartphones and by proxy increase demand for Qualcomm's chipsets.

Webkit is primarily of complementary value to Qualcomm. The main reason for the company to contribute is to improve the web browsers on mobile devices. By creating a mobile web browsing experience that is very similar to browsing on a computer these devices need increasing processing power and internet speeds. Qualcomm is one of the largest suppliers of mobile chipsets and as such will profit from an increased demand for faster chips.

#### Inimitability and Non-substitutability

Due to the open nature of Webkit, and the standards it implements the resource is easy to imitate and substitutable. The software itself is a middleware type application and there are multiple alternatives and Webkit derivatives available in the market. Therefore I have rated these characteristics as low.

### Adopted Open Source strategy

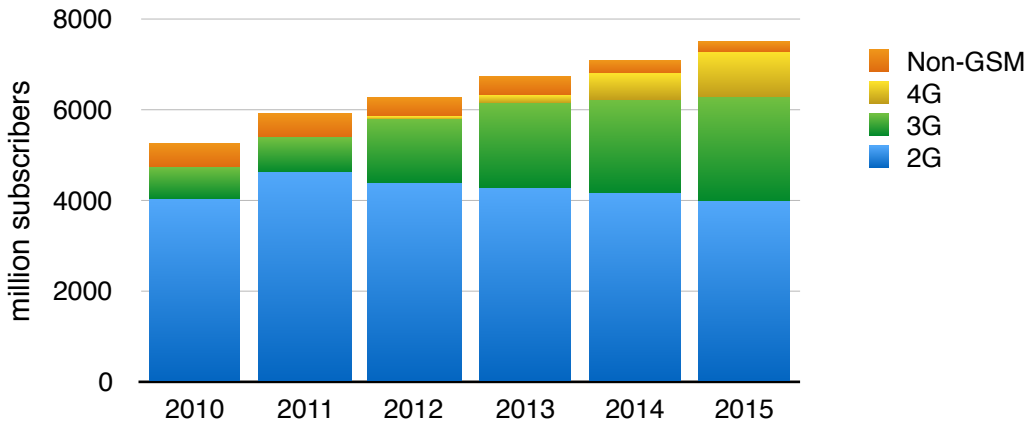
I have rated the strategy for Webkit applied by Qualcomm as self-service. While their value comes from a complementary device, e.g., their chipsets, these do not directly incorporate the Open Source software. Moreover, the Webkit community is primarily a consortia of end user organizations jointly developing applications that can be used by all. While Qualcomm is not an end user, it is engaging as one.

### 4.2.2 Android and GSM, 3G, 4G

Android is an open source mobile operating system for smartphones. It was first created by Android Inc. which was later purchased by Google. Android is based on the Linux Kernel and is released under the Apache V2.0 license, while all Linux kernel improvements are released under the GPL V2 license.

The majority of contributions made by Qualcomm to Android are aimed at creating interoperability between its chipsets and the operating system. The Qualcomm chipsets do not only include the processors needed to power the smartphone, but they also include the modems needed for (among others) GSM, 3G and 4G. These technologies are standardized at 3GPP, where Qualcomm is a key contributing member. The contributions Qualcomm makes to Android ensures that the operating system is able to communicate with these modems and connect to the different networks.

**FIGURE 5: GLOBAL ADOPTION RATE OF MOBILE STANDARDS 2010-2015**




Source: adapted from GSMA (2016)

#### Openness, design paradigm and appropriability regime

3GPP itself is not an official SDO, rather it is a collaborative forum between five regional Telecom SDO's such as ETSI. Hence, in many cases 3GPP shares characteristics and proceedings with these SDO's. The major difference between 3GPP and ETSI is that at 3GPP documents are shared more openly and anyone has access to work-in-progress. Hence, 3GPP scores one point more than ETSI and is awarded 11 out of 13 points.

Currently GSM, 3G and 4G can be considered the dominant design. While not all networks around the world are offering all three technologies, the technologies are closely related and can be considered evolutions of one another. Overall, GSM-based networks represent >95% of the entire market, as can be seen in Figure 8.

While the patents are licensed under FRAND conditions, the patent owners are quite active in seeking out licensing deals and the entire technology is patented (although by a wide variety of firms). While FRAND imposes a reasonable maximum of royalties a patent holder can charge, the past thirty years have shown that the companies active in the development of the technology are very capable of appropriating profits from their inventions. Hence the regime can be considered FRAND on both the technology and interface.

## Complementary assets and appropriation profile

Qualcomm owns a significant portion of patents covering GSM, 3G and 4G. Moreover, it has highly specialized complementary assets in the design and manufacturing of the modems and chipsets, and the corresponding programming skills to create the interoperability with the operating system. While there are companies building their own chipsets such as Apple, these commonly incorporate modems created by Qualcomm (Campbell, 2016).

These assets can be considered quite significant in the smartphone market. By not only designing and manufacturing the chipsets, but also integrating compatibility with the most popular operating system, Android-based smartphone manufacturers have the choice between spending time and resources on creating changes to Android to support their chipset, or use a chipset provided by Qualcomm. For the majority of manufacturers it is far less expensive to rely on Qualcomm for their Chipsets. Because compatibility is ensured at the core of Android, stability is ensured. Moreover, because Qualcomm contributes to Android directly, their updates are included in the new Android releases, resulting in a shorter time to market. This position has been very profitable for Qualcomm and has allowed the company to become the market leader in chipset manufacturing. To ensure this position Qualcomm has had to develop significant specialized assets, not only in the chip/modem manufacturing space, but also on the programming side of operations.

Qualcomm has a very offensive appropriation profile and is the largest net receiver of royalties in the mobile broadband market with over $7 billion in royalties in 2015 (Qualcomm, 2016). On their website Qualcomm has a specific section for licensees, an extensive overview of their patent portfolio, and in their annual report they explicitly write about their patent license income, which are all signs of an offensive IPR policy.

## Resource Characteristics

### Value and Rareness

Looking at the different contributions Qualcomm makes to different Open Source projects the majority of them are to create demand for their core business, which is chipsets. Marcello Lioy (someone at Qualcomm) has stated: "*QuIC is a subsidiary of Qualcomm that brings together a bunch of engineers that are focused on doing open source. So what they do is they optimize open source to run on our platforms such as our Snapdragon chips, and that is part of why we participate in all these open source projects. To make them run on our hardware and optimize them and make the mobile experience better*" (Marcello Lioy, 2011).

As detailed in this quote, the core business for Qualcomm is not the software, but rather the hardware. However, when looking at the contributions made to Android covering GSM, 3G and 4G, these technologies are at the core of Qualcomm's business and Qualcomm is strongly dependent on its licensing fees. While Android per se is not of high value to Qualcomm, the code it contributes covering its patented inventions is.

### Inimitability and Non-substitutability

While Qualcomm is the patent owner of these technologies, the patents are committed under FRAND, hence they are available for anyone on the market. Yet, GSM, 3G and 4G are the dominant design to which Qualcomm owns the patents, making their technology non-substitutable and basically inimitable, as any imitation would still be covered by the patent. Overall these resources can be considered to be highly valuable, and while not necessarily rare in the market, the FRAND conditions make it so that Qualcomm receives royalties for any application or imitation, while the technology itself is non-substitutable.

**Adopted Open Source strategy**

As highlighted by the quote of Marcello Lioy (2011) in the previous section, Qualcomm's main focus is on its hardware business, and contributes only to Open Source projects to optimize them for their hardware. This is a classic example of a complements strategy, where Qualcomm only contributes those resources which ensures better performance of their hardware business.

## 4.3 Cisco

Cisco is an American company specialized in the manufacturing of internet based networking equipment and other products and services related to information technology. The company was founded in 1984 and is headquartered in California. The firm is active in a range of product areas including routing, switching, datacenters, and virtualization. It's routing and switching activities are focused on home networks, enterprise networks and high-end products for fixed and mobile service provider networks. Both the datacenter and routing operations are relevant to the wireless telecommunications industry.

Cisco is active in many fields of the telecommunications industry, but the majority of its activities are not directly relevant to the wireless broadband technology. The firm groups its technologies into the following segments: Switching; Next-Generation Network (NGN) Routing; Collaboration; Service Provider Video; Datacenter; Wireless; Security; and Other Products. Additionally,Cisco provides a range of services.

The majority of Cisco's business is focused on products revolving around the physical network, rather than mobile broadband. Cisco's activities in the wireless telecommunications industry are focused on the intersection between wired- and wireless networks. Cisco produces carrier grade routing and switching hardware and has recently shifted focus towards software defined networking (SDN). SDN decouples the hardware and software in network equipment and allows for the majority of operations to take place within datacenters.

Cisco is a major player in datacenter manufacturing and as a result will claim a more prominent role in the wireless sector in the future. The sale of carrier grade routing equipment resulted in a net revenue of $7.7 billion in 2015, which represented 20.4% of Cisco's total revenue. Cisco does not report on operating margins for its different business segments.

Cisco does not report on its patent portfolio or its licensing income in its annual report, indicating that its portfolio is primarily used for defensive purposes. Cisco reported that the company owned about 13,500 patents in 2012 (Munich Innovation Group, 2013). However, these patents do not cover the 3GPP standards and Cisco does not own any related SEP.

Cisco has had multiple run-ins with the Free Software Foundation (FSF) between 2003 and 2009, mainly centered on the use of GPL-based software while not releasing the derivative works under the same license. The FSF finally decided to file for a copyright infringement in 2008, which was settled by Cisco early 2009. The settlement included a donation from Cisco to the FSF and the appointment of a "free software director" in charge of monitoring license compliance at Cisco (Paul, 2009).

Since the somewhat mandatory engagement with OSS in 2009 Cisco has undergone a strong learning development. After engaging with a small amount of projects in the beginning the company has started to expand its Open Source activities over the years. By now the company is sponsoring several projects and is contributing large amounts of code to a variety of projects. However, the projects Cisco contributes to, are very carefully selected and fit a specific goal. On the one hand the firm wants to showcase the capabilities of its infrastructure and promoting its services under developers, while on the other hand the firm is ensuring that the various software packages operate on its infrastructure and the firm is not dependent on third-party software suppliers.

### 4.3.1 OpenStack

OpenStack is an OSS for the management of cloud-infrastructure. The software was first developed by NASA and Rackspace and was released under the Apache V2.0 license in 2010. A 'cloud' is located in a datacenter, on a large number of different servers. Each individual server runs software that allows to run multiple virtualized servers on the hardware. In the past, every single physical server had to be configured independently, making for a large overhead in datacenters. OpenStack connects to the virtualization software and allows users to configure multiple physical servers from a single interface, reducing this overhead.

While there are proprietary alternatives available, these are owned by competitors such as Microsoft (Azure) and Amazon (AWS). Cloud management software plays an integral part of transforming a datacenter containing numerous individual servers into a cloud operating as a single entity.

OpenStack is not based on a singular standardized technology. Therefore it is not possible to look at a specific technology or SDO to analyze. However, OpenStack itself can be considered a standard in the market, and development is organized through the OpenStack foundation. This foundation is supported by over 150 companies worldwide and behaves much like a SDO. It's processes are organized through a technical steering committee and different user groups.

#### Openness, design paradigm and appropriability regime

As discussed in Section 4.1.2, OpenStack scores 11 points on openness, is in its pre-paradigmatic stage and has an open appropriability regime.

#### Complementary assets and appropriation profile

Cisco does not own a proprietary version of cloud management software, the company is focused on the hardware business, and does not consider itself to be a software vendor (Cooney, 2016). From their perspective it does not matter which software runs the datacenters they help create. Their complementary assets are centered on this idea, they own highly specialized assets in hardware, consulting and services surrounding datacenters.

As Cisco is not primarily focused on the software side of cloud datacenters, I have rated their profile regarding this area of business as defensive. The company does not own any significant patents regarding the software, and is not focused on seeking licensing deals (Cisco, 2016).

## Resource characteristics

### Value and Rareness

As OpenStack is gaining popularity offering compatibility between OpenStack and the Cisco datacenters is increasingly important for Cisco's position. Additionally,by contributing specific functionality Cisco is able to showcase the capabilities of its datacenters. As stated by Lew Tucker (2012), CTO for Cloud at Cisco: "*The idea behind Quantum was to have a networking component in OpenStack that was a peer to compute and storage. Quantum provides the core elements required to build a networking-as-a-service module. That's why it's called Quantum, it is meant to be the most basic elemental part of what is need to create isolated network segments*".

There is value to the competitive advantage for Cisco in that OpenStack helps Cisco market its capabilities, even though OpenStack itself is not very valuable in terms of competitive advantage. Yet, interoperability with its hardware makes the Cisco cloud a more attractive product to many clients. Finally, Cisco expands on OpenStack with a number of proprietary extensions (Cisco, 2016). OpenStack is vital for these extensions, which is why I have rated OpenStacks value to Cisco as low.

### Inimitability and Non-substitutability

As described in the previous sections, there are many alternatives to OpenStack, which are easily obtainable for anyone in the market. Even though at the core cloud management software is quite generic, the add-ons Cisco offers are unique and distinguish Cisco from its competitors. These add-ons are focused on specific functionality and for the majority there are no direct alternatives available. Hence, even though the OpenStack core scores very low on IN characteristics, these add ons have a distinguishing value. Therefore I have rated the IN characteristics as medium.

## Adopted Open Source strategy

A main driver behind Cisco's Open Source activities is to make developers more comfortable with Cisco's software. Cisco is contributing parts of its proprietary work to the OSS to achieve this goal. Moreover, the Open Source projects also provide Cisco the opportunity to showcase the capabilities of their proprietary infrastructure.

While the architecture-level software is made Open Source, the company has a range of proprietary add-ons to differentiate itself from competitors such as Juniper, VMware and Intel which are also using OpenStack. Additionally, rather than promoting a Software-as-a-Service or a Platform-as-a-Service Tucker expects Cisco to be in the "*Everything-as-a-Service*" (Tucker, 2011) business, indicating its goal to transition from a hardware or software vendor towards a service and solutions company. Cooney, senior director of Open Source Strategy since 2015, commented on Open Source generated revenues as "*that will be something we will be discussing in the coming weeks but I would expect it to be more around services and support than having a [software] distribution*" (2016).

While Cisco is moving towards a deployment strategy, for OpenStack its primary strategy is hybridization. The company offers several proprietary extensions to the Open Source core, and is actively contributing to this core to make users more familiar with Cisco's software, thus making it easier to use Cisco's extensions rather than those of competitors.

### 4.3.2 Software Defined Networking

Cisco is also contributing to OpenDayLight, and like Ericsson is a platinum member of the foundation. In 2012 Cisco contributed its ONE controller to OpenDaylight. "*We have contributed our ONE Controller as perhaps the termination point of OpenDaylight […] This is Cisco's largest contribution into open source in the history of the company*" (Ward, 2013). Cisco's assets are primarily centered on networking hardware, which is strongly related to the SDN technology at the core of OpenDayLight.

#### Openness, design paradigm and appropriability regime

As described in Section 4.1.2, SDN scores 10 out of 13 points on openness, is the dominant design and has an open appropriability regime.

#### Complementary assets and appropriation profile

Cisco is specialized in producing networking equipment, which powers both datacenters and network nodes. OpenDayLight lets users move specialized functions from the network node to the datacenter. Hence it can be concluded that Cisco owns specialized complementary assets in the manufacturing of hardware on both sides of this equation.

Cisco's appropriation profile towards SDN is defensive, as described in Section 4.1.2, the technology is defined under a royalty free IPR policy, and a defensive suspension. As Cisco is a key contributor to both OpenDayLight and the ONF SDN standard it cannot assert its patents in an offensive fashion.

#### Resource Characteristics

##### Value and Rareness

The software itself is of low value to Cisco; SDN is just one option to implement on the Cisco hardware, which is of some complementing value to the hardware at best. Moreover, SDN is a technology field with a very open approach and a low level of appropriability. Therefore I have rated the value and rareness as none.

##### Inimitability and Non-substitutability

As discussed in the Ericsson case study, the imitability and substitutability of SDN is very high, due to the free and open IPR policy of the Open Networking Foundation. Therefore I have rated these characteristics as low.

#### Adopted Open Source strategy

The hardware sold by Cisco is not specifically designed to work with OpenDayLight, rather it is one of the applications of the hardware. Hence this strategy is not a complementing strategy, rather Cisco is contributing to the software in order to change the status quo, and offering professional services and consulting around the software. Cisco offers implementation, consulting and support on network infrastructure, including the use of OpenDayLight. I have classified the strategy adopted by Cisco as a deployment strategy.

## 4.4 Telefónica

Telefónica is a Spanish multinational broadband provider operating in 21 countries. The firm was founded in 1924 and is headquartered in Madrid. Telefónica operates brands such as Movistar, O2 and Vivo. The firm is active in both wired and wireless broadband and has approximately 320 million subscribers. It is one of the largest broadband operators in the world.

Telefónica is primarily active as a network operator. The company is active in both wired- and wireless subscriptions. Like the majority of network operators Telefónica is bundling a large part of its wireless subscriptions with the sales of mobile handsets. Telefónica offers networks in many different countries, and its complementary assets are not the same for each country. Telefónica outsources its network management to specialized firms. For example, in Brazil Ericsson is providing and managing both the operations support systems and the business support systems (Laughlin, 2010).

As a result, its complementary assets are primarily focused around sales of subscriptions, and sales of mobile handsets. In 2015 Telefónica's mobile business revenue totaled € 27,936 million, 87.7% of which corresponded to mobile subscriptions, and 12.3% represented revenue generated by handset sales (Telefonica, 2016). Overall, Telefónica's operating margin has been declining over the past years. This is a common trend in the mobile operating market. The technological developments of 3G and 4G has made network operations more expensive, while applications such as Whatsapp and Skype have eroded on the revenues on SMS and phone calls. Simultaneously, customers are not willing to increase spending in the same order of magnitude for their data plans, resulting in decreasing operating margins.

Telefónica owns a total of 545 patents (Telefónica, 2015, p.95). These patents are focused around three pillars: big data, internet of things and network infrastructure.

*"Our Research and Development is focused on two tracks: network innovation and digital services. We have IPR in both worlds. With regard to the latter we innovate in different areas, like personal communications, cloud services, security services, and video services. We produce IP in more or less all the areas where we have some innovation activity. With regard to the networks we have IP in both mobile and fixed networks and network virtualization. The usual suspects nowadays where all the companies in the sector are focusing their innovation."* (Fullea Carrera, 2016)

While the firm is working on development of 5G, it does not own SEP to the current 3GPP standards. The firm's appropriation profile is primarily defensive, filing patents to protect their own activities, rather than seeking licensing deals covering core technologies. The main value of the IP for Telefonica lies in its defensive abilities:

*"Losing the ability to enforce a patent within the Open Source ecosystems we engage in is not an issue for us, we are one of the drivers of the initiative. We do think that IP always needs to be protected as it is a valuable asset. Our intention is to protect out innovations as much as possible. […] Even in those cases where we think that having the IPR is still valuable. In case we provide royalty free licenses we still maintain the defensive value of the patents. Contributing to Open Source does not mean giving up our IP" (Fullea Carrera, 2016).*

### 4.4.1 Network Functions Virtualization

The concept of NFV was discussed in Section 4.1.1. For Telefonica NFV is of particular interest because it is a network operator. Implementing NFV in a network can significantly decrease deploying and maintenance costs and the required investments for building new capabilities into the network (SDX central, 2016). Telefonica's profit margin has been declining for the past five years to a historic low-point of 6% (Telefonica, 2016), as described in the introduction to this case study. Therefore Telefonica has taken an interest in NFV and has created the NFV Reference Labs in 2014 (Rath, 2014) which in 2015 released its OpenMANO NFV orchestration software as OSS (Telefonica, 2015). Together with other operators the firm persuaded ETSI to host its first ever Open Source project (ETSI OSM) aimed at developing a reference software implementation of its management and orchestration standard. However, the standardization process is still ongoing, and the aim is for the Open Source project to test ideas and provide input into the standard. Telefónica's OpenMANO was subsequently integrated into the ETSI OSM project.

#### Openness, design paradigm and appropriability regime

As described in Section 4.1.1, NFV is standardized by ETSI, which scores 10 out of 13 points on openness. The standard has not been finalized yet and currently the market is in a predominant phase. Its current appropriability regime is categorized as commercial licensing under NDA, as the FRAND conditions places upon the patents of firms involved with the development of the standard cannot be enforced until the standard is released.

#### Complementary assets and appropriation profile

Telefónica is primarily active as a seller of internet subscriptions, whereas the actual infrastructure and it's management/maintenance is outsourced to companies such as Ericsson. As a result their complementary assets are focused on sales, and not on network operations. Wikipedia (2016) lists hundreds of different Telecom operators, and in any given country there are often between 4 and 10 competitors in the market, all primarily focused on sales, while outsourcing their networks. Therefore it can be concluded that these are primarily generic SCA's.

Over the past years profit margins for operators have been under pressure, including those of Telefónica, and in an effort to decrease operating costs and increase its operating margins, the firm is looking to expand its complementary assets to include network operations (Thompson & Serrano, 2016).

The firms that own these complementary assets have developed a strong position in these assets through both patent portfolio and the developed expertise. Hence acquiring these assets is capital intensive and has strong barriers. To minimize investments Telefónica decided to use an Open Source approach to the development of these assets, as it will allow them to share development with other operators.

The company owns some patents related to NFV (Fullea Carrera, 2016). However, ETSI OSM is licensed under the Apache V2.0 license. One of the requirements of the Apache V2.0 license is that a contributor supplies a free perpetual grant for any patents it owns that would cover the contributions it has made to the software. Additionally, the Apache V2.0 license prevents users from filing patent litigation over patents covering any functionality of the software through a defensive suspension. "*We contribute to Open Source projects where we don't fear this [losing appropriability], or we don't find it an issue compared to the benefits we get from our contribution to the project. That is always a part of the decision to engage with Open Source projects*" (Fullea Carrera, 2016). This statement indicates that Telefonica's profile towards NFV is defensive.

### Resource characteristics

#### Value and Rareness

For Telefonica, NFV is a key instrument in lowering network costs and capital expenditure. As their profit margin has been declining over the past five years, this is a requirement for the firm, that has some strategic value. NFV is a technological development which can be considered as a prerequisite in order to be able to compete in the market for operators. Therefore, I have categorized NFV as having weak value and rareness characteristics for Telefonica.

#### Inimitability and Non-substitutability

Network operating software is offered by numerous vendors and is a minimum requirement to compete for any operator. The software can be procured from many different vendors, and currently there are multiple Open Source projects working on an implementation. NFV is an infrastructure type software which is currently being standardized, hence it can be concluded that the underlying functionality is widely known. Therefore I have rated these characteristics as low for NFV.

### Adopted Open Source strategy

Telefónica's Open Source strategy is aimed at developing additional complementary assets while minimizing the required investments. They have created OpenMano and subsequently 'donated' the code to the ETSI OSM project, where they are an active participant. Their main goal here is to create community focused on developing something they need themselves as well: "*Basically the creation of an ecosystem is one of the reasons for us to contribute to Open Source. The most usual cases for us to contribute to Open Source is those where we see that we can create a collaborative scene where many companies can work together*" (Fullea Carrera, 2016). This strategy can be classified as a self-service strategy.

## 4.5 Apple

Apple is a US based multinational technology company that is mainly focused on consumer electronics. The firms main products consist of the Mac, iPhone and iPad. The firm was founded in 1976 and is headquartered in California. Apple is the largest IT company in the world based on both revenue and assets, and is the second largest mobile phone manufacturer (behind Samsung). Apple entered the smartphone market in 2007 and has currently released 13 different models.

Apple's main revenue used to be generated by the Mac computer products and the iPod, but the iPhone has become by far its biggest revenue centre over the past decade. Apples total net profit rose from $1,330 million in 2005 to $5,339 million in 2015. The majority of this increase can be attributed to the iPhone. Within 2 years after its initial release the iPhone overtook laptop sales, which was up to then the main revenue centre for Apple. In 2015 iPhone net sales was $155,041 million which represented 66% of Apple's total net sales, whereas the iPhone only represented about 35% of Apples net sales in 2010.

Apple is active in three major complementary assets: chipsets, devices and sales. The chipsets produced by Apple are used exclusively in their own products. While Apple sells some devices through their web-shop and a limited amount of physical stores. Apple stores are primarily located in the US, while the rest of the world is covered by 'premium resellers'. The majority of Apple phones are sold through partnerships with wireless broadband operators in combination with a subscription.

The Apple patent portfolio consists of 10,942 patents (Curtis, 2015), but the firm is not reporting any of their patents on their website, or in their annual report. The financials also do not report on licensing revenue, indicating that Apple's appropriation profile is primarily defensive.

As described in Section 2.6, Apple has been engaging with OSS for a long time. Since 1999, all of it's operating systems are build on top of FreeBSD (Markoff, 1999), a unix-based operating system with a permissive license. However, before FreeBSD is integrated into the Apple operating systems, Apple makes many changes to the kernel. Apple releases these updated kernels as "Darwin" under the Apple Open Source License, which can be classified as a permissive license.

By using a permissive license Apple is able to create a range of proprietary features on top of the kernel, such as the graphical user interface and the entire networking functionality. These additional features are not included in the Open Source releases of Darwin and are what distinguishes OS X from other operating systems.

Apple launched OpenDarwin as a project to increase collaboration between Apple developers and the Open Source community, focused on developing the kernel and creating a community-led operating system based on Darwin. However, over the years the community felt that the project was not supported enough by Apple and was merely used as a "*hosting facility for Mac OS X related projects".* The development of a standalone Darwin operating system failed according to the community website. "*Availability of sources, interaction with Apple representatives, difficulty building and tracking sources, and a lack of interest from the community have all contributed to this*" (OpenDarwin, 2006).

### 4.5.1 XNU

iOS Apple's mobile operating system for its iPhone, iPad and iPod touch. iOS is based on the XNU kernel, which is an Open Source Unix kernel based on a hybrid between two other Open Source unix kernels (FreeBSD and Mach). In many cases Darwin is exactly like FreeBSD, but under the hood there are significant differences changing how internal methods work. Hence XNU can be considered a new standalone kernel and a technology on its own and the underlying technology standard to iOS. Android is the main competitor to iOS and is the market leader in the smartphone market with about 87.6% market share, iOS has a market share of roughly 11.7% (IDC, 2016).

#### Openness, design paradigm and appropriability regime

While XNU is released under the Apple Public Source License, which is has a partial copyleft clause making it incompatible with GPL. It cannot be considered true copyleft because under the license developers are allowed to link to proprietary files. Moreover, there is no standalone developer community building XNU. While XNU is Open Source, and it is possible to submit contributions, Apple has the final say on anything that is submitted to the kernel.

These processes are not transparent to the public, and for non-Apple employees it is not possible to get a seat at the table or vote on any decision. So the processes are completely closed, while the results are completely open. Therefore XNU scores 6 out of 13 points on openness.

Within the Unix world there are already countless different kernels available, such as GNU/Linux, BDS, NetBSD, Solaris and XNU. Additionally,there are also kernels available not based on the Unix architecture such as the Windows NT kernel from Microsoft. Therefore it can be concluded that this technology landscape is fragmented and is in a predominant stage.

The Apple Public Source License is recognized by the Free Software Foundation and the Open Source Initiative as an Open Source license. The License contains a patent grant clause covering any contributions. Moreover, the license stipulates that if someone is making a contribution covered by a patent from a third-party, it is the contributors responsibility to obtain a patent license before redistributing the code (Apple, 2003). Therefore it can be concluded that this license enforces an open appropriability regime.

## Complementary assets and appropriation profile

Apple has developed a range of specialized assets for XNU. More specifically, XNU is the kernel around which all of Apple's hardware and software is created. The Apple chipsets are optimized for XNU, while iOS is a proprietary extension of the kernel. The iPhone, iPad, and iPod touch are all designed to operate iOS and it's XNU kernel. Hence, it can be concluded that Apple owns a number of SCA's regarding XNU.

Apple created the Apple Public Source License, including its patent grant. Therefore it is safe to assume that the firm has a defensive profile regarding XNU. Though, when looking at iOS, the firm is far more protective. Apple regards al its iOS extensions as proprietary products and is actively defending their unique functionalities. Since 2011 Apple has been involved in multiple high-profile cases with Samsung over its mobile software, design concepts and more (Reisinger, 2015). However, it's prime target in these cases is to protect it's own products, rather than to license technologies to as many producers as possible. Regarding iOS Apple's profile is also primarily defensive.

## Resource characteristics

### Value and Rareness

The ecosystems Apple creates for its hardware are a large part of the firms unique offering and is what makes it stand out from Android devices or PC's. When purchasing an Apple product consumers are not buying a piece of hardware, or an operating system; they are buying the complete product. Hence the operating system has a high strategic value for Apple. However, the operating system is a combination of the XNU kernel and the proprietary extensions. As discussed earlier in this chapter, Unix kernel's are far from unique in the world, and the differentiating power lies in the extensions Apple has made to this kernel. I have rated the value and rareness of the technology to be low, because even though the majority of the value lies in the proprietary extensions, they are directly reliant on the XNU kernel itself.

### Inimitability and Non-substitutability

While there are some differences between Apple's iOS and Android, in functionality they are perfect substitutes. Even though there have been a number of patent infringement cases between Apple and other smartphone manufacturers, these cases have either been dealing with design patents, or with the fairness of the licensing terms under FRAND conditions. Operating Systems have been around for long enough to be considered infrastructure type software these days, and their underlying functionality are part of the common knowledge domain. Therefore I have rated the IN characteristics as low.

## Adopted Open Source strategy

I have classified the strategy followed by Apple as a hybridization strategy. iOS is differentiated from XNU by its proprietary extensions which include all of the graphical features, but also some core functionality like networking and other services that run on the system. All of these extensions are created by Apple and are not released to the public.

# 5 Conclusion

Analyzing data from case studies cannot be done using statistical methods, as the sample is to small for such an analysis. Instead, in case study research data is tested by comparing the ranking according to the values of the independent variable with the ranking according to the value of the dependent variable. If the two rank orders follow the pattern described in the proposition, the proposition is confirmed. If the rank orders differ, it must be determined whether the difference is such that it can be concluded that the two rank orders have no relation to each other, or that the rank orders have the same tendency (Dul & Hak, 2008).

### Proposition 1: A more open standard will likely lead to a faster adoption of a dominant design.

The relationship described in this proposition is probabilistic, meaning that on average, open standards will lead to a faster adoption of a dominant design, but this is not necessarily true for every single case.

Not all Open Source projects in the case studies are based on standards: OpenStack and IOS are not directly related to standards, even though some parts of these software packages are. However, a software package such as IOS incorporates many different standards for things like data storage, memory management, interaction with the different wireless modems (Bluetooth, Wifi, GSM, 3G, 4G, NFC), interaction with SD cards, USB interfaces and many more. While many of these standards are created by SDOs, there is not a single standard that takes precedent over the others in mobile operating systems, e.g. Wifi and 3G/4G are both key to the operating system.

The remaining three projects are all based on standards developed in very popular SDOs. Webkit covers the Javascript, HTML and CSS standards developed by W3C, ETSI OSM covers the ETSI NFV standard, GSM/3G/4G is developed by 3GPP and OpenDayLight is based on the OpenFlow standard developed by the Open Network Foundation. As a result, these standards instantly became the dominant design once they were published.

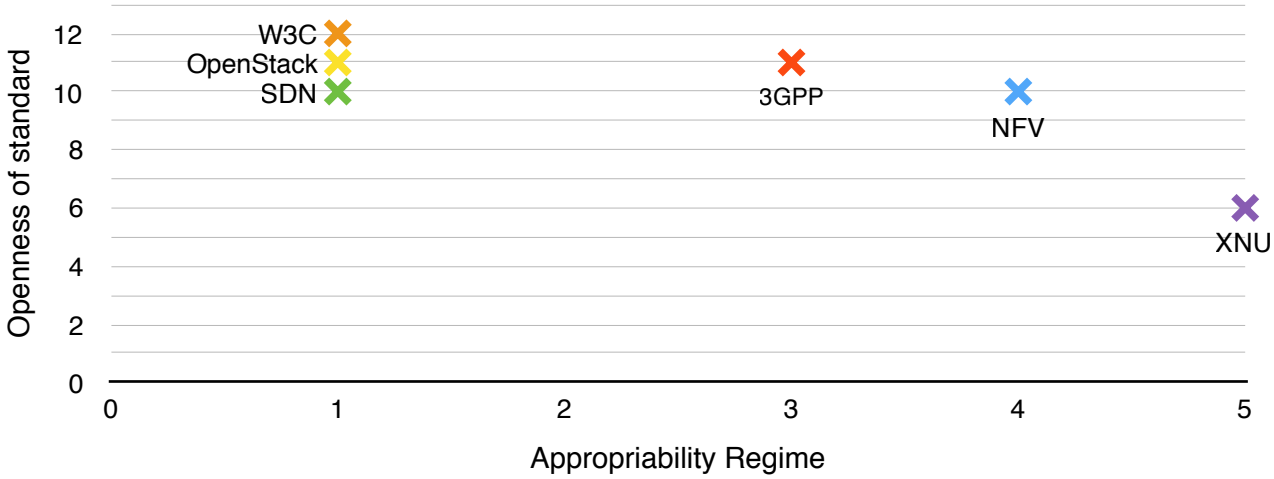
While the measures described in Chapter 3 represent the degree of openness and design paradigm of the standards, it is not possible to draw any conclusions regarding the adoption speed based on these case studies. First of all, the standards described in the case studies have all been developed by very popular SDOs that are globally recognized by the majority of companies. As a result these standards never had any significant competitors, and after completion became the dominant design instantly. Second of all, these case studies provide a snapshot of the technologies, as the study did not allow for a longitudinal approach. The adoption speed of a dominant design is something that manifests over time, and cannot be measured using the chosen research approach.

### Proposition 2: A more open standard will likely result in a weaker appropriability regime than a less open standard.

The relationship described in this proposition is probabilistic, meaning that if the average openness of a pool of standards increases, the mean strength of the appropriability regime for this pool will decrease. However, for an individual case this is not necessarily true. As this study applies the case study approach with a low number of cases, statistical analysis is not a reliable measure. Moreover, this study is a preliminary test of a newly developed model. Therefore the patterns in the data have to be examined to determine if they support the proposition.

The openness of the standard is measured using an adapted version of Krechmer (2005) resulting in a grade between 0 and 13, with 13 being the most open and 0 being the most closed. The appropriability regime has been measured using the IPR policy of the standard ranging from 1 to 5, with 1 being the weakest regime (royalty-free IPR), and 5 being the strongest regime (commercial licensing under NDA). The expected pattern is a negative linear relationship between the two variables. The measured data for the seven technology standards is presented in Figure 6. The pattern observed in this graph is similar to the expected pattern, with the most open standards being situated in the top-left corner of the graph, and the more closed standards being situated further to the bottom right of the graph. Therefore it can be concluded that proposition 2 is confirmed in this study.

**FIGURE 6: A MORE OPEN STANDARD WILL RESULT IN A WEAKER APPROPRIABILITY REGIME THAN A LESS OPEN STANDARD.**




**Proposition 3: Firms are more likely to have specialized complementary assets for technologies in a dominant stage than for technologies in a pre-dominant stage.**
This relationship is probabilistic, meaning that on average, more firms will have specialized assets for technologies in a dominant stage, than for technologies in a pre-dominant stage. When looking at the data presented in Table 6, this is proposition is not supported.

Both Ericsson and Cisco have specialized assets in both pre-dominant technologies and in dominant technologies, while Apple has specialized assets in both its pre-dominant technologies. The emergence of a dominant design is not sudden, rather it is represented by an ever growing market share of the technology. Especially in slow developing technologies companies will not be able to afford waiting for a dominant design to emerge before investing in specialized assets, as this will potentially mean missing out on years of revenues. Moreover, at least in the software industry not all technologies ultimately converge on a dominant design. For example, there are countless of programming languages in the market, and while some are more popular than others, this has been the case for the past sixty years with none of the languages taking the lead in becoming the dominant design. The majority of early years programming languages such as FORTRAN and Basic have been replaced, but today languages like Java, Objective C, Perl, PHP, Ruby and ASP are still highly popular.

**TABLE 6: FIRMS ARE MORE LIKELY TO HAVE SPECIALIZED COMPLEMENTARY**

| Complementary Assets Type | | |
|---|---|---|
| **Specialized** | NFV - Ericsson<br>XNU - Apple<br>OpenStack - Cisco | SDN - Ericsson<br>SDN - Cisco<br>3GPP - Qualcomm |
| **Generic** | NFV - Telefonica<br>OpenStack - Ericsson | W3C - Qualcomm |
| | **Pre-dominant** | **Dominant** |

Design Phase

### Proposition 4: An offensive appropriation profile for a firm requires a tight appropriability regime.

The appropriation profile of a firm depends per technology area, a firm can have a very offensive profile for one technology, while being primarily defensive for another. This is strongly influenced by the initial openness and appropriability regime of the standard. For example, in OpenDayLight the standard is produced under a royalty free agreement, whereas the NFV Mano standard is developed by ETSI under FRAND conditions. This influences the extend to which a company is able to patent the core technologies and to what extend it can maintain an offensive profile for the technology. Ericsson deploys both technologies in its portfolio, however its appropriation profile for ETSI NFV Mano is offensive, while its profile for SDN is defensive. Similarly, Qualcomm has a primarily defensive stance for its Webkit developments, which is based on the Open Source standards for HTML, CSS and Javascript, while it is very offensive for its GSM, 3G and 4G technologies which are developed by 3GPP under FRAND conditions.

**TABLE 7: AN OFFENSIVE APPROPRIATION PROFILE FOR A FIRM REQUIRES A TIGHT APPROPRIABILITY REGIME.**

| | | | |
|---|---|---|---|
| **Offensive** | | 3GPP - Qualcomm | XNU - Apple<br>NFV - Ericsson |
| **Defensive** | SDN - Ericsson<br>W3C - Qualcomm<br>OpenStack - Cisco<br>SDN - Cisco<br>OpenStack - Ericsson | | NFV - Telefonica |
| | **Open** | **RAND** | **Commercial license** |

Appropriability Regime

Proposition 4 describes a necessary condition, meaning that a tight appropriability regime is required for a firm to have an offensive profile in this field, but not every firm operating in a tight appropriability regime will have an offensive profile. In other words, for this proposition no cases in the top left quadrant are expected, as this would indicate a firm having an offensive profile in a technology field where it is unable to charge licensing fees. The data presented in Table 7 shows that none of the firms have an offensive appropriation profile in technologies that have an open appropriability regime, or technologies that have a FRAND technology regime which only allows for royalties on the technology and not the interface. However, as this last category allows for some extend of rent-seeking, this possibility is not excluded by the proposition. Based on the data presented in Table 7 it can be concluded that proposition 4 is confirmed by the data.

### Proposition 5: The value and rareness of a software resource is higher for specialized than for generic assets.

The nature of the generic asset is that it can be acquired easily in the market, while the specialized asset requires significant investments to develop. Therefore, we can assume that specialized assets are harder to come by, and more valuable than generic assets. The expected pattern for this proposition is that specialized assets have either weak, medium or strong Value and Rareness, while generic assets have either no, or score weak in Value and Rareness. The generic assets that are weak in Value and Rareness are not rare, but to some extend they are valuable to the firm. The data presented in Table 8 confirms this pattern, therefore we concluded that proposition 5 is confirmed by the data.

**TABLE 8: THE VALUE AND RARENESS OF A SOFTWARE RESOURCE IS HIGHER FOR SPECIALIZED THAN FOR GENERIC ASSETS.**

| | | | | |
|---|---|---|---|---|
| **Specialized** | | OpenStack- Cisco | XNU - Apple<br>SDN - Cisco | NFV - Ericsson<br>3GPP - Qualcomm |
| **Generic** | W3C - Qualcomm<br>OpenStack - Ericsson | NFV - Telefonica<br>SDN - Ericsson | | |
| | **None** | **Weak** | **Medium** | **Strong** |

### Proposition 6: The inimitability and non-substitutability of a software resource is higher for companies with a strong appropriation profile than for companies with weaker appropriation profiles.

Having a more offensive profile for a certain technology will result in a larger and stronger patent portfolio surrounding this technology. The firm will not only patent its specific applications, rather it will try to create a patent thicket covering a wide array of applications. As a result, the resources will be harder to imitate and more difficult to substitute, as any substitution based on the same basic technology will likely be covered (at least in part) by the firm's patents.

The pattern we are looking for is one where all resources covered by offensive profiles will score medium or strong on inimitability and non-substitutability, while all resources covered by defensive profiles will score none, or weak. The data presented in Table 9 confirms this pattern, hence we concluded that proposition 6 is confirmed by the data.

**TABLE 9: THE INIMITABILITY AND NON-SUBSTITUTABILITY OF A SOFTWARE RESOURCE IS HIGHER FOR COMPANIES WITH A STRONG APPROPRIATION PROFILE THAN FOR COMPANIES WITH WEAKER APPROPRIATION PROFILES.**

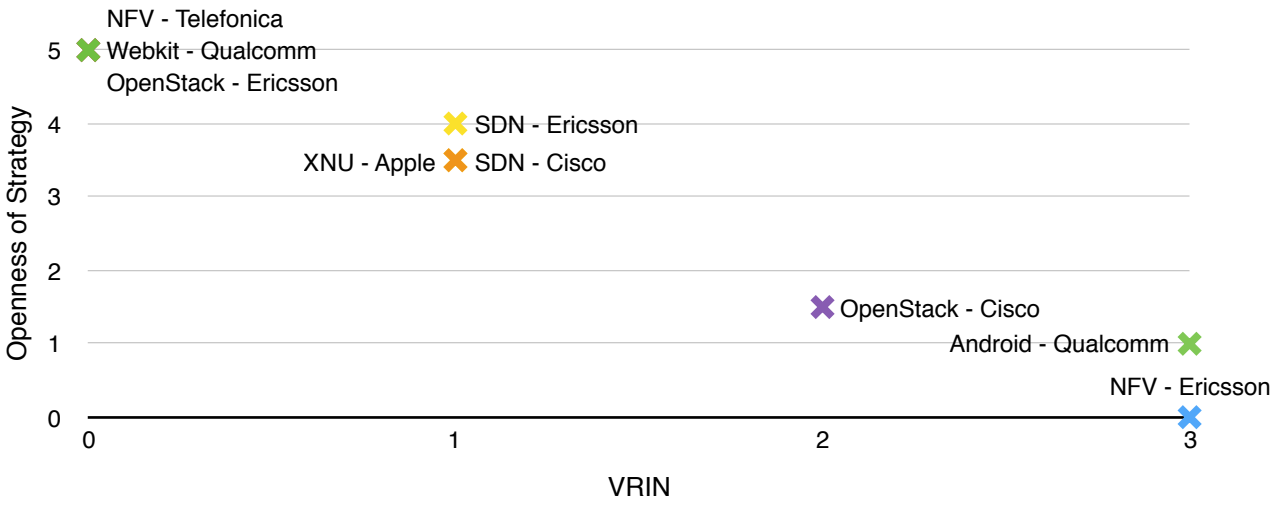
| | | | | |
|---|---|---|---|---|
| **Offensive** | | | XNU - Apple | NFV - Ericsson<br>3GPP - Qualcomm |
| **Defensive** | W3C - Qualcomm<br>OpenStack - Ericsson<br>SDN - Ericsson | NFV - Telefonica<br>SDN - Cisco<br>OpenStack - Cisco | | |
| | **None** | **Weak** | **Medium** | **Strong** |

**Proposition 7: Companies are likely to adopt more open strategies for resources with weak VRIN characteristics, than for resources with strong VRIN characteristics.**

All of the firms engage in Open Source projects that are complementing in an economic sense to their businesses even when the resource is not of direct value to the firm. Ericsson's contributions to OpenStack, or Qualcomm's contributions to Webkit are not valuable in terms of the RBV, but by proxy generate a return to both firms. Apple releasing and maintaining the Swift language helps third party develop applications for their products, increasing the value of these products. Cisco focusses its Open Source activities on datacenter operations, which they provide the hardware for.

Hence the Open Source activities are primarily guided by creating competitive advantage, demand for a complementary asset. Another reason to engage with Open Source can be to gain access to a resource that is necessary to operate, as is the case for Telefonica's ETSI OSM developments. This resource is becoming a requirement for operators in the market, but investments can be substantial for a single operator. As the resource is not a direct source of competitive advantage it makes sense for Telefonica to drive an Open Source project and seek additional developers to share the costs with.

**FIGURE 7: COMPANIES ARE LIKELY TO ADOPT MORE OPEN STRATEGIES FOR RESOURCES WITH WEAK VRIN CHARACTERISTICS, THAN FOR RESOURCES WITH STRONG VRIN CHARACTERISTICS.**




The observed patter between the VRIN characteristics of the software and the openness of the adopted Open Source strategy is illustrated in Figure 7. While there is some spread in the adopted strategies, the observed trend is a negative linear relationship as stated in Proposition 7. Therefore we concluded that proposition 7 is confirmed by the data.

## 5.1 Theoretical Model

Reflecting back on the theoretical model described in Section 2.7, a few changes have to be implemented based on the findings of this study. While the adoption speed of the design paradigm was not possible to measure during this study, the current state of the different design paradigms was. All of the tested propositions were confirmed by the data except for one. The positive relationship between the establishment of a dominant design and the amount of complementary assets owned by the firm was rejected. A possible explanation for this rejection was already provided earlier in this Chapter. However, based on this rejection I propose the following change in the model.

Rather than focussing on complementary assets owned by the firm, this variable should measure the intensity of competition for complementary assets. As the design paradigm is a market characteristic, this will apply to all firms in the market. An increased stimulation for one firm to invest in complementary assets based on market conditions, will apply to all firms. Therefore it can be expected that the establishment of a dominant design will lead to an increase in competition for complementary assets. As a result, there will be more pressure on the value and rareness of an individual firm's resources. When looking at the complete right side of the model this will have the following effect:

A more open standard will lead to a faster adoption of a dominant design, as a result competition for complementary assets will increase, and the value and rareness of an individual firm's resources will decrease. As value and rareness for the resource decrease, it makes more sense to adopt a more open strategy.

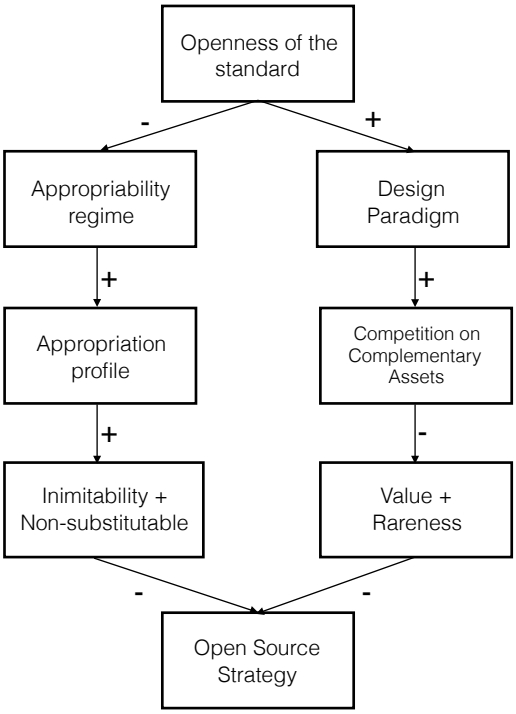



**FIGURE 8: ADJUSTED THEORETICAL MODEL**

Overall it can be concluded that the RBV aspects of value, rareness, inimitability and non-substitutability are valid tools to assess Open Source strategies for specific resources. This is not unexpected, as the RBV is also often applied in outsourcing decisions. However, the aim of this study was to not only place this decision at the resource level, but to take a broader approach and incorporate the scope of the organization and the technology as well. The data has supported the relationships proposed between the PFI framework and the RBV, indicating that the combination of these two fundamental theories provides a good fit and basis for Open Source decision making.

# 6. Discussion

The goal of this study was two-fold, on the one hand I wanted to find out in which cases firms engage with Open Source projects and how this benefits the firm, and on the other hand I wanted to develop a framework that can be used by firms to evaluate software projects and help them decide whether an Open Source approach would be beneficial. The first goal is covered by the theoretical contributions discussed in the next section, while the second goal is discussed in Section 6.2 Managerial implications.

## 6.1 Theoretical contributions

The purpose of this research was to answer the following question: "W*hat variables influence organizations to choose Open Source activities over proprietary development?".* While Open Source is becoming a very common topic for many different technology firms, research on the topic is lacking. Existing studies primarily describe a few success stories, or in what ways companies engage with Open Source projects. However, to date no study has investigated the reasons for companies to do so and the situations in which it makes sense for a company to engage with Open Source projects.

In this study I have developed a model providing clear insights into which variables are key to making Open Source decisions within a firm. By combining the PFI framework and the RBV I have created a model that is not only broader in scope than these original theories, but that is also filling the research gap in Open Source decision making. Additionally, these insights have been linked to specific Open Source monetization strategies. While such strategies were developed in earlier studies, a guideline on when to use what type of strategy was lacking. The model can be used to evaluate Open Source decisions based on both the external technological environment, and the internal configuration of the firm towards this environment. Based on this assessment the model provides insights on what type of Open Source monetization strategy to use.

Furthermore, an initial test of the model was performed providing preliminary insights into the validity of the model and creating additional research avenues, which will be highlighted in Section 6.4.

## 6.2 Managerial implications

Open Source engagements cary both risks and potential rewards. As for every company decision, investments should be evaluated based on its returns and Open Source contributions are only sustainable when it creates value for the company.

These decisions are highly context dependent, therefore they need to be addressed on a case-by-case basis. Each project covers a specific type of technology, that is of different value for each company. The model developed in this study supports managers to make a thorough assessment of the resource based within the context of both the company, and its external environment. Not every software resource is suitable for an Open Source approach, there are those that are simply too specialized to attract external contributors, while others are critical to a companies competitive advantage and should not be shared. Sharing valuable resources through Open Source often destroys this value for the company, as it results in sharing both the asset, and the IP.

Once the decision is made to engage with an Open Source project, the firm has to decide upon a strategy. The model gives an indication of what kind of strategy would fit depending on the VRIN characteristics of the resource. However, there is some leeway in these strategies and the firm has to decide which of these strategies it is best positioned to exploit.

## 6.3 Research limitations

The theoretical contributions notwithstanding, there are several limitations to the study. Firstly, the cases are all situated in the same specific context: the telecommunications industry. Secondly, the telecommunications industry is an industry with both a high rate of standardization and a large amount of patents. Therefore the results are not well suited for generalization. However, large parts of the model have been based on existing and tested theories, such as the PFI framework and the RBV, which allows arguing for some generality. Nonetheless, the main aim of this study was not to prove the general theory, rather this paper is an effort to develop a conceptual model and provide basic theoretical insights in how Open Source activities relate to the firm.

Another limitation to the study was the limited access the researcher had to primary data. While a number of managers and experts from Ericsson have been interviewed, for Qualcomm and Telefonica only one manager could be interviewed, and for Apple and Cisco no primary sources were accessible. While the data is primarily based on the triangulation of several (secondary) sources, verification with a primary source is preferable, and this was not possible for all cases. As a result the reliability of the data especially for the Apple and Cisco case can be questioned.

This research was conducted while the researcher was embedded with the IP licensing department of Ericsson. As this firm has a large patent portfolio and has an overall offensive profile regarding appropriability, this department has a certain agenda. Moreover, within ETSI, there are currently some Open Source developments ongoing on a project competing with one of Ericsson's products. Therefore there is a possibility that the researcher was influenced resulting in some degree of researcher bias. Moreover, the vast amount of data and technological details related to the different standards and patent portfolio may have caused misinterpretation of data by the researcher. These two issues could have negatively influenced the scientific nature of the research. Overall this study is not without its limitations, however as this was primarily a theory building and explorative research these limitations provide avenues for future research, which will be addressed in the following section.

## 6.4 Future research

The main focus of this study was the development of a conceptual model and conducting a preliminary test of the propositions. The next step in this research would be to extend the empirical testing of the model to a larger sample of data across multiple industries. While measures have been created for each variable, these are often of a low level of measurement and a more in depth analysis of the different constructs is required to determine the exact nature of the different relationships.

Within the current scope of the case studies proposition 1 could not be tested, as this requires longitudinal or historical data which was not available to the researcher. However, this is an important relation within the overall model. If a company decides to Open Source a previously closed technology, this will also result in a more open standard, changing the status quo for the entire environment. Therefore additional investigation into this relationship is valuable and recommended for future research. Additionally, I have proposed a change to the complementary assets variable, which will subsequently change the nature of the proposed relationships with that variable as well. Rather than looking at the complementary asset from a single firm’s perspective, I propose that this variable is changed to the intensity of competition on complementary assets. As a result, two new propositions:

**Proposition a:** The adoption of a dominant design will lead to an increase in intensity of competition on complementary assets.

**Proposition b:** An increased intensity of competition on complementary assets will reduce the value and rareness of a firm's resources.

The Open Source strategies described in this study were based on strategies described by Chessbrough and Appleyard (2007) in a Harvard Business Review article based on the four categories found by Perr, Sullivan, and Appleyard (2006). While these strategies seemed to fit to the cases in this study, this is not a peer reviewed source and additional or different strategies might exist within these categories. However, to date no Open Source strategy studies have been conducted. For that reason it would be valuable to investigate whether different or other strategies exist and how they relate to traditional strategies for outsourcing. Outsourcing decisions are fundamentally different from Open Source decisions, in that the firm retains ownership of the end product in outsourcing. Whereas in Open Source this is not the case. However, there are also valuable overlaps which could provide interesting insights into Open Source strategies.

# References

3GPP (2014). 3GPP Working Procedures. Retrieved from: http://www.3gpp.org/ftp/Information/Working_Procedures/3GPP_WP.htm

3GPP. (n.d.) Membership. Retrieved from: http://www.3gpp.org/about-3gpp/membership

Afuah, A., & Tucci, C. L. (2012). Crowdsourcing as a solution to distant search. Academy of Management Review, 37(3), 355-375.

Ahlberg, J. (2014). The implicit patent license grant in some popular Open Source licenses.

Amazon (2016). Fire Tablet, 7" Display, Wi-Fi, 8 GB - Includes Special Offers, Black. Retrieved from: https://www.amazon.com/dp/B00TSUGXKE

Apache Foundation (2004). Apache License Version 2.0. Retrieved from: https://www.apache.org/licenses/LICENSE-2.0

Apple, 2003. Apple Public Source License. Retrieved from: https://opensource.org/licenses/APSL-2.0

Ashby Jones (23 December 2014). "Rockstar Consortium to Sell 4,000 Patents to RPX Corp. for $900 Million". Wall Street Journal

Asklund, U., & Bendix, L. (2002, February). A study of configuration management in open source software projects. In Software, IEE Proceedings- (Vol. 149, No. 1, pp. 40-46). IET.

Baldwin, C.Y., von Hippel, E., 2011. Modeling a paradigm shift: from producer innovation to user and open collaborative innovation. Organization Science 22,1399–1417.

Bapst, M. (07-04-2010). Why is Qualcomm Innovation Center (QuIC) Investing in WebKit? Qualcomm OnQ Blog. Retrieved from: https://www.qualcomm.com/news/onq/2010/04/07/why-qualcomm-innovation-center-quic-investing-webkit

Bapts, M. (2010). Why is Qualcomm Innovation Center (QuIC) Investing in WebKit?. Retrieved from: https://www.qualcomm.com/news/onq/2010/04/07/why-qualcomm-innovation-center-quic-investing-webkit

Barney, J. (1991). Firm resources and sustained competitive advantage. Journal of management, 17(1), 99-120.

Bekkers, R., Duysters, G., & Verspagen, B. (2002). Intellectual property rights, strategic technology agreements and market structure: The case of GSM. Research policy, 31(7), 1141-1161.

Bell, J. (2005) Doing your Research Project (4th edn). Buckingham: Open University Press.

Benkler, Y. (2002). Coase's Penguin, or, Linux and" The Nature of the Firm". Yale Law Journal, 369-446.

Boehm, B. W., Madachy, R., & Steece, B. (2000). Software cost estimation with Cocomo II with Cdrom. Prentice Hall PTR.

Bohm, M. (2016). Open Source and IP in the Digital Society: Open Source and Standards.

Bright, P. (03-04-2013). Google going its own way, forking WebKit rendering engine. ARS Technica. Retrieved from: http://arstechnica.com/information-technology/2013/04/google-going-its-own-way-forking-webkit-rendering-engine/

Business Wire (2001). Websprocket Selected By Supranet Consortium to Enable the Internet With Smart Packet Technology; Platform Unifies Supranet Management Through Java and Oracle. Retrieved from: https://www.thefreelibrary.com/Websprocket+Selected+By+Supranet+Consortium+to+Enable+the+Internet...-a071935216

Butler, B. (17-09-2014). What is Metacloud and why did Cisco buy it? Network World. Retrieved from: http://www.networkworld.com/article/2684801/cloud-computing/what-is-metacloud-and-why-did-cisco-buy-it.html

Cassiman, B. & Veugelers, R. (2002). R&D cooperation and spillovers: some empirical evidence from Belgium. The American Economic Review, 92(4), 1169-1184.

Chesbrough, H. (2006) “Open innovation: A new paradigm for understanding industrial innovation,” in Henry Chesbrough, Wim Vanhaverbeke, and Joel West, eds., Open Innovation: Researching a New Paradigm. Oxford: Oxford University Press, pp. 1-12.

Chesbrough, H. & Bogers, M. (2014). Explicating Open Innovation: Clarifying an Emerging Paradigm for Understanding Innovation. In H. Chesbrough, W. Vanhaverbeke, & J. West (Eds.), New Frontiers in Open Innovation. (pp. 3-28). Section 1.Oxford University Press.

Chesbrough, H. & Di Minin, A. (2014). Open social innovation. New Frontiers in Open Innovation, 169-188.

Chesbrough, H. W. (2006). Open innovation: The new imperative for creating and profiting from technology. Harvard Business Press.

Chesbrough, H. W., & Appleyard, M. M. (2007). Open innovation and strategy. California management review, 50(1), 57-76.

Chiu, M. C., & Okudan, G. (2014). An investigation on the impact of product modularity level on supply chain performance metrics: an industrial case study.Journal of Intelligent Manufacturing, 25(1), 129-145.

Chiu, Y. C., Lai, H. C., Lee, T. Y., & Liaw, Y. C. (2008). Technological diversification, complementary assets, and performance. Technological Forecasting and Social Change, 75(6), 875-892.

Christmann, P. (2000). Effects of “best practices” of environmental management on cost advantage: The role of complementary assets. Academy of Management journal, 43(4), 663-680.

Cnet (02-01-2002). IBM to spend $1 billion on Linux in 2001. Retrieved from: http://www.cnet.com/news/ibm-to-spend-1-billion-on-linux-in-2001/

Coase, R. H. (1937). The nature of the firm. Economica, 4(16), 386-405.

Code Aurora Forum (2014), Legal information.

Colombo, M. G., Grilli, L., & Piva, E. (2006). In search of complementary assets: The determinants of alliance formation of high-tech start-ups. Research Policy, 35(8), 1166-1199.

Corner, S. (12-05-2016). Cisco bets big on open source. Computerworld. Retrieved from: http://www.computerworld.com.au/article/599696/cisco-bets-big-open-source/

Costello, S. (25-06-2015). Operators and vendors predict long life for 2G. Mobile World Live. Retrieved from: http://www.mobileworldlive.com/featured-content/top-three/operators-vendors-forecast-long-life-2g/

Crabtree, B. F. & Miller, W. L. (Eds.). (1999). Doing qualitative research. Sage Publications.

Curtis, S. (16-04-2015). Apple ramps up patent portfolio to take on Samsung. The Telegraph. Retrieved from: http://www.telegraph.co.uk/technology/apple/11542159/Apple-ramps-up-patent-portfolio-to-take-on-Samsung.html

Custer, R. L., Scarcella, J. A., & Stewart, B. R. (1999). The modified Delphi technique: A rotational modification. Journal of Vocational and Technical Education, 15 (2), 1-10.

Dahlander, L., & Gann, D. M. (2010). How open is innovation?. Research policy, 39(6), 699-709.

Dalkey, N., & Helmer, O. (1963). An experimental application of the Delphi method to the use of experts. Management science, 9(3), 458-467.

Dalkey, N.C. (1969). The Delphi Method: An Experimental Study of Group Opinion, prepared for United States Air Force Project Rand, Santa Monica.

Digistan (n.d.) Definition of a free and open standard. Retrieved from: http://www.digistan.org/open-standard:definition

Eats (2016). Annual report 2016: a connected world.

Ecplise Foundation (2004). Eclipse Public License 1.0 (EPL-1.0). Retrieved from: https://opensource.org/licenses/eclipse-1.0.php

Ericsson (2016). Facts and Figures. Retrieved from: https://www.ericsson.com/thecompany/company_facts/facts_figures

Ericsson (27-01-2015). Ericsson reports fourth quarter and full year results 2014. Retrieved from: https://www.ericsson.com/news/1889730

Erlang (n.d.). About. Retrieved from: https://www.erlang.org/about

ETSI (2015). Long-Term Strategy 2016-2021. Retrieved from: http://www.etsi.org/images/files/Brochures/ETSI_LTS%20Brochure_WEB.pdf

European Union, European Interoperability Framework, (2004). European Interoperability Framework for Pan-European eGovernment Service. Retrieved from http://ec.europa.eu/idabc/servlets/Docd552.pdf?id=19529

F.T. Rothaermel, C.W. Hill, Technological discontinuities and complementary assets: a longitudinal study of industry and firm performance, Organ. Sci. 16 (1) (2005) 52–70.

Faems, D., De Visser, M., Andries, P., & Van Looy, B. (2010). Technology Alliance Portfolios and Financial Performance: Value-Enhancing and Cost-Increasing Effects of Open Innovation*. Journal of Product Innovation Management, 27(6), 785-796.

Fairfield Resources International (2007). Review of Patents Declared as Essential to GSM Through June 6, 2007. Retrieved from: http://frlicense.com/GSM_FINAL.pdf

Fairfield Resources International (2009). Review of Patents Declared as Essential to WCDMA Through December, 2008. Retrieved from: http://www.frlicense.com/wcdma1.pdf

Fairfield Resources International (2010). Review of Patents Declared as Essential to LTE and SAE (4G Wireless Standards) Through June 30, 2009. Retrieved from: http://www.frlicense.com/LTE%20Final%20Report.pdf

Fedora Project (2016a). Fedora Sponsors. Retrieved from: https://getfedora.org/sponsors

Gassmann, O., & Enkel, E. (2004). Towards a theory of open innovation: three core process archetypes. In R&D management conference (Vol. 6, No. 0, pp. 1-18).

GNU (2015). What is free software? Retrieved from: http://www.gnu.org/philosophy/free-sw.en.html

Gordon, T. J., & Helmer, O. (1964). Report on a long-range forecasting study.

GSMA (2015). GSMA Mobile Economy 2016. Retrieved from: http://www.gsmamobileeconomy.com/GSMA_Global_Mobile_Economy_Report_2015.pdf

GSMA (2016). Definitive data and analysis for the mobile industry Retrieved from: https://www.gsmaintelligence.com/

Haglund, J. [TelecomTV]. (2016, 03, 16). Ericsson and Intel demonstrate the advantages of open source for SDN and NFV. Retrieved from: https://www.youtube.com/watch?v=swpISov3euY

Heiman, B. A., & Nickerson, J. A. (2004). Empirical evidence regarding the tension between knowledge sharing and knowledge expropriation in collaborations. Managerial and Decision Economics, 25(6-7), 401-420.

Henkel, J. (2006). Selective revealing in open innovation processes: The case of embedded Linux. Research policy, 35(7), 953-969.

Henkel, J., & Baldwin, C. Y. (2011). Modularity for value appropriation: Drawing the boundaries of intellectual property. Boston, MA: Harvard Business School.

Henkel, J., Schöberl, S., & Alexy, O. (2014). The emergence of openness: How and why firms adopt selective revealing in open innovation. Research Policy, 43(5), 879-890.

Hillebrand, F., Rosenbrock, K.H. & Hauser, H. (2013). The Creation of Standards for Global Mobile Communication: GSM, UMTS and LTE from 1982 to 2012. Freiburg: Hillebrand Consulting Engineers GmbH.

IDC. (2002). Windows 2000 Versus Linux in Enterprise Computing. Retrieved from: http://www.microsoft.com/windows2000/docs/TCO.pdf

Intellectual Property Licensing: Forms and Analysis, by Richard Raysman, Edward A. Pisacreta and Kenneth A. Adler. Law Journal Press, 1999-2008. ISBN 978-1-58852-086-9

Jaaksi, A. (2007). Experiences on product development with open source software. In Open source development, adoption and innovation (pp. 85-96). Springer US.

Johnson, S. (27-01-2014). Samsung to pay Ericsson $650 million plus royalties to end patent spat. Reuters. Retrieved from: http://www.reuters.com/article/us-ericsson-samsung-patents-idUSBREA0Q0A120140127

Kemerer, C. F., & Slaughter, S. (1999). An empirical approach to studying software evolution. Software Engineering, IEEE Transactions on, 25(4), 493-509.

Kerner, S.M. (12-04-2013). What is Cisco's Biggest Open Source Contribution Ever? Enterprise Networking Planet. Retrieved from: http://www.enterprisenetworkingplanet.com/datacenter/what-is-ciscos-biggest-open-source-contribution-ever.html

Kerner, S.M. (23-10-2012). Inside Cisco's Open Source OpenStack. Enterprise Networking Planet. Retrieved from: http://www.enterprisenetworkingplanet.com/netsp/inside-ciscos-open-source-openstack.html

Kogut, B., & Metiu, A. (2001). Open-source software development and distributed innovation. Oxford Review of Economic Policy, 17(2), 248-264.

Kratz, R. A. (1992). A comparison of the delphi technique with the nominal group technique (NGT) in determining the criteria to be used for measuring the value-added effect of associate degree graduates of two Pennsylvania community colleges. Retrieved from http://search.proquest.com/docview/303996819?accountid=106781

Landeta, J. (2006). Current validity of the Delphi method in social sciences. Technological forecasting and social change, 73(5), 467-482.

Laughlin, K. (15-05-2010). Telefonica Brazil Outsources Network Management to Ericcson. Nearshore Americas. Retrieved from: http://www.nearshoreamericas.com/telefonica-brazil-outsources-network-management-to-ericcson/

Laursen, K., & Salter, A. J. (2014). The paradox of openness: Appropriability, external search and collaboration. Research Policy, 43(5), 867-878.

Linstone, H. A. (1975). Eight basic pitfalls: A checklist. The Delphi Method. Techniques and Applications, 571-586.

Lioy, M. [O'Reilly]. (2011, 08, 04). Marcello Lioy (Qualcomm Innovation Center), interviewed at OSCON 2011. Retrieved from: https://www.youtube.com/watch?v=xLKvdwUsvjs

M. Tripas, Unraveling the process of creative destruction: complementary assets and incumbent survival in the typesetter industry, Strateg. Manage. J. 18 (1997) 119–142 (Special Issue)

Markoff, J. (17-03-1999). Apple Adopts 'Open Source' Code Strategy. The New York Times. Retrieved from: http://www.nytimes.com/1999/03/17/business/apple-adopts-open-source-code-strategy.html

Marshall, C., & Rossman, G. B. (2014). Designing qualitative research. Sage publications.

Matsumoto, C. (23-02-2015). Cisco Swallows the Red Pill of Open Source Software. SDC Central. Retrieved from: https://www.sdxcentral.com/articles/news/cisco-swallows-red-pill-open-source-software-onos/2015/02/

McMillan, Robert (May 21, 2012). "How Apple and Microsoft Armed 4,000 Patent Warheads". Retrieved November 1, 2013.

Meisnet, J. (Microsoft). (2014, April 8). Chinese Ministry of Commerce approves Microsoft-Nokia deal. Retrieved from: https://blogs.technet.microsoft.com/microsoft_blog/2014/04/08/chinese-ministry-of-commerce-approves-microsoft-nokia-deal/

Miller, M. (Fedora Project). (2016, February 11). Fedora Project Metrics-DevConf.CZ 2016 [Video file]. Retrieved from https://www.youtube.com/watch?v=DklOE_X2dM8

Mullin, J. (Arstechnica) (2014). Chinese gov't reveals Microsoft's secret list of Android-killer patents. Retrieved from: http://arstechnica.com/tech-policy/2014/06/chinese-govt-reveals-microsofts-secret-list-of-android-killer-patents/

Munich Innovation Group (2014). Cisco Systems. Retrieved from: http://www.prime-patent.com/wordpress/wp-content/uploads/2014/03/Cisco_Systems_Patent_Company_Report.pdf

Mygind, L., Jacobsen,R. H., & Swirtun, O. (2006). Introducing Linux and open source. Ericsson Review No. 1. Retrieved from: https://www.ericsson.com/ericsson/corpinfo/publications/review/2006_01/files/linux.pdf

Nordenstam, S., & Swahnberg, O. (21-12-2015). Ericsson signs patent deal with Apple, shares soar. Reuters. Retrieved from: http://www.reuters.com/article/us-ericsson-apple-idUSKBN0U40MT20151221

O'reilly, T. (2007). What is Web 2.0: Design patterns and business models for the next generation of software. Communications & strategies, (1), 17.

Open Networking Foundation (2015). Adopter Membership Agreement. Retrieved from: https://www.opennetworking.org/images/ONF_Membership_Agreement_2015-5.pdf

Open Networking Foundation (2016). Bylaws of Open Networking Foundation. Retrieved from: https://www.opennetworking.org/images/stories/downloads/about/onf-operating-documents/organizational-documents/ONF_BYLAWS_Approved_November_14_2016.pdf

Open Networking Foundation (2016). Intellectual property Rights Policy. Retrieved from: https://www.opennetworking.org/images/stories/downloads/about/onf-operating-documents/organizational-documents/onf-ipr-policy.pdf

OpenDarwin Core Team and Administrators (July 25, 2006). "OpenDarwin Shutting Down". OpenDarwin Project. Archived from the original on August 4, 2006.

Opensignal (2015). Android Fragmentation Visualized. Retrieved from: https://opensignal.com/reports/2015/08/android-fragmentation/

OpenStack (n.d.). The Four Opens. Retrieved from: http://governance.openstack.org/reference/opens.html

Oracle (2016). Oracle Contributor Agreement: Overview and Frequently Asked Questions. Retrieved from: http://www.oracle.com/technetwork/oca-faq-405384.pdf

P. Taylor, J. Lowe, Are functional assets or knowledge assets the basis of new product performance? Technol. Anal. Strateg. Manag. 9 (4) (1997) 473–488

Paul, R. (21-05-2009). Cisco settles FSF GPL lawsuit, appoints compliance officer. ARS Technica. Retrieverd from: http://arstechnica.com/information-technology/2009/05/cisco-settles-fsf-gpl-lawsuit-appoints-compliance-officer/

Paulson, J. W., Succi, G., & Eberlein, A. (2004). An empirical study of open-source and closed-source software products. Software Engineering, IEEE Transactions on, 30(4), 246-256.

Perens, B. (2002). Open Standards: principles and practice. Retrieved from: http://web.archive.org/web/20020803092336/http://perens.com/OpenStandards/Definition.html

Perens, B. (2006). Open standards principles and practice. Retrieved from: http://perens.com/OpenStandards/Definition.html

Perr, J., Sullivan, P., and Appleyard, M.M. (2006) .Open for Business: Emerging Business Models for Open Source Software Companies. Working paper, Lab2Market, Portland State University.

Pettey, C., & van der Meulen, R. (07-06-2016). Gartner Says Worldwide Smartphone Sales to Slow in 2016. Gartner. Retrieved from: http://www.gartner.com/newsroom/id/3339019

Price, C. [TelecomTV]. (2016, 07, 11). Vendors have to prove to SPs that they are trustworthy and transparent. Retrieved from: https://www.youtube.com/watch?v=lwU15kohLFI

Qualcomm (28-06-2012). Qualcomm Announces Plans to Implement a New Corporate Structure. Retrieved from: https://www.qualcomm.com/news/releases/2012/06/28/qualcomm-announces-plans-implement-new-corporate-structure

R. Rosenbloom, C. Christensen, Technological discontinuities, organizational capabilities and strategic commitments, Ind. Corp. Change 3 (3) (1994) 655–686

Rath, J. (03-06-2014). Telefonica, Red Hat and Intel Partner on Open Source NFV for Telcos. Data Center Knowledge. Retrieved from: http://www.datacenterknowledge.com/archives/2014/06/03/telefonica-red-hat-and-intel-collaborate-to-build-an-open-digital-telco/

Raymond, E. "The cathedral and the bazaar." Knowledge, Technology & Policy 12.3 (1999): 23-49.

Raymond, E. S. (1999). The magic cauldron. Retrieved from: http://www.catb.org/esr/writings/magic-cauldron/magic-cauldron.html

Reisinger, D. (2015). Samsung Will Pay Apple for Damages—But Wants the Cash Back. Retrieved from: http://fortune.com/2015/12/04/samsung-pay-apple-patents/

Reisinger, D. (23-08-2016). Here's Why Only Apple and Samsung Know How to Profit Off Smartphones. Fortune. Retrieved from: http://fortune.com/2016/08/23/apple-samsung-smartphone-profits/

Robert Frances Group. (2002). Total Cost of Ownership for Linux in the Enterprise. Retrieved from: http://www.ibm.com/linux/RFGLinuxTCO-vFINAL-Jul2002.pdf

Rockman, S. (15-05-2015). Record smartphones sales, but feature phones far from dead. The Register. Retrieved from: http://www.theregister.co.uk/2015/05/15/record_sales_for_smart_phones_but_the_feature_phone_is_not_dead/

Rohrbaugh, J. (1979). Improving the quality of group judgment: Social judgment analysis and the Delphi technique. Organizational Behavior and Human Performance, 24(1), 73-92.

Rosen, L. (2005). Open source licensing (Vol. 692). Prentice Hall.

Sackman, H. (1974). Delphi assessment: Expert opinion, forecasting, and group process (No. RAND-R-1283-PR). RAND CORP SANTA MONICA CA.

Sanduram, R. (2014). Red Hat Enterprise Linux. Retrieved from: https://fedoraproject.org/wiki/Red_Hat_Enterprise_Linux

Saunders, M., Lewis, P., & Thornhill, A. (2009). Research methods for business students. Harlow: Prentice Hall.

SDX Central (n.d.) What is NFV – Network Functions Virtualization – Definition? Retrieved from: https://www.sdxcentral.com/nfv/definitions/whats-network-functions-virtualization-nfv/

Shapiro, C., & Varian, H. R. (2013). Information rules: a strategic guide to the network economy. Harvard Business Press.

Signals Research Group (2014). THE LTE STANDARD: Developed by a global community to support paired and unpaired spectrum deployments. Retrieved from: https://www.qualcomm.com/media/documents/files/the-lte-standard.pdf

Skulmoski, G. J., Hartman, F. T., & Krahn, J. (2007). The Delphi method for graduate research. Journal of information technology education, 6, 1.

Smith, A., & Garnier, M. (1838). An Inquiry into the Nature and Causes of the Wealth of Nations. T. Nelson.

Spevack, M. (slashdot.org) (2006). Fedora Project Leader Max Spevack Responds. Retrieved from: https://slashdot.org/story/06/08/17/177220/fedora-project-leader-max-spevack-responds

Statcounter, 2016. Statcounter global stats: Top 8 Mobile and Tablet Operating Systems on May 2016. Retrieved from: http://gs.statcounter.com/#mobile+tablet-os-ww-monthly-201605-201605-bar

Statista (2016). Number of mobile phone users worldwide from 2013 to 2019 (in billions) Retrieved from: http://www.statista.com/statistics/274774/forecast-of-mobile-phone-users-worldwide/

Sun, L. (06-09-2016). The 4 Biggest Challenges Facing Qualcomm Inc. The Motley Fool. Retrieved from: http://www.fool.com/investing/2016/09/06/the-4-biggest-challenges-facing-qualcomm-inc.aspx

Swift (n.d.). About Swift. Retrieved from: https://swift.org/about/

Swift (n.d.). Contributing. Retrieved from: https://swift.org/contributing/

Teece, D. J. (1986). Profiting from technological innovation: Implications for integration, collaboration, licensing and public policy. Research policy, 15(6), 285-305.

Telefonica (2015). Results. Retrieved from: https://www.telefonica.com/documents/162467/265875/rdos15t4-eng.pdf/76729a0d-64d7-46e5-b00a-29fed0a6f0bc

Telefonica (26-03-2015). Telefónica NFV reference lab releases OpenMANO NFV orchestration stack. Retrieved from: https://www.telefonica.com/es/web/press-office/-/telefonica-nfv-reference-lab-releases-OpenMano-nfv-orchestration-stack

Thompson, D., & Serrano, R. (2016). Telefonica and Red Hat Sign a Global Agreement to Help Companies Mobilize Business Processes. Redhat Press releases. Retrieved from: https://investors.redhat.com/news-and-events/press-releases/2016/05-06-2016-110129427

Tibirna, C. (2005). The bitter failure named “Safari and KHTML”. Retrieved from: https://blogs.kde.org/2005/04/29/bitter-failure-named-safari-and-khtml

Tucker, L. (2011). Open Stack The Time Is Now [PowerPoint Slides]. Retrieved from: http://www.slideshare.net/ramdurairaj/openstack-the-time-is-now-lew-tucker-cisco

Tuomi, I. 2003. Networks of Innovation: Change and Meaning in the Age of the Internet. Oxford University Press, Oxford, UK.

van der Meulen, R., & Rivera, J. (13-02-2013). Gartner Says Worldwide Mobile Phone Sales Declined 1.7 Percent in 2012. Gartner. Retrieved from: http://www.gartner.com/newsroom/id/2335616

Von Krogh, G., & Von Hippel, E. (2006). The promise of research on open source software. Management science, 52(7), 975-983.

W. Mitchell, Are more good things better: or will technical and market capabilities conflict when a firm expands? Ind. Corp. Change 1 (2) (1992) 327–346.

W. Mitchell, Whether or when? Probability and timing of incumbents' entry into emerging industrial subfields, Adm. Sci. Q. 34 (2) (1989) 208–230.

W3Techs, 2016. Usage of operating systems for websites. Retrieved from: https://w3techs.com/technologies/overview/operating_system/all

Wagner, M. (26-05-2016). Cisco Looks to Open Source for 'Badder Ass' Internet. Light Reading. Retrieved from: http://www.lightreading.com/carrier-sdn/sdn-architectures/cisco-looks-to-open-source-for-badder-ass-internet/d/d-id/723648

West, J. (2003). How open is open enough?: Melding proprietary and open source platform strategies. Research policy, 32(7), 1259-1285.

West, J. (2004, May). What are open standards? Implications for adoption, competition and policy. In Standards and Public Policy Conference.

West, J., & Gallagher, S. (2006). Challenges of open innovation: the paradox of firm investment in open-source software. R&d Management, 36(3), 319-331.

West, J., and Bogers, M. 2014. Profiting from External Innovation: A Review of Research on Open Innovation. Journal of Product Innovation Management.

Wheeler, D. A. (2001). More than a gigabuck: Estimating GNU/Linux's size.

Yarow, J. (Business insider) (2013). Microsoft Is Making An Astonishing $2 Billion Per Year From Android Patent Royalties. Retrieved from: http://www.businessinsider.com/microsoft-earns-2-billion-per-year-from-android-patent-royalties-2013-11?r=US&IR=T&IR=T

Yin, R. K. (1989). Case study research: Design and methods, revised edition. Applied Social Research Methods Series, 5.

# Appendices

## Appendix A Krechmer's Openness of Standards

The following list contains all characteristics and their measurement values.

(1) Open Meeting: everyone is allowed to participate in the meetings. There are two broad indicators of the level of transparency possible in Open Meetings:

1. All stakeholders can pay to become a member
2. Any person or organization can participate in the process at acceptable cost

(2) Consensus: all interests are discussed until an agreement is reached. This right is measured on a 'yes or no' basis, where Yes = 1 and No = 0.

(3) Due Process: using balloting and an appeals process may for resolution. This right is measured on a 'yes or no' basis, where Yes = 1 and No = 0.

(4) One World: the same standard is implemented for the same capability, around the world. This right is measured on a 'yes or no' basis, where Yes = 1 and No = 0.

(5) Open Change: all changes to the organization and the standard are presented and agreed upon in a forum that conforms to all previous rights. This right is measured on a 'yes or no' basis, where Yes = 1 and No = 0.

(6) Open Documents: committee drafts, standards and other documents are easily available.

1. Work-in-progress documents are only available to committee members (standards creators). Standards are for sale.
2. Work-in-progress documents are only available to committee members (standards creators). Standards are available for reasonable or no cost.
3. Work-in-progress documents and standards are available for reasonable or no cost.

(7) Open Interface: open access to standardized interfaces that support migration. This right is measured on a 'yes or no' basis, where Yes = 1 and No = 0.

(8) Open Use: implementers and users need to be able to test whether the standard provides the functionality they require.

1. Open Use via implementer testing events
2. Open Use via certification.

(9) On-going Support: standards are supported until user interest ceases rather than when implementer interest declines (use). The support of an existing standard consists of four distinct phases after the standard is created

1. Create standards: The major task of SSOs
2. Fixes (changes) Rectify problems identified in initial implementations
3. Maintenance (changes) Add new features and keep the standard up to date with related standards work
4. Availability (no changes) Continue to publish, without continuing maintenance
5. Rescission Removal of the published standard from distribution

# Appendix B Interviews

| Interviewed | Position | Date | Duration |
|---|---|---|---|
| **Gunnar Nilsson** | Open Source Strategies, Ericsson | 27-06-2016 | 60 minutes |
| **Magnus Karlsson** | Expert Open systems software architecture, Ericsson | 07-07-2016 | 80 minutes |
| **Christian Hoymann** | Senior Research Engineer, Ericsson | 29-07-2016 | 45 minutes |
| **Jimmy Ahlberg** | Defensive IPR Strategist, Ericsson | 11-10-2016 | 30 minutes |
| **Eduardo Fullea** | Intellectual Property & Standards Specialist, Telefonica | 20-10-2016 | 50 minutes |

## Interview 1: Gunnar Nilsson

Interviewed: Gunnar Nilsson, Ericsson

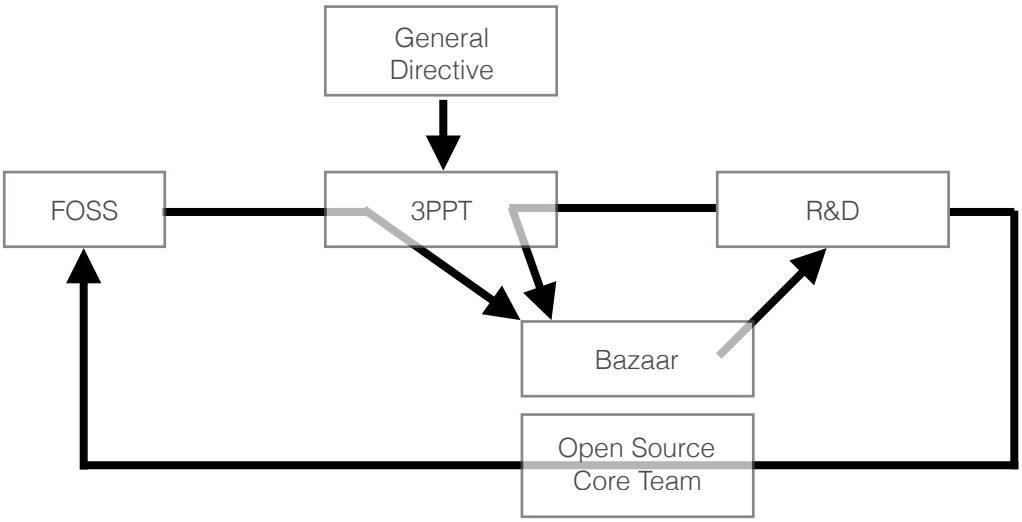
Date: 2016-06-27

Time: 09:00-10:00

Topic: Open Source within Ericsson

**What does the current decision process for open source projects look like?**

For simple small packages we have the following process in place:




All externally sourced software packages are made available trough the Bazaar, which is basically an e-commerce platform for software. Whenever R&D needs something that is not available in the bazaar, sourcing will do the acquisition and make the software available in the bazaar. For FOSS, there is a specific task force within sourcing called 3PPT. 3PPT evaluates all FOSS requests in terms of licenses and effects on existing IP.

**How does Ericsson deal with FOSS contributions by employees/departments?**

When R&D improves a FOSS package and wants to contribute this to the open source version, it has to be evaluated by the Open Source Core Team first. They look at the FOSS license, the global scale and the current (or future) patent licenses Ericsson has.

We are always very careful to contribute to copyleft-type licenses, as we have to determine the scope of the 'virality' of the copyleft. But for permissive licenses we usually don't have any issues. Then we look at the global scale of the project. When it is a project with a small development team and not very wide adaption we have no objections, when the scale is larger we have to investigate more. Finally, we look at our patent portfolio, not only current licensing deals, but also future potential deals, and patents which we determine valuable. The combination of these three factor determines whether or not we allow contributions to the upstream project. But at the same time we are not looking to create our own internal forks of FOSS projects, we want to stay as close to the original project as possible.

**Does this happen often and for what kind of projects?**

We get FOSS contribution requests quite a lot, often just small stuff like bug fixes, or minor functions. But with functions the decision becomes more difficult, as we have to determine how the functionality of the software is changed, whether this is an innovation and how this relates to our patent portfolio.

We also get some requests to open source a project created by one employee or a department. This are usually just tools, nothing major. Usually this is open sourced because the centralized tools department notices the use of a non-authorized tool and has an alternative available. If this happens there will be no more budget to support the tool and the developer / department open sources the tool. They hope that the project will be picked up by some FOSS developer, but this is rarely the case. There is a misconception that FOSS developers are waiting for new projects to pick up, which doesn't happen.

**What do you think about the projects where Ericsson is a key contributor?**

OPNFV, Open Stack, Open Daylight. These decisions become more strategic than tactical.

The current FOSS projects in the TelCo industry are being really hyped. Looking at the Gartner Hype cycle, these projects are probably somewhere close to the hype plateau and somewhere in the future they will experience this disillusion dip. I don't believe in a fully automated, self calibrating network, each network will need specific integration and tuning. Even then there will always remain a human coordination element. While these open source projects are aiming to contribute to this, it will still take a long time for them to reach the stability and level of development needed. In the mean time there are already proprietary solutions available. Probably in the future a mix between proprietary and FOSS elements will be the optimal solution. These large operators have the idea that they themselves can be the ones integrating the FOSS projects into their networks and save costs by this. But looking at current examples, companies are usually hiring outside experts to integrate and maintain their FOSS projects, look at RedHat.

Even for more established projects such as Open Stack, the project is growing and growing because all of the contributors are selfish. Even though the project is mainly developed by corporations, they all take on their own area of functionality and work on this, expanding the functionality. Due to this process the boundaries of the project are getting further and further apart and maintaining and stabilizing the software becomes very difficult. When you look at the more successful projects, they are governed far stricter and the scope of the software is well-defined. This is currently not the case for the majority of TelCo FOSS projects.

**How do OPNFV, OpenDayLight and OpenStack relate to Ericsson's core business?**

They are not the core business, we don't have many patents in the area. While we have our own version Ericsson Cloud Manager, this is not a main income stream. It's more like a side-stream, or a complementary. It's nice to offer it as well.

## Interview 2: Magnus Karlsson

Interviewed: Magnus Karlson, Ericsson

Date: 2016-07-07

Time: 13:30-15:00

Topic: Open Source within Ericsson: Mano, openO and OPNFV

**Why did Ericsson join the OpenO project?**

Our customer department was asked by China Mobile to help them set up an open source project for the management and orchestration layer. Due to geographical reasons it was not desirable to connect the client to the ETSI Mano project. Therefore the client was helped to setup Open O with the Linux Foundation and Huawei. Initially there was no intention from Ericsson to join the project, however due to some internal miscommunications the company did agree to participate and could not back out.

Cloud management and orchestration focusses completely on standardizing interfaces, it enables common computing and communications interfaces and the key methods are part of common engineering knowledge. As such MANO can be classified as infrastructure software and is regarded a commodity. However, the know-how of implementation and maintenance of MANO is not a commodity and is the part where Ericsson captures value.

Telecommunications operators are pressing for open source implementations of MANO. Not necessarily because they want to implement MANO themselves, but because this allows them to create a broader market for implementers and reduce these costs.

**What will be the consequences of a commoditized MANO for Ericsson?**

Currently Ericsson is still earning money through the implementation of cloud management software, therefore it is not actively supporting Open O and Mano, but rather hopes that the projects will delay. However, it is clear that on the long term MANO will become a commodity. Ericsson is contributing to the project by helping define the standardized interfaces and making sure that it is compatible with other Ericsson products, but there will be limited code contributions. While standardization development is underway, the standard is not yet completed and multiple proprietary solutions are currently being sold in the market

**Why is Ericsson contributing to OPNFV?**

From the beginning it was clear that OPNFV will be a commodity, no matter what Ericsson does. In order to ensure the quality and compatibility with other Ericsson systems the company joined the project. OPNFV is the integration of many different things, and for every sub-project Ericsson has made a conscious decision to join, or not join. For instance, OpenDayLight is a project that improves SDN. Ericsson would not be able to keep up with the technological change in a proprietary project, whereas the open source project can. Moreover, ODL is a threat to Cisco's operations, Ericsson joined ODL in part to attack Cisco's eco-system.

## Interview 3: Christian Hoymann

Interviewed: Christian Hoymann, Ericsson

Date: 2016-07-29

Time: 10.15 - 11:00

Topic: 3GPP organization and governance

**How are the Technical Specification Groups in 3GPP organized?**

I work in the TSG for Radio Access Network (RAN), which is one of the biggest groups with the highest workload. At the top level we have RAN plenary, which is where the project management of the technical specification is organized. Here we introduce new features which are then turned into work items. Each feature is defined as a single work item, and each work item is assigned to a single working group. The working group is where the actual detailed specification work is done. RAN plenary meets 4 times per year, while the working groups meet 6 times per year. RAN plenary monitors and steers the work the WG's are doing on the work items, generally they don't interfere but they make sure that the work is progressing according to schedule. When a WG finishes a work item it gets submitted to RAN plenary which approves the specification and publishes it.

**How do technical specifications get submitted to the working groups?**

So the WGs meet 6 times per year, which sets the planning for each sub period. These meetings are very strict and when you miss a deadline it is your own problem. This way we ensure that the work keeps continuing forward. Each meeting takes about a week and the goal is not to spend time on explanations and presentations. Everything that will be discussed and decided upon during the meeting has to be submitted in document form one week before the meeting, and everyone will have to have reviewed all the submissions.

The submissions are not only the proposed technical specifications. Each specification has to be baked up by simulations, tests, throughput studies etc. There are often competing specifications and the decisions are primarily based on technical grounds. However, sometimes specifications can have the same efficiency / quality, when this happens the submitters will try to find technical support for their solutions which are not necessarily scientifically proven (e.g. easier to implement, faster time to market etc). When this happens we will often look for a compromise between the solutions.

**How are the decisions on specifications made?**

Everyone in the room will have a veto'ing power on the decisions, we work with consensus building. Input in the meetings is only accepted from attendees. As a result all of the decisions really feel like group decisions, and everybody owns the decision. When someone tries to go back on decisions in later meetings often even the people who were initially opposed to the decision will argue in favor of the decision. These meetings are often attended by up to 300 people.

**What about the influence of IPR on the processes?**

Of course in the background everything is done with a hidden IPR agenda, but no-one will ever speak about this during the meetings and decisions are primarily based on technical advantages. No-one will ever make a contribution that is not backed by patents, but because of the way the WGs are organized there is a balance among participants.

**Do firms try to influence this process through coalitions?**

There are some small coalitions, but these are not set in stone. These coalitions will vary per topic and will be more based on commonalities in technology than backgrounds. There is a coalition of network operators, which meets before the WG meetings in order to discuss things, but they are not very successful. More often than not the members will not reach an agreement, and event then, everyone has a veto right so their power is limited. Sometimes there is a sort of Chinese coalition, which stems from the ministry research lobby.

**How hard is it for new companies to join and contribute to the WGs?**

For a new firm to join and give input to the process right away is difficult. The technology is very complicated and new companies are often not knowledgable enough to create new inputs. Even for established firms it is still difficult to keep up. Besides that there is a long history in 3GPP with many discussions and decisions. As legacy technology such as GPRS or EDGE still heavily influences 4G technologies a lack of knowledge about this history is a large obstacle for newcomers. Usually it takes firms 2 to 5 years to get up to speed and start making contributions.

## Interview 4: Jimmy Ahlberg

Interviewed: Jimmy Ahlberg, Ericsson

Date: 2016-10-11

Time: 16:30-17:00

Topic: Open Source within Ericsson

**Can you tell me about NFV / MANO, and how ETSI OSM relates to Ericsson?**

So basically there are two things happening currently at ETSI. First we have the standardization track, of which we are a part. This is just like any other standardization track, where operators have shared a goal, and we and other vendors are developing solutions for this problem.

On the other side there is the Open Source pilot called OSM. This is to be a reference implementation of that same standard. However, we have a number of issues with this project. First of all, ETSI is never meant to be building implementations. It is a Standards organization, and even within the statutes of the foundation there is no provision for what they are doing now. So they have called it an Industry Specification Group, while they are actually building the software itself.

Additionally, ETSI OSM is licensed under the Apache V2 license, which has a very broad patent scope. So now we have two different IPR policies within ETSI, the one from ETSI itself, which is FRAND, and the IPR policy as written in the Apache V2 license.

Currently we are working on the standardization process, but we are not willing to engage with the Open Source Mano, as this would imply us granting all the relevant patent licenses for free, while we are also selling our own MANO solution.

**How valuable do you think NFV is to Ericsson in terms of competitive advantage?**

Currently we are offering our own NFV MANO suite called Ericsson Cloud Manager, while this is not a large revenue driver it is an important addition to our offering. Besides, we own some patents covering the NFV / MANO technologies, which we are currently working on to get submitted to the standard. These will become increasingly important once the standardization is complete.

**Is this the same for SDN?**

Not really, no. SDN is something that was developed out in the open from the beginning, so we don't really have any significant patents in this area. Rather we think that SDN is a commodity or will become a commodity in the market, so while we think it is important to participate in the development, this is not a key area for us.

## Interview 5: Eduardo Fullea

Interviewed: Eduardo Fullea Carrera, Telefonica

Date: 2016-10-20

Time: 13:30-14:20

Topic: Open Source within Telefonica & ETSI

*I'd like to start with saying that any opinion that I express during this call is a personal opinion and may not represent the opinion of Telefonica.*

### *Could you tell me a little bit about your job and what you do?*

I work at the IPR department of Telefonica. We deal with the IPR management at the group level. My background is technological, I am a telecommunications engineer. I have worked many years in Innovation projects before I joined the IPR department. We manage the IPR assets from a technological point of view. I have also a lot of experience with standardization, so my personal focus within the team is not only on patents but also on the relationship between patents, standards and Open Source.

### I haven't seen Telefonica in some of the SEP studies I read. On what kind of technology is Telefonica focusing?

Our Research and Development is focused on two tracks: network innovation and digital services. We have IPR in both worlds. With regard to the latter we innovate in different areas, like personal communications, cloud services, security services, and video services. We produce IP in more or less all the areas where we have some innovation activity.

With regard to the networks we have IP in both mobile and fixed networks and network virtualization. The usual suspects nowadays where all the companies in the sector are focusing their innovation

### What are the main Open Source projects Telefonica is engaging with, and how is this perceived within the company?

One of our biggest projects is the FIWARE initiative. It is a large project subsidized by the European Commission that started some years ago. We strive to set up a framework on top of which innovative services can easily be developed. The FI in FIWARE stands for Future Internet. The original focus was on creating a framework for future internet services. in practice the kind of services that can be developed on top of this framework is very rich, like IOT, Smart Cities, Smart Industries can be build on top of FIWAER.

It is a European consortium which has developed a lot of generic enablers. These are Open Source Software projects that implement a specific functions that may be useful to create a specific platform for one of these use cases. The project has had several phases and is still ongoing. The current result is a catalogue of various Open Source reference implementations licensed under different OSS licenses that can be used by anyone interested by anyone to set up a platform. Taking the necessary enablers depending on the use case, where the interoperability is quite straight forward. The different companies will be based on the same enablers based on the same external APIs to other enablers.

So this is probably one of the most significant projects in which Telefonica has been involved. Telefonica has been the leader or enabler of this project, which is a very big European project. We are one of the most relevant drivers, but there is also Orange, and of course there are other types of actors, not only operators.

**Looking to ETSI OSM, normally in ETSI there is a high rate of patenting on the standards, whereas the ETSI OSM license is not very patent friendly. How do you see ETSI OSM compared to the other ETSI activities?**

I think that having Open Source projects linked to paper-ware standardization processes that produce specifications may be positive in many cases. So we think that it can be beneficial to accelerate the time-to-market for the services, it can also be good for interoperability. Also there is a sort of synergy between the specification and the reference implementation development, during the early implementation of the standard, while it is still under development the reference implementation may help improve the quality of the standard as bugs and flaws can be detected at an early stage. Additionally, the reference implementation may be more solid when there is a direct link to the people specifying the standard as they are usually more systematic about defining the system design or architecture, so we see some synergy in this cooperation between the SDOs and the reference implementation as OSS.

Of course the topic about the IPRs is an issue. For companies heavily relying on licensing revenues to finance their innovation it can be a challenge to contribute to this reference implementation when it is licensed that implies an explicit patent license or a non exertion compromise. But of course it depends on a company, for smaller players for which licensing is not an important revenue stream it may more positive. So I think the selection of one model or another depends on the specific situation in each case. Of course the optimal situation would be to find a middle ground, which does not negatively impact some actors over others. It is something to be analyzed case by case.

In ETSI OSM I am aware that there are some companies that are not participating because of the chosen OSS license.

**Telefonica has some IPR in the NFV space, but is engaging with the NFV Open Source project nonetheless. Why did you decide to engage even though you own IPR in the field?**

Losing the ability to enforce a patent within the Open Source ecosystem is not an issue for us, we are one of the drivers of the initiative. We do think that IP always needs to be protected as it is a valuable asset. Our intention is to protect out innovations as much as possible. The decision to contribute to Open Source projects that may have non-assertion commitment because of the selected license. Sometimes creating an Open Source ecosystem where we can collaborate is more important than giving up some IPR that we own.

Basically the creation of an ecosystem is one of the reasons for us to contribute to Open Source. The most usual cases for us to contribute to Open Source is those where we see that we can create a collaborative scene where many companies can work together. Even in those cases where we think that having the IPR is still valuable. In case we provide royalty free licenses we still maintain the defensive value of the patents. Contributing to Open Source does not mean giving up our IP.

**When contributing to Open Source, is there a fear that competitors will reap the benefits from investments made by Telefonica?**

We contribute to Open Source projects where we don't fear this, or we don't find it an issue compared to the benefits we get from our contribution to the project. That is always a part of the decision to engage with Open Source projects.

**Do you think SDO activities and Open Source development will converge in the future?**

We cannot generalize, so maybe not for every case. But I think it will become more and more useful to see such collaborative schemes. I think this pilot and similar ones will become more popular in the future.

**Comparing this to more traditional OSS which has a more direct benefit to the individual contributor, these specialized networking projects will probably attract fewer individuals. Do you think that the same benefits will be gained nonetheless?**

I think we will not be able to attract the interest of individuals as much as other kinds of projects. We expect contributions to come from the enterprise world. When we are talking about networks it may be more valuable to have professional contributions coordinated from companies than a myriad of individual contributors. I don't think this would be a problem, but we cannot expect plenty of individual contributors to be jumping on board to these kinds of projects.

**How have Telefonica's Open Source experiences been so far?**

Overall the experiences have been very positive. For example within FIWARE the ecosystem is gaining quite some traction, not only within Europe. More and more companies from around the world, for example from latin America have adopted the model that has been released as part of the project. They are beginning to use this kind of platforms to develop commercial services such as IOT services, Smart City services, so our experiences have been quite positive. When I think about Smart City services, platforms which are based on the FIREWARE generic enables exposed API's which allowed applications to be portable between different cities that have platforms according to the same model. So I think the ecosystem has not only been positive for us, but for all the companies that are coming on board of the ecosystem. So I think it is helping to promote a de facto standard which will be useful for a more interoperable and faster uptake of services like those in the Smart Cities. ETSI OSM is in an earlier stage, so it to soon to really assess the impact of this project at this time.